

```

;-----
; ANHT/4298 P/N 16134297
;
; MY90 L98 YB AUTO
; ECM 1227727
;
;-----

```

ORG \$8000

```

;-----
L8000:  FDB $13D9  ; EPROM ID, 5181d
;
L8002:  FDB $08BF  ; DATE CODE
L8004:  FDB $0086  ; SEQ NUM
;
L8006:  FDB $DAA5  ; CK SUM
L8008:  FCB $8D    ; ECM ID BYTE
;
L8009:  FCB $00    ; NUM CLY, (8 CYL)
;      4 = $80 1000 0000
;      6 = $C0 1100 0000
;      8 = $00 0000 0000
;
L800A:  FCB $CA    ; CUST ID 1
L800B:  FCB $03    ; CUST ID 2
L800C:  FCB $C1    ; CUST ID 3
L800D:  FCB $07    ; CUST ID 4
;
L800E:  FDB $0CCD  ; VATS MAX, 3277d, 66 Hz
L8010:  FDB $0666  ; VATS MIN, 1638d, 33 Hz
;      30 - 60 Hz
;
L8012:  FDB 0256   ; SEC, IF IGN OFF THIS TIME TURN OFF ECM
;      ARG = TIME*80, (3.2 SEC)
;
L8014:  FCB $00    ; INST PNL DIVIDER ARG
L8015:  FCB 230    ; PULSE/MILE RD SPD CONST
;      ARG = 921600/VAL
;

```

```

*****
* AFR OPTION FLAGS
*
*****

```

```

;
L8016:  FCB $14    ; AFR OPTION WD 1 0001 0100
;-----
;
;      b7, 1 = --
;      b6, 1 = USE A/D BATT FOR INJ Lk Up, (else use PUMP VDC)
;      b5, 1 = OPEN LOOP PURGE
;      b4 ,1 = ENABLE VATS
;
;      b3, 1 = LET LOW TPS DISABLE CCP
;      b2, 1 = 4TH GR SW N.O.
;      b1, 1 = 16 Hz CCP SOLENIOD IN USE
;      b0, 1 = RPM FOR TCC vs MPH
;
;-----
;
L8017:  FCB $04    ; AFR OPT WD 2, 0000 0100
;-----
;
;      b3, N/A
;      b2, 1 = DUAL AIR VALVE/ 0 = SINGLE

```

```

; b1, 1 =
; b0, 1 = EGR DIAG, 1 PASS OPT
;-----
;
L8018: FCB $52 ; AFR OPT WD 3, 0101 0010
;-----
;
; b7, 1 = CCP WHEN OUTPUT ENRGIZED
; b6, 1 = MAG SPD SENSER INSTALLED
; b5, 1 = INT RESET WHEN BLM CELL CHG
; b4, 1 = OPT INV MAT Lk Up
;
; b3, 1 = UNLOCK TCC IN DECEL FUEL C/O
; b2, 1 = CPU Auto Ratio Sel, 0 = TCC
; b1, 1 = INT RESET WHEN Accel Enr IN WORK
; b0, 1 = COOL FOR IDLE SPD TIME OUT
;-----
;
;
L8019: FCB $09 ; AFR OPT WD 4, 0000 1001
;-----
;
; b0 , 1 = ERROR 32 DISABLED IF PASSES 1 TIME
; b1
; b2 , 1 = 4th GR HIWAY MODE SA CHECK
; b3 , 1 = IF CCP CHG, FORCE INT TO 128
;
; b4 , 1 = TCC LOCKED HIWAY S.S. CHK ENAB
; b5 , 1 = N.O. FAN REQ INPUT
; b6 , 1 = PWR STEER SW INSTALLED
; b7
;-----

```

```

*****
* SPARK CALIBRATION PARAMETERS
*
*****

```

```

L801A: FDB 3277 ; 300 RPM, START UP RPM
; ARG = (65536*120)/(RPM/NUM CYL)

;-----
; HI RPM DRP'S REQ FOR ENG RUN vs COOL
;
; TABLE = DRP'S
;-----
L801C: FCB 08 ; -40 DEG C
FCB 08 ; -16
FCB 08 ; 8
FCB 08 ; 32
FCB 08 ; 56
FCB 08 ; 80
FCB 08 ; 104
FCB 08 ; 128
FCB 08 ; 152
;-----

L8025: FCB 17 ; INITAL SPK ADV, 6 DEG

L8026: FDB 0119 ; 39 DEG, (MAX SA ADDED TO INIT TMG)
;

L8028: FDB $FFF5 ; 65525d (MAX SA RETARD FOR KNOCK), 3.5 DEG
;

```

```

;-----
; STALL SAVER
;-----
L802A:  FCB 24      ; 300 RPM, IF RPM < ENABLE STALL SAVER SA
L802B:  FCB 24      ; 300 RPM, IF RPM < & RPM > L802C,
          ; ENABLE STALL SAVER SA
L802C:  FCB 44      ; 550 RPM, IF RPM > & RPM < L802B,
          ; ENABLE STALL SAVER SA
L802D:  FCB 24      ; 300 RPM, IF RPM > DISABLE STALL SAVER SA
L802E   FCB 16      ; 16 BIN ADDED TO SA BLEND EACH DRP

```

```

;-----
; HOT RESTART RETARD S.A.
;-----
L802F:  FCB 113     ; 45 DEG C, IF COOL < DISABLE HOT RETARD, (113F)
L8030:  FCB 1       ; 0.4 DEG SA, HOR RESTART RETARD (256/90)
L8031:  FCB 113     ; 44.5 Deg C, IF COOL >= Disable cold spk, (113F)
L8031:  FCB 255     ; 151 Deg C, if diff cool >= disable cold SA
          ; ARG = TEMP * (256/192)

```

```

;-----
; Dissassembly of ANHT
; 11-15-1990, 21:07:45
;
; >> MAIN SPARK ADVANCE TABLE (90 Vette) <<
;-----

```

ORG \$8033

```

FCB      0       ; Min RPM
FCB      0       ; Min MAP, (20 Kpa)
FCB      12      ; Number BP'S per RPM

```

```

;-----
; 400 RPM           SA           MAP
;-----
FCB 57   ; 20.0      20 Kpa
FCB 57   ; 20.0      30 Kpa
FCB 57   ; 20.0      40 Kpa
FCB 57   ; 20.0      50 Kpa
FCB 57   ; 20.0      60 Kpa
FCB 57   ; 20.0      70 Kpa
FCB 57   ; 20.0      75 Kpa
FCB 51   ; 17.9      80 Kpa
FCB 51   ; 17.9      85 Kpa
FCB 51   ; 17.9      90 Kpa
FCB 51   ; 17.9      95 Kpa
FCB 51   ; 17.9     100 Kpa

```

```

;-----
; 600 RPM           SA           MAP
;-----
FCB 57   ; 20.0      20 Kpa
FCB 57   ; 20.0      30 Kpa
FCB 57   ; 20.0      40 Kpa
FCB 57   ; 20.0      50 Kpa
FCB 57   ; 20.0      60 Kpa
FCB 57   ; 20.0      70 Kpa
FCB 57   ; 20.0      75 Kpa
FCB 51   ; 17.9      80 Kpa
FCB 51   ; 17.9      85 Kpa
FCB 51   ; 17.9      90 Kpa
FCB 51   ; 17.9      95 Kpa

```

FCB	51	; 17.9	100	Kpa

800 RPM		SA	MAP	

FCB	57	; 20.0	20	Kpa
FCB	57	; 20.0	30	Kpa
FCB	85	; 29.9	40	Kpa
FCB	85	; 29.9	50	Kpa
FCB	80	; 28.1	60	Kpa
FCB	68	; 23.9	70	Kpa
FCB	60	; 21.1	75	Kpa
FCB	51	; 17.9	80	Kpa
FCB	51	; 17.9	85	Kpa
FCB	51	; 17.9	90	Kpa
FCB	51	; 17.9	95	Kpa
FCB	51	; 17.9	100	Kpa

1000 RPM		SA	MAP	

FCB	57	; 20.0	20	Kpa
FCB	71	; 25.0	30	Kpa
FCB	108	; 38.0	40	Kpa
FCB	108	; 38.0	50	Kpa
FCB	100	; 35.2	60	Kpa
FCB	97	; 34.1	70	Kpa
FCB	88	; 30.9	75	Kpa
FCB	74	; 26.0	80	Kpa
FCB	60	; 21.1	85	Kpa
FCB	51	; 17.9	90	Kpa
FCB	51	; 17.9	95	Kpa
FCB	51	; 17.9	100	Kpa

1200 RPM		SA	MAP	

FCB	85	; 29.9	20	Kpa
FCB	97	; 34.1	30	Kpa
FCB	114	; 40.1	40	Kpa
FCB	114	; 40.1	50	Kpa
FCB	108	; 38.0	60	Kpa
FCB	100	; 35.2	70	Kpa
FCB	88	; 30.9	75	Kpa
FCB	77	; 27.1	80	Kpa
FCB	65	; 22.9	85	Kpa
FCB	57	; 20.0	90	Kpa
FCB	51	; 17.9	95	Kpa
FCB	51	; 17.9	100	Kpa

1400 RPM		SA	MAP	

FCB	108	; 38.0	20	Kpa
FCB	119	; 41.8	30	Kpa
FCB	119	; 41.8	40	Kpa
FCB	119	; 41.8	50	Kpa
FCB	111	; 39.0	60	Kpa
FCB	105	; 36.9	70	Kpa
FCB	97	; 34.1	75	Kpa
FCB	85	; 29.9	80	Kpa
FCB	74	; 26.0	85	Kpa
FCB	68	; 23.9	90	Kpa
FCB	54	; 19.0	95	Kpa
FCB	51	; 17.9	100	Kpa

1600 RPM		SA	MAP	

FCB	108	; 38.0	20	Kpa
FCB	125	; 43.9	30	Kpa

FCB	125	; 43.9	40	Kpa
FCB	125	; 43.9	50	Kpa
FCB	119	; 41.8	60	Kpa
FCB	108	; 38.0	70	Kpa
FCB	102	; 35.9	75	Kpa
FCB	91	; 32.0	80	Kpa
FCB	85	; 29.9	85	Kpa
FCB	77	; 27.1	90	Kpa
FCB	65	; 22.9	95	Kpa
FCB	51	; 17.9	100	Kpa

; 1800 RPM SA MAP

FCB	108	; 38.0	20	Kpa
FCB	131	; 46.1	30	Kpa
FCB	131	; 46.1	40	Kpa
FCB	131	; 46.1	50	Kpa
FCB	119	; 41.8	60	Kpa
FCB	114	; 40.1	70	Kpa
FCB	102	; 35.9	75	Kpa
FCB	97	; 34.1	80	Kpa
FCB	88	; 30.9	85	Kpa
FCB	80	; 28.1	90	Kpa
FCB	68	; 23.9	95	Kpa
FCB	57	; 20.0	100	Kpa

; 2000 RPM SA MAP

FCB	108	; 38.0	20	Kpa
FCB	137	; 48.2	30	Kpa
FCB	134	; 47.1	40	Kpa
FCB	131	; 46.1	50	Kpa
FCB	117	; 41.1	60	Kpa
FCB	114	; 40.1	70	Kpa
FCB	108	; 38.0	75	Kpa
FCB	97	; 34.1	80	Kpa
FCB	88	; 30.9	85	Kpa
FCB	82	; 28.8	90	Kpa
FCB	71	; 25.0	95	Kpa
FCB	63	; 22.1	100	Kpa

; 2200 RPM SA MAP

FCB	108	; 38.0	20	Kpa
FCB	134	; 47.1	30	Kpa
FCB	134	; 47.1	40	Kpa
FCB	131	; 46.1	50	Kpa
FCB	119	; 41.8	60	Kpa
FCB	111	; 39.0	70	Kpa
FCB	108	; 38.0	75	Kpa
FCB	97	; 34.1	80	Kpa
FCB	85	; 29.9	85	Kpa
FCB	82	; 28.8	90	Kpa
FCB	68	; 23.9	95	Kpa
FCB	63	; 22.1	100	Kpa

; 2400 RPM SA MAP

FCB	108	; 38.0	20	Kpa
FCB	131	; 46.1	30	Kpa
FCB	131	; 46.1	40	Kpa
FCB	131	; 46.1	50	Kpa
FCB	114	; 40.1	60	Kpa
FCB	108	; 38.0	70	Kpa
FCB	100	; 35.2	75	Kpa
FCB	88	; 30.9	80	Kpa

FCB	82	; 28.8	85	Kpa
FCB	77	; 27.1	90	Kpa
FCB	68	; 23.9	95	Kpa
FCB	63	; 22.1	100	Kpa
;-----				
; 2800 RPM		SA		MAP
;-----				
FCB	108	; 38.0	20	Kpa
FCB	131	; 46.1	30	Kpa
FCB	131	; 46.1	40	Kpa
FCB	131	; 46.1	50	Kpa
FCB	114	; 40.1	60	Kpa
FCB	100	; 35.2	70	Kpa
FCB	85	; 29.9	75	Kpa
FCB	80	; 28.1	80	Kpa
FCB	71	; 25.0	85	Kpa
FCB	65	; 22.9	90	Kpa
FCB	57	; 20.0	95	Kpa
FCB	57	; 20.0	100	Kpa
;-----				
; 3200 RPM		SA		MAP
;-----				
FCB	108	; 38.0	20	Kpa
FCB	131	; 46.1	30	Kpa
FCB	131	; 46.1	40	Kpa
FCB	114	; 40.1	50	Kpa
FCB	100	; 35.2	60	Kpa
FCB	88	; 30.9	70	Kpa
FCB	85	; 29.9	75	Kpa
FCB	80	; 28.1	80	Kpa
FCB	71	; 25.0	85	Kpa
FCB	65	; 22.9	90	Kpa
FCB	60	; 21.1	95	Kpa
FCB	57	; 20.0	100	Kpa
;-----				
; 3600 RPM		SA		MAP
;-----				
FCB	108	; 38.0	20	Kpa
FCB	131	; 46.1	30	Kpa
FCB	131	; 46.1	40	Kpa
FCB	120	; 42.2	50	Kpa
FCB	105	; 36.9	60	Kpa
FCB	88	; 30.9	70	Kpa
FCB	82	; 28.8	75	Kpa
FCB	77	; 27.1	80	Kpa
FCB	71	; 25.0	85	Kpa
FCB	65	; 22.9	90	Kpa
FCB	60	; 21.1	95	Kpa
FCB	57	; 20.0	100	Kpa
;-----				
; 4000 RPM		SA		MAP
;-----				
FCB	108	; 38.0	20	Kpa
FCB	131	; 46.1	30	Kpa
FCB	131	; 46.1	40	Kpa
FCB	120	; 42.2	50	Kpa
FCB	105	; 36.9	60	Kpa
FCB	88	; 30.9	70	Kpa
FCB	82	; 28.8	75	Kpa
FCB	77	; 27.1	80	Kpa
FCB	71	; 25.0	85	Kpa
FCB	65	; 22.9	90	Kpa
FCB	60	; 21.1	95	Kpa
FCB	60	; 21.1	100	Kpa
;-----				
; 4400 RPM		SA		MAP

```

;-----
FCB 108 ; 38.0 20 Kpa
FCB 131 ; 46.1 30 Kpa
FCB 131 ; 46.1 40 Kpa
FCB 125 ; 43.9 50 Kpa
FCB 111 ; 39.0 60 Kpa
FCB 99 ; 34.8 70 Kpa
FCB 91 ; 32.0 75 Kpa
FCB 80 ; 28.1 80 Kpa
FCB 77 ; 27.1 85 Kpa
FCB 71 ; 25.0 90 Kpa
FCB 65 ; 22.9 95 Kpa
FCB 65 ; 22.9 100 Kpa

```

```

;-----
; 4800 RPM SA MAP
;-----
FCB 108 ; 38.0 20 Kpa
FCB 131 ; 46.1 30 Kpa
FCB 131 ; 46.1 40 Kpa
FCB 131 ; 46.1 50 Kpa
FCB 108 ; 38.0 60 Kpa
FCB 97 ; 34.1 70 Kpa
FCB 91 ; 32.0 75 Kpa
FCB 85 ; 29.9 80 Kpa
FCB 80 ; 28.1 85 Kpa
FCB 74 ; 26.0 90 Kpa
FCB 68 ; 23.9 95 Kpa
FCB 68 ; 23.9 100 Kpa

```

```

;-----
; EXTENDED SPARK ADVANCE TABLE
; Spark vs RPM and MAP
;
; ANHT
; 90 VETTE
;
; TBL = N * (256/90)
;-----

```

```

ORG $8102 ;
;
L8102: FCB 192 ; RPM min
FCB 0 ; MAP Min
FCB 12 ; Num BP'S/RPM

```

```

;-----
; 4800 RPM SA DEG MAP Kpa
;-----
FCB 131 ;46.05 20
FCB 131 ;46.05 30
FCB 131 ;46.05 40
FCB 131 ;46.05 50
FCB 108 ;37.96 60
FCB 97 ;34.10 70
FCB 91 ;31.99 75
FCB 85 ;29.88 80
FCB 80 ;28.12 85
FCB 74 ;26.01 90
FCB 68 ;23.90 95
FCB 68 ;23.90 100

```

```

;-----
; 5200 RPM
;-----
FCB 131 ;46.05 20

```

```

FCB      131 ;46.05      30
FCB      131 ;46.05      40
FCB      131 ;46.05      50
FCB      108 ;37.96      60
FCB      97  ;34.10      70
FCB      91  ;31.99      75
FCB      85  ;29.88      80
FCB      80  ;28.12      85
FCB      74  ;26.01      90
FCB      68  ;23.90      95
FCB      68  ;23.90     100

```

```

;-----
; 5600 RPM
;-----

```

```

FCB 131      ;46.05      20
FCB 131      ;46.05      30
FCB 131      ;46.05      40
FCB 131      ;46.05      50
FCB 108      ;37.96      60
FCB 97        ;34.10      70
FCB 91        ;31.99      75
FCB 85        ;29.88      80
FCB 80        ;28.12      85
FCB 74        ;26.01      90
FCB 68        ;23.90      95
FCB 68        ;23.90     10

```

```

;-----
; Spark latency correction, (USEC)
;
; TBL = VAL/15.26
;-----

```

```

;-----
; Usec      RPM
;-----
L8129: FCB 8      ; 122      800
FCB 11     ; 168      1200
FCB 12     ; 147      1600
FCB 13     ; 183      2000
FCB 17     ; 259      2400
FCB 15     ; 229      2800
FCB 13     ; 198      3200
FCB 13     ; 198      3600
FCB 13     ; 198      4000
FCB 14     ; 214      4400
FCB 14     ; 214      4800
FCB 14     ; 214      5200
FCB 14     ; 214      5600
FCB 14     ; 214      6000
;-----

```

```

;-----
; COOLANT SA CORR vs LD / 2 AND COOL
;
;-----

```

```

L8137: FCB 57      ; SA BIAS FOR BASE COOL S.A. TABLE
          ; 20 DEG (for next table)

```

```

*****
INCLUDE F2A.SRC
;-----
; COOL SA TBL, 12 BLKS

```



```

; $8138 - $81A7
;
; TBL = (VAL + L8137) * (256/90)
*****

;-----
; HIWAY MODE SA
;-----
L81A7:  FCB 255      ; 151 Deg C, IF COOL < DISABLE HIWAY MODE SA
L81A8:  FCB 43       ; 43 Kpa, IF MAP > DISABLE HIWAY SA MODE
L81A9:  FCB 254      ; 4762 RPM, IF RPM < DISABLE HIWAY SA MODE
L81AA:  FCB 10       ; 10 SEC, ENABLE IF COND'S EXIST THIS PERIOD

;-----
; HIWAY MODE SA vs MAP
; TBL = 256/90
;-----
; S.A.      MAP
;-----
L81AB:  FCB 0      ; 0      30 Kpa
        FCB 0      ; 0      35
        FCB 0      ; 0      40
        FCB 0      ; 0      45
        FCB 0      ; 0      50
        FCB 0      ; 0      55
        FCB 0      ; 0      60
        FCB 0      ; 0      65

;-----
; PWR ENRICH SA
;
; TBL = SA * 256/90
;-----
; SA      RPM
;-----
L81B3:  FCB 0      ; 0 DEG  400
        FCB 3      ; 1      1200
        FCB 6      ; 2      2000
        FCB 14     ; 5      3200
        FCB 20     ; 7      4800

;-----
; START UP SA
;-----
L81B8:  FCB 4      ; INJECTS, START UP SA DECAY RATE

;-----
; START UP SA vs COOL
;
; TBL = SA * 256/90
;-----
; DEG      Deg c
;-----
L81B9:  FCB 14     ; 5.0    -40
        FCB 14     ; 5.0    -28
        FCB 14     ; 5.0    -16
        FCB 14     ; 5.0     -4
        FCB 14     ; 5.0     8
        FCB 14     ; 5.0    20
        FCB 14     ; 5.0    32
        FCB 14     ; 5.0    44
        FCB 14     ; 5.0    56
        FCB 14     ; 5.0    68

```

```
FCB 14      ; 5.0          80
FCB 0       ; 0.0          92
FCB 0       ; 0.0         104
FCB 0       ; 0.0         116
```

```
-----
; START UP SPARK DECAY DELAY vs START UP COOL
; TBL = INJECTS
; TBL = VAL/L81C7
-----
```

```

;
L81C7:  FDB 256      ; TABLE MULT
;
; INJECTS      Deg c
;-----
L81C9:  FCB 100     ; 25,600      -40
        FCB 100     ; 25,600      -28
        FCB 100     ; 25,600      -16
        FCB 100     ; 25,600       -4
        FCB 100     ; 25,600        8
        FCB 100     ; 25,600       20
        FCB 100     ; 25,600       32
        FCB 100     ; 25,600       44
        FCB 100     ; 25,600       56
        FCB 80      ; 25,600       68
        FCB 40      ; 25,600       80
        FCB 40      ; 25,600       92
        FCB 40      ; 25,600      104
        FCB 40      ; 25,600      116
;-----
```

```
-----
; START UP SA DECAY vs START UP COOLANT
;
;
; TBL =
;-----
```

```

; DEG SA      Deg c
;-----
L81D7:  FCB 128     ; 1.4         -40
        FCB 128     ; 1.4         -28
        FCB 128     ; 1.4         -16
        FCB 128     ; 1.4          -4
        FCB 128     ; 1.4          8
        FCB 128     ; 1.4         20
        FCB 128     ; 1.4         32
        FCB 128     ; 1.4         44
        FCB 128     ; 1.4         56
        FCB 128     ; 1.4         68
        FCB 128     ; 1.4         80
        FCB 128     ; 1.4         92
        FCB 128     ; 1.4        104
        FCB 255     ; 2.8        116
;-----
```

```
-----
; IDLE SA PARAMS
;
;-----
```

```
L81E5:  FCB 15      ; 15 SEC, IDLE SA INHIB DISABLE RICH TMR THRESH
;-----
```

```

L81E6:  FCB 64      ; 800 RPM, LO GAIN DESIRED RPM THRESH
L81E7:  FCB 96      ; 1.2 SEC, LO GAIN XISITION DELAY
L81E8:  FCB 192     ; 0.75, LO GAIN UNDER SPD MULT
L81E9:  FCB 128     ; 0.5, LO GAIN OVER SPD MULT
;
L81EA:  FCB 8       ; 100 RPM REQUIRED RPM OFFSET
;

```

```

;-----
; IDLE SA RETARD vs RPM
;
; TBL = SA * (256/90)
;-----

```

```

; S.A.      RPM
;-----
L81EB:  FCB 0      ; 0.0      0 RPM DIFF
        FCB 14     ; 5.0      50
        FCB 14     ; 5.0      100
        FCB 14     ; 5.0      150
        FCB 14     ; 5.0      200
;-----

```

```

;-----
; IDLE SA ADVANCE vs RPM ERROR
;
; TBL = 256/90
;-----

```

```

; S.A.      RPM ERROR
;-----
L81F0:  FCB 14     ; 5.0      0
        FCB 14     ; 5.0      50
        FCB 14     ; 5.0      100
        FCB 14     ; 5.0      150
        FCB 14     ; 5.0      200
;-----

```

```

;-----
; IDLE SA RETARD MULT vs MAP
; TBL = MULT * 256
;-----

```

```

; MULT      MAP Kpa
;-----
L81F5:  FCB 255    ; 0.996    20
        FCB 255    ; 0.996    30
        FCB 255    ; 0.996    40
        FCB 255    ; 0.996    50
;-----

```

```

;-----
; IDLE SA ADVANCE MULT vs MAP
;
; TBL = MULT * 256
;-----

```

```

; MULT      MAP
;-----
L81F9:  FCB 255    ; 0.996    20
        FCB 255    ; 0.996    30
        FCB 255    ; 0.996    40
        FCB 255    ; 0.996    50
;-----

```

```

;-----
; CLOSED THROT MAIN SA vs RPM
;
; TBL = SA DEG * (256/90)
;-----

```

```

;-----
; S.A.      RPM
;-----
L81FD:  FCB 57 ; 20      0
        FCB 57 ; 20      400
        FCB 57 ; 20      800
        FCB 71 ; 25     1200
        FCB 85 ; 30     1600
        FCB 85 ; 30     2000
        FCB 85 ; 30     2400
        FCB 85 ; 30     2800
        FCB 85 ; 30     3200
        FCB 85 ; 30     3600
        FCB 85 ; 30     4000
        FCB 85 ; 30     4400
        FCB 85 ; 30     4800
;-----

*****
* KNOCK PARAMS
*
*****

L820A:  FCB    52      ; 650 RPM, IF >= ENAB KNOCK
        ; ARG = RPM/12.5
        ;
L820B:  FCB    2      ; MPH, IF >= THEN ENAB KNOCK
        ;
L820C:  FCB    48      ; -3.9 DEG C, IF < DISABLE KNOCK
        ; ARG = (DEG C +40) * 256/192
        ;
L820D:  FCB    53      ; 0.24 DEG C, IF DIFF COOL SINCE START >
        ; ENAB KNOCK, ARG = DEG C * (256/192)

;-----
; KNOCK ATTACK RATE vs RPM
;
; TBL = ATK RATE (DEG/MS)/0.0225
;-----
; DEG/MS      RPM
; -----
L820E:  FCB 7      ; 0.158      400
        FCB 8      ; 0.180     1200
        FCB 13     ; 0.293     2000
        FCB 16     ; 0.360     3200
        FCB 16     ; 0.360     4800
;-----

;-----
; KNOCK PCT RECOVERY RATE vs RPM
;
; TBL = (DEG/MSEC)/.0225
;-----
; DEG/MS      RPM
; -----
L8213:  FCB 20     ; 0.450      400
        FCB 20     ; 0.450     1200
        FCB 26     ; 0.697     2000
        FCB 31     ; 0.698     3200
        FCB 31     ; 0.698     4800
;-----

;-----
; MAX KNOCK RETARD WHILE IN PWR ENRICH vs RPM
;
; TBL = VAL * 256/45

```

```

;-----
; SA      RPM
;-----
L8118:  FCB 46  ; 8.8      800
        FCB 51  ; 8.9      1600
        FCB 62  ; 10.8     2400
        FCB 87  ; 15.3     3200
        FCB 68  ; 11.9     4000
        FCB 34  ; 5.9      4800
        FCB 34  ; 5.9      5600
        FCB 34  ; 5.9      6400
;-----

```

```

;-----
; MAX KNOCK RETARD WHILE NOT IN PWR ENRICH vs MAP
;
; TBL = VAL * 256/45
;-----

```

```

; SA DEG  MAP Kpa
;-----
L8220:  FCB 34  ; 5.9      20
        FCB 34  ; 5.9      30
        FCB 34  ; 5.9      40
        FCB 34  ; 5.9      50
        FCB 57  ; 10.0     60
        FCB 114 ; 20.0     70
        FCB 114 ; 20.0     80
        FCB 114 ; 20.0     90
        FCB 114 ; 20.0    100
;-----

```

```

;-----
; LOW OCTAINE KNOCK RETARD MOD PRAMS
;-----

```

```

L8229:  FCB 160 ; LO OCTANE KNOCK RETARD MOD MAP ENABLE
        ; ARG = (Kpa-20) * (256/80), 70 Kpa
        ;
        FCB 2    ; LO OCTANE KNOCK RETARD DIFF MAP AMT
        ; ARG = Kpa * (256/80), 0.625 Kpa
        ;
        FCB 11   ; 1.93 DEG, LO OCTANE KNOCK RETARD
        FCB 50   ; BIN, OCTANE KNOCK RETARD INCREASE AMT
        FCB 2    ; BIN,VLO OCTANE KNOCK RETARD DECREASE AMT
        FCB 17   ; 6 DEG, LO OCTAINE KNOCK BASE RETARD

```

```

;-----
; Low Octain KNOCK mult vs RPM
;
; TBL = MULT * 256
;      Mult  MAP
;-----

```

```

; MULT  RPM
;-----
L822F:  FCB 255 ; 0.966     0 RPM
        FCB 255 ; 0.966     800
        FCB 255 ; 0.966    1600
        FCB 230 ; 0.898     2400
        FCB 205 ; 0.800     3200
        FCB 192 ; 0.750     4000
        FCB 192 ; 0.750     4800
        FCB 192 ; 0.750     5600
        FCB 192 ; 0.750     6400
;-----
;-----

```

; Low Octain KNOCK mult vs MAP

;

; TBL = MULT * 256

;

; Mult MAP

		; MULT	Kpa
		-----	-----
L8338:	FCB 0	; 0.000	20 Kpa
	FCB 0	; 0.000	25
	FCB 0	; 0.000	30
	FCB 128	; 0.500	35
	FCB 128	; 0.500	40
	FCB 128	; 0.500	45
	FCB 255	; 0.996	50
	FCB 255	; 0.996	55
	FCB 255	; 0.996	60
	FCB 255	; 0.996	65
	FCB 255	; 0.996	70
	FCB 255	; 0.996	75
	FCB 255	; 0.996	80
	FCB 255	; 0.996	85
	FCB 255	; 0.996	90
	FCB 255	; 0.996	95
	FCB 255	; 0.996	100

* END OF SPARK CALIB

* DIAGNOSTIC PARAMS

; ERROR MASK WORDS

;

; 1 = ENABLE ERROR RECOGNITION

; 2 = DISABLE

		-----	-----
L8249:	FCB \$F7	; 1111 0111 ERROR WD 1 MASK	
		;	
		; b0 = CODE 23 LO MAT	
		; b1 = CODE 22 LO TPS	
		; b2 = CODE 21 HI TPS	
		; b3 = CODE 16 -	
		;	
		; b4 = CODE 15 LO COOL	
		; b5 = CODE 14 HI COOL	
		; b6 = CODE 13 o2 SENSOR	
		; b7 = CODE 12 NO DRP	

		-----	-----
L824A:	FCB \$CE	; 1100 1110 ERROR WD 2 MASK	
		;	
		; b0 = CODE 35 -	
		; b1 = CODE 34 LO MAP	
		; b2 = CODE 33 HI MAP	
		; b3 = CODE 32 EGR DIAG	
		;	
		; b4 = CODE 31 -	
		; b5 = CODE 26 -	
		; b6 = CODE 25 HI MAT	
		; b7 = CODE 24 VSS SENSOR	

```

;-----
;
L824B:  FCB $7F      ; 0111 1111 ERROR WD 3 MASK
;
; b0 = CODE 51 EPROM
; b1 = CODE 46 VATS
; b2 = CODE 45 HI o2
; b3 = CODE 44 LO o2
;
; b4 = CODE 43 KNOCK
; b5 = CODE 42 EST MONITOR
; b6 = CODE 41 CYL SELECT
; b7 = CODE 36 -
;-----
;
L824C:  FCB $E2      ; 1110 0010 ERROR WD 4 MASK
;
; b0 = CODE 63 -
; b1 = CODE 62 OIL TEMP HI
; b2 = CODE 61 -
; b3 = CODE 56 -
;
; b4 = CODE 55 -
; b5 = CODE 54 - PUMP VOLTAGE
; b6 = CODE 53 - OVER VOLTAGE
; b7 = CODE 52 - OIL TEMP LO
;-----
;
L824D:  FCB 0         ; ERROR WD 5 MASK
;
; B5 = CODE 66 -
; B6 = CODE 65 -
;-----
;
L824E:  FCB 50        ; 50 SUCCESIVE PWR UP'A W/O ERR'S
;
L824F:  FCB 10        ; 1 SEC'S LOG FILTER CONST #1
L8250:  FCB 50        ; 5 SEC'S LOG FILTER CONST #2
L8251:  FCB 100       ; 10 SEC'S LOG FILTER CONST #3
L8252:  FCB 120       ; 12 SEC'S LOG FILTER CONST #4
;
L8253:  FCB 20        ; 2 SEC'S ELITE Light blb tst time
L8254:  FCB 20        ; 2 SEC, CARS LITE tst time
;
;-----
; ERROR 13
;-----
L8255:  FCB 146       ; IF COOL <= 69.5 C, (157F) DISABLE ERROR
; ARG = (DEG C + 40) * (256/192)

L8256:  FCB 60        ; 30 SEC, IF RUN TIME < DISABLE 13
L8257:  FCB 79        ; 0.035 VDC, IF o2 =< DIABLE
L8258:  FCB 124       ; 0.549 VDC, IF o2 > DIABLE 13
L8259:  FCB 13        ; 5% TPS, MIN FOR 13
L825A:  FCB 30        ; 15 SEC MIN COND'S PERIOD

;-----
; ERROR 14
;-----
L825B:  FCB 227       ; IF COOL < 227 DISABLE ERROR

;-----
; ERROR 14/15
;-----
L825C:  FCB 135       ; DEFAULT COOL <= 61C, (142F) IF ERROR 14 or 15
; ARG = (DEG C + 40) * (256/192)

```

```

L825D:  FDB 00      ; NOT USED
;-----

;-----
; ERROR 15
;-----

L825F:  FCB 250    ; IF A/D VAL 250 < = DISABLE ERR 15
;-----

;-----
; ERROR 21
;-----

L8260:  FCB 246    ; IF TPS A/D VAL > SET ERR 21A
L8261:  FCB 128    ; IF TPS A/D VAL < = DISABLE ERR 21A
L8262:  FCB 30     ; CONDITION MUST EXIST FOR 3 SEC TO QUALIFY
L8263:  FCB 15     ; IF AIR FLOW >= 15 gm/sec DISABLE ERR 21A
;-----

;-----
; ERROR 21/22, TPS
;-----

L8264:  FCB 48     ; 48 0.96V A/D IS DEFAULT TPS IF ERR 21
;-----

;-----
; ERROR 22
;-----

L8265:  FCB 12     ; IF TPS A/D > = 12 DISABLER ERR 22
;-----

;-----
; ERROR 23
;-----

L8266:  FCB 2      ; IF MAT < 2 ENABLE ERR 23
L8267:  FCB 120    ; 12 SEC QUALIFICATION TIME
;-----

;-----
; ERROR 23/25
;-----

L8268:  FDB 0240   ; 240 SEC, ENG RUN TIME <= DISABLE 23/25
L826A:  FCB 1      ; 1 MPH, IF > DO 25, ELSE 23
L826B:  FCB 29     ; DEG C MAT DEFAULT IF 25
;-----

;-----
; ERROR 24, Vss
;-----

L826C:  FCB 3      ; 3 MPH, IF > DISABLE 24
L826D:  FCB 40     ; 1000 RPM, IF <= DISABLE 24
L826E:  FCB 184    ; 4600 RPM, IF > DISABLE 24
L826F:  FCB 5      ; 1.9% TPS, IF >= DISABLE 24
L8270:  FCB 22     ; 22 Kpa, IF MAP > DISABLE 24
L8271:  FCB 3      ; 3 SEC, QUALIFICATION TIME
;-----

;-----
; ERROR 25, MAT
;-----

L8272:  FCB 243    ; 139 Deg c, IF MAT > ENABLE 25
L8273:  FCB 120    ; 12 SEC, QUALIFICATION TIME
;-----

;-----
; ERROR 32, EGR
;-----

L8274:  FCB 192    ; 75% DC, SKIP ERR 32 IF EGR < 75%
L8275:  FDB 2700   ; BIN 2700 TO QUALIFY FOR ERR
;
L8277:  FCB 32     ; Disable if ld < 90 Kpa
; ARG = 256 - (Kpa *(256/80))

```



```

L8278:  FCB 208      ; Disable if ld > 1 Kpa
          ;   ARG = 256 - (Kpa *(256/80))
          ;
L8279:  FCB 21       ; Disable if TPS < 7%
L827A:  FCB 77       ; Disable if TPS > 30%
L827B:  FCB 6        ; CLS LP INT WINDOW VALUE
L827C:  FCB 1        ; IF TPS % CHANGES 0.39% DISALLOW TEST
L827D:  FCB 25       ; RUN EGR INT TST 2.5 SEC'S
L827E:  FCB 5        ; 5 IS INT or VAC CHANGE FOR EGR FAULT
L827F:  FCB 30       ; TO ENABLE TST, Vss > 30 MPH
L8280:  FCB 1        ; Dec FAIL COUNTER BY 1
L8281:  FCB 50       ; DELAY 5 SEC
L8282:  FCB 3        ; FAIL COUNTER = 3
          ;
          ;-----
          ; ERROR 33
          ;-----
L8283:  FCB 218      ; PK/NEUT HI MAP/BARO RATIO THRESH, 0.85
          ;   ARG = RATIO * 256
          ;
L8284:  FCB 218      ; DRIVE HI MAP/BARO RATIO THRESH, 0.85
          ;   ARG = RATIO * 256
          ;
L8285:  FCB 5        ; TPS > 1.9% FOR ERR 33 TO BE SET
L8386:  FCB 40       ; THRESHOLD IS 500 RPM TI INCR CNT'R
L8287:  FCB 80       ; 1 SEC ERR TIME THRESH
          ;
          ;-----
          ; ERROR 34, MAF
          ;-----
L8288:  FCB 38       ; MAP HI/BARO LO RATIO THRESH
          ;   ARG = RATIO * 256
          ;
L8289:  FCB 96       ; 1200 RPM THRESHOLD
          ;
L828A:  FCB 26       ; IF RPM >= 1200 & 10% TPS
          ; ERR 34 CAN BE SET
L828B:  FCB 4        ; .05 SEC TIME REG FOR ERR
L828C:  FCB 216     ; DEFAULT MAP IF ERR 34, 51.6 Kpa
          ; ARG = (Kpa * 2.71) - 28.06
          ;
          ;-----
          ; ERROR 33/34 DEFAULT MAP CALC PARAM'S
          ;-----
L828D:  FCB 72       ; MAP DEFAULT COEF
          ;
          ;-----
          ; MAP DEFAULT BIAS vs RPM
          ;
          ; TBL = A/D BIN
          ;-----
L828E:  FCB 160      ;   0 RPM
L828F:  FCB 64       ;   800
L8290:  FCB 40       ;  1600
L8291:  FCB 24       ;  2400
L8292:  FCB 0        ;  3200
          ;
          ;-----
          ; ERROR 41, cyl select
          ;-----
L8293:  FCB 0        ; $00 = 8 CYL
          ; $18 = TBI
          ; $10 = 4 CYL
          ; $08 = 6 CYL
          ;
          ;-----

```

```

; ERROR 42, EST
;-----
L8294:  FCB 4      ; IF RPM < 100 THEN DISABLE ERR 42
L8295:  FCB 2      ; if est toggled, (PA1), >= 2
                ; prior to eng run, enable ERR 42
L8296:  FCB 64     ; 1048 MSEC PULSE WIDTH THRESH
                ; ARG = MSEC * (1000/16384)
L8297:  FCB 3      ;

;-----
; ERROR 43, KNOCK
;-----
L8298:  FCB 100    ; 10 SEC TIME REQUIRED
L8299:  FCB 181    ; 3.62V UPPER VOTLAGE THRESH
L829A:  FCB 72     ; 1.44V LOWER VOTLAGE THRESH

L829B:  FCB 1      ; Add 0.3 deg SA for test
L829C:  FCB 180    ; 95c, Min cool for for ERR 43B
L829D:  FCB 75     ; If MAP < 75 THEN DISABLE ERR 43B
L829E:  FCB 71     ; 24.9 Deg MAX S.A. for ERR 43B TST

;-----
; ERROR 44, o2 SENSOR R/L
;-----
L829F:  FCB 45     ; If o2 VOLTS >= 0.199 DISABLE ERR 44
L82A0:  FCB 50     ; 50 SEC; TIME REQUISITE

;-----
; ERROR 45, o2 LEAN
;-----
L82A1:  FCB 158    ; If o2 VOLTS >= 0.699 DISABLE ERR 45
L82A2:  FCB 50     ; 50 SEC; TIME REQUISITE
L82A3:  FCB 5      ; ENABLE ERR 45 IF TPS > 1.9%
L82A4:  FCB 0      ; ENABLE ERR 45 IF TPS < 0%

;-----
; ERROR 46, VATS
;-----
L82A5:  FCB 50     ; IF NO VATS IN 0 SEC SET ERR 46

;-----
; ERROR 52/62
;-----
L82A6:  FCB 1800   ; IF ENG RUN > 1800 SEC DISABLE ERR 52/62

;-----
; ERROR 52
;-----
L82A8:  FCB 245    ; If oil temp A/D val <= 245 disable ERR 52

;-----
; ERROR 54, BATTERY VOLTAGE
;-----
L82A9:  FCB 15     ; IF VOLTAGE LO FOR 1.5 SEC SET ERR
L82AA:  FCB 20     ; IF PUMP VOLTAGE < 2V SET ERR 54
L82AB:  FCB 20     ; IF PUMP VOLTAGE < 2V SET ERR 54
L82AC:  FCB 20     ; IF IGN VOLTAGE < 2V SKIP ERR 54 CHK

;-----
; ERROR 62 OIL TEMP
;-----
L82AD:  FCB      245 ; IF OIL < ___ DEG C DISABLE ERR 62
                ; ARG = ( DEG C-40) * (256/192)

```

* END OF DIAGNOSTICS

* EGR PARAMS

*

;

; EGR QUAL, TPS HYST PAIR

;

L82AE: FCB 8 ; ENABLE EGR IF TPS > 3.25%
; LOWER HYST, CK IF EGR CD NOT 0

;

L82AF: FCB 10 ; ENABLE EGR IF TPS > 3.9%
; UPPER HYST, CK IF EGR DC = 0

;

L82B0: FCB 30 ; IF MAT < 30 DEG C DISABLE EGR

L82B1: FCB 16 ; INC EGR ?? BY 16 EVERY 100 MSEC

* EGR DC TABLE

; EGR DC Vs PRM Vs VAC LOAD TABLE

;

; Dissassembly of ANHT

; 11-24-1991, 21:16:09

;

; TBL = %EGR * 2.56

ORG \$82B2

L82B2:

FCB 32 ; Min PRM

FCB 0 ; Min Load Value

FCB 9 ; lines /block

; 800 RPM PCT Kpa

FCB 0 ; 0.0 80

FCB 0 ; 0.0 70

FCB 0 ; 0.0 60

FCB 0 ; 0.0 50

FCB 0 ; 0.0 40

FCB 0 ; 0.0 30

FCB 0 ; 0.0 20

FCB 0 ; 0.0 10

FCB 0 ; 0.0 0

; 1200 RPM PCT Kpa

FCB 00 ; 0.0 80

FCB 128 ; 50.0 70

FCB 255 ; 99.6 60

FCB 255 ; 99.6 50

FCB 255 ; 99.6 40

FCB 255 ; 99.6 30

FCB 255 ; 99.6 20

FCB 255 ; 99.6 10

FCB 00 ; 0.0 0

; 1600 RPM PCT Kpa

FCB 00 ; 0.0 80

FCB 128 ; 50.0 70

```

FCB 255 ; 99.6 60
FCB 255 ; 99.6 50
FCB 255 ; 99.6 40
FCB 255 ; 99.6 30
FCB 255 ; 99.6 20
FCB 255 ; 99.6 10
FCB 00 ; 0.0 0

```

```

;-----
; 2000 RPM      PCT      Kpa
;-----
FCB 00 ; 0.0 80
FCB 128 ; 50.0 70
FCB 255 ; 99.6 60
FCB 255 ; 99.6 50
FCB 255 ; 99.6 40
FCB 255 ; 99.6 30
FCB 255 ; 99.6 20
FCB 255 ; 99.6 10
FCB 00 ; 0.0 0
;-----

```

```

;-----
; EGR D.C MULT (0 - 2) vs COOLANT
; tbl = 7 lines
;
; TBL = MULT * 128
;-----

```

```

L82D9: ;
; MULT      Deg c Cool
;-----
FCB 0 ; 0.0      8
FCB 0 ; 0.0      20
FCB 0 ; 0.0      32
FCB 0 ; 0.0      44
FCB 128 ; 1.0      56
FCB 128 ; 1.0      68
FCB 128 ; 1.0      80
;-----

```

```

*****
* END OF EGR CALIB
*****

```

```

*****
* CAN PURGE PARAMS
*****

```

```

L82E0: FCB 230 ; IF VAC > __ kpa DON'T CHANGE PURGE D.C.
; ARG = 256 - (Kpa * (256/80))
;
L82E1: FCB 1 ; 0.1 SEC BETWEEN PURGE D.C. UPDATES
;
L82E2: FCB 125 ; IF INTIGRATOR GT EQ 125 INRC D.C.
L82E3: FCB 95 ; IF INTIGRATOR LT EQ 90 DECR D.C.
;
L82E4: FCB 5 ; 1.9 PCT CCP DC.
L82E5: FCB 2 ; 0.8 PCT CCP DC.
L82E6: FCB 128 ; 50.0 PCT CCP DC.
;
L82E7: FCB 64 ; 20 MPH, SAVE PURGE IF MPH >
; ARG = MPH/(16/5)
;
L82E8: FCB 115 ; 115 BLK LRN MLT, SAVE PURGE IF >
L82E9: FCB 107 ; 107 BLK LRN MLT, DISABLE PURGE IF <=
L82EA: FCB 133 ; 60c, PURGE COOL ENAB THRESH, (140 F)

```

```

;-----
; PURGE ON to OFF PARAMS
;-----
L82EB:  FCB 6      ; 1.8 MPH, ENABLE IF >
          ; ARG = MPH*(16/5)
          ;
L82EC:  FCB 0      ; 0 KPA, ENABLE IF >
L82ED:  FCB 8      ; 3.1% TPS, ENABLE IF >
;-----

;-----
; PURGE OFF to ON PARAMS
;-----
L82EE:  FCB 3      ; 0.9 MPH, DISABLE IF <=
          ; ARG = MPH*(16/5)
          ;
L82EF:  FCB 0      ; 0 Kpa, DISABLE IF <=
L82F0:  FCB 6      ; 2.3% TPS, DISABLE IF <=
;-----

;-----
; PURGE DUTY CYC Vs AIR FLOW (MAF)
;   TBL = 9 LINES
;
; Table = D.C. * 2.56
;-----
L82F1:  FCB 8      ; 9 TABLE  VAL'S
          ;
          ; DC          GMS/SEC
          ; -----
          FCB 52    ; 20.3      0
          FCB 52    ; 20.3      4
          FCB 52    ; 20.3      8
          FCB 64    ; 25.0     12
          FCB 128   ; 50.0     16
          FCB 255   ; 99.6     20
          FCB 255   ; 99.6     24
          FCB 255   ; 99.6     28
          FCB 255   ; 99.6     32
;-----

*****
* END OF CAN PURGE PARAMS
*
*****

*****
* XMISSION PARAMS
*
* TYPE $8D ECM
*****
L82FB:  FCB      255      ; 0.996, Vss FILTER COEF

;-----
; GEAR DETERMINATION RATIOS
;-----
L82FC:  FCB 255      ; 1st GEAR RATIO
L82FD:  FCB 36       ; 4th GR MAX RATIO
L82FE:  FCB 31       ; 4th GR MIN RATIO
L82FF:  FCB 8        ; 0.8 SEC FOR 4th DETERMINE
;-----

;-----
; Computer Aided Ratio Selection

```

; MANUAL XMISSION

L8300: FCB 255 ; IF COOL GT 151c CK FOR CARS ACTIVE
;
L8301: FCB 0 ; RESET SPEED
L8302: FCB 0 ; ENABLE SPEED
L8303: FCB 0 ; DISABLE SPEED
;
L8304: FCB 0 ; RPM DISABLE
L8305: FCB 0 ; %TPS FOR DISABLE
;
L8306: FCB 0 ; Kpa LOW BARO DISABLE
L8307: FCB 0 ; Kpa LOW BARO FOR RE ENABLE

; PASS BY UNLOCK CALIB (disabled)

L8308: FCB 255 ; 99.6% TPS
;
L8309: FCB 28 ; 28 MPH
L830A: FCB 32 ; 32 MPH
;
L830B: FCB 255 ; 6375 RPM
;
L830C: FDB 65535 ; SEC. DELAY/10 BEFORE PASS BY LK UP
;
LD80E: FDB 0240 ; KEEP TCC LK'ED 24 SEC'S FOR PB
;
L8310: FCB 43 ; RETARD SPK IF IN Pass By NOISE

; TCC CALIB
; 700R4

L8311: FCB 6 ; UNLOCK TCC IF DIFF %TPS G.T. 2.3% IN 12,5 Msec
L8312: FCB 9 ; UNLOCK TCC IF %TPS G.T 3.5%
L8313: FCB 29 ; DIFF TPS UNLOCK TIME (12 SEC'S)
L8314: FCB 120 ; LOWER TEMP LIMIT, (50 C)
L8315: FCB 32 ; COAST RELEASE, TPS SW POINT, (32 MPH)
;
L8316: FCB 8 ; LO NPH COAST LOAD LIMIT, 3% TPS
L8317: FCB 0 ; HI NPH COAST LOAD LIMIT, 3% TPS
;
L8318: FCB 3 ; TPS HYST FOR
;
L8319: FCB 2 ; 0.2 SEC DELAY AFETER 4-3 DN SHIFT
L831A: FCB 0 ; 0 SEC DELAY BEFORE LOCK, LO GR AFTER COAST REL
L831B: FCB 0 ; 0 SEC DELAY BEFORE LOCK, HI GR AFTER COAST REL
L831C: FCB 0 ; 0 MPH LOCK DELAY BYPASS MPH THRESH
;
L831D: FCB 76 ; UNLOCK PREVENTION THRESH, (76 MPH)
L831E: FCB 0 ; OFFSET TO Vss WHEN 3RD GR REQ, (0 MPH)

; TCC LOCK/UN-LOCK XISITION Vss CALIBRATION
; (MPH/1)
; 700R4

; TCC UN-LOCK LO GEAR

L831F: FCB 40 ; 40 MPH, LOW MPH TEST
L8320: FCB 255 ; 255 MPH, HI MPH TEST

;-----
; TCC LOCK LO GEAR
;-----

L8321: FCB 38 ; 38 MPH, LOW TEST
L8322: FCB 255 ; 255 MPH HI TEST

;-----
;-----
; LOAD LIMIT vs MPH or RPM, 12 lines
; (LOW GEARS UPPER THRESHOLD)
;
; TCC LOCK/UN-LOCK vs MPH or RPM
; 700R4
; tbl = %full ld * 2.56
;-----

		; % LD	MPH	RPM
L8323:	FCB 64	; 25	12	400
	FCB 64	; 25	20	800
	FCB 64	; 25	28	1200
	FCB 64	; 25	36	1600
	FCB 97	; 38	44	2000
	FCB 123	; 48	52	2400
	FCB 138	; 54	60	2800
	FCB 143	; 56	68	3200
	FCB 148	; 58	76	3600
	FCB 148	; 58	84	4000
	FCB 148	; 58	92	4400
	FCB 148	; 58	100	4800

;-----
;-----
; LOAD LIMIT vs MPH or RPM, 12 lines
; (LOW GEARS LOWER THRESHOLD)
;
; TCC LOCK/UN-LOCK vs MPH or RPM
; 700R4
;tbl = %full ld * 2.56
;-----

		; % LD	MPH	RPM
L832F:	FCB 36	; 14	12	400
	FCB 36	; 14	20	800
	FCB 36	; 14	28	1200
	FCB 36	; 14	36	1600
	FCB 56	; 22	44	2000
	FCB 82	; 32	52	2400
	FCB 102	; 40	60	2800
	FCB 113	; 44	68	3200
	FCB 123	; 48	76	3600
	FCB 123	; 48	84	4000
	FCB 123	; 48	92	4400
	FCB 123	; 48	100	4800

;-----
;-----
; TCC LOCK/UN-LOCK XISITION Vss CALIBRATION
; (MPH/1)
; 700R4
;-----

;-----

; TCC UN-LOCK HI GEAR

;-----

L833B FCB 40 ; 40 MPH, LO MPH TSET

L833C FCB 255 ; 255 MPH, HI MPH TEST

;-----

; TCC LOCK HI GEAR

;-----

L833D FCB 38 ; 40 MPH, LO MPH TEST

L833E FCB 255 ; 255 MPH, HI MPH TEST

;-----

;-----

; LOAD LIMIT vs MPH or RPM, 12 lines

; (HI GEARS UPPER THRESHOLD)

;

; TCC LOCK/UN-LOCK vs MPH or RPM

; 700R4

;

; TBL = %FULL LD * 2.56

;-----

; % LD MPH RPM

;-----

L833F:	FCB 64	; 25	12	400
	FCB 64	; 25	20	800
	FCB 64	; 25	28	1200
	FCB 64	; 25	36	1600
	FCB 97	; 38	44	2000
	FCB 123	; 48	52	2400
	FCB 138	; 54	60	2800
	FCB 143	; 56	68	3200
	FCB 148	; 58	76	3600
	FCB 148	; 58	84	4000
	FCB 148	; 58	92	4400
	FCB 148	; 58	100	4800

;-----

;-----

; LOAD LIMIT vs MPH or RPM, 12 lines

; (HI GEARS LOWER THRESHOLD)

;

; TCC LOCK/UN-LOCK vs MPH or RPM

; 700R4

;

; TBL = %FULL LD * 2.56

;-----

; % LD MPH RPM

;-----

L834B:	FCB 36	; 14	12	400
	FCB 36	; 14	20	800
	FCB 36	; 14	28	1200
	FCB 36	; 14	36	1600
	FCB 56	; 22	44	2000
	FCB 82	; 32	44	2000
	FCB 102	; 40	52	2400
	FCB 113	; 44	60	2800
	FCB 123	; 48	68	3200
	FCB 123	; 48	76	3600
	FCB 123	; 48	84	4000
	FCB 123	; 48	92	4400
	FCB 120	; 47	100	4800

;-----

;-----

; LIGHT PARAMETERS

;


```

;-----
L8357:  FCB 47      ; ___ DEG C THERSH FOR ENG LAMP ENABLE
L8358:  FCB 0       ; 0 MPH THERSH FOR ENG LAMP ENABLE
L8359:  FCB 0       ; 0% TPS THERSH FOR ENG LAMP ON
L835A:  FCB 56      ; 1400 RPM THERSH FOR ENG LAMP ON
L835B:  FCB 208     ; 5200 RPM THERSH FOR ENG LAMP ON

```

```

;-----
; LIGHT ON TABLE
; TBL = % FULL LD * 2.56
;-----

```

```

; % LD      RPM
;-----
L835D:  FCB 13      ; 5%      800
        FCB 46      ; 18      1600
        FCB 118     ; 46      2400
        FCB 192     ; 75      3200
        FCB 192     ; 75      4000
        FCB 255     ; 100     4800
        FCB 255     ; 100     5600
;-----

```

```

L8364:  FCB 52      ; 1300 RPM, THRESHOLD FOR ENG LIGHT ON
L8365:  FCB 204     ; 5100 RPM, THRESHOLD FORCE ENG LIGHT ON

```

```

;-----
; LIGHT OFF TABLE
; TBL = % FULL LD * 2.56
;-----

```

```

; % LD      RPM
;-----
L8366:  FCB 18      ; 7%      800
        FCB 51      ; 20      1600
        FCB 123     ; 48      2400
        FCB 197     ; 77      3200
        FCB 197     ; 77      4000
        FCB 255     ; 100     4800
        FCB 255     ; 100     5600
;-----

```

```

;-----
; LIGHT ON DELAY MODIFER
; TABLE
; DELAY vs %TPS
;
; TABLE = (SEC*10)/L836D
;-----

```

```

L836D:  FCB 2       ; SCALER

; DELAY      TPS
;-----
L836E:  FCB 150     ; 30      0.0 %
        FCB 150     ; 30      12.5
        FCB 150     ; 30      25.0
        FCB 150     ; 30      37.5
        FCB 150     ; 30      50.0
;-----

```

```

*****
* XMISSION PARAMS
*****

```

```

*****
* FAN PARAM'S

```

```
*****
L8373:  FCB 225      ; 45 SEC A/C CLUTCH OF TIME FOR DISABLE
L8374:  FCB 35       ; 35 MPH FAN 1 ENABLE SPD
L8375:  FCB 50       ; 50 MPH FAN 2 ENABLE SPD
L8376:  FCB 55       ; 55 MPH FAN 2 DISABLE W/High/C PRESS
```

```
;-----
; FAN 1 TEMP'S
;-----
```

```
L8377:  FCB 193      ; FAN 1 DISABLE THRESH, 104c, (226f)
L8378:  FCB 197      ; FAN 1 ENABLE THRESH, 107c, (220f)
                ; ARG = (DEG C+40) * (256/192)
```

```
;-----
; FAN 2 TEMP'S
;-----
```

```
L8379:  FCB 197      ; FAN 2 DISABLE THRESH, 107c, (220f)
L837A:  FCB 204      ; FAN 2 ENABLE THRESH, 113c, (235f)
```

```
L837B:  FCB 10       ; 2 SEC FAN 1 REQ TURN ON DELAY
L837C:  FCB 100      ; 20 SEC FAN 1 DELAYED RUN ON AFTER OFF
L837D:  FCB 25       ; 5 SEC FAN 1 TO FAN 2 TURN ON DELAY
L837E:  FCB 75       ; 15 SEC FAN 2 DELAYED RUN ON AFTER REQ OFF
```

```
;-----
*****
* END OF FAN PARAM'S
*****
```

```
;-----
; A/C clutch params
;-----
```

```
L837F:  FCB 255      ; 99.6 %TPS DISABLE THRESH
L8380:  FCB 218      ; 124c COOLANT DISABLE THRESH
L8381:  FCB 255      ; 151c OIL DISABLE THRESH
L8382:  FCB 0        ; 0 DEC COIL HYST
L8383:  FCB 255      ; DISABLE AC IF MPH > 255
L8384:  FDB 180      ; DISABLE IF HI MPH AFTER 18.0 SEC
L8386:  FCB 30       ; HI MPH 30 SEC BEFORE RE-ENABLE
L8387:  FCB 200      ; IF RPM > 5000 DO NOT ENAGE A/C
```

```
;-----
; A/C IAC LEARNING GAIN PARAM'S
;-----
```

```
L8388:  FCB 20       ; 20 STEPS DEFAULT IF Non Vol MEM FAIL
L8389:  FCB 40       ; 40 STEPS MAX A/C LEARN
L838A:  FCB 10       ; 10 STEPS MIN A/C LREAN
L838B:  FCB 5        ; 5 STEPS MAX+ A/C LEARN EACH ON/OFF CYCLE
                ;
L838C:  FCB 40       ; 0.8 Msec MIN ERR DEAD TIME LEARN EN/DISABLE
L838D:  FCB 128      ; MULT PK/NEUT GAIN FOR LEARN
```

```
*****
* AIR MANAGEMENT
*****
```

```
L838E:  FCB 0        ; IF IN RWR ENRICH GT EQ 0 SEC'S, DIVERT AIR
L838F:  FCB 10       ; IF RPM GT 1 SEC DIVERT AIR
                ;
L8390:  FCB 170      ; IF FILT o2 GT 0.782V FOR TIME = LB392, DIVERT
L8391:  FCB 56       ; IF FILT o2 LT 0.248V FOR TIME = LB392, DIVERT
                ;
L8392:  FCB 0        ; R/L DIVERT TMR
```

```

;
L8393:  FCB 200      ; 20 SEC R/L DIVERT TIMER
;
L8394:  FCB 177      ; IF GT 4425 RPM & TIME GT LB392, DIVERT
L8395:  FCB 10       ; 1 SEC ENAB CAT AIR IF TIME GT L8399 SINCE NEG MAP
L8396:  FCB 20       ; 20 Kpa MAP DIVERT IF MAP LT
;
L8397:  FCB 60       ; 60 Kpa, DIVERT AIR IF MAP GT
L8398:  FCB 60       ; 60 MPH, DIVERT AIR IF MPH GT 60 & MAP L8397
;
L8399:  FCB 64       ; 64 Kpa ENAB CAT AIR IF OPN LP & MAP DROPED 64 KPA
; IN 100 msec LOOP
L839A:  FCB 73       ; IF COOL < 14.7C, DIVERT
L839B:  FCB 254      ; IF IN Opn Lp GT 25.4 SEC'S IF PRIOR Clsd Lp

```

* START OF FUEL PARAMS

*

* ANHT, TYPE \$8D

; CRANKING PARAMS

; CRANK FUEL DELIVERY DELAY vs MAT

;

; TBL = DELAY IN DRP'S

```

;
;          DEG C
;-----
L839C:  FCB 2      ; 2      -40 C
        FCB 2      ; 2      -26
        FCB 1      ; 1      -16
        FCB 0      ; 0      -4
        FCB 0      ; 0       8
        FCB 4      ; 4      20
        FCB 4      ; 4      32
        FCB 4      ; 4      44
        FCB 4      ; 4      56
        FCB 6      ; 6      68
        FCB 7      ; 7      80
        FCB 8      ; 8      92
        FCB 8      ; 8     104
        FCB 8      ; 8     116
        FCB 8      ; 8     128
        FCB 8      ; 8     140
        FCB 8      ; 8     152

```

; CRANK PW vs COOLANT TEMP

; ANHT TYPE \$8D ECM

;

; TBL = (usec * 65.536)/(SCALAR)

; SCALAR = 13108

```

L83AD:  FDB 13108   ; SCALAR FOR MAX PW
; CAL = SCALAR * 65.536 (200 MSEC)
;
;          usec          deg c Cool
;-----

```

```

L83AF:  FCB 205    ;          41001          -40

```

```

FCB 179 ; 35801 -28
FCB 96 ; 19200 -16
FCB 54 ; 10800 -4
FCB 38 ; 7600 8
FCB 20 ; 4000 20
FCB 15 ; 3000 32
FCB 10 ; 2000 44
FCB 10 ; 2000 56
FCB 10 ; 2000 68
FCB 8 ; 1600 80
FCB 8 ; 1600 92
FCB 8 ; 1600 104
FCB 8 ; 1600 116

```

; CRANK FUEL PW MULT vs TPS

;

; MULT * 64 (0-4)

L83BD: FCB 8 ; 9 LINE TABLE

;

; MULT % TPS

```

FCB 64 ; 1 0
FCB 64 ; 1 12.5
FCB 64 ; 1 25.0
FCB 64 ; 1 37.5
FCB 64 ; 1 50.0
FCB 64 ; 1 62.5
FCB 64 ; 1 75.0
FCB 0 ; 0 87.5
FCB 0 ; 0 100.0

```

; FOLLOWING THREE TABLES BY INDEX LK UP

; CRANK FUEL PW MULT vs REF PULSES 1-16

;

;

;

; MULT PULSE

```

L83C7: FCB 255 ; 1 1
FCB 255 ; 1 2
FCB 128 ; .5 3
FCB 0 ; 0 4
FCB 0 ; 0 5
FCB 0 ; 0 6
FCB 0 ; 0 7
FCB 0 ; 0 8
FCB 255 ; 1 9
FCB 255 ; 1 10
FCB 0 ; 0 11
FCB 0 ; 0 12
FCB 0 ; 0 13
FCB 0 ; 0 14
FCB 0 ; 0 15
FCB 0 ; 0 16

```

```

; EXTENTION OF ABOVE TABLE FOR REF'S GT 17
;
; TABLE = MULT * 256
;-----
;
;      ;      MULT      PULSE
;-----
L83D7: FCB 204      ; 0.80      1
      FCB 0        ; 0          2
      FCB 0        ; 0          3
      FCB 0        ; 0          4
      FCB 204     ; 0.80     5
      FCB 0        ; 0          6
      FCB 0        ; 0          7
      FCB 0        ; 0          8
;-----

;-----
; CRANK FUEL MULT vs BARO
;
;   TBL = MULT * 128
;-----
;      ; MULT      BARO Kpa
;-----
L83DF: FCB 103      ;          65
      FCB 111      ;          75
      FCB 118      ;          85
      FCB 125      ;          95
      FCB 128      ; 0.50     105
;-----

*****
*   AIR FUEL PARAMS
*
* ANHT, TYPE $8D
*****
L83E4: FCB 105      ; 41% TPS FOR TPS HI to TPS LOW
L83E5: FCB 48       ; 19% TPS IS LOW TPS
;
L83E6: FCB 1        ; 3.9% LOW TPS FILTER COEF
;
L83E7: FCB 16       ; 6.3% COOL FILTER COEF FOR INV COOL VALUE
;
L83E8: FCB 64       ; 25% OIL FILTER COEF

;-----
; FUEL OUT PARAMS
;   TBL = msec * 65.536
;   TBL = usec * 0.065536
;-----
L83E9: FCB 0000     ; 0 usec, MIN BPW
L83EB: FCB 0072     ; 1098 usec, SINGLE FIRE BPW DISABLE THRESH
L83ED: FCB 0056     ; 854 usec, SINGLE FIRE BPW ENABLE THRESH
L83EF: FCB 0056     ; 854 usec, MIN ASYNC PULSE

;-----
; o2 SENSOR PARAM'S
;-----
L83F1: FCB 240      ; 0.937, 12.5 MSEC FILTER COEFF
L83F2: FCB 5        ; 0.195, 100 MSEC FILTER COEF
L83F3: FCB 102     ; 0.398, 12.5 MSEC o2 FILTER COEF
;-----

;-----
; FUEL CUT OFF/ON PARAMS
;-----

```

```

L83F4:  FCB 255      ; 255 MPH FUEL CUT OFF
L83F5:  FDB 0098    ; 9080 RPM FUEL CUT OFF
;
L83F7:  FCB 254      ; 254 MPH FUEL RETURN
L83F9:  FDB 0103    ; 9544 RPM FUEL RETURN
;-----

```

```

;-----
; INJECTOR CORRECTION vs BATTERY
;
; ANHT TYPE 8D ECM
;
; (USE PUMP or BATTERY VDC, CK AFR OPTION WD 1,b6)
; ADD'S TO BPW (TWICE)
;
; TBL = usec * 0.032768
;-----

```

ORG \$02FA			
		usec	Vbatt
L83FA:	FCB 23	702	0.0
	FCB 23	702	1.6
	FCB 255	7782	3.2
	FCB 255	7782	4.8
	FCB 146	4456	6.4
	FCB 67	2045	8.0
	FCB 48	1465	9.6
	FCB 35	1068	11.2
	FCB 26	793	12.8
	FCB 18	549	14.4
	FCB 12	366	16.0
	FCB 9	275	17.6
	FCB 4	122	19.2
	FCB 0	0	20.8
	FCB 0	0	22.4
	FCB 0	0	24.0
	FCB 0	0	25.6

```

;-----
; LOW PULSE WIDTH INJECTOR OFFSET
; 15 line tbl
;
; TBL = MSEC * 65.536
; TBL = usec * 0.065536
;-----

```

			usec	PW Msec
L840B:	FCB 18	274		0.488
	FCB 10	153		0.732
	FCB 6	92		0.976
	FCB 5	76		1.220
	FCB 4	61		1.460
	FCB 3	46		1.708
	FCB 2	31		1.950
	FCB 1	15		2.197
	FCB 0	0		2.440
	FCB 0	0		2.685
	FCB 0	0		2.929
	FCB 0	0		3.170
	FCB 0	0		3.410
	FCB 0	0		3.660
	FCB 0	0		3.900

```

;-----
*****
* OPEN LOOP FUEL/AIR PARAMS
*
* ANHT, TYPE $8D
*
* Single Fire #/h = 20317.6/cal val
*****
L841A:  FDB 0445      ; STOCH AF RATIO 14.7
                        ; (ARG = 6553.6/RATIO)
                        ;
L841C:  FDB 0460      ; SEC/GRAM PROD OF INJ FLOW Rate
                        ; (0.359 SEC/GRAM, 2.786 g/Sec), 22#/HR
                        ; ARG = ((SEC/GRAM * 256) * 5)
                        ;
L841E:  FCB 4          ; 4 INJ'S, START UP INJ STARTUP DECAY RATE
                        ;
L841F:  FCB 0          ; 0% CHG OPN LP F/A % CHG AT IDLE
                        ; (%/2.56)
;-----

```

```

;-----
; OPEN LOOP FUEL/AIR % CHG vs MAP
;
; ANHT/4298 P/N 16134297
; ECM P/N 1227727
;
; TBL = PCT CHG * 2.56
;-----

```

		% CHG	MAP
L8420:	FCB 5	2	20
	FCB 5	2	25
	FCB 5	2	30
	FCB 5	2	35
	FCB 5	2	40
	FCB 8	3	45
	FCB 10	4	50
	FCB 13	5	55
	FCB 18	7	60
	FCB 18	7	65
	FCB 23	9	70
	FCB 26	10	75
	FCB 31	12	80
	FCB 36	14	85
	FCB 36	14	90
	FCB 36	14	95
	FCB 36	14	100

```

;-----
; START UP ENRICH vs COOLANT TEMP
; *** CHOKE ***
;
;
; Dissasemby of ANHT, LINES = 17
; 09-06-1994, 10:04:41
;
; TBL = 2.56 * %CHG
;-----
ORG $0431 :
; %CHG Deg c
;-----

```

```

L8431 FCB 243 ; 94.9 -40.0
L8432 FCB 192 ; 75.0 -28.0
L8433 FCB 154 ; 60.2 -16.0
L8434 FCB 154 ; 60.2 -4.0
L8435 FCB 128 ; 50.0 8.0
L8436 FCB 64 ; 25.0 20.0
L8437 FCB 64 ; 25.0 32.0
L8438 FCB 64 ; 25.0 44.0
L8439 FCB 49 ; 19.1 56.0
L843A FCB 38 ; 14.8 68.0
L843B FCB 38 ; 14.8 80.0
L843C FCB 13 ; 5.1 92.0
L843D FCB 13 ; 5.1 104.0
L843E FCB 13 ; 5.1 116.0
L843F FCB 2 ; 0.8 128.0
L8440 FCB 0 ; 0.0 140.0

```

; START UP ENRICHMENT DECAY DELAY vs COOLANT

;

; TBL = NUM INJECTS

;

FDB 0512 ; SCALE FACTOR

;

; INJECT Deg c

;

```

L8441: FCB 0 ; 0 -40
      FCB 0 ; 0 -28
      FCB 25 ; 25 -16
      FCB 25 ; 25 -4
      FCB 25 ; 25 8
      FCB 25 ; 25 20
      FCB 25 ; 25 32
      FCB 50 ; 50 44
      FCB 50 ; 50 56
      FCB 50 ; 50 68
      FCB 50 ; 50 80
      FCB 50 ; 50 92
      FCB 50 ; 50 104
      FCB 50 ; 50 116

```

; START UP ENRICHMENT DECAY DECAY AMOUNT vs COOLANT

;

; Dissassembly of ANHT, LINES = 14

; 09-06-1994, 09:33:44

;

; TBL = 6.5536 * %CHG

ORG \$044F ;

; %CHG

Deg c

;

```

L844F FCB 255 ; 38.9 -40
L8450 FCB 229 ; 34.9 -28
L8451 FCB 82 ; 12.5 -16
L8452 FCB 82 ; 12.5 -4
L8453 FCB 66 ; 10.1 8
L8454 FCB 118 ; 18.0 20
L8455 FCB 118 ; 18.0 32
L8456 FCB 66 ; 10.1 44

```



```

L8457   FCB  66      ;    10.1      56
L8458   FCB  66      ;    10.1      68
L8459   FCB  66      ;    10.1      80
L845A   FCB 164      ;    25.0      92
L845B   FCB 164      ;    25.0     104
L845C   FCB 164      ;    25.0     116

```

```

; OPEN LOOP AIR/FUEL PCT CHANGE vs COOLANT

```

```

; Dissassembly of ANHT LINES = 17

```

```

; 09-06-1994, 09:39:22

```

```

; TBL = (PCT CHG * 2.56) * SCALAR

```

```

    ORG $045D ;

```

```

L845D:  FCB 221 ; SCALAR,
          ; ARG = 256 - VALUE, (35d)

```

```

          ; %CHG          Deg c

```

```

-----
L845E   FCB 242 ;    94.5      -40
L845F   FCB 153 ;    59.8      -28
L8460   FCB  76 ;    29.7      -16
L8461   FCB  50 ;    19.5       -4
L8462   FCB  38 ;    14.8        8
L8463   FCB  25 ;     9.8       20
L8464   FCB  25 ;     9.8       32
L8465   FCB  25 ;     9.8       44
L8466   FCB  25 ;     9.8       56
L8467   FCB  25 ;     9.8       68
L8468   FCB  25 ;     9.8       80
L8469   FCB  25 ;     9.8       92
L846A   FCB  25 ;     9.8      104
L846B   FCB  25 ;     9.8      116
L846C   FCB  25 ;     9.8      128
L846D   FCB  25 ;     9.8      140
L846E   FCB  25 ;     9.8      152

```

```

; STARTUP ENRICH OFFSET vs MAP

```

```

; TBL = PCT CHG * 2.56

```

```

          ; PCT CHG          DEG C

```

```

-----
L846F:  FCB  0 ;     0          -40 DEG C
          FCB  0 ;     0          -28
          FCB  0 ;     0          -16
          FCB  0 ;     0           -4
          FCB  0 ;     0           8
          FCB  0 ;     0          20
          FCB  0 ;     0          32
          FCB  0 ;     0          44
          FCB  0 ;     0          56
          FCB  0 ;     0          68
          FCB 38 ;    15          80
          FCB 76 ;    30          92
          FCB 76 ;    30         104
          FCB 76 ;    30         116
          FCB 76 ;    30         128

```

FCB 76 ; 30 140
FCB 0 ; 0 152

* HIGHWAY FUEL PARAMS

*

* DISABLED AT L8484 **

L8480: FCB 65 ; DISABLE HI WAY IF Kpa GT 60 Kpa
L8481: FCB 167 ; DISABLE HI WAY IF COOL =< 82,3C, (185F)
;
L8482: FCB 10 ; LOCK OUT HIWAY FOR 10 SEC
L8483: FCB 60 ; 60 SEC MAX TIME IN HIWAY
L8484: FCB 255 ; DISABLE HIWAY IF SPD =< 255 MPH

; A/F RATIO vs MAP FOR HI WAY MODE

;

; TBL = AFR/10

	; AFR	PA
L8485: FCB 160	; 16:1	30
FCB 155	; 15.5	40
FCB 155	; 15.5	50
FCB 147	; 14.7	60

* CLOSED LOOP FUEL CONTROL

* ENABLE PARAMS

L8489: FCB 120 ; 50C, (122F) MIN COOL FOR CLS LP
L848A: FCB 100 ; IF COOL >= 35c, (95f) USE HOT CLS LP TIMER
L848B: FCB 73 ; IF COOL <= 14c, (59f) USE COLD CLS LP TIMER

;

; CLOSED LOOP TIMERS

; INDEXED LK UP

;

L848C: FCB 90 ; 45 Sec COLD CLS LP TIMER
L848D: FCB 90 ; 45 Sec WARM CLS LP TIMER
L848E: FCB 25 ; 12.5 Sec HOT CLS LP TIMER

;

; o2 QUALIFERS, (WINDOW)

;

L848F: FCB 158 ; IF o2 < 699 mvdc THEN o2 IS READY
L8490: FCB 45 ; IF o2 > 199 mvdc THEN o2 IS READY
;
L8491: FCB 50 ; IF o2 IN ABOVE WINDOW >= 10 Sec's then o2 is NOT READY

* CLOSED LOOP AUTH LIMITS

* (FOR INTIGRATOR VAL'S)

L8492: FCB 80 ; CLS LP MIN CORRECTION VALUE
L8493: FCB 180 ; CLS LP MAX MIN CORRECTION VALUE

* CLOSED LOOP FUEL CNT'L PARAMS & TABLES

*

L8494: FCB 6 ; 2.3% TPS MAX FOR IDLE
L8495: FCB 48 ; 15 MPH MAX SPD FOR IDLE
;
L8496: FCB 136 ; 600 mvdc, FAST o2 R/L THRES AT IDLE
L8497: FCB 136 ; 600 mvdc, UPPER ZERO ERROR o2 AT IDLE
L8498: FCB 128 ; 570 mvdc, LOWER ZERO ERROR o2 AT IDLE
;
L8499: FCB 0 ; MULT PORP STEP SIZE GAIN, ARG = VAL*256
L849A: FCB 0 ; 0 SEC'S ADD CORR TO INT DELAY AT IDLE
L849B: FCB 160 ; 0.750 MULT TO CLS LP GAIN FACTOR
;
L849C: FCB 12 ; 053 mvdc, DIFF TO MAKE R/L WINDOW (FAST o2)
L849D: FCB 23 ; 100 mvdc, DIFF R/L WHEN AIR DIVERTED
L849E: FCB 4 ; 17 mvdc, MIN ERROR TO MAKE INTIG CORR
;
L849F: FCB 224 ; 0.870 MULT GAIN FACTOR FOR POS ERRORS
L84A0: FCB 154 ; 0.600 COEF FOR o2 FILTER CONSTANT
;
L84A1: FCB 96 ; 32c, (90f) OFF SET FOR COOL INTIG GAIN

* FUEL PID PARAMS

* MY 90 L98 Y & F CAR TYPE \$8D CALIB.

; UPPER ZERO ERROR REF FOR SLOW o2 R/L vs MAP
;
;
;
; TBL = o2 VOLTS * 266

	ORG L84A2	; SLOW o2	
		; ERROR	Kpa MAP
		-----	-----
L84A2:	FCB 140	; 0.619	20
	FCB 148	; 0.655	30
	FCB 152	; 0.672	40
	FCB 152	; 0.672	50
	FCB 148	; 0.655	60
	FCB 144	; 0.637	70
	FCB 132	; 0.584	80
	FCB 128	; 0.566	90
	FCB 125	; 0.553	100

; FAST o2 R/L THRESHOLD vs MAP
;
;
;
; TBL = o2 VOLTS * 266

	ORG L84AB	; mvdc	
		; o2	Kpa MAP
		-----	-----
L84AB:	FCB 120	; 531	20
	FCB 128	; 566	30
	FCB 132	; 584	40
	FCB 132	; 584	50
	FCB 128	; 481	60

```
FCB 124      ; 466      70
FCB 112      ; 421      80
FCB 109      ; 411      90
FCB 106      ; 469      100
```

```
; INTEGRATOR DELAY vs AIR FLOW
```

```
;
```

```
;
```

```
;
```

```
; TBL = SEC'S * 80
```

```
                ; sec's      AIR FLOW
                ;-----
L84B4:  FCB 130  ; 1.624      0
        FCB 138  ; 1.725      8
        FCB 142  ; 1.775     16
        FCB 142  ; 1.775     24
        FCB 138  ; 1.725     32
        FCB 134  ; 1.675     40
        FCB 122  ; 1.525     48
        FCB 120  ; 1.500     56
        FCB 118  ; 1.475     64
```

```
; SLOW o2 FILTER COEF vs AIR FLOW
```

```
;
```

```
;
```

```
;
```

```
; TBL = Sec's * 80
```

```
                ; Sec's      AIR FLOW
                ;-----
L84BD:  FCB 48   ; 0.600      0
        FCB 44   ; 0.550      8
        FCB 24   ; 0.300     16
        FCB 22   ; 0.275     24
        FCB 20   ; 0.250     32
        FCB 18   ; 0.225     40
        FCB 16   ; 0.200     48
        FCB 14   ; 0.175     56
        FCB 12   ; 0.150     64
```

```
; SLOW o2 FILTER COEF vs FLOW ??
```

```
;
```

```
; TBL = MULT * 256
```

```
                ; MULT      FLOW
                ;-----
L84C6:  FCB 16   ; 0.063      0
        FCB 24   ; 0.094      8
        FCB 32   ; 0.125     16
        FCB 36   ; 0.141     24
        FCB 40   ; 0.156     32
        FCB 44   ; 0.172     40
        FCB 44   ; 0.172     48
        FCB 44   ; 0.172     56
        FCB 44   ; 0.172     64
```

;-----

;-----

; PORPORTIONAL STEP SIZE GAIN vs FLOW?

;

; (0 - 1) * 256

;-----

		; STEP	FLOW
L84CF:	FCB 255	; 99.6	0
	FCB 255	; 99.6	8
	FCB 255	; 99.6	16
	FCB 255	; 99.6	24
	FCB 255	; 99.6	32
	FCB 255	; 99.6	40
	FCB 255	; 99.6	48
	FCB 255	; 99.6	56
	FCB 255	; 99.6	64

;-----

;-----

; PORPORTIONAL STEP SIZE GAIN vs RPM

;

; TBL = SEC * 80

;-----

		; GAIN	RPM
L84D8:	FCB 20	; 0.25	400 RPM
	FCB 12	; 0.15	800
	FCB 8	; 0.10	1200
	FCB 4	; 0.05	1600
	FCB 3	; 0.04	2000
	FCB 2	; 0.03	2400

;-----

;-----

; PORPORTIONAL STEP SIZE vs ERROR

;

; TBL = COUNTS

;-----

		; BIN	ERROR
L84DE:	FCB 4	; 4	0
	FCB 4	; 4	8
	FCB 4	; 4	16
	FCB 4	; 4	24
	FCB 4	; 4	32
	FCB 4	; 4	40
	FCB 4	; 4	48
	FCB 4	; 4	56
	FCB 6	; 6	64
	FCB 8	; 8	72
	FCB 10	; 10	80
	FCB 16	; 16	88

;-----

;-----

; PORPORTIONAL STEP DURATION vs ERROR

;

; TBL = SEC * 80

;-----

		; BIN	ERROR
L84EA:	FCB 4	; 0.05	0
	FCB 4	; 0.05	8

```

FCB 4      ; 0.05      16
FCB 4      ; 0.05      24
FCB 4      ; 0.05      32
FCB 8      ; 0.10      40
FCB 12     ; 0.15      48
FCB 24     ; 0.30      56
FCB 32     ; 0.40      64
RCB 64     ; 0.80      72
FCB 128    ; 1.60      80
FCB 240    ; 3.00      88

```

```

; R/L OFFSET vs COOLANT
; 14 LINES

```

```

; TBL = COUNTS

```

```

; BIN      Deg c
;-----
L84F6: FCB 0      ; 0      -40
      FCB 0      ; 0      -28
      FCB 0      ; 0      -16
      FCB 0      ; 0      -4
      FCB 1      ; 1       8
      FCB 16     ; 16     20
      FCB 16     ; 16     32
      FCB 16     ; 16     44
      FCB 16     ; 16     56
      FCB 16     ; 16     68
      FCB 0      ; 0      80
      FCB 0      ; 0      92
      FCB 0      ; 0     104
      FCB 0      ; 0     116

```

```

; INTEGRATOR DELAY MULT vs ERROR

```

```

; TBL = MULT VAL

```

```

;          ERROR
;-----
L8504: FCB 255    ;          0
      FCB 255    ;          8
      FCB 255    ;         16
      FCB 255    ;         24
      FCB 128    ;         32
      FCB 64     ;         40
      FCB 64     ;         48
      FCB 64     ;         56
      FCB 64     ;         64
      FCB 64     ;         72
      FCB 64     ;         80
      FCB 64     ;         88

```

* BLOCK LEARN MEM CNT'L PARAM'S

* ANHT 5.71 V8, \$8D ECM

_____ | _____ | _____ | _____

		12	13	14	15	HI MAP 45 Kpa
		8	9	10	11	MID MAP 36 Kpa
		4	5	6	7	LO MAP 28 Kpa
		0	1	2	3	
700	LO RPM					
1200	MID RPM					
2000	HI RPM					

```

L8510:  FCB    28      ; 700 RPM, LOW BLM CELL EDGE
L8511:  FCB    48      ; 1200 RPM, MID BLM CELL EDGE
L8512:  FCB    80      ; 2000 RPM, HI BLM CELL EDGE
;
L8513:  FCB    25      ; 28 kpa, LOW BLM CELL EDGE
L8514:  FCB    50      ; 36 kpa, MID BLM CELL EDGE
L8515:  FCB    80      ; 45 kpa, HI BLM CELL EDGE
;
L8516:  FCB    4       ; 100 RPM, EDGE TO EDGE HYSYERESIS
;
L8517:  FCB    2       ; 2 kPA, EDGE TO EDGE HYSYERESIS
;
L8518:  FCB   120      ; If Cool lt eq 50c, disable BLM update
L8519:  FCB   240      ; If Cool lt eq 140c, disable BLM update
;
L851A:  FCB    0       ; IF MAP L.T. 0 then disable BLM update
;
L851B:  FCB    28      ; 400 msec, BLM update rate, (Sec * 20)+1
L851C:  FCB    2       ; BLM update amount
;
L851D:  FCB   160      ; Max BLM value
L851E:  FCB   108      ; Min BLM value
;
L851F:  FCB    5       ; If INIT (Clsd Lp) GT 128+5, enable BLM update
L8520:  FCB    5       ; If INIT (Clsd Lp) LT 128-5, enable BLM update
;
L8521:  FCB    0       ; DISABLE IDLE CELL LRN IF CCP DC GT 0
L8522:  FCB    51      ; FORCE IDLE CELL LRN IF CCP DC LT 51

```

```

*****
* STAY ALIVE MEM PARAM'S, (SAM Cells)
* ANHT 5.71 V8, $8D ECM
*****

```

```

L8523:  FCB  118      ; SAM CELL 0, LO LIMIT, (BLM INIT VAL'S)
L8524:  FCB  150      ; SAM CELL 0, HI LIMIT
;
L8525:  FCB  118      ; LO LIMIT FOR OTHER CELLS)
L8526:  FCB  150      ; HI LIMIT FOR OTHER CELLS)
;
L8527:  FCB    4       ; IDLE CELL NUMBER, (A CELL)
L8528:  FCB   10      ; NON IDLE CELL NUMBER, (B CELL)
;
L8529:  FCB  150      ; STOP CELL 0 UPDATE AFTER 30 SEC'S
;
L852A:  FCB  171      ; IF Cool LT 88c, SKIP SAM UPDATE
L852B:  FCB  192      ; IF Cool GT 104c, SKIP SAM UPDATE

```

```

;
L852C:  FCB 3      ; 0.0117, SAM A FILTER COEF
L852D:  FCB 3      ; 0.0117, SAM B FILTER COEF

```

```

*****
* DIFF TPS ACCEL ENRICH TABLES & PARAMS
* ANHT 5.7l V8, $8D ECM
*****

```

```

L852E:  FCB 3      ; 13.5% TPS MIN FOR ACCEL ENRICH PROP TO BASE PW
L852F:  FCB 1      ; 1 INJ IN FUEL LIMITING
L8530:  FCB 12     ; ACCEL ENR LASTS 12 REF PULSES
L8531:  FCB 128    ; 50% TPD DIFF FOR V6 ACELL ENR CALC
;
L8532:  FDB 454    ; 6.92 MSEC MAX ACELL ENR PW LIMIT
;
L8534:  FCB 32     ; 0.125 MULT FOR 12.5 Msec TPS Acell Enr
L8535:  FCB 32     ; 0.125 MULT FOR DIFF TPS * 32
;
L8536:  FCB 1      ; USE 4-7 OF TBL L8563 FOR 1 REF PULSE FROM AE or DE
L8537:  FCB 3      ;
L8538:  FCB 8      ;
;
L8539   FCB 3      ;
L853A:  FCB 10     ;

```

```

;-----
; ACELL ENRICH FACTOR PER BPW (0 - 4)
; (PUMP SHOT)
; TBL = FACTOR * 64
;
; ANHT 5.7l V8, $8D ECM
;-----

```

		;	FACTOR	PULSE
L853B:	FCB 255	;	3.984	1
	FCB 32	;	0.500	2
	FCB 32	;	0.500	3
	FCB 32	;	0.500	4
	FCB 32	;	0.500	5
	FCB 32	;	0.500	6
	FCB 32	;	0.500	7
	FCB 32	;	0.500	8

```

;-----
; ASYNC FACTOR vs DIFF TPS
; (PUMP SHOT)
;
; 03-12-1996 Dissasemby of ANHT Lines= 17
;
; TBL = 128 * MULT
;-----

```

	ORG	\$0543	;	MULT	%TPS
L8543	FCB	128	;	1.000	0.0
L8544	FCB	128	;	1.000	3.1
L8545	FCB	128	;	1.000	6.3
L8546	FCB	80	;	0.625	9.4
L8547	FCB	48	;	0.375	12.5
L8548	FCB	48	;	0.375	15.6
L8549	FCB	48	;	0.375	18.8
L854A	FCB	48	;	0.375	21.9
L854B	FCB	32	;	0.250	25.0
L854C	FCB	32	;	0.250	28.1


```

L854D  FCB 32      ; 0.250      31.3
L854E  FCB 32      ; 0.250      34.4
L854F  FCB 32      ; 0.250      37.5
L8550  FCB 32      ; 0.250      40.6
L8551  FCB 32      ; 0.250      43.8
L8552  FCB 32      ; 0.250      46.9
L8553  FCB 32      ; 0.250      50.0

```

; ASYNC FACTOR vs COOLANT

;

; TBL = FACTOR * 128

```

                ; FACTOR      DEG C
                ;-----
L8554:  FCB 160    ; 1.250      -40
        FCB 160    ; 1.250      -28
        FCB 160    ; 1.250      -16
        FCB 160    ; 1.250       -4
        FCB 160    ; 1.250       8
        FCB 144    ; 1.125      20
        FCB 112    ; 0.875      32
        FCB 96     ; 0.750      44
        FCB 80     ; 0.625      56
        FCB 80     ; 0.625      68
        FCB 64     ; 0.500      80
        FCB 56     ; 0.478      92
        FCB 48     ; 0.375     104
        FCB 48     ; 0.375     116
        FCB 48     ; 0.375     128

```

; FUEL LIMIT FACTOR vs COOLANT

;

; MSB ARE DRP TO L8536

; LSB ARE DRP G.T. or E.Q. TO L8536

; (INDEX LK UP)

;

; TBL = FACTOR *

```

                ; FACTOR      COOL DEG c
                ;-----
L8563:  FCB 221    ;          -40
        FCB 221    ;          -16
        FCB 221    ;           8
        FCB 204    ;          32
        FCB 170    ;          56
        FCB 170    ;          80
        FCB 170    ;         104
        FCB 170    ;         128

```

; TPS FILTER COEF vs RPM

; (Make Filtered TPS for Delta TPS Acell Enrich)

;

; TBL = COEF * 256

```

                ; COEF      RPM
                ;-----
L856B:  FCB 16     ; 0.063      000

```

```

FCB 16      ; 0.063      400
FCB 16      ; 0.063      800
FCB 16      ; 0.063     1200
FCB 16      ; 0.063     1600
FCB 16      ; 0.063     2000
FCB 16      ; 0.063     2400
FCB 16      ; 0.063     2800
FCB 16      ; 0.063     3200
FCB 16      ; 0.063     3600
FCB 16      ; 0.063     4000
FCB 16      ; 0.063     4400
FCB 16      ; 0.063     4800
FCB 16      ; 0.063     5200
FCB 16      ; 0.063     5600
FCB 16      ; 0.063     6000
FCB 16      ; 0.063     6400

```

```

*****
* DIFF MAP ACCEL ENRICH TABLES & PARAM'S
* ANHT, TYPE 8D ECM
*****

```

```

L857C: FCB      6 ; 0.023, FILTER COEF, ACELL ENRICH FILT MAP TERM
L857D: FCB     10 ; FILTER COEF, ACELL ENRICH FILT RPM TERM
      ;
L857E: FCB      3 ; 75 RPM, NEG DECELL RPM THRESH FOR USING L857F
L857F: FCB    128 ; 50%, L859A THRESH MULT ON DECEL RPM

```

```

; MAP ACEL ENRICH FACTOR vs DIFF MAP
;
; DIFF MAP ABOVE MIN

```

```

      ; MULT      KPA diff
      ;-----
L8580: FCB 24    ; 9.375      0
      FCB 32    ;           10
      FCB 36    ;           20
      FCB 40    ;           30
      FCB 51    ;           40
      FCB 64    ; 25.000     50

```

```

;-----
; MAP Acell Enrich COOLANT MULT
;
; VAL = MULT

```

```

L8586: FCB 8      ; 9 LINE TBL
      ;
      ;           Deg c
      ;-----
      FCB 55    ;           -40
      FCB 55    ;           -16
      FCB 55    ;            8
      FCB 55    ;           32
      FCB 32    ;           56
      FCB 26    ;           80
      FCB 19    ;          104
      FCB 19    ;          128
      FCB 19    ;          152

```

; Acell Enrich COOLANT MULT vs COOLANT
;
; VAL = MULT * 128

FCB	MULT	Deg c
FCB 8	9	LINE TBL
FCB 55	0.429	-40
FCB 55	0.429	-16
FCB 55	0.429	8
FCB 55	0.429	32
FCB 32	0.250	56
FCB 26	0.203	80
FCB 19	0.148	104
FCB 19	0.148	128
FCB 19	0.148	152

; Acell Enrich DECAY FACTOR vs COOLANT
;
; VAL = %CHG PER INJECT * 2.56

L8590:

FCB	%CHG	Deg c
FCB 8	9	LINE TBL
FCB 102	39.8	-40
FCB 102	39.8	-16
FCB 102	39.8	8
FCB 102	39.8	32
FCB 102	39.8	56
FCB 102	39.8	80
FCB 128	50.0	104
FCB 128	50.0	128
FCB 128	50.0	152

; DIFF MAP for Acell Enrich Enable
;
; VAL = DIFF MAP * (16/5)

L859A:

FCB	Kpa	%TPS
FCB 10	3.13	00.0
FCB 10	3.13	12.5
FCB 10	3.13	25.0
FCB 10	3.13	37.5
FCB 10	3.13	50.0

; MAP Acell Ebr FACTOR MULT vs TPS LOAD
;
; VAL = FACTOR * 64

FACTOR	TPS %
--------	-------

```

L859F:  FCB 64      ; 1.00      00.0
        FCB 64      ; 1.00      12.5
        FCB 64      ; 1.00      25.0
        FCB 64      ; 1.00      37.6
        FCB 64      ; 1.00      50.0

```

* A/C ON to OFF ENRICH PARAMM'S

```

L85A4:  FDB      128      ; 1.95 SEC A/C ON > OFF XISSITION PW

```

* DECELL ENLEAN PARAM'S

```

L85A6:  FCB 1      ; 0.4% TPS
L85A7:  FCB 6      ; DIFF A/D MAP VALUE

L85A8:  FCB 6      ; 0.023 COEF FOR MAP FILTER
L85A9:  FCB 255    ; 0.50 DECEL MULT IF TCC LOCKED
L85AA:  FCB 7      ; 0.027 COEF 12.5 MSEC FILTER

L85AB:  FCB 0      ; ENABLE DECELL INT IF < or = 0 Kpa
L85AC:  FCB 36     ; ENABLE DECELL INT IF G.T. 900 RPM
L85AD:  FCB 16     ; 6.25% TPS DECEL ENLEAN THT FOLLOWER

```

; DIFF ENLEAN FACTOR N vs NUM OF INJ'S IN
; DECELL LEAN

; (INDEXED LK UP)

; TBL = FACTOR * 256

	;	FACTOR	INJ'S
L85AE:	FCB 0	;	0
	FCB 16	; 0.063	1
	FCB 16	; 0.063	2
	FCB 16	; 0.063	3
	FCB 16	; 0.063	4
	FCB 16	; 0.063	5
	FCB 16	; 0.063	6
	FCB 16	; 0.063	7
	FCB 16	; 0.063	8
	FCB 16	; 0.063	9
	FCB 16	; 0.063	10
	FCB 16	; 0.063	11
	FCB 16	; 0.063	12
	FCB 16	; 0.063	13
	FCB 16	; 0.063	14
	FCB 16	; 0.063	15
	FCB 8	; 0.031	16

; DIFF ENLEAN TERM MULT Vs Diff MAP, (A/D)

; TBL = A/D MAP * 256

	;	FACTOR	A/D
L85BF:	FCB 16	; 0.625	0
	FCB 16	; 0.625	16

```

FCB 24      ; 0.094      32
FCB 32      ; 0.125      48
FCB 64      ; 0.250      64
FCB 64      ; 0.250      80
FCB 64      ; 0.250      96
FCB 64      ; 0.250     112
FCB 64      ; 0.250     128
FCB 64      ; 0.250     144
FCB 64      ; 0.250     160
FCB 64      ; 0.250     176
FCB 64      ; 0.250     192
FCB 64      ; 0.250     208
FCB 64      ; 0.250     224
FCB 64      ; 0.250     240
FCB 64      ; 0.250     256

```

```

;-----
; Diff enlean term mult vs Cool
;
; TBL = MULT * 64
;-----

```

```

; MULT      Deg c
;-----
L85D0: FCB 64      ; 1.00      -40
      FCB 64      ; 1.00      -28
      FCB 64      ; 1.00      -16
      FCB 64      ; 1.00       -4
      FCB 64      ; 1.00       8
      FCB 32      ; 0.50      20
      FCB 32      ; 0.50      32
      FCB 32      ; 0.50      44
      FCB 48      ; 0.75      56
      FCB 48      ; 0.75      68
      FCB 32      ; 0.50      80
      FCB 32      ; 0.50      92
      FCB 32      ; 0.50     104
      FCB 32      ; 0.50     116
      FCB 32      ; 0.50     128
      FCB 32      ; 0.50     140
      FCB 32      ; 0.50     152 c
;-----

```

* DECELL FUEL C/O PARAMS

```

L85E1: FCB 56      ; IF RPM > 1400, ENABLE C/O (upper hyst)
L85E2: FCB 44      ; IF RPM > 1100 STAY IN C/O, (low hyst)

L85E3: FCB 20      ; IF Kpa < 20, ENABLE C/O (low hyst)
L85E4: FCB 28      ; IF Kpa < 28, STAY IN C/O (upper Hyst)

L85E5: FCB 4       ; 50 RPM MAX INCR TO REMAIN IN C/O, (12.5 INTERVAL)
L85E6: FCB 7       ; 87.5 RPM FOR STALL SAVER

L85E7: FCB 80      ; 0.25 SEC C/O REQUISITE TIME FOR ENABLE

L85E8: FCB 0       ; DEG C FOR DISABLE OF C/O
L85E9: FCB 15      ; 15 MPH MIN SPD FOR C/O

L85EA: FCB 0       ; %TPS C/O STALL SV'R
L85EB: FCB 30      ; 0.38 SEC STALL SV'R TPS FOLLOW TIME

L85EC: FDB 128     ; DECEL FUEL C/O STALL SVR

```

```

; ACCEL ENRICH PW = 1.953 Msec
L85EE:  FCB      1      ; PULSE CNT TIMER

L85EF:  FDB     160     ; ___ SEC MIN BETWEEN CONSEC DECEL C/O's
L8FF1:  FCB      5      ; 1.95% TPS DECEL FUEL C/O DEFAULT
L85F2:  FCB      2      ; SA BLEND VAL IF IN DECEL FUEL C/O

L85F3:  FCB      2      ; 2 CNT'S ADDED TO SA BLEND/DRP
L85F4:  FCB      3      ; 1.17% TPS HYST TO TABLE L85F5
;-----

```

```

;-----
; DECELL FUEL CUT OFF
; TPS THRESH vs RPM
;
; VAL = %TPS * 2.56
;-----

```

			%TPS	RPM
L8FF5:	FCB	3	1.17	0
	FCB	3	1.17	400
	FCB	3	1.17	800
	FCB	3	1.17	1200
	FCB	3	1.17	1600
	FCB	5	1.95	2000
	FCB	8	3.12	2400
	FCB	13	5.07	2800
	FCB	13	5.07	3200
	FCB	13	5.07	3600

```

*****
* PWR ENRICH PARAMS
* ANHT TYPE $8D
*****

```

```

L85FF:  FCB 230      ; 8.1 Kpa VAC to ENABLE PWR ENR
; ARG = 256 - (Kpa * (256/80))

L8600:  FCB 10       ; 3.1 Kpa, VAC HYST FOR PWR ENRICH Kpa
; ARG = Kpa * (256/80)

L8601:  FCB 16       ; 6.25% TPS HYST, PE off if TPS LT (L8603 - L8601)

L8602:  FCB 209      ; USE TBL L8608 IF COOL = or GT (116c), (244F)

```

```

;-----
; PWR ENRICH TPS THRESHOLD vs RPM
; FOR NORMAL COOLANT
; *** WOT ***
;
; TBL = TPS% * 2.56
;-----

```

			TPS%	RPM
L8603:	FCB	128	50	400
	FCB	128	50	1200
	FCB	128	50	2000
	FCB	154	60	3200
	FCB	180	70	4800

```

;-----
; PWR ENRICH TPS THRESHOLD vs RPM

```

```

; FOR HIGH COOLANT
; *** WOT ***
;
; TBL = TPS% * 2.56
;-----
; %TPS      RPM
;-----
L8608:  FCB 128      ; 50.0      400
        FCB 128      ; 50.0      1200
        FCB 38       ; 14.8      2000
        FCB 38       ; 14.8      3200
        FCB 38       ; 14.8      3800
;-----

```

```

*****
* WOT AFR = 6553.6/((TBL L8608 + TBL L860D + 128) * 445)/256)
*   or
* WOT LAMBDA = (LAMBDA C + LAMBDA W)
*   or
* WOT AFR = 14.7/(LAMBDA C + LAMBDA W)
*****

```

```

;-----
; PWR ENRICH F/A %CHG vs COOLANT
; *** WOT ***
;
; ANHT, TYPE $8D
;
; TBL = (LAMBDA * 256) + 128
;-----

```

```

L860D:  FCB 8          ; TABLE LENGTH +1
        ;
        ; LAMBDA C    DEG c
        ;-----
        FCB 140       ; 0.546      -40
        FCB 125       ; 0.488      -16
        FCB 112       ; 0.438       8
        FCB 86        ; 0.336      32
        FCB 58        ; 0.227      56
        FCB 58        ; 0.227      80
        FCB 58        ; 0.227     104
        FCB 58        ; 0.227     128
        FCB 58        ; 0.227     152
;-----

```

```

;-----
; POWER ENRICH FUEL/AIR % CHG vs RPM
; *** WOT ***
;
; ANHT, TYPE $8D
;
; TBL = LAMBDA+128/256
; TBL = (% CHG * 1.28) + 128
;-----

```

```

; LAMBDA W      RPM
;-----
L8617:  FCB 113      ; 0.941       0
        FCB 113      ; 0.941      400
        FCB 113      ; 0.941      800
        FCB 113      ; 0.941     1200
        FCB 115      ; 0.949     1600
        FCB 115      ; 0.949     2000
        FCB 115      ; 0.949     2400

```

```

FCB 137      ; 1.035      2800
FCB 128      ; 1.000      3200
FCB 160      ; 1.125      3600
FCB 150      ; 1.086      4000
FCB 157      ; 1.113      4400
FCB 138      ; 1.039      4800
FCB 138      ; 1.039      5200
FCB 138      ; 1.039      5600
FCB 138      ; 1.039      6000
FCB 138      ; 1.039      6400

```

* IDLE AIR CONTROL CONSTANTS & TABLES FOR PID F.B.

* ANHT CALIBRATION

```

L8628:  FCB 48      ; 0.188 COEF RPM/12.5 FILTER
          ;
L8629:  FCB 160     ; 160 STP'S IDLE RPM START UP POSIT, PARK
L862A:  FCB 20      ; 20 STP'S ADDED TO WARK PK IF A/C ON
L862B:  FCB 2       ; 2 STP'S ADDED FOR FAN 1
L862C:  FCB 2       ; 2 STP'S ADDED FOR FAN 2
L862D:  FCB 5       ; 5 STP'S ADDED IF COLD ENG SA IS DONE
          ;

```

; CMD SPEED DEF

;

; TBL = ARG/12.5

;

```

L862E:  FCB 8       ; 100 RPM ADDED IF IN PK/NEUT
          ;
L862F:  FCB 48      ; 600 MSEC STRT UP DELAY
          ;
L8630:  FCB 1       ; 12.5 RPM ADDED IF A/C ON
L8631:  FCB 80      ; 1000 RPM IAC CMD SPD OPN LP OFFSET
          ;

```

; IAC CMD SPEED vs COOL

;

; TBL = RPM/12.5

;

; RPM DEG C

```

L8632:  FCB 84      ; 1050      -40
          FCB 84      ; 1050      -28
          FCB 84      ; 1050      -16
          FCB 84      ; 1050       -4
          FCB 76      ; 950        8
          FCB 68      ; 850       20
          FCB 64      ; 800       32
          FCB 56      ; 700       44
          FCB 56      ; 700       56
          FCB 52      ; 650       68
          FCB 48      ; 600       80
          FCB 48      ; 600       92
          FCB 48      ; 600      104
          FCB 48      ; 600      116
          FCB 48      ; 600      128

```



```

FCB 48      ;      600      140
FCB 48      ;      600      152
;-----

;-----
; COMAND SPEED DEAD BAND
;
; RPM CALC's ARE RPM/12.5
;-----
L8643: FCB 3      ; 37.5 RPM MAX ERROR FOR D.B. IN PK/NEUT
FCB 3      ; 37.5 RPM MAX ERROR FOR D.B. IN DRIVE
;
FCB 4      ; 50    RPM MAX ERROR FOR D.B. FOR A/C LEARN
FCB 4      ; 50    RPM MAX ERROR FOR D.B. FOR MOTER POSIT
;-----

;-----
; PID ENABLING PARAMS
;-----
L8647: FCB 2      ; ENAB IF TPS > 1.2%
L8648: FCB 1      ; DONT STORE NEW TPS IF ID DIFF VAL < 1
L8649: FCB 3      ; ENAB IAC PID IF Vss > 0.9 MPH
; TBL = MPH * 3.2
;
L864A: FCB 1      ; ENAB IS UNDER SPEED ERR > 12.5 RPM
; MPH * 3.2
;-----

;-----
; PROPORTIONAL GAINS, (PID)
; ** NOT USED IN ANHT **
;-----
L864B: FCB 16     ; 0.781, UNDERSPEED ERROR PROPPORT GAIN, STEPS
; ARG = VAL * (256 * 12.5)
;
L844C: FCB 16     ; 0.781, OVERSPEED ERROR PROPORT GAIN, STEPS
; ARG = VAL * (256 * 12.5)
;
L864D: FCB 255    ; MIN RPM UNDERSPEED ERROR TO ENAB
; HI PROPORT GAIN ADDITION
;
L864E: FCB 0      ; HI PROPORT GAIN ADDITION, STEPS
; ARG = VAL * (256 * 12.5)
;-----

;-----
; DERIVATIVE GAINS, (PID)
;-----
L864F: FCB 24     ; 600 RPM/(RPM/SEC)
;
L8650: FCB 24     ; 600 RPM/(RPM/SEC)
;
L8651: FCB 32     ; 343 RPM/(RPM/SEC), MIN DEC RATE TO EN HI GAIN
;
L8652: FCB 255    ;
;
L8653: FCB 8      ; 50 RPM Min under spd error to enable hi prop &
; deriv gain additions, (RPM/12.5)
;-----

;-----
; INTERGAL GAINS, (PID)
; *** NOT USED IN ANHT ***
;-----
L8654: FCB 48     ; 48 steps Pk/Nut int gain if RPM rate <
L8655: FCB 64     ; 64 steps
L8656: FCB 12     ; 187.5 RPM/(RPM/SEC), PK/NUT max RPM rate

```

```

L8657  FCB  12      ; 187.5 RPM/(RPM/SEC), Drive max RPM rate
;-----

;-----
; PARK NEUT SCALING GAINS
;-----
L8658:  FCB  255      ; 0.996, (Neut stp's/Drive stp's)
; ARG = VAL/256
;-----

*****
* THROTTLE FOLLOWER GAINS
*
* (MSEC/6.25)
*****
L8659:  FCB  10      ; 62.5 msec between stp's in Pk/Neut
L865A:  FCB  6       ; 37.5 msec between stp's in Drive
;
L865B:  FCB  7       ; 0.5msec add to time between follower in Drive
L865C:  FCB  0       ; 0 msec add to time between follower in open loop
;
L865D:  FCB  192     ; 150% follower slope gain
;
L865E:  FCB  40     ; 40 steps max in drive
;
L865F:  FCB  255     ; .996, (Neut steps/Drive steps)
; ARG = VAL/256
;-----

*****
* STEPPER MOTOR PARAMS
* *** NOT USED IN ANHT ***
*****
L8660:  FCB  32      ; Quantization gain to convert algo out from
; linear motor gains to motor steps

L8661:  FCB  8       ; Idle speed inverse quantization gain to form fract
; Steps

*****
* TRANSITIONAL DELAYS
*****
L8662:  FCB  0       ; Disable clsd lp PID for 0.00 msec after OPEN LOOP DISABLE
L8663:  FCB  10      ; Disable clsd lp PID for 0.20 msec after
L8664:  FCB  16      ; Disable clsd lp PID for 0.32 msec after NEUT TO DRIVE
L8665:  FCB  0       ; Disable clsd lp PID for 0.00 msec aeter FAN ON
L8666:  FCB  0       ; Disable clsd lp PID for 0.00 msec after COLD ENG SPK
;-----
; LOW INTEGRATOR GAIN
;-----
L8667:  FCB  64      ; PK/NEUT LOW INT GAIN if RPM error in dead band
; (STEPS/(RPM/SEC))
;
L8668:  FCB  128     ; DRIVE LOW INT GAIN if RPM error in dead band
; (STEPS/(RPM/SEC))
;
L8669:  FCB  0       ; IF RPM < 0 RPM DISABEL CLS'D LP PID FOLLOWING END OF COLD ENGINE SPA
; TBL = ARG * 128 * (12.5/0.05)
;-----
; MOTOR RESET PARAMS
;-----
L866A:  FCB  160     ; MAX STEP MOTOR POSIT POSSIBLE WORST CASE VALUE, (from orifice)
;-----

```

```

; RPM RATE DEAD BAND
;-----
L866B:  FCB 16      ; MAX RPM RATE FOR DEVIVITIVE
          ; (250 RPM), RMP/SEC = RPM/15.625

;-----
; CLUTCH ANTICIPATE
;-----
L866C:  FCB 255    ; Skip pwr steer read if TPS =< 99.6%
L866D:  FCB 255    ; Skip pwr steer read if TPS > 99.6%
L866E:  FCB 8      ; Skip pwr steer read if Vss >= 8 MPH
;-----

;-----
; PWR STEER ANTICIPATE
;-----
L866F:  FCB 0      ; Pwr steer antcipate steps
L8670:  FCB 0      ; Pwr steer antcipate steps if A/C ON

;-----
; MANUAL PARAMS
;-----
L8671:  FCB 0      ; MANUAL VECH OPT FLAG (MANUAL IF ARG NON - 0)

;-----
; PID HOT SA RETARD PARAM'S
;-----
L8672:  FCB 25     ; IF ENG RUN TIME < 25 SEC ENABLE HOT SA RETARD

;-----
; HOT RESTART CMD SPEED vs MAT
;
; VAL = RPM/25
;-----
          ; SPD          DEG C
L8673:  FCB 6      ; 75          -40
          FCB 6      ; 75          -28
          FCB 6      ; 75          -16
          FCB 6      ; 75           -4
          FCB 6      ; 75           8
          FCB 6      ; 75          20
          FCB 6      ; 75          32
          FCB 6      ; 75          44
          FCB 6      ; 75          56
          FCB 8      ; 100         68
          FCB 12     ; 150         80
          FCB 16     ; 200         92
          FCB 16     ; 200        104
          FCB 16     ; 200        116
          FCB 16     ; 200        128
          FCB 16     ; 200        140
          FCB 16     ; 200        152
;-----

;-----
; IAC TABLE MULT'S (0 - 2)
;
; TBL = MULT * 128
;-----
          ; MULT          Kpa
L8684:  FCB 205    ; 1.6          65
          FCB 179    ; 1.4          75
          FCB 154    ; 1.2          85

```

FCB 128 ; 1.0 95
FCB 128 ; 1.0 105

; WARM PARK POSIT vs COOLANT
;
; TBL = STEPS

L8689: FCB 8 ; NUM OF LINES IN TABLE -1
;
; STEPS Deg c

FCB 125 ; 125 -40
FCB 100 ; 100 -16
FCB 75 ; 75 8
FCB 50 ; 50 32
FCB 25 ; 25 56
FCB 10 ; 10 80
FCB 5 ; 5 104
FCB 5 ; 5 128
FCB 5 ; 5 152

; PID DESIRED RPM ADJ FOR BATTERY VOLTAGE

L8693: FCB 117 ; ADD O.S. TO DESIRED RPM IF BATT =< 11.7 V
L8694: FCB 135 ; SUB O.S. TO DESIRED RPM IF BATT > 13.5 V
;
L8695: FCB 5 ; ADD OFFSET IF BATT LO 5 SEC'S
L8696: FCB 1 ; ADD 12.5 RPM TO OFFSET IF BATT LO
;
L8697: FCB 20 ; CLR BATT IAC WHEN V_{ss} > 20 MPH
;
L8698: FCB 6 ; LIMIT IAC OFFSET FOR LO BAT TO 75 RPM.
; ARG = RPM/12.5

; IAC MIN MOTOR KEEP ALIVE LEARNING

L8699: FCB 40 ; DEFAULT KEEP ALIVE MOTOR STEPS
L869A: FCB 171 ; LEARN WHN COOL > 86.7 C, (188 F)
;
L869B: FCB 192 ; LEARN WHN COOL < 103.3 C, (216 F)
; ARG = (DEG C+40) * 256/192)
;
L869C: FCB 10 ; MIN KEEP ALIVE MOTOR STEPS
L869D: FCB 70 ; MAX KEEP ALIVE MOTOR STEPS
;
L869E: FCB 4 ; 4 INJ'S, IDLE SPEED INJECT TIME OUT DECAY RATE
;
L869F: FCB 1 ; MAX DELTA STEPS/IDLE FOR KA LEARN
L86A0: FCB 50 ; SEC TO LEARN MIN MOTOR KA STEPS

; IDLE SPEED TIME OUT IAC STEPS vs COOLANT or MAT
;
;
; TBL = STEPS

;
; STEPS Deg c

L86A1: FCB 30 ; 30 -40
FCB 30 ; 30 -28

```

FCB 30      ; 30      -16
FCB 30      ; 30      -4
FCB 30      ; 30       8
FCB 30      ; 30      20
FCB 30      ; 30      32
FCB 40      ; 40      44
FCB 40      ; 40      56
FCB 45      ; 45      68
FCB 50      ; 50      80
FCB 60      ; 60      92
FCB 60      ; 60     104
FCB 60      ; 60     116
FCB 60      ; 60     128
FCB 60      ; 60     140
FCB 60      ; 60     152

```

```

;-----
; IAC  MANUAL Throttle Follower vs MPH
;
;
; TBL = MPH/12.5
;-----

```

```

      ORG $06B2      ;
                  ; STEPS      MPH
;-----
L86B2: FCB 0        ; 0          0
      FCB 4        ; 50         4
      FCB 8        ; 100        8
      FCB 12       ; 150        12
      FCB 12       ; 150        16
      FCB 12       ; 150        20
      FCB 12       ; 150        24
      FCB 12       ; 150        28
      FCB 12       ; 150        32
      FCB 12       ; 150        36
      FCB 12       ; 150        40
      FCB 12       ; 150        44
      FCB 12       ; 150        48
      FCB 12       ; 150        52
      FCB 12       ; 150        56
      FCB 12       ; 150        60
      FCB 12       ; 150        64

```

```

* BARO FORM PARAM'S
*
```

```

L86C3: FCB 96      ; MIN TPS FOR BARO UPDATR, 37.5 %
                  ; ARG = VAL * 2.56
                  ;
L86C4: FCB 1       ; MAX DIFF MAP FOR BARO UPDATE 0.31 Kpa
                  ; ARG = VAL * (256/80)
                  ;
L86C5: FCB 144     ; READ BARO THERSH 3600 RPM
                  ; ARG = VAL/25
                  ;
L86C6: FCB 11     ; MAX BARO OFFSET THRESH, 4 Kpa
                  ; ARGD= VAL * 2.71

```

```

; BARO PRESS vs RPM & TPS
;
```

```

;   ORG $86C7
;
; Table = Kpa * 2.71
;-----
L86C7:  FCB      16      ; RPM min,
        FCB      48      ; TPS min
        FCB       6      ; Num line in table

;-----
; 1200 RPM      Kpa      TPS
;-----
        FCB  8      ; 2.952      37.5%
        FCB  3      ; 1.107      50.0
        FCB  1      ; 0.369      62.5
        FCB  0      ; 0          75.0
        FCB  0      ; 0          87.5
        FCB  0      ; 0          100.0
;-----
; 2400 RPM      Kpa      TPS
;-----
        FCB 16      ; 5.904      37.5%
        FCB 16      ; 5.904      50.0
        FCB  8      ; 2.952      62.5
        FCB  5      ; 1.845      75.0
        FCB  3      ; 1.107      87.5
        FCB  1      ; 0.369      100
;-----
; 3600 RPM      Kpa      TPS
;-----
        FCB 16      ; 5.904      37.5%
        FCB 16      ; 5.904      50.0
        FCB 16      ; 5.904      62.5
        FCB 16      ; 5.904      75.0
        FCB 11      ; 4.059      87.5
        FCB  5      ; 1.845      100
;-----

*****
* MAP SENSOR PARAMS
*****
L86DC:  FCB 240      ; MAP FILTER COEF, 0.9375

;-----
; SPD DENS CALC PARAMS
;-----
L86DD:  FDB 159      ; CYL VOL & UNIT CONV, (0.7125 LITRE/CYL)
        ; (CAL = L/CYL * 223.157)
        ;
L86DF:  FCB 48      ; RPM, VE filtering TPS THRESH
L86E0:  FCB 3       ; % TPS VE Filtering thresh
L86E1:  FCB 6       ; VE filtering Coef

;-----
; EGR PARTIAL PRESSURE vs VACUUM
;
; TYPE $8D ECM
;
; Dissassembly of ANHT LINES = 9
; 09-06-1994, 09:13:27
;
; TBL = 4 * Kpa Part Press
;-----
        ORG      $06E2

```

```

;-----
;
;                               Kpa Part Press   Kpa VAC
;-----
L86E2  FCB  6      ;    1.5           80.0
L86E3  FCB  8      ;    2.0           70.0
L86E4  FCB 12     ;    3.0           60.0
L86E5  FCB 16     ;    4.0           50.0
L86E6  FCB 16     ;    4.0           40.0
L86E7  FCB 16     ;    4.0           30.0
L86E8  FCB 16     ;    4.0           20.0
L86E9  FCB 16     ;    4.0           10.0
L86EA  FCB  0      ;    0.0            0.0
;-----

```

```

;-----
; VE1.SRC, (FL1)
;
; Dissasembly of ANHT, 13 x 13
; TYPE $8D ECM
; 11-15-1990, 22:09:46
;
; VE table 400 - 1600 RPM
;
;-----

```

ORG \$86EB

```

L86EB:  FCB    0      ; Min RPM Val
        FCB    0      ; Min Load, (kpa) Value
        FCB   13     ; Number of lines in each table

```

```

;-----
; 400 RPM                    VE            Kpa
;-----

```

```

FCB  81      ; 31.6      20
FCB  92      ; 35.9      25
FCB 102     ; 39.8      30
FCB 112     ; 43.8      35
FCB 124     ; 48.4      40
FCB 137     ; 53.5      45
FCB 147     ; 57.4      50
FCB 149     ; 58.2      55
FCB 152     ; 59.4      60
FCB 157     ; 61.3      70
FCB 166     ; 64.8      80
FCB 173     ; 67.6      90
FCB 185     ; 72.3     100

```

```

;-----
; 500 RPM                    VE            Kpa
;-----

```

```

FCB  92      ; 35.9      20
FCB 102     ; 39.8      25
FCB 111     ; 43.4      30
FCB 121     ; 47.3      35
FCB 133     ; 52.0      40
FCB 144     ; 56.3      45
FCB 151     ; 59.0      50
FCB 154     ; 60.2      55
FCB 158     ; 61.7      60
FCB 163     ; 63.7      70
FCB 169     ; 66.0      80
FCB 174     ; 68.0      90
FCB 185     ; 72.3     100

```

```

;-----
; 600 RPM                    VE            Kpa
;-----

```

FCB	100	;	39.1	20
FCB	110	;	43.0	25
FCB	119	;	46.5	30
FCB	132	;	51.6	35
FCB	142	;	55.5	40
FCB	155	;	60.5	45
FCB	160	;	62.5	50
FCB	163	;	63.7	55
FCB	167	;	65.2	60
FCB	174	;	68.0	70
FCB	178	;	69.5	80
FCB	184	;	71.9	90
FCB	192	;	75.0	100

; 700 RPM VE Kpa

FCB	111	;	43.4	20
FCB	121	;	47.3	25
FCB	130	;	50.8	30
FCB	139	;	54.3	35
FCB	155	;	60.5	40
FCB	160	;	62.5	45
FCB	165	;	64.5	50
FCB	170	;	66.4	55
FCB	172	;	67.2	60
FCB	180	;	70.3	70
FCB	184	;	71.9	80
FCB	190	;	74.2	90
FCB	196	;	76.6	100

; 800 RPM VE Kpa

FCB	115	;	44.9	20
FCB	127	;	49.6	25
FCB	139	;	54.3	30
FCB	149	;	58.2	35
FCB	160	;	62.5	40
FCB	165	;	64.5	45
FCB	171	;	66.8	50
FCB	175	;	68.4	55
FCB	178	;	69.5	60
FCB	183	;	71.5	70
FCB	186	;	72.7	80
FCB	190	;	74.2	90
FCB	197	;	77.0	100

; 900 RPM VE Kpa

FCB	119	;	46.5	20
FCB	130	;	50.8	25
FCB	141	;	55.1	30
FCB	152	;	59.4	35
FCB	164	;	64.1	40
FCB	169	;	66.0	45
FCB	174	;	68.0	50
FCB	177	;	69.1	55
FCB	180	;	70.3	60
FCB	184	;	71.9	70
FCB	186	;	72.7	80
FCB	192	;	75.0	90
FCB	198	;	77.3	100

; 1000 RPM6 VE Kpa

FCB	124	;	48.4	20
-----	-----	---	------	----

FCB	135	;	52.7	25
FCB	146	;	57.0	30
FCB	156	;	60.9	35
FCB	166	;	64.8	40
FCB	171	;	66.8	45
FCB	176	;	68.8	50
FCB	180	;	70.3	55
FCB	182	;	71.1	60
FCB	185	;	72.3	70
FCB	189	;	73.8	80
FCB	192	;	75.0	90
FCB	200	;	78.1	100

;-
; 1100 RPM VE Kpa

FCB	128	;	50.0	20
FCB	139	;	54.3	25
FCB	150	;	58.6	30
FCB	159	;	62.1	35
FCB	168	;	65.6	40
FCB	173	;	67.6	45
FCB	179	;	69.9	50
FCB	182	;	71.1	55
FCB	185	;	72.3	60
FCB	188	;	73.4	70
FCB	190	;	74.2	80
FCB	194	;	75.8	90
FCB	202	;	78.9	100

;-
; 1200 RPM VE Kpa

FCB	130	;	50.8	20
FCB	141	;	55.1	25
FCB	152	;	59.4	30
FCB	162	;	63.3	35
FCB	171	;	66.8	40
FCB	176	;	68.8	45
FCB	181	;	70.7	50
FCB	184	;	71.9	55
FCB	186	;	72.7	60
FCB	190	;	74.2	70
FCB	192	;	75.0	80
FCB	195	;	76.2	90
FCB	202	;	78.9	100

;-
; 1300 RPM VE Kpa

FCB	133	;	52.0	20
FCB	143	;	55.9	25
FCB	154	;	60.2	30
FCB	163	;	63.7	35
FCB	173	;	67.6	40
FCB	178	;	69.5	45
FCB	182	;	71.1	50
FCB	184	;	71.9	55
FCB	186	;	72.7	60
FCB	190	;	74.2	70
FCB	192	;	75.0	80
FCB	196	;	76.6	90
FCB	202	;	78.9	100

;-
; 1400 RPM VE Kpa

FCB	136	;	53.1	20
FCB	145	;	56.6	25
FCB	154	;	60.2	30

FCB	163	;	63.7	35
FCB	173	;	67.6	40
FCB	178	;	69.5	45
FCB	182	;	71.1	50
FCB	184	;	71.9	55
FCB	186	;	72.7	60
FCB	190	;	74.2	70
FCB	192	;	75.0	80
FCB	196	;	76.6	90
FCB	202	;	78.9	100

```

;-----
; 1500 RPM           VE           Kpa
;-----

```

FCB	139	;	54.3	20
FCB	147	;	57.4	25
FCB	156	;	60.9	30
FCB	165	;	64.5	35
FCB	174	;	68.0	40
FCB	180	;	70.3	45
FCB	182	;	71.1	50
FCB	184	;	71.9	55
FCB	186	;	72.7	60
FCB	190	;	74.2	70
FCB	192	;	75.0	80
FCB	196	;	76.6	90
FCB	202	;	78.9	100

```

;-----
; 1600 RPM           VE           Kpa
;-----

```

FCB	141	;	55.1	20
FCB	151	;	59.0	25
FCB	161	;	62.9	30
FCB	167	;	65.2	35
FCB	174	;	68.0	40
FCB	180	;	70.3	45
FCB	182	;	71.1	50
FCB	184	;	71.9	55
FCB	186	;	72.7	60
FCB	190	;	74.2	70
FCB	192	;	75.0	80
FCB	196	;	76.6	90
FCB	202	;	78.9	100

```

;-----
; Dissasembly of ANHT, FL2
;
; VE2.SRC
;
; 11-15-1990, 22:19:52
;-----

```

```

L8797:      ORG          $8797
            FCB          0          ; Min RPM Val
            FCB          0          ; Min Load, (kpa) Value
            FCB          9          ; Number of BP'S per RPM

```

```

;-----
; 1600 RPM BLK
;-----

```

FCB	141	;	55.1	20	Kpa
FCB	161	;	62.9	30	Kpa
FCB	174	;	68.0	40	Kpa
FCB	182	;	71.1	50	Kpa

FCB	186	; 72.7	60	Kpa
FCB	190	; 74.2	70	Kpa
FCB	192	; 75.0	80	Kpa
FCB	196	; 76.6	90	Kpa
FCB	202	; 78.9	100	Kpa

; 2000 RPM BLK

FCB	146	; 57.0	20	Kpa
FCB	166	; 64.8	30	Kpa
FCB	179	; 69.9	40	Kpa
FCB	186	; 72.7	50	Kpa
FCB	190	; 74.2	60	Kpa
FCB	195	; 76.2	70	Kpa
FCB	197	; 77.0	80	Kpa
FCB	201	; 78.5	90	Kpa
FCB	204	; 79.7	100	Kpa

; 2400 RPM BLK

FCB	159	; 62.1	20	Kpa
FCB	180	; 70.3	30	Kpa
FCB	191	; 74.6	40	Kpa
FCB	194	; 75.8	50	Kpa
FCB	198	; 77.3	60	Kpa
FCB	202	; 78.9	70	Kpa
FCB	204	; 79.7	80	Kpa
FCB	208	; 81.3	90	Kpa
FCB	212	; 82.8	100	Kpa

; 2800 RPM BLK

FCB	181	; 70.7	20	Kpa
FCB	202	; 78.9	30	Kpa
FCB	209	; 81.6	40	Kpa
FCB	211	; 82.4	50	Kpa
FCB	214	; 83.6	60	Kpa
FCB	220	; 85.9	70	Kpa
FCB	222	; 86.7	80	Kpa
FCB	227	; 88.7	90	Kpa
FCB	229	; 89.5	100	Kpa

; 3200 RPM BLK

FCB	189	; 73.8	20	Kpa
FCB	209	; 81.6	30	Kpa
FCB	214	; 83.6	40	Kpa
FCB	216	; 84.4	50	Kpa
FCB	222	; 86.7	60	Kpa
FCB	223	; 87.1	70	Kpa
FCB	225	; 87.9	80	Kpa
FCB	232	; 90.6	90	Kpa
FCB	238	; 93.0	100	Kpa

; 3600 RPM BLK

FCB	187	; 73.0	20	Kpa
FCB	207	; 80.9	30	Kpa
FCB	213	; 83.2	40	Kpa
FCB	217	; 84.8	50	Kpa
FCB	220	; 85.9	60	Kpa
FCB	225	; 87.9	70	Kpa
FCB	228	; 89.1	80	Kpa
FCB	232	; 90.6	90	Kpa
FCB	239	; 93.4	100	Kpa

; 4000 RPM BLK

```
-----  
FCB 180 ; 70.3 20 Kpa  
FCB 196 ; 76.6 30 Kpa  
FCB 201 ; 78.5 40 Kpa  
FCB 211 ; 82.4 50 Kpa  
FCB 220 ; 85.9 60 Kpa  
FCB 222 ; 86.7 70 Kpa  
FCB 225 ; 87.9 80 Kpa  
FCB 232 ; 90.6 90 Kpa  
FCB 236 ; 92.2 100 Kpa  
-----
```

; 4400 RPM BLK

```
-----  
FCB 167 ; 65.2 20 Kpa  
FCB 181 ; 70.7 30 Kpa  
FCB 192 ; 75.0 40 Kpa  
FCB 198 ; 77.3 50 Kpa  
FCB 201 ; 78.5 60 Kpa  
FCB 208 ; 81.3 70 Kpa  
FCB 213 ; 83.2 80 Kpa  
FCB 216 ; 84.4 90 Kpa  
FCB 219 ; 85.5 100 Kpa  
-----
```

; 4800 RPM BLK

```
-----  
FCB 155 ; 60.5 20 Kpa  
FCB 168 ; 65.6 30 Kpa  
FCB 180 ; 70.3 40 Kpa  
FCB 186 ; 72.7 50 Kpa  
FCB 192 ; 75.0 60 Kpa  
FCB 200 ; 78.1 70 Kpa  
FCB 202 ; 78.9 80 Kpa  
FCB 204 ; 79.7 90 Kpa  
FCB 206 ; 80.5 100 Kpa  
-----
```

; 5200 RPM BLK

```
-----  
FCB 155 ; 60.5 20 Kpa  
FCB 168 ; 65.6 30 Kpa  
FCB 180 ; 70.3 40 Kpa  
FCB 186 ; 72.7 50 Kpa  
FCB 192 ; 75.0 60 Kpa  
FCB 200 ; 78.1 70 Kpa  
FCB 202 ; 78.9 80 Kpa  
FCB 204 ; 79.7 90 Kpa  
FCB 206 ; 80.5 100 Kpa  
-----
```

; 5600 RPM BLK

```
-----  
FCB 155 ; 60.5 20 Kpa  
FCB 168 ; 65.6 30 Kpa  
FCB 180 ; 70.3 40 Kpa  
FCB 186 ; 72.7 50 Kpa  
FCB 192 ; 75.0 60 Kpa  
FCB 200 ; 78.1 70 Kpa  
FCB 202 ; 78.9 80 Kpa  
FCB 204 ; 79.7 90 Kpa  
FCB 206 ; 80.5 100 Kpa  
-----
```

; MAT COMP Vs MAT

;

; TABLE is MAT COMP COUNTS

```

;-----
; BIN          MAT Deg c
;-----
L87FD:  FCB 0      ; 0          -40
        FCB 12     ; 12         -28
        FCB 24     ; 24         -16
        FCB 36     ; 36          -4
        FCB 48     ; 48           8
        FCB 60     ; 60          20
        FCB 72     ; 72          32
        FCB 84     ; 84          44
        FCB 96     ; 96          56
        FCB 108    ; 108         68
        FCB 120    ; 120         80
        FCB 128    ; 128         92
        FCB 128    ; 128        104
        FCB 128    ; 128        116
        FCB 128    ; 128        128
        FCB 128    ; 128        140
        FCB 128    ; 128        152
;-----

```

```

;-----
; (COOLANT - MAT) DELTA MULT
; FOR INV TERM L.U. vs AIR FLOW
;
; TBL = MULT * 128
;-----

```

```

;-----
; MULT          GMS/SEC AIR FLOW
;-----
L880E:  FCB 128    ; 1.000        0
        FCB 96     ; 0.750        16
        FCB 64     ; 0.500        32
        FCB 36     ; 0.281        48
        FCB 27     ; 0.210        64
        FCB 20     ; 0.156        80
        FCB 16     ; 0.125        96
        FCB 16     ; 0.125       112
        FCB 16     ; 0.125       128
        FCB 16     ; 0.125       144
        FCB 16     ; 0.125       160
        FCB 16     ; 0.125       176
        FCB 16     ; 0.125       192
        FCB 16     ; 0.125       208
        FCB 16     ; 0.125       224
        FCB 16     ; 0.125       240
        FCB 16     ; 0.125       256
;-----

```

```

;-----
; VE Mult vs BARO
;
; TBL = MULT * 128
;-----

```

```

;-----
; MULT          Kpa BARO
;-----
L881F:  FCB 152    ; 1.188        65
        FCB 146    ; 1.141        75
        FCB 139    ; 1.086        85
        FCB 133    ; 1.039        95
        FCB 128    ; 1.000       105
;-----

```

```

* SERIAL DATA CALIB
* TYPE $8D ECM P/N 1227730
* 8192 Baud Comm
*****

```

```

L8824:  FCB 118          ; INJ FLOW RATE 3776 gal/hr
          ; (GAL/HR)/32, IP DISPLAY

```

```

;-----
; Broadcast Message Scheduling table
; SELECT MSG ADDRESS FOR EACH MINOR LOOP NUMBER
; If address = 0000 the the message is ignored
;
; table = Address
;-----

```

```

L8825:  FDB $0000      ; 0
          FDB $0000      ; 1
          FDB $0000      ; 2
          FDB $0000      ; 3
          FDB $0000      ; 4
          FDB $0000      ; 5
          FDB $0000      ; 6
          FDB $0000      ; 7
          FDB $0000      ; 8
          FDB $0000      ; 9
          FDB $0000      ; 10
          FDB $0000      ; 11
          FDB $0000      ; 12
          FDB $0000      ; 13
          FDB $0000      ; 14
          FDB $0000      ; 15

```

```

;-----
;-----
; MSG TO TX IN B.C. REMOTE MODE
; TBL VAL = ADDR
; DEVICE ID $80
;-----

```

```

L8845:  FDB $0000      ; NEXT MSG ENTRY ADDR
          FCB $80        ; DEVICE ID
          FCB 0          ; MEM FLAG
          FCB 5          ; NUM BYTES TO TX
          ;
          FDB $014E      ; ADDR OF OUTPUT MSG BUFFER
          FDB $0000      ; ADDR OF INPUT MSG BUFFER
          ;
          FDB $0038      ; 1 MW2
          FDB $0058      ; 2 RPM/25
          FDB $0094      ; 3 TPS, (A/D)
          FDB $0000      ; 4
          FDB $0079      ; 5 MAP

```

```

;-----
;-----
; MSG TO TX IN B.C. REMOTE MODE
;
; DEVICE ID $F4
;-----

```

```

L8858:  FDB $0000      ; NEXT MSG ENTRY ADDR
          FCB $10        ; DEVICE ID
          FCB $0         ; MEM FLAGS
          FCB 5          ; NUM BYTES TO TX
          ;
          FDB $014E      ; ADDR OF OUTPUT MSG BUFFER
          FDB $0000      ; ADDR OF INPUT MSG BUFFER

```

FDB \$005D ; 1. COOLANT
FDB \$3130 ; 2. IP FUEL
FDB \$3131 ; 3. IP FUEL
FDB \$8824 ; 4. INJ FLOW RATE 3776 gal/hr
FDB \$0038 ; 5. MW 2

FDB \$0000 ; 6
FDB \$0000 ; 7
FDB \$0000 ; 8
FDB \$0000 ; 9
FDB \$0000 ; A
FDB \$0000 ; B
FDB \$0000 ; C

; POLLING MESSAGE
; DEVICE ID \$F0

L8879: FDB \$0000 ; NEXT MSG ENTRY ADDR
FCB \$F0 ; DEVICE ID
FCB 0 ; MEM FLAG
FCB 0 ; NUM BYTES TO TX

FDB \$014E ; ADDR OF OUTPUT MSG BUFFER
FDB \$0000 ; ADDR OF INPUT MSG BUFFER

; CIM, DEVICE ID \$40

; MSG TO TX IN BC REMOTE MODE
; TBL VALL = ADDR

L8882: FDB \$88A3 ; NEXT MSG ENTRY ADDR
FCB \$40 ; DEVICE ID
FCB \$80 ; MEM FLAG, OUTPUT IS ROM TBL
FCB 12 ; NUM BYTES TO TX

FDB \$014E ; ADDR OF OUTPUT MSG BUFFER
FDB \$014E ;

FDB \$0058 ; 1 RPM
FDB \$0074 ; 2 MAP, A/D
FDB \$0095 ; 3 TPS
FDB \$005D ; 4 COOLANT
FDB \$0063 ; 5 MAT
FDB \$0049 ; 6 CCM Mode Wd
FDB \$01B8 ; 7 CCM REVS
FDB \$3130 ; 8 IP FUEL
FDB \$3131 ; 9 IP FUEL
FDB \$8124 ; 10
FDB \$0081 ; 11 MPH
FDB \$01BD ; 12 OIL TEMP

```
* SERIAL DATA REC'ED MSG TABLES
* MESSAGE ID $42
*
*TYPE $8D ECM P/N 122730
*****
```

```
;-----
; SERIAL DATA RX MSG
; RESPONDS TO A CCM POLL, $42
; ECM RESPONDS WITH ID $43
;
; DEVICE ID = $42
;-----
```

```
L89A3:  FDB $88B2    ; NXT MSG ENTRY ADDER
        ;
        FCB $42     ; DEVICE ID
        ;
        FCB $80     ; MEM FLAG, OUTPUT IS ROM TBL
        FCB $03     ; NUM BYTE TO OUTPUT
        ;
        FDB $014E   ; ADDR OF OUTPUT BUFFER
        FDB $014E   ; ADDR OF INPUT BUFFER
        ;
        FDB $0058   ; 1  RPM/25
        FDB $0074   ; 2  MAP A/D VALUE
        FDB $0095   ; 3  TPS * 2.56
;-----
```

```
;-----
; SERIAL DATA RX MSG
; RESPONDS TO A CCM POLL $82
; ECM RESPONDS WITH $83
;
;-----
```

```
L89B2:  FDB $88BB    ; NXT MSG ENTRY ADDER
        ;
        FCB $82     ; DEVICE ID
        FCB $00     ; MEM FLAG
        FCB $80     ; NUM BYTE TO OUTPUT

        FDB $014E   ; ADDR OF OUTPUT BUFFER
        FDB $014E   ; ADDR OF INPUT BUFFER
;-----
```

```
*****
* SERIAL DATA REC'ED MSG TABLES
* MESSAGE ID $Fx
*****
```

```
*****
* MODE 0, REVERT TO NORMAL MODE.
*
* ALDL DEV MUST REQUEST MODE 0 BY
* XMITING THE FOLLOWING MSG TO
* THE ECM:
*
*   DEVICE ID           = $F4
*   MSG LEN 1+85       = $56
*   MODE                = $00
*   CKSUM              = $29
*
* THE ECM WILL RESPOND WITH:
*
*   DEVICE ID           = $F4
```



```
* MSG LEN 64+85 = $95
* MODE = $00
* CKSUM = $cc
*
*****
```

```
;-----
; SERIAL DATA RX MSG
; DEVICE ID = $F4, MODE 0
;
; ECM RESPONDS WITH $F4 AND RESUMES
;
; NORMAL COMM.
; TBL VAL = ADDR
;-----
```

```
L88BB: FDB $0000 ; NXT MSG ENTRY ADDER
      ;
      FCB $F4 ; DEVICE ID
      ;
      FCB $00 ; MEM FLAG
      FCB $80 ; NUM BYTE TO OUTPUT

      FDB $014E ; ADDR OF OUTPUT BUFFER
      FDB $014E ; ADDR OF INPUT BUFFER
;-----
```

```
*****
```

```
* MODE 1 FIXED DATA DATA STREAM
* ALDL DEV MUST REQUEST MODE 1 BY
* XMITING THE FOLLOWING MSG TO
* THE ECM:
```

```
*
* DEVICE ID = $F4
* MSG LEN 1+85 = $56
* MODE = $01
* CKSUM = $29
*
```

```
* THE ECM WILL RESPOND WITH:
```

```
*
* DEVICE ID = $F4
* MSG LEN 64+85 = $95
* MODE = $01
* DATA BYTE 1 = $XX
*
* .
* .
* BATA BYTE 63 = $XX
* CKSUM = $29
*
```

```
*****
```

```
;-----
; SERIAL DATA RX MSG
; DEVICE ID = $4F, MODE 1
; tbl vall = addr
;-----
```

```
L88C4: FDB $0000 ; NXT MSG ENTRY ADDER
      ;
      FCB $F4 ; DEVICE ID
      ;
      FCB $80 ; MEM FLAG, OUTPUT IS ROM TBL
      FCB 64 ; NUM BYTE TO OUTPUT
      ;
      FDB $014E ; ADDR OF OUTPUT BUFFER
```

```

FDB $014E ; ADDR OF INPUT BUFFER
;-----
FDB $8000 ; 1 PROM ID MSB
FDB $8001 ; 2 PROM ID LSB
;-----
FDB $0004 ; 3 MAL FUNCT WD 1
;
; b7, 1
; b6, 1
; b5, 1
; b4, 1
;
; b3, 1
; b2, 1
; b1, 1
; b0, 1
;-----
FDB $0005 ; 4 MAL FUNCT WD 2
;
; b7, 1
; b6, 1
; b5, 1
; b4, 1
;
; b3, 1
; b2, 1
; b1, 1
; b0, 1
;-----
FDB $0006 ; 5 MAL FUNCT WD 3
;
; b7, 1
; b6, 1
; b5, 1
; b4, 1
;
; b3, 1
; b2, 1
; b1, 1
; b0, 1
;-----
FDB $0007 ; 6 MAL FUNCT WD 4
;
; b7, 1
; b6, 1
; b5, 1
; b4, 1
;
; b3, 1
; b2, 1
; b1, 1
; b0, 1
;-----
FDB $0008 ; 7 MAL FUNCT WD 5
;
; b7, 1
; b6, 1
; b5, 1
; b4, 1
;
; b3, 1
; b2, 1
; b1, 1
; b0, 1
;-----
FDB $005F ; 8. COOLANT TEMP, (DEG C +40) * (256/192)

```

```

;
;-----
FDB $0060 ; 9. TEMP, (DEG C +40) * (256/192)
;
;-----
FDB $0094 ; 10. A/D TPS
;
;-----
FDB $0058 ; 11. RPM, RPM/25
;
;-----
FDB $30B3 ; 12. NEW REF PER + $3000, (65536 * 15)/RPM
FDB $30B4 ; 13. NEW REF PER + $3001, (65536 * 15)/RPM
;
;-----
FDB $0081 ; 14. MPH
;
;-----
FDB $0003 ; 15. FUEL/AIR MODE WD 1
;
;-----
FDB $0140 ; 16. TRANS RATIO
;
;-----
FDB $008E ; 17. FILTERED O2 A/D VAL
;
;-----
FDB $00FC ; 18. ALDL CNT'R
;
;-----
FDB $00E8 ; 19. COR RCL
;
;-----
FDB $00E0 ; 20. BLM VALUE
;
;-----
FDB $00DF ; 21. BLM CELL NUMBER
;
;-----
FDB $00EB ; 22. INT
;
;-----
FDB $002B ; 23. CURRENT IAC
;
;-----
FDB $0095 ; 24. TPS, %TPS * 2.56
;
;-----
FDB $0112 ; 25. RPM/12.5
;
;-----
FDB $0075 ; 26. MAP A/D VALUE
;
;-----
FDB $0030 ; 27. SC1 SDI
;
;-----
FDB $0033 ; 28. FMD SDI
;
;-----
FDB $0002 ; 29. Non Vol MW1
;
;-----
FDB $013C ; 30. MAT A/D
;
;-----
FDB $0123 ; 31. EGR D.C.

```

```

;
;-----
FDB $0125 ; 32. PURGE D.C.
;
;-----
FDB $0041 ; 33. DIAG MW2
;
;-----
FDB $0103 ; 34. BATTERY A/D VALUE
;
;-----
FDB $0104 ; 35. PUMP VDC
;
;-----
FDB $0043 ; 36. DIAG MW4
;
;-----
FDB $002D ; 37. CURRENT IAC
;
;-----
FDB $01BD ; 38. OIL A/D VALUE
;
;-----
FDB $3128 ; 39. SA (256/90)
FDB $3129 ; 40. SA
;
;-----
FDB $30BD ; 41. SA + REF
FDB $30BE ; 42. SA + REF
;
;-----
FDB $00C1 ; 43. OLD PA3 CNT
FDB $00C4 ; 44. KNK RETARD (256/90)
;
;-----
FDB $30E4 ; 45. INJ, Msec * 65.536
FDB $30E5 ; 46. INJ, Msec * 65.536
;
;-----
FDB $30F3 ; 47. AFR, MSB, (445 = 14.7)
FDB $30F4 ; 48. AFR, LSB
;
;-----
FDB $3130 ; 49. IP FUEL, Msec * 0.256
FDB $3131 ; 50. IP FUEL, Msec * 0.256
;
;-----
FDB $0134 ; 51. ACUM DIST
;
;-----
FDB $3019 ; 52. TIME, SECONDS, MSB
FDB $301A ; 53. TIME, SECONDS, LSB
;
;-----
FDB $0038 ; 54. MW 2
;
;       b7, 1
;       b6, 1
;       b5, 1
;       b4, 1
;
;       b3, 1
;       b2, 1
;       b1, 1
;       b0, 1
;-----
FDB $004A ; 55. TCC MW

```

```

;
;       b7, 1
;       b6, 1
;       b5, 1
;       b4, 1
;
;       b3, 1
;       b2, 1
;       b1, 1
;       b0, 1
;-----
FDB $0031 ; 56. FMD BYTE 1
;
;       b7, 1, A/C CLUTCH OFF
;       b6, 1, HEAD PRESS HI
;       b5
;       b4
;
;       b3, 1, PWR STEER SW, (CRAMP)
;       b2
;       b1, 1 = IN 3rd or 4th GR
;       b0, 1 = Pk/Neut, 0 = Over Drive
;-----
FDB $0037 ; 57. MW 1
;
;       b7, 1
;       b6, 1
;       b5, 1
;       b4, 1
;
;       b3, 1
;       b2, 1
;       b1, 1
;       b0, 1
;-----
FDB $0001 ; 58. NON VOL MW
;
;       b7, 1
;       b6, 1
;       b5, 1
;       b4, 1
;
;       b3, 1
;       b2, 1
;       b1, 1
;       b0, 1
;-----
FDB $004B ; 59. CARS MW
;
;       b7, 1
;       b6, 1
;       b5, 1
;       b4, 1
;
;       b3, 1
;       b2, 1
;       b1, 1
;       b0, 1
;-----
FDB $003E ; 60. CLC MW
;
;       b7, 1
;       b6, 1
;       b5, 1
;       b4, 1
;

```

```

;      b3, 1
;      b2, 1
;      b1, 1
;      b0, 1
;-----
FDB $003F ; 61. AIR MW
;
;      b7, 1
;      b6, 1
;      b5, 1
;      b4, 1
;
;      b3, 1
;      b2, 1
;      b1, 1
;      b0, 1
;-----
FDB $003C ; 62. LCC PWM
;
;-----
FDB $0047 ; 63. FUEL/AIR MODE WD 1
;
;-----

```

```

;-----

```

```

*****

```

```

* MODE 2 SELECTABLE MEMORY DUMP
*
* ALDL DEV MUST REQUEST MODE 2 BY
* XMITTING THE FOLLOWING MSG TO
* THE ECM:
*

```

```

*   DEVICE ID           = $F4
*   MSG LEN 3+85       = $58
*   MODE               = $02
*   START ADD MSB     = $aa
*   START ADD LSB     = $aa
*   CKSUM              = $nn
*

```

```

* THE ECM WILL RESPOND WITH:
*

```

```

*   DEVICE ID           = $F4
*   MSG LEN 65+85      = $96
*   MODE               = $03
*   ADD CONT           = $dd
*   ADD CONT           = $dd
*   .
*   .
*   ADD (+63)CONT     = $dd
*   ADD (+63) CONT    = $dd
*   CKSUM              = $29
*

```

```

*****

```

```

;-----
; SERIAL DATA RX MSG
; DEVICE ID = $4F, MODE 2
; TBL VALL = ADDR
;-----

```

```

L894B:  FDB $0000 ; NXT MSG ENTRY ADDER
        FCB $F4   ; DEVICE ID
        FCB $40   ; MEM FLAG, OUTPUT IS RAM TBL
        FCB 65    ; NUM OF BYTES TO TX

```

FDB \$014E ; ADDR OF OUTPUT BUFFER
FDB \$014E ; ADDR OF INPUT BUFFER

* MODE 3 RAM DUMP 0 to 8 ADDRESSES
*
* ALDL DEV MUST REQUEST MODE 3 BY
* XMITTING THE FOLLOWING MSG TO
* THE ECM:

*
* DEVICE ID = \$F4
* MSG LEN 2n+85 = \$58
* MODE = \$03
* ADD MSB = \$aa
* ADD LSB = \$aa
* .
* .
* ADD MSB = \$aa
* ADD LSB = \$aa
* CKSUM = \$cc

* THE ECM WILL RESPOND WITH:

*
* DEVICE ID = \$F4
* MSG LEN n+1+85 = \$nn
* MODE = \$03
* ADD CONT = \$dd
* ADD CONT = \$dd
* .
* .
* ADD CONT = \$dd
* ADD CONT = \$dd
* CKSUM = \$cc

; SERIAL DATA RX MSG
; DEVICE ID = \$4F, MODE 3
;
; SELECTABLE MEMORY DUMP

L8954: FDB \$0000 ; NXT MSG ENTRY ADDR
 ;
 FCB \$F4 ; DEVICE ID
 ;
 FCB \$40 ; MEM FLAG, OUTPUT IS RAM TBL
 FCB 17 ; NUM OF BYTES TO TX

 FDB \$014E ; ADDR OF OUTPUT BUFFER
 FDB \$014E ; ADDR OF INPUT BUFFER

* MODE 4 MODIFIER OP'S
*
* MODE 4 WILL MODIFY ECM OPERATION
* & XMIT THE CONTENTS OF RAM,
* (0 -> 8 LOC'S) AS REQUESTED
*

* ALDL DEV MUST REQUEST MODE 4 BY
* XMITTING THE FOLLOWING MSG TO
* THE ECM:

*
* DEVICE ID = \$F4
* MSG LEN 2n+1+10+85 = \$nn
* MODE = \$04
* CNTL WD 1 = \$cc
* .
* .
* CNTL WD 10 = \$cc
* ADD 1 MSB = \$aa
* ADD 1 LSB = \$aa
* .
* .
* ADD N MSB = \$aa
* ADD N LSB = \$aa
* CKSUM = \$cc
*

* THE ECM WILL RESPOND WITH:

*
* DEVICE ID = \$F4
* MSG LEN n+1+85 = \$nn
* MODE = \$04
* ADD CONT = \$dd
* .
* .
* ADD CONT = \$dd
* CKSUM = \$cc
*

; SERIAL DATA RX MSG
; DEVICE ID = \$4F, MODE 4
; TBL VAL = ADDR

L895D: FDB \$0000 ; NXT MSG ENTRY ADDER
 FCB \$F4 ; DEVICE ID
 FCB \$40 ; MEM FLAG, OUTPUT IS RAM TBL
 FCB 1 ; NUM OF BYTES TO TX

 FDB \$014E ; ADDR OF OUTPUT BUFFER
 FDB \$014E ; ADDR OF INPUT BUFFER

; SERIAL DATA RX MSG
; DEVICE ID = \$F4, MODE 7
;
; COMMAND MESSAGE
;
; TBL VAL = ADDR

L8966: FDB \$0000 ; NXT MSG ENTRY ADDER
 FCB \$F4 ; DEVICE ID
 FCB \$80 ; MEM FLAG, OUTPUT IS ROM TBL
 FCB 12 ; NUM OF BYTES TO TX

 FDB \$014E ; ADDR OF OUTPUT BUFFER
 FDB \$014E ; ADDR OF INPUT BUFFER

```

;-----
; SERIAL DATA RX MSG
; DEVICE ID = $F4, MODE 10
;
; Reset err's
;
; TBL VAL = ADDR
;-----
L896F:  FDB $0000    ; NXT MSG ENTRY ADDER
        FCB $F4     ; DEVICE ID
        FCB $80     ; MEM FLAG, OUTPUT IS ROM TBL
        FCB 128     ; NUM BYTES IN OUTPUT BUFFER

        FDB $014E   ; ADDR OF OUTPUT BUFFER
        FDB $014E   ; ADDR OF INPUT BUFFER
;-----

;-----
; TABLE POINTERS FOR ALDL MODE
; DEVICE ID $F4
;
;-----
L8978:  FDB $88BB    ; Ser data RX DEVICE ID $F4, MODE 0
        FDB $88C4    ; Ser data RX DECICE ID $F4, MODE 1
        FDB $894B    ; Ser data RX DEVICE ID $F4, MODE 2
        FDB $8954    ; Ser data RX DEVICE ID $F4, MODE 3
        FDB $895D    ; Ser data RX DEVICE ID $F4, MODE 4
        FDB $8966    ; Ser data RX DEVICE ID $F4, MODE 7
        FDB $896F    ; Ser data RX DEVICE ID $F4, MODE 10
;-----

L8986:  FDB $FFFF    ; ALLOW MODE 4, 65,535 SEC'S
L8988:  FDB 0000     ; PREVENT MODE 4, 0 SEC'S
;-----

*****
* END OF CALIBRATION
*****

*****
* ANHT_ALG.SRC
*
* START OF ALGO
*****
LIST    E,F,4

;-----
; 1990 VET COUPE SOURCE CODE 6/9/93
;
; RE-ASSEMBLED OK 8/31/93 WAS
;
;
; POWER UP OR RESET INIT.
;
; HARDWARE VECTORS HERE ON POWER UP OR ON
; AN UNEXPECTED EVENT, (RESET).
;
; ALL RAM IS INIT'ED, I/O PORTS RESTARTED
; ADDITIONAL TEST ARE RUN TO VALIDATE THE SOFTWARE

```

```

;
;
;-----
*****
INCLUDE 8D_EQU.SRC
*****
      ORG      $B000
LB000: LDS      #$01FF          ; Set user stack

      JSR      LEC65          ; GO DO FACTORY TEST

;-----
; ASSIGN LOG RAM
; Bit 3, (1=ECM, 0=I2U)
;-----
      LDAA     #8              ; b3
      STAA     L5000          ; Log RAM CNT'L FLAG ADDR

;-----
; CK EPROM OK FLAG, IF GOOD
; SEND $000A (CE LAMP OFF) TO
; CPU else $0002 TO CPU
;-----
      LDX      #$000A          ; CE lite off
      BRCLR   L0003,#8,LB015  ; Br if ERR 51 not set
      ;
      ; Else
      ;
      LDX      #$0002          ; Ck eng lamp ON
LB015: STX      L3FFC          ; CPU CNT'L REG
;-----

;-----
; CLEAR CPU RAM
; $3FC0 - $3FFA
;-----
      LDX      #$3FC0          ; CPU
      CLRA
      CLR     CLR B          ; CLR D REG
LB01D: STD      0,X           ; Zero Reg

      INX
      INX
      CPX     #$3FFA          ; Ck for done
      BNE     LB01D           ; Till done
      ; ... else
;-----

;-----
; SET UP CPU I/O
;
;-----
      LDAA     #$8C           ; Bits 2,3 & 7
      STAA     L400B          ; SER I/O CSR,
      ; Set prescaler port value=64Khz

      LDAA     #$FF           ; Set for output
      STAA     L4003          ; CPU DDR

```

```

LDAA    #$90                ; b4 & b7,, SET BAUD RATE, 8192
STAA    L4004              ; BAUD I/O REG, (SCI)

LDAA    #$B0                ; Ch B for A/D TEST 1st TIME THRU
JSR     LF0D2              ; GET A/D RESULT FOR TEST CH
;-----

;-----
; CLEAR RAM
; $002E - $01C1, SKIP $0100
;-----
LDX     #$01C2              ; LAST RAM +1
LB03D:  CLR     0,X          ; Clear RAM
LB03F:  DEX     ; Next address
CPX     #$0100             ; Don't clear CKSUM, SKIP AROUND
BEQ     LB03F              ; Till done

CPX     #$002E             ; Done ?
BNE     LB03D              ; If not done, loop
; .... else
;-----

;-----
; FMD Init
; (must delay for glitch
; to settle)
;-----
LDX     #$4000              ; Init FMD
BSET    2,X,#4             ; Sel FMD, ( PRP2)

LDAA    #$04                ; SET EST enable BIT
STAA    L0033              ; Save status in FMD status wd
JSR     LF0C0              ; XMIT ON SERIAL LINE

BCLR    2,X,#4             ; Deselect FMD
;-----

; SC1 Init
;-----
LB05A:  CLRA

;
; CK AFR OPT WD 3, 0101 0010
; b6, 1 = MAG SPD SENSER INSTALLED
;
LDAB    $8018              ; AFR OPT WD #3
BITB    #$40                ; b6
BNE     LB064              ; Br if b6, MAG SPD SENSOR
; .... else
LDAA    #$10                ; Disable SXR, VSS follows VSS1 (B4)
LB064:  STAA    L002F        ; Set up for xmit

BCLR    2,X,$80            ; Select SC1 (RPR7)
JSR     LF0C0              ; Send serial byte

BSET    2,X,$80            ; Deselect SC1 input data
STAA    L0030              ; Save inital SCI input data
;-----

```

```

; SOCKETED DEVICE CHECK
; ERR 51, (EPROM)
;
; TYPE $8D ECM
;-----
LB071: LDAA    $8008          ; ID BYTE ADDRESS
      CMPA    #$AA          ; Engeneering Ck Byte
      BEQ     LB083         ; BR if Engeneering Ck Byte
      ; ... else
      CMPA    #$8D          ; Production ck sum Arg
      BNE     LB088         ; Go set ERR, wrong EPROM
      ; .... else
      LDX     #$0100        ; Point to fact test word
      BRSET   0,X,$10, LB088 ; Br is Cksum wrong

LB083: BCLR    L0003,#8      ; Clear EPROM ERR 51 BIT
      BRA     LB08B         ;
;-----

LB088: BSET    L0003,#8      ; Set EPROM ERR 51 (Failed) BIT

LB08B: LDX     #$FB1A        ; Init CPU
      BRCLR   L0003,#8, LB095 ; Br if No ERR 51 BIT set
      ; ... else
      LDX     #$FB12        ; 64,274d
LB095: STX     L3FFC         ; CPU CNT'L REG
;-----
; INITITILIZE IM
;
;-----
LDAA    #8                ; Bit 3
STAA    L5000             ; LOG RAM TO ECM, (I^2U)

LDAA    $8025             ; REF ANGLE, (INIT SA, 6 Deg)
CLRB
STD     L01A5             ; SAVE SA TDC

CLRA
STD     L0463             ; Base Injection PW
;-----
; DO Non Vol RAM CK SUM
;-----
LB0A8: JSR     LE4A9         ; NV RAM CKSUM

      CPD     L0017         ; CK CURRENT DIAG ERR CK SUM
      BEQ     LB0CE         ; IF MATCH, CK SUM OK
      ; ... else
;-----
; Initilize Non Volitile RAM
;-----
;-----
; Clear NV RAM $0000 $002E
;-----
LB0B0: LDX     #$002E        ; End address

LB0B3: CLR     0,X           ; Clear RAM
      DEX
      BNE     LB0B3         ; Till done
      ; .... else
;-----

```

```

;-----
; DO Non Vol RAM CK SUM
;
;-----
JSR    LE4A9          ; DO Non Vol RAM CKSUM
STD    L0017         ; Update Non Vol ERR CK SUM
;-----

JSR    LE548          ; Go init BLM cells

LDAA   $8629          ; 160, SET PRESENT IAC MOTOR S/U POSIT, (park)
STAA   L002B         ; CURRENT IAC POSIT

LDAA   $8699          ; 40, DEFAULT KEEP ALIVE IAC MOTOR STEPS
STAA   L002D         ; KEEP ALIVE IAC POSIT

LDAA   #$40           ; B6 1 = Non Vol MEM BAD
STAA   L003E         ; Set Non Vol Mem Bombed flag,
                   ; cleared when ram cleared

LB0CE: BRCLR    L0003,#8,LB0EE ; Br if ERR 51 not set

JSR    LE5AC          ; Else... turn on CE light

BSET   L0006,#1      ; Set ERR 52

;-----
; DO Non Vol RAM CK SUM
;-----
JSR    LE4A9          ; Non Vol RAM CKSUM
STD    L0017         ; Up date Non Vol ERR CK SUM
;-----

LDAA   L0100          ; FACTORY TEST MODE WD
BMI    LB0EE         ; Br if in fact test

LDAA   #$70           ; Set A/D ch 7, Diag ALCL
JSR    LF0D2         ; Go do A/D

CMPA   #40           ; < 40d, (NEAR GND)
BCS    LB0EE         ; Br if in diag mode
                   ; ... else

;-----
; WAIT HERE FOR INTERUPT
; SWI VECTOR TO $6000
;-----
LB0EB: SWI           ; Force reset
BRA    LB0EB         ; LOOP HERE
;-----

*****

LB0EE: JSR    LD74C          ; Read MAT so it can be used as default

BSET   L003D,$80      ; SET b7, 4K Pull up for cool
JSR    LE4B8         ; Read coolant w/4k PU (MJR LP SEG 6)

BRSET  L0001,#8,LB128 ; Ck Bit 3 of Mode wd 1, (Abnorm shut dwn)
                   ; EST ERR

LDAA   L005D         ; FLT COOLANT TEMP

```

```

BRCLR    L0046, #80, LB103    ; BR IF NOT b7
                                ; B7 1 = USE MAT FOR FUEL CALC
                                ; ... else

LB103:   LDAA    L0063          ; INV A/D MAT
        PSHA                    ; SAVE INV A/D MAT
        CMPA    #208           ; CK TEMP LIMIT, (92c)
        BLS     LB10A          ; BR IF L.T. 92C
                                ; .... else
        LDAA    #208           ; LIMIT TO 92 C, (198 F)

;-----
; LK UP START UP vs ENRICH
;
;-----
LB10A:   PSHA                    ; SAVE INV COOL TEMP TO STX
        LDX     #8431          ; START UP vs ENRICH TABLE,
                                ; (Init start up)
        JSR     LE3D0          ; LOOK UP

        CLRB
        STD     L000D          ; TIME OUT AFR

        PULB
        PULA
        PSHB

;-----
; LK UP OPEN LP AFR vs COOLANT
;
;-----
        LDX     #845E          ; OPEN LP AFR vs COOLANT TABLE
        JSR     LE3D0          ; LOOK UP

        STAA    L00F0          ; SAVE RESULT

;-----
; LK UP START UP vs COOLANT SPK ADV
;
;-----
        PULA
        LDX     #81B9          ; START UP vs COOLANT SPK ADV TABLE
        JSR     LE3D0          ; LOOK UP

        STAA    L0012          ; START UP vs COOLANT SPK ADV

LB128:   LDAA    L005D          ; FLT COOLANT TEMP

;
; CK AFR OPT WD 3, 0101 0010
; b0, 1 = COOL FOR IDLE SPD TIME OUT
;
        LDAB    $8018          ; AFR OPT WD 3
        RORB
        BCS     LB132          ; BR IF ....
                                ; ... else
        LDAA    L0063          ; INV MAT

;-----
; IDLE SPEED TIME OUT IAC STEPS vs COOLANT or MAT
;
;
; TBL = STEPS
;-----
LB132:   LDX     #86A1          ; INDEX IAC IDLE SPD TIME OUT STPS vs

```

```

; COOLANT or MAT TABLE
JSR     LE3D0      ; LOOK UP
;
STAA    L01B1     ; IDLE SPD TIME OUT STEPS
;
LDAA    L005F     ; COOLANT TEMP, (DEG C +40) * (256/192)
;
BCLR    L003C,#$10 ; CLR b4, 2nd pass coolant flag
;
CMPA    $8031     ; 44c, MAX COOL FOR COLD ENG SPK ADV
BCS     LB148     ; IF COOLANT WARM BYPASS $8031 LOGIC
; ... else
BSET    L0001,#$40 ; SET WARM KICK DOWN ENABLED BIT

;-----
; INIT o2 Var's
;-----
LB148: LDAA    $83F3      ; 12.5Msec o2 FIL COEF.
STAA    L008E     ; MNR LP FILTERED A/D o2
STAA    L0090     ; Slow filt o2 val
STAA    L0092     ; Diag Flt o2 val

;-----
; INIT TPS Var's
;-----
LDAA    #$50      ; Set A/D Ch for TPS read, (CH 5)
JSR     LF0D2     ; Go do A/D

STAA    L0094     ; AD TPS BUFFER

;-----
; INIT TPS
;-----
LDAA    $83E5     ; MIN TPS, (%TPS * 2.56, 35d)
STAA    L0099     ; Flt A/D TPS OFF
JSR     LE45C     ; TPS LOAD
;
LDAA    L0095     ; TPS LOAD AXIS VARIABLE, %TPS * 2,56
STAA    L00AB     ; Flt TF TPS DE
STAA    L00A0     ; Flt TPS Acell Enr

JSR     LDE49     ; Init Batt voltage & Ign flag

;-----
; INIT MAP Var's
;-----
LDAA    #$30      ; Set A/D Ch for MAP read, (ch 3)
JSR     LF0D2     ; Go do A/D

STAA    L0074     ; Raw MAP A/D VAL
STAA    L0076     ; Flt MAP val
CMPA    L002E     ; MAP BAR VAL
; Init Baro to MAP if
; MAP > Current NV BARO

BLS     LB178

STAA    L002E     ; Up date to MAP BAR VAL

;-----
; SPD DENS MAP = (1 ATMOSP)
; = (((A/D MAP/694) * 65536)+2647)/256
;-----
LB178: PSHA          ; Save MAP on STACK
TAB          ; A to B
CLRA        ; A = 0
LDX         #0694 ;

```

FDIV
PSHX

PULA ; GET MAP fm STX
PULB
ADDD #2647 ;
STD L0079 ; MAP for Spd/Dens calc
;-----

;-----
PULA ; Get MAP fm STACK
SUBA #28 ;
LDAB #151 ;
JSR LE5F7 ; MAKE PRESS LD VAR'S

STAA L0071 ; Init normalized MAP Ld Var
STAA \$00A6 ; Flt MAP for MAP Acell Enr
STAA \$00AD ; Flt MAP for MAP Decel Enlean
;-----

;-----
;
; = (((A/D BARO/694) * 65536)+2647)/256
;-----
LDAB L002E ; Get MAP BAR VAL for Calc'ed A/D
; Baro Calc, (16 bit)

CLRA
LDX #694 ;
FDIV
PSHX
PULA
PULB
ADDD #2647 ;
STD L007B ; CALC'ED BARO kPa VALUE
;-----

;-----
; Calc MAP/Baro ratio
;-----
PSHB
PSHA

PULX ; GET BARO Kpa FM STX
LDD L0079 ; MAP for Spd/Dens calc
; (1 Kpa/BIT)

FDIV
PSHX ; SAVE RESULT TO STX
PULA ; RESLUT TO A REG
INS
STAA L007D ; MAP to BARO RATIO
;-----

;-----
; OTHER RAM VAR'S
;-----
LDD L3FC8 ; PA2 CNT'R
STD L00D5 ; OLD PA2 CNT'R

LDD L3FEE ; PA4 CNT'R, RD SPD PULSE CNT'R
JSR LE4B7 ; SHORT DELAY

LDX L3FC2 ; B Ctr
STX L0087 ; Last Vss count


```

STD      L0085          ; Last(+1) Vss count

LDD      $841A         ; A/F RATIO 14.4, 0445d
STD      L00F1         ; FINAL TOTAL AF VAL (AFR)

LDAA     #14           ; Set Mnr Lp CNT'R = 14 so ML 15 is 1st
STAA     L0000         ; NINOR LP CNT'R, (DO MJR LP 15 1st)
;-----

;-----
;Initialize CL corr terms
;-----
LDAA     #128          ; Mid point
STAA     L00E0         ; BLM
STAA     L00E8         ; Clsd lp corr
STAA     L00EB         ; Clsd lp intrgrator
;-----

;-----
;Initialize Vss
;-----
LDAA     #39           ; MAX FOR NO RD SPD Ctr
STAA     L0084         ; RD SPD NO PULSE CNTr

;-----
;Initialize IAC posit
;-----
LDAA     $8629         ; COLD IAC PARK POSIT
ADDA     L002D         ; KEEP ALIVE MIN MOTOR STEPS
BCS      LB1E2         ; PREVENT OVER FLOW
; ... else
ADDA     L01B1         ; ADD in IDLE SPD TIME OUT STEPS
BCC      LB1E4         ;
; ... else
LB1E2:   LDAA     #255   ; limit to 255
LB1E4:   STAA     L011D   ; INIT IAC WARM PARK POSIT

;-----
; Toggle COP1 just prior
; to enable interupts
;-----
LDX      #$FF00        ; Set to toggle COP1,
;      00 = COP Clr,
;      FF = COP ARM
STX      L400B         ; Hardware

;-----
; Exit initalization:
; start 6.25 msec int's
;-----
LDAA     L4008          ; CPU TX/RX STAT REG, (8192 BAUD SCI)
; Ld to CLR PENDING CLK INT'S
;
LDAA     L4005          ; CPU CNT'R, REAL TIME COUNTER
ADDA     #02           ; Wait 30-60 usec FOR FIRST IRQ*
STAA     L4006          ; CPU CAPT REG

;
; GET 8192 BAUD SCI RX Data
;
LDAA     $4009          ; GET RX Data, (8192 BAUD SCI)

;

```

```

; SET UP 8192 SCR RX
;
LDX      #$4000          ; INDEX CPU REG'S

;
; Enable 8192 SCI Rx interupts, $4007
;   b0 CNT'R INTERUPT
;   b1 RX WAKE UP
:   b2 RX ENABLE
:   b5 RX INTERUPT ENABLE
;
BSET     7,X,#$27        ; $4027, ENABLE CLK & RCV INT'S

JMP      LB6C0           ; Wait for 1st INTERUPT
*****

*****
* IRQ1.SRC
*
* Executed once for every IRQ1 which is
* conneceted to CPU RT int line.
*
* All 'minor' loop routines are performed in
* this routine including A/F, EST, IAC
*
*
*****
LB204:   LDAA     L0100          ; FACTORY TEST MODE WD
        BPL      LB20C          ; If not in factory test mode
        ; ... else
        JMP      LED23          ; IN FACTORY TST MODE, GO EXE

;-----
; CHECK 8192 SCI STATUS
;
;
;-----
LB20C:   LDX      #$4007          ; INDEX 8192 BAUD SCI REGESTERS
        BRSET   1,X,#$01,LB24D  ; BR IF b0, CK CNT'R INTERUP
        ; (TO 6.25 Msec INIT)

*****
* RX STATUS CHECKS
*
*****
;
; 8192 RX INTERUPT CHECKS
; CK IF RX INT ENABLED & IF RX DATA PRESENT
;
BRCLR   0,X,#$20,LB21F        ; BR IF NOT b5, RX INT ENAB
        ; ... else
BRCLR   1,X,#$20,LB24C        ; BR IF NOT b5, RX DAT TEG FULL
        ; ... else
JSR     LEB93                 ; TO RX INT HANDLER

RTI                                ; RETURN FM INTERUPT

*****
* TX STATUS CHECKS
*
*****
;

```

```

; CK IF 8192 TX INT ENABLED
;
LB21F: BRCLR    0,X,#$80, LB22B      ; BR IF NOT b7,
; ... else

;
; CK IF TX DATA REG EMPTY
;
BRCLR    1,X,#$80, LB24C      ; BR IF NOT b7, TX DATA REG EMPTY
; ... else
JSR      LEB93                ; TX INTERUPT HANDLER

RTI                                           ; RETURN FM INTERUPT

;
; CK IF 8192 TX COMPLETE INT ENABLED , else EXIT
;
LB22B: BRCLR    0,X,#$40, LB24C      ; BR IF NOT b6, 8192 TX DONE
; ... else

;
; CK IF XMIT COMPLETE, else EXIT
;
BRCLR    1,X,#$40, LB24C      ; BR IF NOT b6, 8192 TX DONE
; ... else

;-----
; SET UP 8192 SCI FOR RX
;
;-----
LDX      #$4000                ; INDEX CPU REG'S

;
; Select the SCI (Via THE SXR CHIP)
;
BCLR     2,X,#$80              ; CLR b7, PORT DDR, (MAKE IT AN INPUT)
BCLR     L002F,#$01            ; CLR b0, DISABLE SER DATA BIT

LDAA     L002F                  ; Disable SXR Xmitter
JSR      LFOC0                  ; Xmit S3 Bit

BSET     2,X,#$80              ; SET b7, PORT DDR, (MAKE IT AN OUTPUT)

;
; Enabel Rx interupts
; b0 CNT'R INTERUPT
; b1 RX WAKE UP
: b2 RX ENABLE
: b5 RX INTERUPT ENABLE
;
LDAA     #$27                  ; b0, 1, 2 & 5
STAA     L4007                  ; CPU TX/RX CNT'L REG, (8192 BAUD SCI)
; Enable Rx interupts

BCLR     L0048,#$08            ; Clr b3, XMIT IN WORK

LB24C: RTI
*****

```

; CRANK FUEL CALC & DELIVERY LOGIC

;-----

LB29B: BRSET L0036,#\$10,LB305 ; BR IF IGN OFF, (NOT b4)
; ... else

BRSET L0037,#\$80,LB321 ; BR IF b7, EMG RUNNING
; (SKIP CRANK FUEL IF RUNNING)
; ... else

BRCLR L0003,#\$20,LB305 ; BR IF NOT b5, 1 = VATS OK
; ... else

;-----
; LK UP CRANK FUEL DELAY vs MAT
; (DELAY vs DRP'S)
;-----

LB2A7: LDAA L0063 ; GET INV MAT
LDX #\$839C ; CRANK FUEL DELAY vs MAT Table
JSR LE3D0 ; LOOK UP FUEL DELIV DELAY

CMPA \$013F ; DRP CNT'R
BHI LB305
; ... else
LDAB L012B ; DRP CNTR FOR COLD START ELIMN
ANDB #15 ; Assume 1st 16 ref's
LDX #\$83C7 ; CRANK FUEL PW MULT vs Num DRP'S, (1-16)

BRCLR L003E,#\$02,LB2C5 ; Br if < 16 ref's
; B1 1 = USE ALT ___ TBL
; ... else

LDX #\$83D7 ; CRANK FUEL PW MULT vs DRP'S TBL
ANDB #7 ;

LB2C5: ABX ;
LDY L00E2 ; BPW
BEQ LB30F ; DON'T OUT PUT IF BPW = 0

LDAA 0,X ; GET MULTIPLIER
BEQ LB313 ; DON'T OUT FUEL OR UPDATE HUD
; IF MULTIPLIER = 0
LDX #L00E2 ; BASE PULSE WIDTH
JSR LE33F ; ADJ BPW, (8 X 16 MULT)

PSHB
PSHA
ADDD L012D+1 ; UPDATE ACCUM FUEL
BCC LB2DF ; BR IF NO OVERFLOW
; ... else
INC L012D ; ACCUM FUEL

LB2DF: STD L012D+1 ; UPDATE ACCUM FUEL

PULX ; GET CURRENT BPW
LDAB L00EE ; ADJ BPW FOR BATTERY VDC
ABX ; ADD TO BPW (twice)
ABX
STX L0436 ; BASE PULSE WIDTH
STX L3FF2 ; ASYN PW
JSR LE4B7 ; CPU DELAY

LDD L3FFC ; CPU CNT'L REG
JSR LE4B7 ; CPU DELAY

ORAA #4 ; TRIGGER ASYNC PULSE
STD L3FFC ; CPU CNT'L REG

```

JSR      LE4B7                ; CPU DELAY

ANDA    #$FB                ; 1111 1011
STD     L3FFC                ; CPU CNT'L REG
BRA     LB313

;-----
; VATS FAILED/IGN OFF DO 0 FUEL
;-----
LB305:  LDX      #0
        STX     L00E2        ; BASE PULSE WIDTH
        STX     L0436        ; BASE PULSE WIDTH
        BRA     LB321
;-----

LB30F:  STY     L0436        ; BASE PULSE WIDTH

LB313:  LDAB    L012B        ; DRP CNTR FOR COLD START ELIMN
        CMPB   #14          ;
        BLS    LB31D        ; BR IF L.T. 14 DRP'S
        ; ... else
        BSET   L003E,#2     ; SET ALT TABLE FLAG

;-----
; SPK
;-----

LB31D:  INCB                    ; INCR CNT'R
        STAB   L012B        ; DRP CNTR FOR COLD START ELIMN

LB321:  BRCLR   L0033,#$04,LB344 ; BR IF not b2
        ; B2 1 = EST ENABLE
        ; ... else

        BRSET  L00BB,#80,LB344 ; BR IF b7
        ; SA RETARDED
        ; ... else

        BRSET  L0016,$$02,LB344 ; BR IF b1,
        ; IN DECEL FUEL C/O STALL SAVE
        ; IF ERR CNTR NON ZERO
        ; ... else

        LDAA   $85F3        ; 2 CNT'S ADDED TO SA BLEND/DRP
        LDX   L009B        ; STALL SAVER TIMER
        BNE   LB337        ;
        ; ... else
        LDAA   $802E        ; 16 CNTS ADDED TO SA BLEND EACH DRP

LB337:  ADDA   L013B        ; SPK ADV BLEND MULT
        BCC   LB341        ; BR IF BLEND NOT FINISHED
        ; ... else

        LDAA   #255
        BCLR  L0039,$$20    ; clr b5, SPARK BLEND IN PROG FLAG

LB341:  STAA   L013B        ; SPK ADV BLEND MULT

LB344:  BRSET  L003C,$$02,LB352 ; BR IF b1,
        ; ALREADY STARTED
        ; B1 1 = TIME 1st DRP TO ENG RUN
        ; ... else

        BRSET  L0037,$$80,LB352 ; Br b7, IF ENGINE RUNING
        ; ... else
        BSET   L003C,$$02    ; SET TIME 1ST REF TO ENG RUN FLAG

```

```

CLR      L045E                ; CLEAR THE START TIME

LB352:  BRCLR  L0036,#$10,LB359    ; BR IF IGN ON
        ; ... else
BCLR    L0038,#$04            ; RESET DRP THIS 6.25 ms PERIOD INDICATOR

;-----
; CK VATS
; ECM 1227730 TYPE $8D
;-----
LB359:  BRSET  L0003,#20,LB387    ; DON'T CK VATS IF ALREADY PASSED
        ; ... else

;
; CK AFR OPTION WD 1 0001 01    00
;   b4, 1 = ENABLE VATS
;
LDAA    $8016                ; AFR OPTION WD 1, 0001 0100
BITA    #$10                 ; b4
BEQ     LB352                ; Br IF NOT b4
        ; ... else
LDD     L3FF8                ; PA1 TMR
CLV
SUBD    $0135                ; OLD VATS VAL
BEQ     LB387                ; BR IF NO PULSE
        ; ... else

LDX     L3FF8                ; PA1 TMR
STX     $0135                ; OLD VATS VAL
PSHB
PSHA

SUBD    L800E                ; 3277d, VATS HI FREQ, 66 HZ

PULA
PULB
BHI     LB387                ; BR IF DELTA CNTS >
        ; ... else
SUBD    L8010                ; 1638d, VATS LO FREQ, 33 HZ
BCS     LB352                ; Br IF DELTA CNTS <
        ; ... else

LB381:  BSET   L0003,#$20        ; SET VATS BIT OK

CLR     L00D1                ; ERR 46, (VATS FAIL TMR)
;-----

LB387:  LDX    L3FC0            ; LAST DRP PERIOD
        STX    L00B3            ; CURRENT MNR LOOP DRP PERIOD
;-----

*****
* CK ALDL MODE for MODE 4 REQUEST
*
* ANHT/4298 P/N 16134297
* ECM TYPE $8D, ECM P/N 1227727
*
*****
BRCLR  L0038,#$20,LB3D1      ; Br if not b5, (IN ALDL MODE)
        ; ... else
LB390:  LDAA   L0170            ; GET ALDL MODE NUMBER
        CMPA  #$04            ; CK IF MODE 4
        BNE  LB3BF            ; BR IF NOT MODE 4

```

```

; ... else
LDD     L01A2           ; GET TIME MD 4 IS LOCKED OUT
BNE     LB3BF          ; BR IF NZ, MODE 4 LOCKED OUT
; ... else
BSET    L0048,#$01     ; SET b0, (IN ALDL MODE 4)
*****

```

```

;-----
; XFER 1/0 CONTROL BYTES FROM AL Serial
; Out I/F Cnt's Block BUFFER TO CONTROL WORDS
; {10 BYTES}
;
;-----

```

```

LDD     L0171           ; ALSO ICB+1
STD     L0192           ; ALDL DISCRETE EN

```

```

LDD     L0173           ; ALSO ICB+3
STD     L0194           ; ALDL MD EN

```

```

LDD     L0175           ; ALSO ICB+5
STD     L0196           ; ALDL PW EN

```

```

LDD     L0177           ; ALSO ICB+7
STD     L0198           ; ALDL FN MD

```

```

LDD     L0179           ; ALSO ICB+9
STD     L019A          ; ALDL DS AF

```

```

BRA     LB3D1           ;
;-----

```

```

;-----
; ALDL MODE 4, CK FOR MODE ZERO
;
;
; CLR L0048: afr status wd
;     b0 IN ALDL MD 4
;     b4 BLM RESET
;     b5 IAC R/S
;     b6 ERR FLG CLR'ED
;
; CLR L0039: mode wd 2
;     b1
;
;
; ANHT/4298 P/N 16134297
; ECM TYPE $8D, ECM P/N 1227727
;
;-----

```

```

LB3BF: BRCLR   L0048,#$01,LB3D1   ; BR IF ALREADY OUT OF ALDL MODE 4
; ... else
LDAA   L0170           ; GET CURRENT ALDL MODE <-----****
BNE    LB3D1          ; BR IF NZ
; (don't clr mode 4 till mode 0 req)
; .... else
BCLR   L0048,$71      ; CLR 0111 0001, MODE 4 FLAGS
BCLR   L0039,$01      ; CLR b0, SKIP ERR 42 FLAG
CLR    L00D7          ; CLR EST FEEDBACK CNT'R
;-----

```



```

;-----
;
;
;-----
LB3D1: LDAA    $0000          ; MINOR LOOP COUNTER
      INCA          ; INC MINOR LOOP CNT'R
      CMPA    #160        ; 1 SECOND YET
      BEQ     LB3DB       ;
      ; ... else
      JMP     LB465       ; ELSE JMP BY MISC 1 SEC LOGIC

;
; MISC 1 SECOND LOGIC
;
LB3DB: BRCLR   L0037,#$02,LB3E2 ; BR IF NOT b1,1 = HIWAY FUEL TMR RUN'G
      ; ... else

      INC     L0080       ; INCR FUEL TMR

;
; TOGGLE 1 SECOND FLAG, (BIT 5)
;
LB3E2: LDAB    L003D          ; ALCL MODE
      EORB    #$20          ; b5, TOGGLE 1 SEC FLG
      STAB    L003D          ; SAVE TO ALCL MODE
      ;-----
      ; ... else
      BRCLR   L0037,#$80,LB3F1 ; Br IF NOT B7, 1 = ENGINE RUNNING
      ; ... else

;-----
; INCREMENT ENGINE
; RUN TIME COUNTER
;-----
LDX     L0019          ; Eng run time
INX          ; Inc eng run time counter
STX     L0019          ; Eng run time
;-----

;-----
; SERIAL DATA LINK INACTIVE
; (TIME OUT LOGIC)
;
;
; ANHT/4298 P/N 16134297
; ECM TYPE $8D, ECM P/N 1227 727
;
;-----
LB3F1: BRCLR   L0038,#$28,LB432 ; BR IF NOT b3 & b5
      ; B3 1 = IN CCM MODE
      ; B5 1 = IN ALDL MODE
      ; ... else

      LDAA    L019E          ; GET SERIAL DATA LINK INACTIVE TMR
      INCA          ; INCR LINK INACTIVE TIMER
      BEQ     LB402       ; Br IF TMR OVERFLOW
      ; ... else
      STAA    L019E          ; SERIAL DATA LINK INACTIVE TIMER
      CMPA    #5           ; 5 Seconds
      BCS     LB432       ; Br if time LT 5 sec
      ; .. else

```

```

;
; CLR FLAGS
;
;
LB402: CLR      L0170          ; FORCE NORMAL MODE, (MODE 0)
      BCLR     L0038,#$28    ; Clr ALDL & CCM Mode flags
      BCLR     L0048,#$79    ; Clr MODE 4 FLAGS etc
      BCLR     L0039,#$01    ; Clr SKIP ERR 42 FLAG
      CLR      L00D7          ; Clr EST FEEDBACK CNT'R

      LDX      #$4000        ; INDEX CPU REG'S
      SEI                      ; TURN OFF INT'S

;
; SET SCI RX Via SXR CHIP
;
      BCLR     2,X,$$80      ; CLR b7, MAKE PORT b7 AN INPUT
      BCLR     L002F,#1      ; CLR b0,
      LDAA     L002F          ; DISABLE SCI Xmitter (Via SXR CHIP)
      JSR      LF0C0          ; GO Xmit S3 SERIAL

;
; DESELECT SCI RX Via SXR CHIP
;
      BSET     $02,X,$$80    ; SET b7, PORT b7 AND OUTPUT

      CLI                      ; RESTORE INTERUPT SERVICE

;
; Enabel Rx interupts
;   b0 CNT'R INTERUPT
;   b1 RX WAKE UP
;   b2 RX ENABLE
;   b5 RX INTERUPT ENABLE
;
      LDAA     $$27           ; BITS 0,1,2 & 5
      STAA     $4007         ; CPU TX/RX CNT'L REG, (8192 BAUD SCI)

      LDD      #0            ;
      STD      L014C          ; CLR SCI DATA MSG TABLE PTR
      STD      L014A          ; CLR SCI DATA BYTE COUNTER
;-----

;-----
; ALDL Mode 4 LOCKOUT TIMER
;
; ANHT/4298 P/N 16134297
; ECM TYPE $8D, ECM P/N 1227727
;
;-----
LB432: BRCLR   L0048,$$01,LB44D ; BR if NOT IN MODE 4
      ; else
      LDX      L01AD          ; TMR
      INX                      ; INCR TMR
      STX      L01AD          ; SAVE NEW TMR VALUE

      CPX      $8986          ; ALLOW MODE 4 PERIOD, 65,535 SEC'S
      BCS      LB464          ; Br if TIMER NOT EXPIRED
      ; else
      BCLR     L0048,$$01    ; CLR MODE 4 TST BIT

      LDX      #01           ; INIT TMR

```

```

STX      L01A2                ; START LOCK OUT TIMER

BRA      LB464
;-----

;-----
;
;
;-----
LB44D:   LDX      L01A2                ; LOCK OUT TIMER
        BEQ      LB464                ; Br IF MODE 4 NOT LOCKED OUT
        ; ... else

        INX
        STX      L01A2                ; LOCK OUT TMR

        CPX      $8988                ; PREVENT MODE 4, 0 SEC'S
        BCS      LB464                ; if l.o. time < calib max
        ; ... else

        LDD      #0
        STD      L01A2                ; LOCKOUT TIMER DONE,
        ;      Re-ENABLE MODE 4

        STD      L01AD
;-----

;-----
;
;
;-----
LB464:   CLRA                        ; SET MINOR LOOP CNT'T = 0

LB465:   STAA     $0000                ; MINOR LP CNTR

        ;
        ; READ FMD 2 DEVICE AT $40002 via SPI
        ;
LB467:   JSR      LE2FA                ; CPU SERIAL READ, READ THE FMD DESCRETES
        STAA     L0031                ; Save results to FMD 1 MODE WD

        JSR      LE2FF                ; FMD SERIAL READ
        ; READ THE FMD BYTE 2 DESCRETES

        STAA     L0032                ; SAVE FMD 2 DEVICE RESULTS

        ;
        ; READ FMD 2 DEVICE AT $40002 via SPI
        ; I/O D
        ;
        LDX      #$4000                ; POINT TO I/O D REG'S
        SEI                        ; TURN OFF INTERRUPTS

        BCLR     2,X,#$80              ; SELECT I/O D SC1
        LDAA     L002F                ; GET SC1 OUTPUT
        JSR      $F0C0                ; XMIT ON SERIAL LINE

        BSET     2,X,#$80
        CLI                        ; CLR & TUEN ON INT'S
        STAA     L0030                ; SAVE SC1 INPUT DATA

        LDAA     L0031                ; GET FMD 1 MODE WD
        COMA

```

```

;
; CK AFR OPTION WD 1, 0001 0100
;
LDX      #$8016          ; INDEX TO AFR OPTION WD'S
BRSET    $02,X,$04,LB493 ; BR IF NOT b2, (AUTO XMISH)
; ... else
BRCLR    0,X,#4,LB493   ; BR IF NOT b2, N.C. 4TH GR SW
; ... else
EORA     #$04           ; INVERT 4TH GR SWITCH

LB493:   BRCLR    $03,X,$20,LB499 ; BR IF NOT b5, (N.O. FAN REQ
; INPUT)
; ... else
EORA     #$40           ; TOGGLE FAN REQUEST, b6

LB499:   BRSET    3,X,$40,LB49F   ; BR IF b6, (PWR STEER SW
; INSTALLED)
; .... else
LB49F:   STAA     L0038           ; CLR b3, PER STEER BIT
; MD WD

LDAA     L0095           ; TPS LOAD AXIS VARIABLE, %TPS * 2,56
CMPA     $866C          ; SKIP PWR STEER READ
; IF TPS =< 99.6%

BCS      LB4BD          ; .... else
CMPA     $866D          ; SKIP PWR STEER READ
; IF TPS > 99.6%

BCC      LB4BD          ; .... else
LDAA     L0081           ; FILTERED MPH
CMPA     $866E          ; SKIP PWR STEER READ
; IF Vss >= 8 MPH
;

LDAA     L0031           ; GET FMD 1 MODE WD
COMA
ANDA     #$08           ; CLR b3, PWR STEER SW, (CRAMP)
ORAA     L0038
STAA     L0038

LB4BD:   JSR      LE616          ; GO DO IAC 6.25 MSEC ROUTINE

;
; Read Throttle posit
;
LB4C0:   LDAA     #$50          ; Load TPS A/D Channel, (Ch 5)
JSR      LF0D2            ; Go do A/D

STAA     L0094           ; Save TPS result
JSR      LE45C           ; GO CALC TPS LD VAL
; RET W/ RESULTR AT L0095

;-----
; LK UP TPS FILT COEF vs RPM
;
; Make Filtered TPS for
; Delta TPS Acell Enrich
; (0.063 ALL VAL'S)
;-----
LDAA     L0058           ; Get RPM/25
LDX      $856B          ; TPS filt coef tbl
JSR      LE3D0          ; Go look up coef

TAB

```

```

LDX      L00A0                ; Get old Delta TPS A.E.
LDAA     L0095                ; TPS LOAD AXIS VARIABLE, %TPS * 2,56

JSR      LE31C                ; Apply lag filt coef

;-----
; Set FILTERED TPS AE = TPS LD if ....
;   TPS LD < FLTERED TPS AE
;-----
CMPA     L0095                ; TPS LD VAL
BLS      LB4E0                ;
;... else
LDAA     L0095                ; TPS Ld VAL
LB4E0:   STD      L00A0        ; UpDATE FILTERED TPS AE

;-----

*****
* DIFF TPS Acell Enrichment
*
* If (N TPS LD - FLT TPS AE) > TPS AE enable AE &
* save (N TPS LD - FLT TPS AE) as DELTA TPS.
*
*   ORG $B4E2
*****
ORG $B4B2

LB4E2:   BRCLR   L0037,#$80,LB535    ; BR IF NOT b7, 1 = ENGINE RUNNING
; ... else

LDAA     L0038                ; DIAG MOD WD
BITA     #$24                 ; CK BITS 2 & 5
BNE      LB535                ; DISSABLE DIF-AE IF ERR 21 or 22
; ... else
LDAA     L0095                ; TPS Ld val
BRCLR   L0045,#$80,LB521      ; Br IF NOT TPS ACELL ENRICH
; ... else

LDAB     L00A3                ; Diff TPA Acell Enr Cntr, (DRP's)
CMPB     $8530                ; 12, ACEL ENR LASTS 12 REF PULSES
BCC      LB535                ; Br IF CNT'S E.Q. GT 12
; ... else

SUBA     L0096                ; MAKE DIFF TPS
BCC      LB505                ; BR if DIFF IS Negitive
; ... else

COMA     ; Invert DIFF TPS
CMPA     $853A                ; ZERO ASYNC PWR ENR IF
; DIFF TPS G.T. 3.9%
BCC      LB535                ; ... else
BRA      LB53E                ; EXIT

LB505:   CMPA     $8538        ; 3.13% TPS FOR L853B TBL ENTRY
; IF ACELL ENR ENABLED
BLS      LB516                ; BR IF L.T. 3.13% TPS
; ... else

LDAA     $8539                ; INITIAL ACELL ENRICN CNT
; FOR RE-ENTRY TO TBL L853B
CMPA     L00A2                ; Acell Enr COUNTER
BCC      LB513                ;

```

```

;... else
STAA    L00A2    ; Acell Enr COUNTER
LB513:  CLR      L00A3    ; Diff TPA Acell Enr Cntr, (DRP's)
LB516:  LDAA     L0095    ; TPS Ld val
        SUBA     L00A0    ; Make diff TPS
        CMPA     $8537    ; USE MSB OF TBL L853B FOR 3 DRP'S
        ; FROM START OF ACELL ENR OR
        ; DECEL ENLEAN IF 1.2 % TPS
        BLS      LB535    ; BR IF TPS L.T. 1.2%
        ; ... else
        BRA      LB52E    ;

;
; BR HERE IF IF NOT TPS ACELL ENR
;
LB521:  SUBA     L00A0    ; MAKE DIFF TPS
        CMPA     $852E    ; %TPS MIN DIFF FOR Acell Enr PORP to BPW
        BLS      LB535    ; CLR FLGS & EXIT TPS Acell Enr
        ; ... else

        BCLR     L0045,$$10 ; CLR b4,
        ; USE LO ORDER NIBBLE OF FUEL LIMIT TBL

        CLR      L00A3    ; CLEAR Diff TPS Acell Enr Cntr, (DRP's)
LB52E:  STAA     L009E    ; DIFF TPS FOR ENR

        BSET     L0045,$$C0 ; SET b6 & b7
        ; B7 1 = TPS ACCEL ENRICH ON
        ; B6 1 = TPS ACCEL ENRICH LIMITING EXT ON

        BRA      LB53E    ; EXIT TPS Acell Enr

LB535:  CLR      L00A2    ; FUEL TRIM DRP CNT'R
        CLR      L00A3    ; Diff TPS Acell Enr Cntr, (DRP's)
        BCLR     L0045,$$90 ; CLR b4 & b7, TPS Acell Enr & LIMIT FLAGS
        ; b4 1 = USE LO ORDER NIBBLE OF
        ; FUEL LIMIT TBL
        ; b7 1 = TPS ACCEL ENRICH ON

;
; READ MAP SENSOR
;
LB53E:  LDAA     $$30     ; Sel MAP A/D ch
        JSR      LF0D2    ; Go read A/D

        STAA     L0075    ; MAP A/D result

;-----
; ERR 33, HIGH MAP
;-----
LDAA     L0038    ; DIAG MD WD 1
BITA     $$24     ; BIT 2 & 5
BNE      LB576    ; IF ERR 21 or 22, (TPS or )
; .... else
LDAA     $8283    ; RATIO, HI MAP/BARO THRESH

BRCLR    L0031,$$01,LB555 ; BR IF NOT b0
; b0 1 = Pk/Neut, 0 = Over Drive
; ... else

LDAA     $8284    ; DRIVE HI MAP/BARO RATIO THRESH, 0.85

```

```

LB555:  CMPA    L007D      ; BARO/MAP PRESS RATIO
        BHI     LB573      ; CLR ERR 33 TMR & EXIT
        ; ... else
        LDAA   L00CE      ; ERR 33, HI MAP TIMER
        CMPA   $8287      ; TIME, ERR 33 THRESH
        BCC    LB57B      ; EXIT WITH ERR 33 SET
        ; ... else
        LDAA   L0059      ; RPM/12.5 (un-filt)
        CMPA   $8286      ; RPM THRESH TO INCR ERR 33 TMR
        BCS    LB576      ; CLR ERR 33 TMR & EXIT
        ; ... else
        LDAA   L0095      ; TPS Ld val
        CMPA   $8285      ; %TPS, UPPER LIMIT TO SET ERR 33
        BCC    LB576      ; CLR ERR 33 TMR & EXIT
        ; ... else
        INC    L00CE      ; BUMP UP ERR 33, HI MAP TMR
        BRA    LB581      ; EXIT ERR 33 TEST

;-----
; CLR ERR 33
;-----

LB573:  BCLR    L0038,$$40 ; CLR b6, ERR 33 FLAG
        ;      B6 1 = HIGH BATT, (OUTPUTS OFF)

LB576:  CLR     L00CE      ; CLR ERR 33, HI MAP TMR
        BRA    LB581      ; EXIT ERR 33 TST VIA ERR 34 TST

;-----
; SET ERR 33
;-----

LB57B:  BSET    L004E,$$04 ; SET b2, ERR 33, MAP HIGH BIT
        BSET    L0038,$$40 ; SET b6, HIGH BATT, (OUTPUTS OFF)

;-----
; ERR 34, LOW MAP
;-----

LB581:  BRSET   L0038,$$04, LB5AE ; EXIT ERR 34 IF TPS ERR 21
        ; ... else

        LDAA   L007D      ; BARO/MAP PRESS RATIO
        CMPA   $8288      ; MAP HI/BARO LO RATIO THRESH
        BCC    LB5AE      ; CLR ERR 34 IF RATIO HIGH
        ; ... else
        LDAB   L00CF      ; ERR 34, LO MAP TIMER
        CMPB   $828B      ; 0.05 SEC'S TO SET ERR 34
        BCC    LB5A6      ; SET ERR 34 & EXIT
        ; ... else
        LDAB   L0059      ; RPM/12.5, (un-filt)
        CMPB   $8289      ; RPM THRESH TO SET ERR 34, (600 RPM)
        BCS    LB5A1      ; INCR ERR TMR & EXIT
        ; ... else
        LDAB   L0095      ; TPS Ld val
        CMPB   $828A      ; %TPS MIN TPS FOR ERR 32, (20%)
        BLS    LB5AE      ; CLR ERR 34 & EXIT
        ; ... else

LB5A1:  INC     L00CF      ; BUMP UP ERR 34, LO MAP TIMER
        BRA    LB5B4      ;

;-----
; SET ERR 34
;-----

LB5A6:  BSET    L004E,#2    ; SET ERR 34, MAP LOW BIT
        BSET    L0038,$$80 ; SET b7, SHIFT LIGHT ON

        BRA    LB5B4      ;

```

```

;-----
; CLR ERR 34
;-----
LB5AE: CLR      L00CF      ; CLR ERR 34, LO MAP TIMER
      BCLR     L0038, #80 ; CLR b7, SHIFT LIGHT ON

;-----
; ERR 33/34
;-----
LB5B4: LDAA     L0075      ; MAP A/D RESULT
      BRCLR   L0038, #C0, LB5E9 ; BR IF NOT b6 & b7,
      ; SHIFT LIGHT ON & HI BAT

      LDAA     $828C      ; Kpa, DEFAULT if ERR 34, (84.5 Kpa)
      BRCLR   L0037, #80, LB5E9 ; BR IF NOT b7, 1 = ENGINE RUNNING
      ; ... else

      LDAA     $828D      ; 28%, MAP DEFAULT COEF, (NOT RUNNING)
      LDAB     L0095      ; TPS Ld val
      MUL      ; COEF * TPS Ld
      ASLD     ; MULT * 2
      BCS      LB5E6      ; IF OVER FLOW, USE DEFAULT MAP
      ; ... else

      ASLD     ; IF OVER FLOW, USE DEFAULT MAP
      BCS      LB5E6      ; ... else

      PSHA     ; SAVE RESULT TO STX

      LDAA     L0085      ; Last(+1) Vss count
      LSRA     ; DIV BY 2
      CMPA     #64        ; LMT TO 3200 RPM FOR LK UP
      BLS      LB5D7      ; BR IF L.T. or E.Q 64
      ; ... else

      LDAA     #64        ; USE 3200 RPM FOR LK UP

;-----
; LK UP MAP DEFAULT BAIS vs RPM
; 5 LINE TBL
;-----
LB5D7: LDX      #828E      ; INDEX MAP DEFAULT BAIS TBL
      JSR      LE3D0      ; 2d lk up, no off set

      PULB

      ABA      ; ADD B TO A
      BCS      LB5E6      ; IF OVER FLOW, USE DEFAULT MAP
      ; ... else

      CMPA     $828C      ; MAP DEFAULT if ERR 34, 85 Kpa
      BLS      LB5E9      ; BR IF E.Q. or L.T. TO DEAFULT

LB5E6: LDAA     $828C      ; 84.5 Kpa, DEFAULT if ERR 34
LB5E9: STAA     L0074      ; CURRENT A/D MAP

;
; FILTER OLD/NEW MAP VALUE
;
      LDAB     $86DC      ; 93.75%, MAP FILTER COEF,
      LDX      L0076      ; GET OLD MAP
      JSR      LE31C      ; LAG FILTER ROUTINE

      STD      L0076      ; NEW Filt A/D MAP MAP

;-----
; MAKE MAP LD VALUE
;

```



```

; MAP LD = ((AD MAP/694) * 65536) + 2647)/256
;-----
TAB                                ; A/D MAP fm A
CLRA
LDX      #694
FDIV
PSHX
PULA
PULB
ADDD     #2647                      ;
STD      L0079                      ; MAP for Spd/Dens calc
;-----

;-----
; MAKE  NORMILIZED LOAD MAP VAL
;
; RESULT = 1.18 * A/D MAP-26)
;-----
LDAB     #151                        ;
LDAA     L0076                      ; FILT A/D MAP VAL
SUBA     #26                          ;
JSR      LE5F3                      ; MAKE PRESSURE LOAD VAR'S

STAA     L0071                      ; NORMILIZED LOAD MAP VAL

;-----
; FILT MAP FOR ACELL ENRICH
;
;-----
LDX      L00A6                      ; OLD MAP
LDAA     L0071                      ; NORMILIZED LOAD MAP VAL
LDAB     $857C                      ; COEF, Acell Enr MAP Filtr
JSR      LE31C                      ; LAG FILTER ROUTINE

CMPA     L0071                      ; NORMILIZED LOAD MAP VAL
BLS      LB61E                      ; IF RESULT E.Q. or L.T.
; ... else
LDAA     L0071                      ; USE NORMILIZED LOAD MAP VAL
LB61E:  STD      L00A6                      ; SAVE FILT MAP FOR ACEL ENRICH

;-----
; MAKE MAP/BARO PRESS RATIO
;
;-----
LDAB     L0075                      ; MAP A/D RESULT
CLRA
LDX      #694
FDIV
PSHX
PULA
PULB
ADDD     #2647                      ; 1 Kpa/BIT
LDX      L007B                      ; 1 Kpa/BIT BARO
FDIV     ; CALC MAP/BARO, (0-1)
PSHX     ; RESULT TO STX

PULA
INS
STAA     L007D                      ; SAVE BARO/MAP PRESS RATIO

;-----
; VACUUM VARIABLES
;

```

```

; CURRENT MANIFOLD VACUUM = 1.18 * (246 - A/D VAC)
;-----
LDAA    L002E                ; A/D BARO
SUBA    L0076                ; BARO - MAP
BCC     LB63C                ; BR ID NO UNDER FLOW
                        ; ... else
                        ; A = 0
LB63C:  ADDA    #29           ;
                        ;
                        ; IF NO OVERFLOW
                        ; ... else
LDAA    #255                ; MAX VAL
LB642:  STAA    L007E        ; SAVE A/D VACUUM

LDD     #$F697              ; LD A = 246
                        ; LD B = 1.18 + 128
SUBA    L007E                ; 246 - A/D VAC
JSR     LE5F3                ; TO PRESS LD SUBROUTINE

STAA    L007F                ; CURRENT MANIFOLD VACUUM
;-----

;-----
; CALCULATE EGR PARTIAL PRESSURE
;
;-----
CLRA
LDAB    L0123                ; LK'ED UP EGR D.C.
BEQ     LB66E                ; IF Z, EXIT

LDAA    L007F                ; CURRENT MANIFOLD VACUUM
LSRA                    ; SCAL FOR LK UP, (DIV by 2)

;-----
; LK UP EGR PART PRESS vs VAC
;
;
;-----
LDX     #$86E2                ; INDEX EGR Part Press vs VAC TBL
JSR     LE3D0                ; 2d lk up, (No off set)

MUL                    ; EGR D.C. * EGR PARTIAL PRESS
ADCA    #0                   ; ROUND OFF
LSRD                    ; DIV by 8
LSRD
PSHB                    ; SAVE RESULT TO STX
PSHA
TSX                    ; STX to X Reg
LDAA    L0124                ; EGR D.C.
JSR     LE33F                ; 8x16 MULT SUB ROUTINE

PULX                    ; GET RESULT FM STX
ADCA    #0                   ; ROUND OFF
LB66E:  STD     L0069        ; SAVE PARTIAL PRESS DUE TO EGR

;-----
; CALC EGR PART PRESSURE
; IN MAP LD VALUE
;-----
ASLB                    ; MULT by 2
ADCA    #0                   ; ROUND OFF
PSHA                    ; SAVE UPPER NIBBLE RESULT

LDAB    #$33                ; UPPER NIBBLE RESULT * 51
                        ; (333/256)
MUL
ADCA    #0                   ; ROUND

```

```

PULB                                ; GET PART PRESS RESULT FM STX
ABA                                  ; B + UPPER NIBBLE * 3
ABA
ABA
TAB                                  ; A -> B Reg
LDAA    L0071                        ; NORMILIZED LOAD MAP VAL
SBA                                  ; EGR P/P - MAL LD
BCC     LB684                        ; BR IF NO OVERFLOW
                                           ; ... else
CLRA                                       ; A = 0
LB684: STAA    L0078                  ; EGR CORR MAP VALUE

;-----
; 12.5 MSEC FUEL
;-----
BRSET   0, #01, LB6A4                ; BR IF b0, EXE FUEL,
                                           ;   ODD MINOR LOOP'S
                                           ; ... else
BCLR    L003C, #08                  ; CLR b3,  BACKUP FUEL Flag

;-----
; ALDL MODE 4,  COMMAND BACKUP FUEL
;
; CNT'L WD $0194, b..
;
; ANHT/4298 P/N 16134297
; ECM TYPE $8D, ECM P/N 1227727
;
;-----
BRCLR   L0048, #01, LB69E           ; BR IF NOT b0, MODE 4
                                           ; ... else
LDAA    L0194                        ; CNT'R WD
BPL     LB69E                        ; BR IF ...
                                           ; ... else
BSET    L003C, #08                  ; SET b3, BACKUP FUEL Flag
BSET    L0039, #01                  ; SET SKIP ERR 42 FLAG

BRA     LB6A1                        ; GO EXE SPK ADVANCE
;-----

;-----
; COP 2 TOGGLE
; Every odd minor loop, 12.5 MSEC
;-----
LB69E: JSR     LE2F4                  ; Toggle COP2 to  FMD

*****
* EXECUTE 12.5 Msec SPK
*****
LB6A1: JMP     LB6F9                  ; MINOR LOOP, EST SPK TMG

*****
* EXECUTE 12.5 Msec FUEL
* >> TO MAJOR LOOP EXE <<
*****
LB6A4: JMP     LC4F9                  ; TO Air Fuel MAJOR LOOP

LB6A7: BRCLR   L0000, #1, LB6B0      ; BR IF NOT b0, TO MAJOR LP EXEC

```

```

LDAA    #$FF                ; Set to arm COP
STAA    $400B              ; COP ARM
*****

```

```

*****
* MAJOR LOOP EXEC, TYPE $8D ECM
*
* Schedules 100 Msec into 6.25 Msec RT interupt dead
* time, Minor loop (6.25 Msec returns here)
*****

```

```

ORG $B6B0

```

```

LB6B0: LDAB    L0000                ; MINOR LOOP COUNTER
        ANDB   #$0F                ; MASK 0000 1111, (0 - 15)
        LDX   #LB6D8              ; SEGMENT TABLE
        ASLB                   ;
        ABX                       ; ADD LP CNT TO X FOR ADDR POINTER
        LDX   0,X                 ; GET CURRENT SEGMENT ADDRESS
        JSR   0,X                 ; JSR TO SLECTED SEGMENT ROUTINE

```

```

        BCLR  L0039,#$04           ; CLR b2, CLR TIMING Error

```

```

LB6C0: CLI                       ; TURN ON INTERUPTS

```

```

;-----
; RAM Refresh $0121 th $0000
;-----

```

```

LDX     L0121                    ; RAM Loc
BEQ     LB6D3                    ; Do RE-FRESH

```

```

LB6C6: DEX                       ;
        DEX
LB6C8: SEI                       ; DISABEL INT'S WHILE RE-FRESH
        LDD   0,X                 ; GET OLD DATA
        STD   0,X                 ; REFRESH LOCATION
        CLI                       ; ENABLE INT'S
        STX   L0121
        BNE   LB6C6              ; BR IF N/Z

```

```

;         ; .... else
LB6D3: LDX     #L01C2              ; STACK OVER WRITE TEST ADDR
        BRA   LB6C8              ;

```

```

;-----
; SEGMENT TABLE
; TYPE $32
;-----

```

```

LB6D8: FDB     $CE5C              ; SEG 0, MAKE A/D BARO FROM MAP
        FDB     $CEBD              ; SEG 1, DESCRETE & PWM OUTPUT ROUTINE
        FDB     $D121              ; SEG 2, OIL TEMP
        FDB     $D270              ; SEG 3, LOG RAM TO I^2U
        FDB     $D161              ; SEG 4, MISC 100 Msec ROUTINE
        FDB     $D27D              ; SEG 5, A/C, C/L, TEST OF FAN
        FDB     $E4B8              ; SEG 6, LOG RAM TO CPU, COOLANT, A/D CON
        FDB     $D50C              ; SEG 7, LK UP COOL VAR'S
        FDB     $D55C              ; SEG 8, ESC LOW OCTANE SPK MOD LOGIC
        FDB     $D5C6              ; SEG 9, AIR MANAGMENT
        FDB     $D6C1              ; SEG A, KNOCK, MAT IAC VAR'S, ETC...
        FDB     $D812              ; SEG B, EGR
        FDB     $D87D              ; SEG C, CCP
        FDB     $D9FC              ; SEG D, DIAGNOSTICS
        FDB     $DE49              ; SEG E, BATT VDC, TCC, CARS, E LITE
        FDB     $E1C6              ; SEG F, AIR/FUEL MAJOR

```

```

RTS

```

* MINOR LOOP, EST SPK TMG
* XMIT SERIAL DATA
*
* EXECUTED EVERY OTHER IRQ1, (25 Msec)
*

ORG \$B6F9

```
LB6F9: LDAA    L0038          ; MODE WD 2
      ANDA    #$04          ; CLR BIT 2, DRP THIS 6.25 ms PRTIOD
      ORAA    L0041          ;
      STAA    L0041          ; DIAG MOD WD 2

      LDAA    L006B          ; LIMITED AIR FLOW FM IDEAL GAS LAW
      CMPA    #64           ; CK UPPER LIMIT
      BLS     LB709          ; BR IF L.T. or E.Q. 64
                          ;... else
      LDAA    #$64          ; LIMIT TO 64

      ;-----
      ; LK UP SLOW o2 FILT COEF  vs AIR FLOW
      ;
      ; TBL = Mult  * 256
      ;-----

LB709: ASLA                    ; MULX * 2
      LDX     #$84C6          ; SLOW o2 FILTER COEF vs FLOW TBL
      JSR     LE3D0          ; 2D LOOK UP, NO OFF SET

      PSHA                    ; FILT COEF TO STX

      LDAA    L005D          ; FLT COOLANT TEMP

      LDAB    $84A0          ; 0.6 COEF FOR o2 FILTER CONSTANT
      MUL
      ADDA    L005D          ; FLT COOLANT TEMP
      BCC     LB71D          ; IF NO OVERFLOW
                          ;... else
      LDAA    #255           ; MAX VAL

LB71D: PULB                    ; GET SLOW o2 vs FLOW RESULT FM STX
      MUL
      TAB
      LDAA    L008E          ; A/D 02
      LDX     L0090          ; A/D O2 AF SC
      JSR     LE31C          ; LAG FILTER ROUTINE

      STD     L0090          ; SAVE FILTERED A/D o2 RESULT

      BRCLR   L0038,$#04,LB746 ; MODE WD 2, b2, BR if no ref
                          ; - in last 12.5 msec

      LDD     L3FC8          ; PA2 CNT'R, SPK Period
      SUBD    $00D5          ; OLD PA2 CNT'R
      TSTA
      BNE     LB73A          ; BR IF N/Z
                          ; .... else
      CMPB    $8296          ; 1048 Msec PULSE WIDTH THRESH
      BLS     LB742          ; BR IF PW L.T. 1048 Msec
                          ; ... else

LB73A: INC     L00D7          ; INCR EST FB CNT'R, (Err #42)
      BNE     LB742          ; BE IF CNT'R NZ
                          ; ... else
```

```

DEC      L00D7                ; EST FB CNT'R, (Err #42)

LB742:  ADDD   L00D5                ; OLD PA2 CNT'R
        STD    L00D5                ; OLD PA2 CNT'R

LB746:  BRSET  L0037,#$80,LB757      ; BR IF b7, ENGINE RUNNING
        ; ... else
        LDX   L00B3                ; CURRENT MNR LOOP DRP PERIOD
        CPX   $801A               ; 300 RPM, START UP RPM
        BCS   LB757                ;
        ; ... else

        BCLR  L0038,#$04           ; MODE WD 2, b2, DRP THIS LOOP
        CLRA                      ; A = 0
        BRA   LB78B

LB757:  BRCLR  L0038,#$04,LB79B      ; BR IF NOT b2, NO DRP'S THIS LOOP
        ; (MODE WD 2)
        ; ... else
        BCLR  L0038,#$04           ; CLR b2,
        ; ... and

;-----
; CALC DYNAMIC DWELL
; = (DYNAMIC DWELL/8) +1
;-----
LDD     L00B7                ; DYNAMIC DWELL
LSRD                    ; DIV by 8
LSRD
LSRD
COMA
COMB
ADDD    L00B7                ; DYNAMIC DWELL + 1
BPL     LB76C                ; BR IF NO UNDER FLOW

LB76C:  LDD     #0                ; ELSE USE 0
        STD    L00B7                ; SAVE DYNAMIC DWELL

        CLR   L00BF                ; NUM MNR LP'S BETWEEN DRP'S

        BRSET L0037,#$80,LB7B7      ; BR IF b7, ENGINE RUNNING
        ; ... else
        BRCLR L0036,#$08,LB7AA      ; BR IF NOT b7, 1st DRP

        LDAA  L005D                ; FLT COOLANT TEMP
        LSRA                      ; DIV BY 2

;-----
; LK UP HI RPM DRP'S REQ FOR ENG RUN vs COOL
;
; TABLE = DRP'S
;-----
LDX     #$801C                ; HI RPM DRP'S REQ FOR ENG RUN vs COOL TBL
JSR     LE3D0                ; 2D LOOK UP, NO OFF SET

CMPA    L0148                ; DRP'S AT QUAL'ED RPM
BLS     LB790                ; BR IF QUAL BR HI DRP VAL
; ... else
LDAA    L0148                ; GET DRP'S AT QUAL'ED RPM CNT'R
INCA                      ; INCR CNTR
LB78B:  STAA   L0148                ; SAVE DRP'S AT QUAL'ED RPM CNT'R

```

```

BRA      LB7AD                      ; ENG RUNNING
;-----

LB790:   BSET      L0037,#$80        ; SET b7, ENG RUNNING FLAG
        BCLR      L003C,#2          ;
        BSET      L0001,#8          ; SET b3, BAD SHUT DN

        BRA      LB7B7              ; EST MINOR LP, (RUNING)

;
; WAIT FOR ENG TO BE RUNNING
;
LB79B:   LDAA      L00BF              ; NUM MNR LP'S BETWEEN DRP'S
        CMPA      #23                ;
        BCS      LB7B0              ; BR IF L.T. 23
; .... else

        INX
        LDX      #$FFFF             ; CLR CNT REG
        STX      L3FC0              ; Last Ref period

;-----
; LOOP HERE TILL
; NEXT INTERUPT
; SWI VECTOR TO $6000
;-----
LB7A7:   SWI
; WAIT FOR INTRUPT,

        BRA      LB7A7

LB7AA:   BSET      L0036,#8          ; SET b7, 1ST DRP BIT

LB7AD:   JMP      LBD9B

LB7B0:   BRCLR    L0037,#$80,LB7AD   ; BR IF NOT b7, 1 = ENG RUNNING
; {BACK IF ENG NOT RUNNING}
; ... else
        INCA
; BUMP CNT'R
        STAA     L00BF              ; NUM MNR LP'S BETWEEN DRP'S

*****
* END OF IRQ1 ROUTINE
*
*****

*****
* EST Minor loop
*
* Eng running logic
*
*
*****
LB7B7:   LDAA      #0                  ; CLR COP
        STAA     L400C              ; CPU COP

;
; 8 cyl equiv RPM val
;
        LDX      #L00B3             ; CURRENT MNR LOOP DRP PERIOD

```

```

LDAA    $8009                ; 0 = 8 CYL
BNE     LB7C8                ; BR if Cyl NE 8
                        ; .... else
LDD     L00B3                ; CURRENT MNR LOOP DRP PERIOD
BRA     LB7CB                ; Save cnts

LB7C8:  JSR     LE33F        ; No, scale to 8 cyl ref
                        ; (8 x 16 MULT)

LB7CB:  STD     L0467        ; ECM IM RAM, NUM OF 65.5Khz cnts
                        ; between DRPs

;-----
; RPM, RPM P, RPM/25 & RPM/12.5
;
;      Q = (153.6/RPM RER) * 2 ^8
;
; PER = 1/65.5 Khz per bit
;-----
PSHB                    ;
PSHA                    ;
                        ;
ASLD                    ; SET DIVIDE OPER
PSHB                    ;
PSHA                    ;
                        ;
PULX                    ;
LDD     #0307            ; 15 * (512/25) or 153.6 * 2
FDIV                    ; do division
PSHX                    ;
TSX                    ;
LDD     0,X              ;
CMPA   #96              ; 2400 RPM
BCS    LB7E9            ; Br IF RPM <= 2400 RPM, (LO RANGE)
                        ; .... else
ADDD   #16512           ; ROUND Q & ADJ to HI RANGE (2400-4800)
BCC    LB7F0            ; IF IN HI RANGE RPM/25
                        ; ... else
LDAA   #255             ; If > 4800 RPM limit to 4800, (255d)
BRA    LB7F0            ;
                        ;

;-----
; Mult by 2 in lo range, 1 bit = 12.5 RPM
; Round (2Q) and adj for lower range
;-----
SUBD   #8064            ;
                        ;
LB7E9:  ASLD                    ; x2, RPM.12.5
BCC    LB7F0            ;
                        ; ... else
CLRA                    ; CLEAR RPM

;-----
; (-400 RPM ($20) + 128 to round)
; 0 CTS = 400 RPM
;-----
LB7F0:  STAA   L0057        ; RPM/25

PULA                    ; Pull RPM/25
PULB

PSHB                    ;
PSHA                    ;
ADDD   #128             ; Round off
BCC    LB7FD            ; BR if no overflow
                        ; ... else

```



```

LDAA    #255                ; limit to 255
LB7FD:  STAA    L0085        ; Last(+1) Vss count

;-----
; MAKE UN-FILT RPM/12.5
;-----
PULA                                ;
PULB                                ;
ASLD                                ; MULT * 2
BCS     LB809                ; IF OVERFLOW LIMIT TO 255, (3187 RPM)
; .... else
ADDD    #128                ; ROUNDED
BCC     LB80B                ; IF NO OVERFLOW
; ... else
LB809:  LDAA    #255        ; 3187 RPM
LB80B:  STAA    L0059        ; SAVE UNFILT 12.5 RPM Var

;-----
; Compute FILTERED RPM Var
; FORKNOCK & IAC USE
;-----
LB80D:  CLR    CLRB          ; 1st TIME thru FILTERED RPM = RPM/12.5
LDX     L0059                ; RPM/12.5, (UN-FILT)
BEQ     LB818                ; IF 0, EXIT

LDAB    $8628                ; IAC RPM/12.5 FILTER COEF, (50%)
JSR     LE31C                ; Do lag filt

LB818:  STD     L0059        ; Filt RPM/12.5

;-----
; COMPUTE FILTERED RPM &
; DELTA Map A.E.
;-----
LDAA    L0059                ; RPM/12.5
CLR    CLRB                  ; 1st TIME THRU FLT RPM AE = RPM/12.5
LDX     L013D                ; GET OLD FILT DELTA MAP A.E.
; RPM FILTER COEF
BEQ     LB828                ; Br IF FIRST TIME THRU

LDAB    $85D7                ; FILTER COEF
JSR     LE31C                ; GO DO LAG FILT

LB828:  STD     L013D        ; SAVE NEW FILTERED VALUE

;-----
; STALL SAVER SPARK
;-----
BRCLR   L0001,#$80,LB834     ; BR IF NOT b7, NO EST ERR, (42)
; (SKIP STALL SVR
; ... else
BCLR    L0039,#3             ; CLR STALL SVR BITS 0 & 1
BRA     LB89B

LB834:  CMPA    $802A        ; Enab Stall Saver thresh,
BCS     LB845                ; Br if RPM < L.T. 300 RPM
; ... else
CMPA    $802B                ; 300 RPM, IF RPM < & RPM >
; ENABLE STALL SAVER SA
BCC     LB869                ; BR, NO STALL SAVER SA
; ... else
LDAB    L005C                ; OLD RPM (FOR DIFF RPM)
CMPB    $802C                ; Stall Saver RPM THRESH
BLS     LB869                ; Br if RPM LT

;-----

```

```

; ENABLE STALL SAVER
;-----
LB845: BRSET    L0039,#$02,LB866          ; Br IF IN SS Spark

      CLR      L00D7                      ; EST FB CNT'R, (Err #42)
      LDD      L3FFC                      ; CPU CR
      ANDB     #$EF                        ; CLR b4
      JSR      LE4B7                      ; Short delay

      STD      L3FFC                      ; CPU CR

      BCLR     L0033,#4                   ; CLR b2, EST ENABLE BIT, OFF

      SEI
      JSR      $E2FF                      ; TURN OFF INTERRUPT
                                          ; FMD SERIAL BYTE READ

      CLI

      LDD      L3FC8                      ; Get SPK Period, (PA2 CNT'R)
      STD      L00D5                      ; OLD PA2 CNT'R

      LDAA     L0059                      ; RPM/12.5

LB866: BSET     L0039,#3                   ; Set b0 & B1, STALL SAVER SPK FLAG

;-----
; NO STALL SAVER SA
;-----
LB869: BRCLR   L0039,#2,LB89B            ; Br IF NOT B1, IN STALL SVR SPARK

;
; Disable Stall SVR Lmt, ( ... RPM)
;
      CMPA     $802D                      ; STALL SVR RPM THRESH
      BLS     LB89B                      ; BR IF RPM GT ... rpm

      BCLR     L0039,#3                   ; Clear Stall SVR SPARK FLAG, b0 & b1
      INC     L00D7                      ; EST FB CNT'R, PREVENT ERR 42

;
; ZERO VAL'S
; EST DWELL, EST PULSE WIDTH
;
      CLRA
      CLRB
      STD     L3FDC                      ; TO SPK Dwell Cnt'r
      JSR     LE4B7                      ; SHORT DELAY

      STD     L3FE6                      ;
      JSR     LE4B7                      ; SHORT DELAY

      STD     L3FE8                      ; CURRENT EST FALL - LAST EST FALL
      JSR     LE4B7                      ; SHORT DELAY

      STD     L3FF6                      ; EST FALL CNT'R
      JSR     LE4B7                      ; SHORT DELAY

      LDX     L3FEC                      ; B CNT'R LAST DRP
      JSR     LE4B7                      ; SHORT DELAY
      STX     L3FE4                      ; B CNT'R START NEXT DWELL

*****
* Compute Dwell
*****
;-----
; IF LOAD CHANGE DURING LAST

```

```

; 25 msec > 10 Kpa SET DWELL
; to $FFxx & Br to LB8B2
;-----
LB89B: LDAA    #255                ; NEW DWELL MSB

LDAB    L0071                ; New MAP Load,  NORMILIZED LOAD MAP VAL
SUBB    L0073                ; NORMILIZED LOAD MAP VAL, 25 Msec OLD
BCS     LB8A7                ; BR IF OLD G.T. NEW
; ... else
CMPB    #32                  ; Diff MAP > 10 Kpa
BCC     LB8B2                ; br if G.T.
; ... else

;-----
; IF DELTA RFPER > DYNAMIC DWELL
; set  DYNAMIC DWELL = -2 * DELTA RFPER
;-----
LB8A7: LDD     L00B5                ; OLD REF PER
SUBD    L00B3                ; CURRENT MNR LOOP DRP PERIOD
ASLD                    ; DIV by 2
SUBD    L00B7                ; DYNAMIC DWELL
BMI     LB8B4                ;

LB8B2: ADDD    L00B7                ; DYNAMIC DWELL
STD     L00B7                ; DYNAMIC DWELL

LB8B4: LDD     L0071                ; NORMILIZED MAP LD Var,
STD     L0072                ; NORMILIZED MAP LD Var, (12.5ms old)

;-----
; Limit  DYNAMIC DWELL to
; REF PERIOD/8, (11 DEG FOR V8)
;-----
LDD     L00B3                ; CURRENT MNR LOOP DRP PERIOD
LSRD                    ; DIV BY 8
LSRD
LSRD
SUBD    L00B7                ; DYNAMIC DWELL
BCC     LB85C                ; Br if DYNAMIC DWELL > RFPER/8
; ... else

;
; IF DYNAMIC DWELL > RFPER/8 set  DYNAMIC DWELL to RFPER/8
;

ADDD    L00B7                ; DYNAMIC DWELL
STD     L00B7                ; Save new DYNAMIC DWELL

LB85C: LDD     L00B3                ; CURRENT MNR LOOP DRP PERIOD
STD     L00B5                ; OLD REF PER

LSRD                    ; DIV BY 2
SUBD    #229                ; 3.49 Msec, (15.25 Msec/bit)
BCC     LB8D4                ; BR IF D G.T. 3.5 Msec,
; ... else
ADDD    #308                ; ADD 4.697 Msec

BRA     LB8E4                ; EXIT via BATTERY COMP

LB8D4: LSRD                    ;
SUBD    #2957                ; 4.5 Msec
BCS     LB8DF                ; BR IF D L.T. 4.5 Msec
; ... else
ADDD    #382                ; ADD 5.825 Msec

```

```

BRA      LB8E4                ; EXIT via BATTERY COMP

;-----
; D = 1/4 * (D + 23.3 msec)
;-----
LB8DF:   ADDD      #1527        ; 23.3 Msec
        LSRD
        LSRD                ; DIV BY 4

*****
* BATTERY VOLTAGE COMPENSATION
*
*****

LB8E4:   PSHB
        PSHA
        LDAA      #124         ; 12.4 VDC
        SUBA      L0103        ; A/D BATTERY VOLTAGE
        BCC       LB8EE        ; BR IF BATTERY L.T. 12.4 V
                                   ; ... else
        CLRA                ; NEGITIVE RESULT

;-----
; LOW BATTERY DWELL
; If Batt < 12.4 VDC
;-----
LB8EE:   LDAB      #4          ;
        MUL                ; 630 usec/Volt
        TSX
        ADDD      0,X
        PULX
        ADDD      L00B7        ; DYNAMIC DWELL
        STD       L00B9        ; DWELL TO CPU, (15.25 Msec/BIT)

        LDD       L00B3        ; CURRENT MNR LOOP DRP PERIOD
        SUBD      #39          ; LIMIT NEW DWELL to (REFPER - 600 MSEC)
        SUBD      L00B9        ; DWELL TO CPU, (15.25 Msec/BIT)
        BCC       LB906        ; EXIT
                                   ; ... else
        ADDD      L00B9        ; DWELL TO CPU, (15.25 Msec/BIT))
        STD       L00B9        ; DWELL TO CPU, (15.25 Msec/BIT)

*****
* MAIN SPARK ADVANCE
*
*****
ORG $B906

LB906:   LDX       #$8033      ; POINT TO MIN SPK TBL

        *****
        * 400 RPM - 4800 RPM
        * MAIN SPK TBL
        *****
        BRSET     L0001,$$40,LB924 ; Br IF WARM Kickdown enabled

        LDAB      L005D        ; FLT INV COOLANT TEMP
        CMPB      $8033        ; COOL LOW ENOUGH FOR COLD SPARK ?
        BCC       LB91D        ; Br aROUND LOGIC IF NOT COLD ENOUGH
                                   ; ... else
        SUBB      L0060        ; CALC DELTA COOL TEMP SINCE START UP
                                   ; TEMP, (DEG C+40) * (256/192)
        BCS       LB920        ; Br If DELTA NOT BIG ENOUGH
                                   ; .... else
        CMPB      $8022        ; TEST POS DELTA
        BCS       LB920        ; Br IF DELTA NOT BIG ENOUGH

```

```

; .... else
LB91D:  BSET      L003C,#$40      ; WARM KICK Dn REQ

LB920:  LDAB      #176            ; Max LOAD AXIS ON 1ST WARM PASS
      BRA       LB946            ;

;-----
; DETERMINE IF CLOSED THROTTLE
; SPARK ADVANCE OR NOMAL SPK ADV
;-----
LB924:  LDAA      L0095            ; TPS Ld val
      CMPA      $8647            ; 1.2% TPS
      BHI       LB93B            ; IF TPS > 1.2% do normal SPK

;-----
; Closed throttle
; main spark advance
;-----
      LDAA      L0058            ; RPM/25
      CMPA      #192            ; RPM Limit for lkup, 4800 RPM
      BLS       LB933            ; IF L.T. 4800 RPM
      ; ... else
      LDAA      #192            ; USE 4800 RPM FOR LK UP

;-----
; LK UP CLOSED THROTTLE S.A. vs RPM
;
;-----
LB933:  LDX       #$81FD          ; CLOSED THROTTLE S.A. TABLE, (2d)
      JSR       LE3D0            ; (SPK ADV vs RPM)
      ; Go do LK UP
      ; 2D LOOK UP, NO OFF SET
      BRA       LB95C            ; STORE S.A.
;-----

*****
* NORMAL SPK ADV
*
*****

;-----
; PROCESS MAP for
; BRK POINTS
;-----
LB93B:  LDAB      L0078            ; LINEARIZED MAP
      CMPB      #160            ; > 70 Kpa MAP
      BCC       LB944            ; Br IF YES.. MORE WORK TO DO
      ; ... else
      LSRB      ; PERFORM SCALING, DIV / 2
      BRA       LB946            ; .. & Br to do LK UP

LB944:  SUBB      #80             ; SUB OFFSET FOR CHANGE TO 5 Kpa
      ; Bk POINTS

;-----
;-----
; PROCESS RPM for
; tbl CHOICE
;-----
LB946:  LDAA      L0057            ; RPM/25
      CMPA      #192            ; ABOVE 4800 RPM ?

```

```

BCS      LB957      ; BR IF L.T. 4800 RPM
           ; ... else
CMPA     #224      ; LIMIT RPM to 5600 for LK UP
BLS      LB952
           ; ... else
LDAA     #224      ; FORCE TO 5600 RPM

LB952:  LDX      #$8102

*****
* IF RPM G.T. 4800 RPM
* 4800 RPM - 5600 RPM
* SA EXT TABLE
*****

BRA      LB959      ; TO SPK ADV LK UP

LB957:  LDAA     L0057      ; RPM/25
LB959:  JSR      LE353      ; 3D LK UP

LB95C:  STAA     L0456      ; Save Main spark advance, (HU ADDR)
PSHA
*****

*****
* COOLANT SA COMP
*
*****
LDAA     L005D      ; FLT COOLANT TEMP
CMPA     #208      ; Limit COOL DEG FL to 115 C
BLS      LB968      ; BR IF COOL LT 115c

LDAA     #208      ; FORCE TO 115c FOR LK UP

*****
* COOL SA COMP TABLE
*****

LB968:  LDX      #$8138

LDAB     L007F      ; CURRENT MANIFOLD VACUUM
LSRB
JSR      LE353      ; SCALE 10 Kpa PER TABLE BK Pt
           ; GO LK UP VAL

STAA     L0457      ; COOL COMP SPK ADV FM 3D LU (HU ADDR)
PSHA
           ; SAVE COOL SA ON STX
*****

*****
* HIGHWAY MODE S.A.
*
*****

;
; CK AFR OPT WD 4, 0000 1001
; b4, 1 = TCC LOCKED HIWAY S.S. CHK ENAB
;
LDAB     $8019      ; AFR OPT WD 4
BITB     #$10      ; b4
BEQ      LB980      ; BR IF NOT b4
           ; ... else
BRCLR    L004A,$$01,LB9AF ; Br if TCC NOT LOCKED

```

```

;
; CK AFR OPT WD 4, 0000 1001
; b2, 1 = 4Th GR HIWAY MODE SA CHECK
;
LB980: BITB    #$04          ; b2
      BEQ     LB98C        ; Br if not b2
                                ; ... else
      LDAA   $0140        ; GEAR RATIO, HIGEST GEAR
      CMPA   $82FE        ; 4TH GR, MIN RATIOH
      BHI    LB9AF        ; No, zero Hy way mode spark
                                ; ... else
LB98C: LDAA   L0079        ; MAP for Spd/Dens calc
      CMPA   $81A8        ; UPPER LIMIT FOR HI WAY MODE sa
      BHI    LB9AF        ; Br if MAP > limit
                                ; ... else
      LDAB   L005D        ; FLT Coolant temp
      CMPB   $81A7        ; Lo COOLANT LIMIT FOR hw sa
      BLS    LB9AF        ; Br if Cool to low
                                ; ... else
      LDAB   L0057        ; RPM/25
      CMPB   $81A9        ; 4762 RPM Lo LIMIT FOR HIWAY SA
      BCS    LB9AF        ; Br if RPM to lo
                                ; ... else
      LDAB   L00C5        ; HIWAY MODE TMR
      CMPB   $81AA        ; 10 SEC, ENABLE IF COND'S EXIST
                                ; ...THIS PERIOD
      BHI    LB9B5        ; IF ALL TESTS PASSED
                                ; ... else
      LDAA   $0000        ; NINOR LP CNTR
      BNE    LB9B0        ; INC HI WAY MODE TIMER EVERY SEC
                                ; ... else
      INCB
      BRA    LB9B0        ; INCR HIWAY MODE TMR

LB9AF: CLRB          ; CLEAR HI WAY MODE TIMER
LB9B0: STAB    L00C5    ; Update HI WAY MODE TIMER
      CLRA

      BRA    LB9C5        ; EXIT
;-----

;-----
; Adj SA by EXTRACTING S.A. HIWAY MODE
;
; else SET S.A. HM = 0
;-----
LB9B5: LDAA   L0078        ; EGR CORR MAP VALUE
      CMPA   #144        ; CK MAX VAL
      BLS    LB9B9        ;
                                ; ... else
      LDAA   #144        ; USE MAX VAL

;-----
; LK UP HIWAY SA vs MAP
;
;-----
LB9B9: LDAB   #32          ; MAX MAP LD FOR LK UP
      LDX   #$81AB        ; POINT TO TBL
      JSR   LE3CC        ; 2D LK UP

LB9C5: STAA   L0458        ; HI WAY SA, (HU RAM)
      PSHA          ; SA FOR PGM

```

```

;-----
;-----
; HOT RE-START RETARD
;
;-----
BRSET    L0001, #20, LB9DB          ; Br if IAC kick dn enab
; ... else
LDAB     L0060                      ; START UP COOLANT THRESH
CMPB     $802F                      ; IF COOL LT 45c, DISABLE HOT RE START
BCS      LB9DB                      ; BR IF S/U COOL LT 45c
; .... else
LDAB     L001A                      ; ENG RUN TIME TIME + 1
SUBB     $8672                      ; IF ENG RUN TIME < 25 SEC ENABLE
; HOT SA RETARD

BCS      LB9E1                      ; BR IF RT LT
; .... else
LB9DB:   CLRB                       ;

BSET     L0001, #20                 ; SET, IAC KICK DN ENABLE FLG

BRA      LB9E4                      ;

LB9E1:   LDAB     $8030              ; HOT RE-START SPK RETARD, 0.4 DEG SA

LB9E4:   BCLR     L003A, #8          ; CLR HOT RESTART RETARD BIT
STAB     L0459                      ; SAV SKP FOR HU
PSHB     ; SAV SKP FOR PGM

BEQ      LB9F0                      ;

BSET     L003A, #08                 ; SET HOT RESTART RETARD ENABLE
;-----

*****
* PWR ENRICH SPARK ADVANCE
*
*****

LB9F0:   CLRA                      ; ASSUME NOT IN PWR ENRICH
BRCLR    L0046, #20, LB9FF          ; BR IF NOT b5, B5 1 = IN PWR ENRICH

LDAA     L0057                      ; RPM/25
LSRA     ; DIV BY 4, SCALE FOR LK UP
LSRA

;-----
; LK UP PWR ENR SPK ADV vs RPM
;
;-----
LDX      #$81B3                    ; SA vs RPM TABLE, (5 lines)
; (400 - 4800 RPM)
JSR      LE3D0                      ; 2D LOOK UP, NO OFF SET

LB9FF:   PSHA                      ; SAVE PWR RNRICH SA TO STX
;-----

;-----
; PWR ENRICH KNOCK TEST
; ERR 38
;-----
CLRB
BRSET    L0038, #10, LBA21          ; BR IF IN DIAGNOSTICS

```



```

; ... else
BRCLR   L0033,#4,LBA21      ; BR IF EST NOT ENABLED
; ... else
BRSET   L003A,#8,LBA21      ; IF HOT RE-START RETARD ACTIVE
; ... else
BRSET   L0002,#0,LBA21      ; IF ERR 43, (PRIOR)
; ... else
BRCLR   L0002,#1,LBA24      ; IF ERR 43 NOT IN WORK
; ... else
LDAA    L005D                ; FLT COOLANT TEMP
CMPA    $829C                ; IF COOL < 95.3C DISABLE TEST
BLS     LBA24                ;
; ... else
LDAB    L01A7                ; ERR 43 TEST S.A.
BRA     LBA24                ;

LBA21:  BCLR   L0002,#1      ; CLR TEST IN PROGRESS FLAG
LBA24:  PSHB                   ; VAVE EST TEST S.A.
;-----

*****
* SUM THE SPARK ADVANCES
*
*****
LDX     #0                    ; ZERO SPK ADV ACCUM
ABX                                         ; + TEST SPK ADV VAL

PULB                                       ;
ABX                                         ; + KNOCK TEST SA

PULB                                       ; + PE SA, (2D look up)
ABX                                         ;

PULA                                       ; THROW AWAY ???
PULB                                       ; + HY MODE S.A., (2D look up)
ABX                                         ;

PULB                                       ; + COOL COMP SPK ADV, (3D look up)
ABX                                         ;

PULB                                       ; + MAIN SA FM TBL 3D look up
ABX                                         ;

LDAB    L0012                ; + START UP vs COOLANT SPK ADV
ABX

PSHA                                       ; ??????
PSHX                                       ; WITH TOTAL SPK ADV
*****

*****
* IDLE S.A.
*
*****

;-----
; IDLE SPARK MODE DETERMINATION
;-----
BCLR    L0045,#$20            ; CLR b5, (MAP ACCEL ENRICH DONE 1ST TIME)
BCLR    L0039,#$08            ; CLR b3, (CLR IDLE SA LO GAIN ENAB FLG)

BRCLR   L0038,#$24,LBA4A      ; IF ERR 21 or 22

```

```

BRCLR    L0081, #\$FF, LBA89                ; IF Vss = 0

JMP      LBB00                              ; EXIT TO SPARK OUTPUT ROUTINES
; {BY PASS IDLE SPK RPM CNT'L}
;-----

;-----
; ALDL MODE 4,
;
; ANHT/4298 P/N 16134297
; ECM TYPE \$8D, ECM P/N 1227727
;
; CNT'L WD \$0198, b0 1 = MOD IAC
; CNT'L WD \$0198, b1 1 = RPM MOD IAC
;
;-----

LBA4A:   BRCLR    L0048, #\$01, LBA59        ; IF NOT IN ALDL MD 4
; ... else
LDAA     L0198                              ; CNT'L WD.., b0,
BITA     #\$01                              ; b0,
BEQ      LBA59                              ; IF NOT MOD'ING IAC
; ... else
BITA     #\$02                              ; CNT'L WD.., b1,
BEQ      LBA6B                              ; DISABLE IF IDLE S.A. FORCING IAC POSIT
; ... else

LBA59:   LDAA     L0095                      ; CURRENT TPS LD VAL
CMPA     \$8647                              ; TPS Thresh
BHI      LBA6B                              ; If TPS GT
; ... else
BRSET    L00FD, #\$04, LBA6E               ; BR IF b2, MOTOR RESET WORKING
; ... else
LDX      L0019                              ; Eng run time
CPX      #10                                ; 10 SEC'S
BHI      LBA6E                              ; IF LT 10 SEC'S SINCE START
; ... else
;

;-----

LBA6B:   JMP      LBB00                      ; EXIT TO SPARK OUTPUT ROUTINES
; {BY PASS IDLE SPK RPM CNT'L}
;-----

;-----
; IDLE SPK ADV LO GAIN
; QUALIFIERS
;-----

LBA6E:   LDAA     L0081                      ; FILTERED MPH
BNE      LBA7E                              ; IF Vss GT 0
; .... else
BRSET    L00FD, #\$10, LBA7E               ; BR IF b4, OLD PK/NEUT, LAST PID
LDAB     L0112                              ; REQUESTED ENG SPEED
CMPB     \$81E6                              ; 800 RPM, LO GAIN DESIRED RPM THRESH
BCS      LBA89                              ;
; ... else
;
; LO GAIN
;

LBA7E:   LDAB     \$81E7                      ; 1.2 SEC, LO GAIN XISITION DELAY TIME
STAB     L011C                              ;
BSET     L0039, #8                          ; SET IDEL SPK LO GAIN MODE
BRA      LBA96                              ;

LBA89:   LDAB     L011C                      ; LO GAIN XISITION DELAY TIME

```

```

        BEQ      LBA93                ; IF DELAY DONE
        DEC      L011C                ; ... else
        BRA      LBA96                ; DECR DELAY TIME
        ;
LBA93:  BSET     L0045,#$20            ; SET b5, SPK HI GIAN MODE ENABLE
        ;-----
        ; CALCULATE IDLE SPK ADV CNT'L RPM
        ;-----
LBA96:  LDAA     L0059                ; RPM/12.5
        LDAB     L0112                ; IAC CNT'L SPEED

        BRCLR    L0081,$$FF,LBAA6     ; IF Vss = 0, NO OFFSET
        ; ... else
        ADDB     $81EA                ; ADD 100 RPM REQUIRED RPM OFFSET
        BCC      LBAA6                ;
        ; ... else
        LDAB     #255                 ; MAX LIMIT
LBAA6:  SBA      ;

*****
* CALCULATE IDLE SPK ADV CORR BASED ON RPM DIFF'S
*
*****
;-----
; LK UP IDLE SA RETARD vs RPM
;-----
LDX     $$81EB                ; IDLE SA RETARD vs RPM
BCLR    L003E,#8              ; CLR b3, FOR OVERSPEED
BCC     LBAB6                  ; ... else
NEGA    ;                      ; ABS VALE OF SPEED ERROR
BSET    L003E,#8              ; SET b3 FOR UNDERSPEED

;-----
; LK UP IDLE SA  ADVANCE vs RPM ERROR
;-----
LDX     $$81F0                ; IDLE SA  ADVANCE vs RPM ERROR
LBAB6:  CMPA     #16              ; 200 RPM
        BLS     LBABC            ; IF L.T. or E.Q. 16d

        LDAA     #16              ; LIMT LK UP VAL TO 200 RPM

LBABC:  ASLA    ;                  ; SCALE RPM FOR Lk UP
        ASLA    ;                  ; Mult * 2
        JSR     LE3D0             ; 2D LOOK UP, NO OFF SET

        PSHA    ;                  ; SAVE IDLE SPD MODIFIER ON STX
        ;-----

;-----
; LOOK UP & DO MAP CORRECTION TO IDLE SPK ADV MOD
;-----

;-----
; LK UP IDLE SA RETARD MULT vs MAP
;-----
LDX     $$81F5                ; IDLE SA RETARD MULT vs MAP TBL

```

```

BRCLR    L003E,#8,LBACC          ; BR IF NOT UNDER SPD IDLE SPK
; ... else
LBACC:   LDX      #$81F9          ; IDLE SA ADVANCE MULT vs MAP TBL
LDAA    L0071                    ; NORMILIZED LOAD MAP VAL
CMPA    #96                      ; CK IF G.T. 50 Kpa
BLS     LBAD4                    ; BR IF L.T or E.Q.
; ... else
LDAA    #96                      ; USE LIMIT VALUE, (50 Kpa)

LBAD4:   LSRA                    ;
JSR     LE3D0                    ; 2D LOOK UP, NO OFF SET

PULB
MUL
ADCA    #0                      ; ROUND OFF

;
; DO LOW GAIN MODE ADJUST
; IF ENABLED
;
BRCLR    L0039,#8,LBAED          ; Br if not in idel spk Lo gain

LDAB    $81E8                    ; 75%, UNDER SPD Lo GAIN Mult
BRSET   L003E,#8,LBAEA          ; BR IF b3, UNDER SPD
;
LDAB    $81E9                    ; OVER SPD LO GAIN MULT
;
LBAEA:   MUL
ADCA    #0                      ; ROUND OFF
;
LBAED:   PSHA                    ; SAVE FINE SPK ADV Mod
CLRA
BRSET   L003E,#8,LBAF8          ; Br IF b3, UNDER SPD
;
DECA
PULB
COMB
BRA     LBAF9                    ;
;
;
LBAF8:   PULB
;
LBAF9:   TSX                    ; ADD OR SUB SPK FM TOTAL SPK
ADDD    0,X
INS
INS
;
BRA     LBB02                    ;
;-----

;-----
; SPARK OUTPUT ROUTINES
; FINISH UP DETAILS
;
;-----
LBB00:   PULA                    ; WHEN NOT IN IDLE
PULB
;
;
LBB02:   TSX
SUBB    $8137                    ; 20 DEG SA BIAS VALUE FOR COOL SPK COMP
SBCA    #0
SUBB    0,X                      ; CALIB Value

```

```

SBCA      #0                                ;
                                                ;
BRCLR     L004A,#$10,LBB15                 ; Br IF TCC PASS BY LOCK INACTIVE
                                                ; ... else
SUBB      $8310                             ; RETARD SPK IF IN P.B. NOISE
SBCA      #0                                ;
LBB15:    INS                               ;
                                                ;
; SCALE & DO LOW OCTAINE S.A. RETARD
;
PSHB                                     ; Save SA on stack
PSHA

;-----
; LK UP LOW OCTAINE KNOCK MULT vs RPM
;
;-----
LDAA      L0058                             ; RPM/25
LSRA      ; Scale for 800 RPM break points
LDX       #$822F                             ; Low Octain KNOCK mult vs RPM
JSR       LE3D0                             ; 2D LOOK UP, NO OFF SET
;
TAB                                     ; MULT to B Reg

;-----
; LK UP LOW OCTAINE KNOCK MULT vs MAP
;
;-----
LDAA      L0071                             ; NORMILIZED LOAD MAP VAL
LDX       #$8238                             ; Low Octain KNOCK mult vs MAP TBL
JSR       LE3D0                             ; 2D LOOK UP, NO OFF SET
;
MUL                                     ; RPM * MAP (Mult's)
ADCA      #0                                ; SCALE
LDAB      L01C0                             ; LO OCTANE RETARD MOD'ER, TEMPORARY
MUL                                     ;
ADCA      #0                                ;
STAA      L01C1                             ; LO OCTANE RETARD MOD'ER, FINAL
;
PULA                                     ; GET SPK AVD
PULB                                     ;
SUBB      L01C1                             ; LO OCTANE RETARD MOD'ER, FINAL
SBCA      #0                                ;
STD       L0128                             ; SPK ADV

;-----
; HEADS UP FOR SPARK MOD
;
;-----
BRCLR     L0036,#$80,LBB47                 ; BR IF NOT b7, HEADS UP CONNECTED
                                                ; ... else
JSR       L5803                             ; To HU for SPK ADV MOD Modification
;-----

;
; SUB OFF SPK INITAL ADVANCE
;
LBB47:    SUBB      $8025                     ; 6 DEG REF SPK ANGLE
SBCA      #0
STD       L00BB                             ; SPK ADV Rel to DRP, (un-lim)

;
; LIMIT SPK ADV TO 40 DEG

```

```

;
LDD      $8026      ; 39 DEG, (MAX ADDED SA LMT)
SUBD    L00BB      ; SPK ADV Rel to DRP, (UN-LIM)
BGT     LBB59

ADDD    L00BB      ; SPK ADV Rel to DRP, (UN-LIM)
STD     L00BB      ; SPK ADV Rel to DRP, (UN-LIM)

;
; KNOCK MINOR LOOP
;
LBB59:  LDD      L3FCA      ; READ KNOCK SENSOR INPUT
        PSHB      ; SAVE TO STX
        PSHA

        SUBD     L00C1      ; CALC CURRENT KNOCK CNT,
        ; (PA3 CNT'R LAST MNR LP)
        TSTA     ; CK IF Z
        BEQ     LBB65      ; BR IF Z

        LDAB    #255      ; MAX VAL

LBB65:  PULX
        PSHB      ; SAVE CURRENT KNOCK COUNT TO STX
        STX     L00C1      ; CPU PA3 CNT'R LAST MNR LP

        CLRA
        BRSET   L0003,#$80,LBB8A      ; BR IF b7, KNOCK NOT DISABELED

;
; CALC DIFF COOL
;
LDAB    L005D      ; FLT COOLANT TEMP
SUBB    L0060      ; START UP COOLANT
BLS     LBB7E      ; BR IF L.T. or E.Q. DIFF COOL
        ; ... else
        ; 0.24 DEG C, IF DIFF COOL SINCE START >

CMPB    $820D
BLS     LBB7E      ; ... else
        ; SET b7, KNOCK DISABELED BY DIFF COOLANT

BSET    L0003,$80
BRA     LBB8A

;
; CK KNOCK CONTROL QUAL'S
;
LBB7E:  LDAB    L005D      ; FLT COOLANT TEMP
        CMPB    $820C      ; IF COOL < 3.9 DEG C, DISABLE KNOCK
        BCC     LBB8A      ; BR IF L.T.
        ; ... else
        ; CLR b7, KNOCK ENABLED BY COOL

BCLR    L0041,$80

BRA     LBBE0

LBB8A:  BSET    L0041,$80      ; SET b7, KNOCK ENABLED BY COOL BIT

        LDAB    L0059      ; RPM/12.5
        CMPB    $820A      ; IF >= 650 RPM ENAB KNOCK
        BCC     LBB9B      ;
        ; ... else
        ; GET FILT RPM
        LDAB    L0081      ; GET FILT RPM
        CMPB    $820B      ; IF >= 2 MPH THEN ENAB KNOCK
        BCS     LBBE0      ;
        ; ... else
        ; USE LIMIT

LBB9B:  LDAA    #255

```

```

;
; CK ERR #43B STATUS
;
BRSET    L0002, #80, LBBC0          ; BR IF b7, ERR 43B TEST
                                           ; FAILED THIS START UP
                                           ; ... else

;
; CK BATTERY
;
LDAB     L0103                      ; A/D BATTERY VOLTAGE, (0.0968/V)
CMPB     #93                        ; 9.3 VDC
BCS      LBBC0                      ; BR IF BATT L.T. 9.3 VDC
                                           ; ... else

;
; CK ERR #43A STATUS
;
BRSET    $003C, #20, LBBC0         ; BR IF b5, KNOCK FAIL ERR 43A
                                           ; ... else

;-----
; LK UP KNOCK ATTACK RATE vs RPM
;
; TBL = ATTACK RATE IN Deg/MSEC/0.0225,
; 400 - 4800 RPM
;-----
LDAA     L0057                      ; GET RPM/25
LSRA                                          ; DIV BY 4 FOR LK UP
LSRA

LDX      #$820E                    ; KNOCK ATTACK RATE vs RPM TBL
JSR      LE3D0                      ; 2D LOOK UP, NO OFF SET

PULB                                          ; GET KNOCK CNT FM STX
PSHB

MUL                                          ; KNOCK CNT * ATTACK RATE
ASLD                                          ; MULT * 2
ADDA     L00C4                      ; SUM TO KNOCK RETARD
BCC      LBBC0                      ; IF NO OVER FLOW
                                           ; ... else
LDAA     #255                       ; LIMIT

LBBC0:  PSHA

BRCLR    L0046, #20, LBBDB         ; BR IF NOT b5, PWR INRICH
                                           ; ... else

;-----
; LK UP MAX KNOCK RETARD WHILE IN PWR ENRICH ..
; vs RPM
;-----
LDX      #$8218                    ; MAX KNOCK RETARD WHILE IN
                                           ; PWR ENRICH vs RPM TBL

LDAA     L0085                      ; RPM/25
LSRA                                          ; RPM/2
LDAB     #16                        ; Offset for Lk Up
JSR      LE3CC                      ; 2d LK UP

BRA      LBBDB

;-----
;-----

```

```

; LK UP MAX KNOCK RETARD WHILE not IN PWR ENRICH ..
; vs MAP
;-----
LBB2:  LDX      #$8220          ; MAX KNOCK RETARD while NOT
;                               ; in PWR ENRICH vs MAP TBL
;                               ; NORMILIZED LOAD MAP VAL
      LDAA     L0071
      LSRA
      JSR      LE3D0          ; 2D LOOK UP, NO OFF SET

LBBDB: PULB          ;
      CBA          ; COMP LK'ED UP RESULT to B reg
      BCS      LBBE0      ; BR IF B G.T. K'ED UP RESULT
;                               ; .... else
      TBA

LBBE0: STAA     L00C4          ; KNOCK RETARD
      LSRA          ; KNOCK RETARD/2
      PSHA          ; SAVE ON STX

;-----
; CK FOR HEADS UP
; {SPARK ADVANCE}
;-----
BRCLR  L0036,$$80,LBBEC      ; BR IF NOT b7, HEADS UP CONNECTED
;                               ; ... else
BRSET  L004C,#1,LBBF5        ; BR IF b0, HEADS UP FLG FOR SLEW
;-----

LBBEC: LDD      L00BB          ; SPK ADV Rel to DRP, (UN-LIM)
      TSX
      SUBB     0,X          ;
      SBCA     #0
      STD      L00BB          ; SPK ADV Rel to DRP, (UN-LIM)

;
; PWR ENRICH KNOCK TEST
;
LBBF5: PULA          ;
      PULB          ; GET KNOCK CNT FM STX
      TSTB          ;
      BEQ      LBC0F          ; BR IF KNOCK CNT = 0
;                               ; ... else
      BRCLR   L0002,$$01,LBC03 ; BR IF ERR 43B TST NOT IN WORK
;                               ; ... else
      BCLR    L0002,$$01      ; CLR ERR 43B TEST IN WORK BIT
      BRA     LBC09

LBC03: LDAB     L013B          ; SPK ADV BLEND MULT
      INCB          ; INCR BLEND VAL
      BNE     LBC0C          ; BR ID BLEND IS N/Z
;                               ; ... else
LBC09: BSET     L0002,$$40      ; SET b6, SKP IAC MOTOR RESET BIT
LBC0C: BCLR     L0002,$$80      ; SET b7, EST ERR 43B BIT
;                               ; (TEST FAILED THIS START UP)

;
; LIMIT SPK RETARD
;
LBC0F: LDX      $8028          ; Max Retard, (65525d) 3.5 DEG
      CPX      L00BB          ; SPK ADV Rel to DRP, (UN-LIM)
      BLT     LBC18          ;
;                               ; ... else

```


STX L00BB ; SPK ADV Rel to DRP, (UN-LIM)
;-----

;-----
; ALDL MODE 4
;
; ANHT/4298 P/N 16134297
; ECM TYPE \$8D, ECM P/N 1227727
;
; SET SPK ADV/SPK RETARD FLAGS AND APPLY SPK ADV
;
; CNTL WD \$0198, b4
; CNTL WD \$0198, b5
; CNTL WD \$019B, SPK ADV
;
;-----

LBC18: BRCLR L0048,#\$01,LBC4E ; BR IF NOT IN MODE 4
; ... else
LDAA \$0198 ; CNT'L WD
BITA #\$08 ; b3,
BEQ LBC4E ; Br IF NOT CNT'LING SPK, (EXIT)
; ... else
LDAB \$019B ; CNT'L WD, ALDL SPK ADV
BITA #\$10 ; b4,
BNE LBC36 ; Br IF MOD'ING sPK
; ... else
BITA #\$20 ; b5,
BEQ LBC33 ; Br IF IN ADV SA
; ... else
LDAA #128
NEGB
BRA LBC4A ; Br IF IN RETARD SA

LBC33: CLRA ;

BRA LBC4A

LBC36: BITA #\$20 ; b5
BNE LBC43

; ... else
LDD L00BB ; SPK ADV Rel to DRP, (UN-LIM)
ADDB L019B ; CNT'L WD, ALDL SPK ADV
ADCA #0 ; ROUND
BRA LBC4A

LBC43: LDD L00BB ; SPK ADV Rel to DRP, (UN-LIM)
SUBB L019B ; CNT'L WD, ALDL SPK ADV
SBCA #0 ; ROUND

LBC4A: STD L00BB ; SPK ADV Rel to DRP, (UN-LIM)

BRA LBC61
;-----

;-----
;
;
;-----

LBC4E: BRSET L0042,#\$80,LBC61 ; BR IF b7, IN DIAGNOSTICS

```

LDX      L00BB                ; SPK ADV REL TO DRP, (UN-LIM)
BMI      LBC61

; ... else

PSHX
TSX
LDAA     L013B                ; SPK ADV BLEND MULT
JSR      LE33F                ; 8 x 16 MULT

PULX
STD      L00BB                ; SPK ADV Rel to DRP, (UN-LIM)

LBC61:  LDD      L00BB                ; SPK ADV Rel to DRP, (UN-LIM)
        STD      L00BD                ; FINAL SPK ADV REL TO DRP
        BMI      LBC6C

; ... else
        BCLR     L0037, #01           ; CLR b0, 1 = SPK RETARD
; {IF ADVANCE CLR RETARD FLG}
        BRA      LBC70

LBC6C:  BSET     L0037, #01           ; SET b0, IF RETARD, 1 = RETARD
        NEGB

LBC70:  PSHB

;-----
; CK HEADS UP MAKE SPK ADV REL TO TDC
;
;-----
BRCLR    L0036, #080, LBC78        ; BR IF NO HEADS UP CONNECTED
; ... else
JSR      L5806                    ; DISP SA IF IM IS CONNECTED
;-----

*****
* SPARK OUT ROUTINE
*
* EXECUTED EVERY OTHER 1RQI
* ALL MNR LOOP SPARK FUNCTIONS and Vss
*
*****

LBC78:  PULA                ;
        TSX                ;
        JSR      LE33F        ; 8x16 MULT
        ;
        PULX                ;
        PSHB                ;
        PSHA                ;
        TSX                ;
        LDAB     L0037        ; MINOR LOOP MD WD 1
        RORB                ;
        BCS      LBC8E        ; TO SPK OUT ROUTINE
        ; .... else
        CLRA                ;
        CLRB                ;
        SUBD     0, X        ;
        PULX                ;
        BRA      LBC90

LBC8D:  RTS
;-----

```



```

        ADDD    #1
        SUBD    0,X
        BCC     LBCD5
; ... else
        BRA     LBCD9
LBCD1:
        SUBD    0,X
        BCC     LBCD9
; ... else
LBCD5:  ADDD    0,X
        STD     0,X
LBCD9:  LDD     0,X
        SUBD    L3FF6
        BSR     LBC8D
;
; EST FALL CNT'R
; SHORT DELAY
        STD     L3FE8
        BSR     LBC8D
; CURRENT EST FALL
; SHORT DELAY
        ADDD    L3FDC
        BSR     LBC8D
; EST DWELL, (PULSE WIDTH)
; SHORT DELAY
        SUBD    L00B9
        STD     L3FE6
        BSR     LBC8D
; DWELL TO CPU, (15.25 msec/BIT)
; LAST EST FALL
; SHORT DELAY
        LDD     L00B9
        STD     L3FDC
        BSR     LBC8D
; DWELL TO CPU, (15.25 msec/BIT)
; EST DWELL, (EST PULSE WIDTH)
        PULA
        PULB
        STD     L3FF6
; EST FALL CNT'R
        LDAA   L0036
        BITA   #$08
        BEQ    LBD3A
; b3, 1st DRP BIT = 1 IF 1ST DRP NOW
; BR IF NOT 1st DRP
;-----
; ERROR 42 ROUTINE
; SPK MON EANBLED
;-----
        BITA   #$10
        BNE    LBD2B
; b4 of L0036 = 1 IF IGN ON
; BR IF IGN OFF
; .... else
        LDAB   $824B
        BITB   #$20
        BEQ    LBD2B
; ERROR WD 3 MASK, 0111 1111
; BIT 5 = CODE 42 EST MONITOR
; BR IF IF ERR 42 NOT ENABLED
        LDAB   L0037
        ASLB
        BMI    LBD2B
; MINOR LOOP MD WD
;
; ... else
        ASLA
        BMI    LBD20
; GO SET ERR 42A
; ... else
        LDAB   L00D7
        CLR    L00D7
; GET EST FB CNT'R
; EST FB CNT'R
        CMPB   $8295
        BLS    LBD28
; 4 EST ERR'S, P/O ERR PARAM, EST,
; IF CNTS G.T. 4, GO SET ERR 42A BIT
; ... else
LBD20:  BSET    L0036, #$40
; SET ERROR 42B (b6)
        LDD     L3FFC
; CPU CR

```

```

        BRA        LBD6C                ;
LBD28:  BSET      L0037, #$40          ; BYPASS CHECK ENABLED
LBD2B:  LDX       L3FEC                ; B CNT'R LAST DRP

        LDAB     L0036                ;
        ANDB     #$F7                 ; CLR 1st DRP BIT
        STAB     L0036

        INCB
        INCB
        STX      L3FE4                ; B CNT'R START NEXT DWELL
        INCB

LBD3A:  MUL
        LDD      L3FFC                ; CPU CR
        ORAB     #$10                 ; SET b4

        BRSET    L0033, $04, LBD4D    ; BR IF EST ENABLED

        CLR      L013B                ; SPK ADV BLEND MULT

        BSET     L0039, #$20          ; SET BLEND SPK
        BSET     L0033, #4            ; ENABLE EST BIT

LBD4D:  BRSET    L0039, #2, LBD55     ; BR IF IN IN Stall Saver SPK BYPASS
        BRCLR   L0001, #$80, LBD6F   ; NOT EST ERR, (42)

LBD55:  BCLR     L0033, #4            ; CLEAR EST ENABLE BIT
        ANDB     #$EF
        BRCLR   L0039, #1, LBD6C     ; BR IF NOT b0,
        ; SKIP ERR STALL SAVER SPK BYPASS

        PSHB
        LDAB     $00D7                ; EST FB CNT'R
        ; IF EST TOGGLED, (PA1), >= 2
        CMPB    $8295                ; 4 EST ERR'S, P/O ERR PARAM, EST
        PULB
        BLS     LBD6F

        BSET     L0001, #$8           ; SET BAD SHUTDOWN
        BRA     LBD6F

LBD6C:  BSET     L004F, #$20          ; SET b5, ERR 42, EST

; -----
; ALDL MODE 4,  COMMAND
;
; ANHT/4298 P/N 16134297
; ECM TYPE $8D, ECM P/N 1227727
;
; CNT'L WD $0194, b..
;
;
; ECM 8D, ECM P/N 1227730
; -----
LBD6F:  BRCLR   L0048, #$01, LBD96    ; BR IF NOT b0, IN ALDL MODE 4

        PSHA
        LDAA    L0194                ; MODE ENABLES, ALDL
        BITA    #$08                 ; b3, 1 = SPK BACK UP CMD'ED
        BEQ     LBD95

```

```

ANDB    #$EF                ; CLEAR BIT 7
BCLR    L0033, #$04         ; CLEAR EST ENABLE BIT
PULA
PSHA
STD     L3FFC                ; CPU CR
SEI
JSR     LE2FF                ; FMD SERIAL BYTE READ

CLI                    ; CLEAR & RESTORE INTERUPTS

CLR     L00D7                ; EST FB CNT'R

BSET    L0039, #$01         ; SET  SKIP ERR STALL SAVER SPK BYPASS

LDX     L3FC8                ; PA2 CNT'R, SPK Period
STX     L00D5                ; OLD PA2 CNT'R

LBD95:  PULA
;-----

;-----

LBD96:  STD     L3FFC                ; CPU CR
        BRA     LBDF7
        *****
        * END OF SPK OUT
        *
        *****

        *****
        * LOW RPM SPARK
        *
        *****

;-----
; CLEAR VARS
;-----

LBD9B:  LDD     #0                ; A = 0, B = 0
        STAA   L400C              ; CPU COP

        LDX    #$FFFF
        STX    L0467              ;

        STAA   L0057              ; RPM/25
        STAA   L0085              ; RPM/25
        STAA   L0059              ; RPM/12.5
        STD    L0059              ; RPM/12.5
        STD    L013D              ; filt delta MAP A.E.
        STD    L00F8              ; CLR ACCUM Acell Enr FUEL

        BRSET  L0001, #$08, LBDBA ; BR IF b3, BAD SHUTDOWN

        STD    L0019              ; ENG RUN TIME

LBDBA:  STAA   $0114
        STAA   $00A5

        BRSET  L00FE, #$04, LBDEF ; IAC Md Wd #2

        BSR    LBDC7

        BRA    LBDEF

LBDC7:  LDAB   L003E              ; AIR MODE WD
        ASLB

```

```

BPL      LBDD1
LDAB     $8388
STAB     L002C
; ... else
; A/C LRN, 30 STPS
; IDLE SPD A/C ANTICIPATE
LBDD1: LDAB     $8629
; 160 STP'S IDLE RPM START UP
; POSIT, PARK, (COLD)
;
STAB     L0119
JSR      LC01F
LDAA     #255
STAA     $0112
; LIMIT
; IAC CNT'L SPEED
LDAA     #$80
STAA     L00FD
; SET b7, AC STATUS OFF
; IAC MODE WD
CLRA
;
; CK PL/NEUT STATUS
;
LDAB     L003B
BITB     #$01
BEQ      LBDEC
; CPU INPUT STATUS WD
; b0, PK/NEUT
; BR IF NOT b0
; ... else
;
; b1 1 = MOTOR RESET DONE
; b4 1 = OLD PK/NEUT IN NEUT, LAST PID
; b5 1 = PID CMD STEP ACTIVE
; b6 1 = START UP DELAY IN WORK
;
LDAA     #$72
LBDEC: STAA     L00FE
; b1, b4,5 & 6
; IAC Md Wd #2
RTS
;-----

LBDEF: LDAA     L00EF
JSR      LE5AE
; TIME OUT AF CNT'R
BCLR     L0033,$04
; CLR b2, EST ENABLE BIT
LBDF7: SEI
JSR      LE2FF
; FMD serial byte read
CLI
; CLEAR & RESTORE INTERUPTS
BRCLR    L0037,$80,LBE06
JSR      LE8FA
JSR      LCD6C
; BR IF NOT b7. 1 = ENGINE RUNNING
; TO FUEL OUT
; TO ASYNC OUT
;-----

;-----
; SERIAL DATA FOR MINOR LOOP MESSAGE SCHEDULT TABLE
;
;-----
LBE06: BRCLR    L0038,$28,LBE0C
; BR IF NOT b3 & b5, MODE WD 2
; ... else
BRA      LBE6F
; EXIT TO ALDL MODE 10 SCI
LBE0C: LDX      #$8825
LDAB     L0000
; INDEX MESSAGE SCHEDULING TABLE <---**
; NINOR LP CNTR

```

```

LBE11:  ANDB    #$1F          ; MASK 0001 1111, PERMIT MSG 0 -> 16
        ABX          ; ADD MNR LP CNT'R TO POINTER
        LDX     0,X         ; GET ADDRESS FOR THIS MNR LP NUM

        BEQ     LBE6F       ; BR IF ADDR IS Z, NO MSG AT THIS TIME
                                ; (EXIT TO ALDL MODE 10 SCI)
                                ; ... else

        LDAA   2,X         ;
        BEQ     LBE6F       ; BR IF Z, MSG ID NUM
                                ; (EXIT TO ALDL MODE 10 SCI)
                                ; ... else

        BRSET  L0048,$#08,LBE68 ; BR IF b3, XMIT IN WORK
                                ; ... else

        BSET  L0048,$#08    ; SET XMIT IN WORK BIT
        STX  L014C         ; SAVE DATA TBL POINTER

        LDAB  L4008        ; CPU TX/RX STATUS REG, CLR INTERRUPTS

        STAA  L400A        ; TX DATA, (8192 BAUD SCI) <-----*****

        STAA  L014B        ; SCI DATA CHECK SUM

        LDAB  4,X         ; GET NUM OUTPUT BYTES
        STAB  L019D        ; 8192 MSG LENGTH
        BEQ  LBE49        ; BR IF NO DATA HAS TO BE XFERED FM PROM
                                ; ... else

        LDY   5,X         ; GET OUTPUT BUFFER ADDRESS

        ;
        ; XFER DATA FROM EPROM TO OUTPUT BUFFER
        ;

LBE39:  PSHX
        LDX   9,X         ; GET ADDR FM CALIBRATION ROM
        LDAA 0,X         ; GET DATA TO XFER
        STAA 0,Y         ; SAVE DATA IN OUTPUT BUFFER
                                ; INCR OUTPUT BUFFER ADDRESS
        INY
        PULX
        INX          ; INCR ADDRESS INDEX
        INX
        DECB        ; DECR BYTE COUNT
        BNE   LBE39    ; BR IF NZ, (TILL DONE)
                                ; ... else

LBE49:  LDAB   #1         ; SET BYT COUNT = 1
        STAB  L014A        ; SCI DATA BYTE COUNTER
        SEI

        ;
        ; SET UP 8192 SCI FOR XMIT
        ;

        LDX   #$4000      ; INDEX CPU REG'S
        BCLR  2,X,$#80    ; CLR b7, MAKE PORT b7 AN INPUT
        BSET  $32,$#01    ; SET b0, $4032, TX ENAB)

        LDAA  L0032        ; FMD 2 MODE WD
        JSR  LF0C0        ; XMIT ON SERIAL LINE

        BSET  2,X,$#80
        CLI

        ;
        ; CPU TX/RX CNT'L REG
        ; b0 COUNTER INTERRUPT ENABLE
        ; b3 TX ENABLE
        ; b7 TX INTERRUPT ENABLE

```



```

;
LDAA    #$89                ; b0, 3, 7
STAA    L4007              ; CPU TX/RX CNT'L REG, (8192 BAUD SCI)

BRA     LBE6F              ; EXIT TO ALDL MODE 10 SCI

LBE68:  BRSET    L0048,$04,LBE6F ; BR IF b2, XMISSION IN WORK
; ... else
BSET    L0048,$04        ; SET b2, XMISSION IN WORK BIT
;-----

*****
* ALDL MODE 10 CONTROL ROUTINE
* {CLEAR ERRORS}
*
*****

;
; CK IF MODE 10
;
LBE6F:  LDAA    L0170        ; GET CURRENT ALDL MODE NUMBER
CMPA    #10              ; MODE 10
BNE     LBE8D            ; BR IF NOT IN MODE 10
; ... else

;
; CK IF SYSTEM ERROR ALREADY CLEARED
;
BRSET   L0038,$01,LBE90  ; BR IF b0, ERR'S CLEARED
; (GO CK MODE 4)
; ... else
BSET    L0038,$01        ; SET b0, ERR'S CLEARED, MODE 10 FLAG

;
; CLEAR SYSTEM ERRORS
;
LDD     #0                ; CLR ERR WD'S
STD     L0004            ; ERR WD 1 & 2
STD     L0006            ; ERR WD 3 & 4
STAA   L0008            ; ERR WD 5

LDD     #01              ; RESET ...
STD     L0017            ; K/A ERR CK SUM TO CLR'ED STATE

BRA     LBEE2            ; EXIT MODE 4/MODE 10

LBE8D:  BCLR    L0038,$01  ; CLR b0, ERR'S CLEARED
;-----

;-----
; MODE 4 CONTROL, CONTROLLER MODE
;
;
; ANHT/4298 P/N 16134297
; ECM TYPE $8D, ECM P/N 1227727
;
; CNT'L WD $0194, b6
;
;-----
LBE90:  BRCLR   L0048,$01,LBEE2 ; BR IF NOT b0, IN ALDL MODE 4

```

```

; ... else

LDX    #L0194                ; ALDL MD ENABLES
BRCLR  0,X,#$40,LBEB2       ; BR IF NOT b6, (SPK BACK UP CMD'ED)
; ... else
BRSET  L0048,#$40,LBEB5     ; BR IF b6, (ERR FLG CLR PRIOR LP)
; ... else
BSET   L0048,#$40          ; SET b6, (ERR FLG CLR PRIOR LP BIT)

;
; CLEAR SYSTEM ERR'S
;
LDD    #0
STD    L0004                ; CLR ERR WD 1 & 2
STD    L0006                ; CLR ERR WD 3 & 4
STAA   L0008                ; ERR WD 5

LDD    #1
STD    L0017                ; K/A ERR CK SUM TO CLR'ED STATE

BRA    LBEB5                ;

;
; NOT SPK BACK UP CMD'ED
;
LBEB2: BCLR    L0048,#$40    ; CLR b5, (ERR FLG CLR PRIOR LP BIT)

;
; ERR FLG CLR PRIOR LP
;
LBEB5: BRSET  L00FE,#$04,LBECF ; BR IF b2, IAC RESET IN WORK
; (IAC MODE WD 2)
; ... else
BRCLR  0,X,#$20,LBECC       ; BR IF NOT b5, (ASKING FOR IAC RESET)
; ... else
BRSET  L0048,#$20,LBECF     ; BR IF b5, (IAC RESET CMD'ED, PRIOR LP)
; ... else
BSET   L0048,#$20          ; SET b5, (IAC RESET CMD'ED, PRIOR LP BIT)
BSET   L00FE,#$04          ; SET b2, IAC RESET STARTED
; IAC Md Wd #2
BSET   L002B,#$FF          ; SET b0->b7, (IAC CURRENT POSITION = 255)

BRA    LBECF                ;

LBECF  BCLR    L0048,#$20    ; CLR b5, (IAC RESET CMD'ED, PRIOR LP BIT)

LBECF  BRCLR  0,X,#$10,LBEDF ; br IF NOT b4, (DISABLE EST)
; ... else
BRSET  L0048,#$10,LBEE2     ; BR IF b4, (BLM RESET CMD'ED, PRIOR LP)
; ... else
BSET   L0048,#$10          ; SET b4, (IAC RESET CMD'ED, PRIOR LP BIT)

JSR    LE548                ; Go init BLM cells

BRA    LBEE2                ; EXIT TO RD SPD

LBEDF: BCLR    L0048,#$10    ; SET b4, (BLM RESET CMD'ED, PRIOR LP BIT)
;-----

```

```

*****
* ROAD_SPD.SRC
*

```

```

* DONE EVERY 25 Msec
*
*
*****
LBEE2: BRSET    0,#$02,LBEE9          ; BR IF b1, EXE 25 Msec ROUTINE

      JMP      Lbfd6                 ; EXIT Vss TO IAC MINOR LP

      ;
      ; Vss FOR INSTRUMENT
      ; PNL DIVIDE INIT
      ;
LBEE9: SEI          ; TURN OFF INTERUPTS

      LDAA    L0032                 ; FMD 2 MODE WD
      ; SC1 STATUS, 1 = TX ENAB)
      ANDA    #$0F                 ; MASK Vss CONFIG BITS, (LOWER NIBBLE)

      ;
      ; CK AFR OPT WD 3, 0101 0010
      ; b6, 1 = MAG SPD SENSER INSTALLED
      ;
      LDAB    $8018                 ; AFR OPT WD 3
      BITB    #$40                 ; IF b6, 1 = MAG SPD SENSER INSTALLED
      BNE     LBEF7                 ; BR IF MAG SPD SELECTED
      ; ... else
      ORAA    #$10                 ; ELSE SET OPTICAL SENSOR

LBEF7: ORAA    $8014                 ; INST PNL DIVIDER ARG, (0d)
      STAA    L0032                 ; FMD 2 MODE WD
      ; SC1 STATUS, 1 = TX ENAB)

      CLI          ; TURN OFF INTERUPTS

      ;
      ; DIFF RD SPD PULSES & DIFF B CNTR
      ; CNTS CALCULATION
      ;
      LDX     L3FC2                 ; B Ctr VALUE
      PSHX                    ; SAVE TO STACK
      BRN     LBF06                 ; CPU DELAY, (3 CYC)

LBF06: LDD     L3FE0                 ; GET CUM RD SPD PULSE CNTR VALUE
      NOP                    ; DELAY
      NOP
      NOP
      NOP
      CPX     L3FC2                 ; B Ctr, CK IF ANY PULSES
      BEQ     LBF19                 ; IF NOT
      ; ... else
      PULX                    ; CLEAN GET OLD VALUE OFF STX
      LDX     L3FC2                 ; B Ctr, LN NEW B CNT'R VAL
      PSHX                    ; NE VAL TO STX

      BRN     LBF19                 ;

      LDD     L3FE0                 ; RD SPD PULSE CNTR

LBF19: PSHB                    ; SAVE PA4 CNT'R TO STX
      PSHA
      SUBD    L0085                 ; CALC DIFF RD SPD PULSES
      BNE     LBF37                 ; IF DIFF N.E. 0
      ; .... else

      ;
      ; NO RD SPD PULSE THIS LOOP

```

```

;
PULX          ; CLR DATA OFF STX
PULX          ;

LDAA  L0084   ; RD SPD NO PULSE CNTR
INCA                   ; INCR NO PULSE CNTR
CMPA  #39     ; LIMIT Vss CNT'R TO 39
BCS  LBF31    ; BR IF NO Vss IN 1 SEC
; ... else
LDD   #0      ; SET ALL SPD VARS TO 0
STAA  L0083   ; VSS TABLE INPUT
STD   L0081   ;

LDA   #39     ; LIMIT Vss CNT'R TO 39
LBF31: STAA  L0084 ; NO Vss PULSE CNT'R

LDAA  L0083   ; VSS TABLE INPUT
BRA   LBF7B   ; TO JMP BRIDGE, ($BFC9)

;
; RD SPD PULSE HERE THIS LOOP
;
LBF37: PULX   ; GET CNT'R VAL
STX   L0085   ; PA4 OLD VAL
STAB  L008D   ; NEW VAL AS DIFF
PSHB                   ; SAVE TO STK
ADDB  L0134   ; ADD OLD ACCUM DISTANCE
STAB  L0134   ; NEW ACCUM DISTANCE

TSX                   ;
LDD   1,X     ; GET CNT, (3FC3)
SUBD  L0087   ; SUB OLD VALUE
DEC   0,X     ; CK IF DIFF = 1
INS                   ;
PSHB                   ; SAVE CUTTEN DIPP PULT COUNT TO STX
PSHA                   ;
BNE   LBF66   ; BR IF N/Z, (NO AVG OF CNT'S IF > 1
; ... else
BRSET L0037,#$10,LBF58 ; BR IF b4, 1 = 1st PASS W/Vss PULSE
; ... else
BSET  L0037,#$10 ; SET b4, 1 = 1st PASS W/Vss PULSE

BRA   LBF69   ; BY PASS AGV'ING

;
; IF ONLY 1 PULSE LAST LP AVG LAST 2
;
LBF58: LDD    L0089 ;
LSRD                   ; DIV by 2
PSHB                   ; SAVE AVG TO STX
PSHA                   ;
TSX                   ;
LDD   2,X     ; GET CURRENT PULSE CNT
LSRD                   ; DIV by 2
ADDD  0,X     ; CK IF DIFF = 1
PULX                   ; CLR OLD DATA FM STX
BRA   LBF69   ;

LBF66: BCLR   L0037,#$10 ; CLR b4, 1 = 1st PASS W/Vss PULSE
LBF69: STD    L008B     ; SAVE CURRENT Vss COUNT

PULX                   ; CURRENT Vss COUNT as OLD
STX   L0089

PULX                   ; GET OLD CNTR VALUE

```

```

STX      L0087                ; SAVE as OLD CNTR VALUE

LDAA     L0084                ; RD SPD NO PULSE LP CNTR
CLR      L0084                ; RD SPD NO PULSE LP CNTR

CMPA     #39                  ; CK
BCS      LBF7D                ;
                                ; ... else
CLRA                                ; A = 0

LBF7B:   BRA      LBF7D

;-----
; Vss CALCULATION
;
;-----
LBF7D:   LDAB     L008D                ; GET Vss COUNT

;
; CK AFR OPT WD 3, 0101 0010
; b6, 1 = MAG SPD SENSER INSTALLED
;
LDAA     $8018                ; 0101 0010, AFR OPT WD 3
BITA     #40                  ; b6
BNE      LBF9A                ; BR IF b6, MAG SENSOR
                                ; ... else

;
; OPTICAL SENSOR
;
LDX      #461                 ; 461 = 1 RD SPD PULSE
DECB                                ;
BEQ      LBF95                ; BR IF 1 Vss PULSE
                                ; ... else
LDX      #922                 ; 922 = 2 RD SPD PULSE'S
DECB                                ;
BEQ      LBF95                ; BR IF 2 Vss PULSE'S
                                ; ... else
LDX      #1382                ; 1382 = 3 RD SPD PULSE'S

LBF95:   PSHX                    ; SAVE MULT CONST
PULA                                ; CONST TO A Reg
PULB
BRA      LBF9E

;
; MAG SENSOR
;
LBF9A:   LDAA     $8015                ; PULSE/MILE RD SPD CONST, 230d
                                ; ARG = 921600/VAL, (4006 PPM)
MUL

LBF9E:   LDX      L008B                ; GET COUNT VALUE
FDIV
PSHX

PULA
PULB
TSTB
BPL      LBFAB                ; BR OF LSB L.T. 128, ($80)
                                ; ... else
INCA                                ; ROUND UP
BNE      LBFAB                ; IF NO OVERFLW
                                ; ... else
DECA                                ; PREVENT OVERFLOW

```

```

;
; FILTER Vss,
;
LBFAB: LDX      L0081      ; GET Vss
        LDAB     $82FB     ; 99.6%, Vss FILTER COEF
        JSR      LE31C     ; LAG FILTER ROUTINE

        STD      L0081     ; SAVE FILTERED Vss

;
; CALC MPH
;
LDX      #L0081          ; FILTERED Vss
LDAA     #205            ;
JSR      LE33F          ; MULT 8 * 16

CMPA     #47
BCC      LBFC7          ; BR IF
; ... else

ASLD
BCS      LBFC7          ; BR IF OVERFLOW
; ... else

ASLD
BCC      LBFC9          ; BR IF NO OVERFLOW
; ... else

LBFC7:  LDAA     #192     ; MAX VAL
;
;
LBFC9:  LDAB     $00CA     ; SPD CNTR
        CMPB     $8271     ; ERR 24, 3 SEC, QUALIFICATION TIME
        BLS      LBFD4     ; BR IF ERR 24
; ... else

        LDAA     $8649     ; ENAB PID if Vss > 0.9 MPH
        INCA
;

LBFD4:  STAA     L0083     ; SAVE Vss DEFAULT VALUE
;-----

```

```

*****
* IAC MINOR LOOP 12.5 Msec ROUTINE
* PID
*****

```

```

;
; IGN OFF or NOT RUNNING
;
LBFD6:  BRSET    L00FE,#$04,LBFE2      ; BR IF b2, MOTOR RESET IN WORK
; (IAC Md Wd #2)
; ... else
        BRSET    L0002,#$10,LC031     ; BR IF b4, SKP IAC MOTOR RESET BIT
; ... else
        BRCLR   L0036,#$10,LC031     ; BR IF NOT b4, IGN ON
; ... else

;
; IAC RESET & PARK SHUT DN
;
LBFE2:  LDAA     L002B          ; CURRENT IAC POSITION

        BRSET    L00FE,#$02,LBFFD     ; BR IF b1, MOTOR CLOSED
; (IAC Md Wd #2)
; ... else
        BRSET    L00FE,#$04,LBFF7     ; BR IF b2, RESET IN WORK
; (IAC Md Wd #2)
; ... else
        BSET     L00FE,#$04           ; SET b2, RESET IN WORK
; (IAC Md Wd #2)

```

```

        LDAA    #255                ; MAX OF - 127 STEPS
        STAA    L002B              ; IAC CURRENT POSITION

LBFF3:  LDAA    #255

        BRA     LC005

LBFF7:  TSTA    ; IS MOTOR CLOSED ORIFICE
        BNE     LBFF3             ; BR IF N/Z
        ; .... else
        BSET    L00FE,#$02        ; SET b1,, MOTOR CLOSED
        ; (IAC Md Wd #2)

        ;
        ; SET UP FOR COLD START
        ;
LBFFD:  SUBA    $8629              ; 160 STP'S IDLE RPM START UP POS, PARK
        NEGA
        BPL     LC005

        LDAA    #$7F

LC005:  STAA    L0111              ; REQUESTED MOTOR STEPS
        BNE     LC016

        BSET    L0002,$$10        ; ... else
        ; SET b4, SKIP MOTOR RESET
        ; (IAC MOTOR RESET BIT)
        BRSET   L0036,$$10,LC016  ; BR IF b4, IGN OFF

        JSR     LBDC7              ;

        BRA     LC031

        ;
        ; GENERAL IAC MOTOR  PARK ROUTINE
        ;
LC016:  CLR     L0115              ; TPS FOLLOWER DIFF
        BCLR    L00FD,$$10        ; CLR b4, OLD PK/NEUT,
        ; .. LAST PID, 1 = NEUT

        JMP     LC4F1

LC01F:  LDAA    L002B              ; IAC CURRENT POSITION
        SBA
        BCC     LC025              ; SUB OFF PARK POSIT
        ; BR IN NO UNDERFLOW
        ; ... else
        NEGA    ; INVERT VAL

LC025:  BPL     LC029              ;

        LDAA    #127                ; MAX LIMIT

LC029:  BCS     LC02D

        ; .... else
        ORAA    #$80                ; SET b7

LC02D:  STAA    L0111              ; CURRENT REQUEST MOTOR STEPS

        RTS

```

```

;-----
; THROTTLE FOLOWER CODE
;
;-----
;

```

```

; CK IF ENG RUNNING
;
LC031: LDAA    L0037      ; MINOR LP MD WD 1
      BPL     LC016      ; BR IF ANY BIT SET, BR IF NOT RUNNING

      LDAA    L0097      ; TF TPS
      STAA    L0098      ;

      LDAA    L0095      ; TPS Ld val
      TAB     ; SAVE OLD IN B Reg
      SUBB    L0097      ; TF TPS
      BCC     LC041      ;
      ; ... else
      NEGB    ; GET ABS VAL OF DIFF TPS

LC041: CMPB    $8648      ; DONT STORE NEW TPS IF DIFF VAL < 1
      BLS     LC048      ; BR IF
      ; ... else
      STAA    L0097      ; TF TPS

;
; CK IF ERR 21 or 22
;
LC048: LDAB    L0038      ; MODE WD 2
      ANDB    #$24      ; CLR b2 & b5
      BEQ     LC054      ;
      ; ... else
      LDAA    $8647      ; ENAB IF TPS > 1.2%
      ASLA    ; MULT * 2
      STAA    L0097      ; TF TPS

LC054: BRSET   L00FE,$#08,LC05B ; BR IF b3, STARUP DELAY DONE
      ; (IAC Md Wd #2)
      ; ... else
      JMP     LC247      ; TO 50 Msec IAC PID ROUTINE,

;-----
; IAC FOR FAN1, FAN 2 AND WARM KICK DOWN
;
;-----
LC05B: BRSET   L0037,$#08,LC063 ; BR IF b3, 1 = FAN 1 PID STPS DONE
      ; ... else
      BRSET   L0039,$#40,LC08B ; BE IF b5, FAN 1 OUT REQUESTED
      ; ... else
LC063: BRSET   L003F,$#40,LC06B ; BR IF b5, ALL FAN PID STP'S ADDED
      ; ... else
      BRSET   L0045,$#01,LC0A4 ; BR IF b0, FAN 2 OUT REQ'ED
      ; ... else
LC06B: BRSET   L0001,$#40,LC0D7 ; BR IF b5, BAD SHUTDOWN
      ; ... else
      BRCLR  $003C,$#40,LC0D7 ; BR IF NOT b5,
      ; ... else
      LDAA    $862D      ; 5 STP'S ADDED IF COLD ENG SA IS DONE
      ASLA    ; MULT * 2
      SUBA    L011F      ; CK IF ALL STEPS DONE
      BHI     LC087      ;
      ; ... else
      BSET    L0001,$#40 ; SET b6, BAD SHUTDOWN
      BCLR    L003C,$#40 ; SET b6
      LDAA    $8666      ; DISABLE CLSD LP PID FOR 0 MSEC AFTER
      ; COLD ENG SPK
      BRA     LC0D4

LC087: LDAA    #$82      ; DO 2 EXTEND STEPS
      BRA     LC0BD

```



```

;
; ENABLE FAN BUT NO ADDED STEPS IF TPS GT THRES
;
LC08B: LDAA    L0097          ; TF TPS
      CMPA    $8647        ; ENAB IF TPS GT 1.2%
      BCC     LC0C9        ; ENABLE FAN CODE
                          ; ... else

;
; ENABLE FAN BUT NO ADDED STEPS IF Vss GT THRES
;
      LDAA    L0083          ; VSS TABLE INPUT
      CMPA    $8649        ; PID ENAB IF Vss GT 0.9 MPH
      BHI     LC0C9        ; ENABLE FAN CODE
                          ; ... else

;
; ENABLE FAN AND ADD 2 STEPS
;
      LDAA    $862B          ; 2 STP'S ADDED FOR FAN 1
      ASLA
      SUBA    L011F
      BLS     LC0C9        ; ENABLE FAN CODE

      BRA     LC0BB

LC0A4: LDAA    L0097          ; TF TPS
      CMPA    $8647        ; ENAB IF TPS > 1.2%
      BCC     LC0CE

                          ; ... else
      LDAA    L0083          ; VSS TABLE INPUT
      CMPA    $8649        ; ENAB IF Vss > 0.9 MPH
      BHI     LC0CE        ; BR IF MPH G.T. 0.9

      LDAA    $862C          ; 2 STP'S ADDED FOR FAN 2
      ASLA
      SUBA    L011F
      BLS     LC0CE

LC0BB: LDAA    #02          ;
LC0BD: STAA    L0111        ; REQUEST 2 RETRACT STEPS

      BSET    L00FD, #$08   ; SET b3, COMP MODE
      BCLR   L00FD, #$10   ; CLR b4, TF ACTIVE FLAG

      JMP     LC247        ; TO IAC PID ROUTINE, (50 Msec)

LC0C9: BSET    L0037, #$08 ; SET b3, ENABLE FAN #1
      BRA     LC0D1        ; TO LD FAN XISSITION DELAY

;
LC0CE: BSET    L003F, #$40 ; SET b6, ENABLE FAN #2

;
; GET FAN COMP TRANSISION DELAY
;
LC0D1: LDAA    $8665        ; DISABLE CLSD LP PID FOR 0 MSEC AETER
                          ; FAN ON
LC0D4: STAA    L011B        ; DISABLE CLSD LP PID TMR

LC0D7: BCLR   L00FD, #8    ; START UP DELAY DONE, CLR b3
      CLR     L011F

```

```

;
; IAC THROTTLE FOLLOWER CODE
;
BRCLR   L00FE,#$20,LC0E4      ; BR IF NOT b5, PID CMD'ING A STEP
; (IAC Md Wd #2)
; ... else
JMP     LC247                 ; TO 50 Msec IAC PID ROUTINE,
;

;
; CK IF IN DECEL FUEL C/O STALL SAVER
;
LC0E4:  LDAA    L0097          ; TF TPS
LDAB    L003E                ; AIR MODE WD
BITB    #$10                 ; b4 STALL SAVER
BEQ     LC0F8                ; BR IF NOT b4
; ... else
LDAB    L00E7                ; STALL SVR TIMER
BEQ     LC0F8                ; BR IF TIMER Z
; ... else
DECB    ; DECR STALL SVR TIMER
STAB    L00E7                ; STALL SVR TIMER

ADDA    $85EA                ; %TPS C/O STALL SV'R, (0d)
BCS     LC10A                ; BR IF
; .... else
LC0F8:  BRCLR   L0046,#$02,LC101 ; BR IF NOT b1, IN DECEL F C/O
; ... else
ADDA    $85F1                ; 1.95% TPS DECEL FUEL C/O DEFAULT
BCS     LC10A                ; BR IF OVERFLOW
; .... else
LC101:  BRCLR   L003F,#$80,LC10C ; BR IF NOT b7, DECEL ENLEAN ACTIVE
; ... else
ADDA    $85AD                ; 6.25% TPS DECEL ENLEAN THT FOLLOWER
BCC     LC10C                ; BR IF NO OVERFLOW
; ... else
LC10A:  LDAA    #255          ; USE MAX LMT

LC10C:  LDAB    L00FD          ; START UP DELAY DONE

;
; BYPASS TPS CK IF A/C BEING CONT'ED
;
CMPA    $8647                ; ENAB PID IF TPS > 1.2%
BHI     LC132                ; BR IF

;
; CK FOR XISITION DELAY
;
LDAA    L0098                ; TF TPS, (OLD)
CMPA    $8647                ; ENAB PID IF TPS > 1.2%
BLS     LC120                ; BR IF

;
; TPS JUST CLOSED
; DISABLE CLSD LP PID FOR 0.20 MSEC
;
LDAA    $8663                ; 0.20 MSEC XISSITION DELAY
STAA    L011B                ; DISABLE CLSD LP PID TMR

LC120:  CLRA    ; 0 TPS VALUE
BITB    #$50                 ; b4 & b6, A/C CNT'LED & TF ACTIVE
BNE     LC142                ; BR IF b4 & b6, (A/C BEING CNT'ED)
; ... else

;
; CK FOR PWR STEER SW ACTIVE BY HI PRESSURE

```

```

;
PSHB
LDAB    L003B           ; CPU INPUT STATUS WD
BITB    #$08           ; b3, HI PWR STEER PRESS
PULB
BNE     LC142           ; BR IF b3,
; ... else

;
; PREVENT NOISE MOD'ING A/C ADJ
;
TST     L0115           ; TPS FOLLOWER DIFF
BEQ     LC197           ; BR IF TF NOT DONE
; ... else
LC132: LDAB    $865D     ; 150% follower slope gain
MUL
ASLD
BCC     LC13B           ; IF NO OVERFLOW
; ... else
LDAA    #255           ; LIMIT TO 255

LC13B: LDAB    $865E     ; 40 steps max in drive
CBA
BLS     LC142
; ... else

TBA

LC142: LDAB    L003B     ; CPU INPUT STATUS WD
BITB    #$08           ; b3, HI PWR STEER PRESS
BEQ     LC160
; ... else
ADDA    $866F         ; PWR STEER ANTCIPATE STEP, 0 STP'S
BCC     LC14F
; ... else
LDAA    #255           ; LIMIT

LC14F: LDAB    L00FD     ;
BMI     LC15A
ADDA    $8670         ; PWR STEER ANTCIPATE STEPS IF
; A/C ON, 0 STP'S
BCC     LC15A         ; If no overflow
; ... else
LDAA    #255           ; Limit

LC15A: ORAB    #$40     ; SET b6, START UP DELAY IN WORK
ANDB    #L00FE
BRA     LC18B

LC160: LDAB    L00FD     ;
BMI     LC185
LDAB    L002C         ; IDLE SPD A/C ANTICIPATE
CMPB    $8389         ; MAX A/C LRN, 32 STEPS
BHI     LC170
CMPB    $838A         ; 20 STP'S ADDED TO WARK PK IF A/C ON
BCC     LC17B
; ... else
LC170: LDAB    $8388     ; A/C LRN, 30 STPS
STAB    L002C         ; IDLE SPD A/C ANTICIPATE
LDAB    L003E         ; AIR MODE WD
ORAB    #$40         ; Bit 6
STAB    L003E         ; AIR MODE WD

```

```

LC17B: LDAB    L00FD
      ADDA    L002C           ; IDLE SPD A/C ANTICIPATE
      BCC    LC18B
                                     ; ... else
      LDAA   #$00FF
      BRA    LC18B

LC185: BITB    #$40
      BEQ    LC18B
                                     ; ... else
      ORAB   #$0020

LC18B: STAB    L00FD
      LDAB   L00FE           ; IAC Md Wd #2
      BPL   LC199
                                     ;
      LDAB   $865F           ; .996, (Neut steps/Drive steps)
      MUL
      BRA    LC1B9
                                     ;
                                     ;
LC197: BRA     LC1FA
                                     ;

;-----
; ADJ OVER COMP
; ERR IN DRIVE ONLY
;-----

LC199: LDAB    L0083           ; VSS TABLE INPUT
      CMPB   $8649           ; ENAB IF Vss > 0.9 MPH
      BLS    LC1B9
                                     ;
                                     ; ... else
      PSHA
      LDD    L0081
      CMPA   #64
      BCS    LC1AA
                                     ;
                                     ; .... else
      LDD    #L3FFF

;-----
; LK UP IAC MANUAL T.F.
; OFFSET vs MPH
;-----

LC1AA: ASLD
      ASLD
      LDX    #$86B2           ; IAC MANUAL T.F. OFFSET vs MPH
      JSR    LE3D0           ; 2D LOOK UP, NO OFF SET

      TAB
      PULA
      ABA
      BCC    LC1B9           ; If no overflow
                                     ; ... else
      LDAA   #255           ; Else use 255

LC1B9: LDAB    L00FD           ; IAC MODE WD
      ORAB   #$10           ; BIT 4, OLD PK/NEUT, LAST PID
      SUBA   L0115           ; TPS FOLLOWER DIFF
      BCS    LC1C8
                                     ; .... else
      BPL   LC1DB

      LDAA   #$7F
      BRA    LC1DB

LC1C8: BMI     LC1CC

```

```

        LDAA      #L0081
LC1CC:  NEGA
        PSHB
        CMPA      #$0003
        BCC      LC1D8
                                ; ... else
        LDAB      $8663
                                ; DISABLE CLSD LP PID FOR 0.20 MSEC AFTER
        STAB      L011B
                                ; DISABLE CLSD LP PID TMR
LC1D8:  ORAA      #$80
        PULB
LC1DB:  STAA      L0111
                                ; REQUEST MOTOR STEPS
        ASLA
        BEQ      LC1EE
                                ; ... else
        BCS      LC245
                                ; .... else
;-----
; RESET TPS FOLLOWER MODE
;-----
        LDAA      L002B
                                ; IAC CURRENT POSITION
        CMPA      $866A
                                ; MAX STEP MOTOR POSIT POSSABLE
                                ; WORST CASE VALUE, ( FRM ORIFICE)
        BCS      LC245
                                ; .... else
        ANDB     #$EF
                                ; 1110 1111, CLR BIT 4
        BRA      LC1FA
LC1EE:  ANDB     #$CF
                                ; 1010 1111, CLR BITS 4 & 6
        BPL      LC1FA
                                ; ... else
        LDAA      L003B
                                ; CPU INPUT STATUS WD
        BITA     #$08
                                ; b3, HI PWR STEER PRESS
        BNE      LC1FA
                                ; .... else
        ANDB     #$00BE
;-----
; OPEN LOOP IAC
;
; (COLD AND START UP)
;-----
;-----
; BGN OPEN LOOP
; START UP CNT'L
;-----
;-----
; ALDL MODE 4,  COMMAND IAC
;
; ANHT/4298 P/N 16134297
; ECM TYPE $8D, ECM P/N 1227727
;
;
; ECM 8D, ECM P/N 1227730
;-----
LC1FA:  BRCLR    L0048,$01,LC210
                                ; BR IF NOT IN ALDL MODE 4
                                ; ... else
;
; CK ALDL MD 4 CNT'L WD'S
; L0194, b1 1 = IAC OPN/CLSD LP CNT'L ENABLED

```

```

; L0195, b1
;
LDAA    L0194                ; CNT'L WD
BITA    #$02                ; b1
BEQ     LC210                ; BR IF NOT b1
; ... else
LDAA    $0195                ; CNT'L WD
ANDB    #$FB                ; CLR b2
BITA    #$02                ; b1
BNE     LC245                ; BR IF b1
; ... else
ORAB    #$04                ; set b2
;-----

LC210:  BITB    #$04          ; b2
        BNE     LC245        ; BR IF b2
; ... else
LDAA    L0083                ; VSS TABLE INPUT
BNE     LC241
; ... else
LDAA    L0097                ; TF TPS
CMPA    $8647                ; ENAB IF TPS > 1.2%
BCC     LC241
; ... else
;-----
; OPEN LOOP MODE
;
;-----
LDAA    L0112                ; IAC CNT'L SPEED
ADDA    $8631                ; 1000 RPM IAC CMD SPD  OPN LP OFFSET
BCC     LC229
; ... else
LC229:  LDAA    #255          ;
        CMPA    L0059        ; RPM/12.5
        BCC     LC233
;
LDAA    #L0082
ORAB    #2
;
BRA     LC242                ; EXIT W/ SAVE
;-----
; CHK RPM G.T. FLAG
;-----
LC233:  LDAA    #L0002
        BITB    #L0002
        BEQ     LC242
; ... else
;-----
; DISABLE CLSD LP PID
;-----
ORAB    #4                  ; b2
;
LDAA    $8662                ; DISABLE CLSD LP PID FOR 0 MSEC AFTER
; OPEN LOOP DISABLE
STAA    L011B                ; DISABLE CLSD LP PID TMR

LC241:  CLRA
;
LC242:  STAA    L0111        ; REQUEST MOTOR STEPS
LC245:  STAB    L00FD        ; IAC STATUS
;-----

```

```

* IAC PID ROUTINE, (50 Msec)
*
*
*****
LC247: LDAA    L0000                ; NINOR LP CNTR
      ANDA    #$07
      BEQ     LC250
      JMP     LC4F6
      ; ... else

LC250: PSHX
      PSHX
      PSHX
      PSHX
      TSY
      BCLR   L00FE, #$20          ; CLR b5, PID CMD STEP ACTIVE
      ; (IAC Md Wd #2)

      LDAA   L005D                ; FLT COOLANT TEMP
      LDX    #$8632              ; IAC CMD SPEED vs COOL TABLE
      JSR    LE3D0               ; 2D LOOK UP, NO OFF SET

      TAB
      BRCLR  L00FE, #$80, LC269  ; BR IF NOT b7, PK/NEUT, 1 = NEUT
      ; (IAC Md Wd #2)
      ; ... else
      ADDB   $862E               ; 100 RPM ADDED IF IN PK/NEUT

LC269: BRSET  L00FD, #$80, LC270 ; BR IF AC STATUS OFF
      ADDB   $8630               ; 12.5 RPM ADDED IF A/C ON

LC270: BRCLR  L003A, #8, LC27E   ; HOT RE-STRART REATRD OFF
      LDAA   L0063                ; INV MAT
      LDX    #$8673              ; HOT RESTART CMD SPEED vs MAT TBL
      JSR    LE3D0               ; 2d LOOK UP, NO OFF SET

      ABA
      TAB
LC27E: ADDB   L01B5                ; DIFF DESIRED RPM FOR LOW BATTERY
      ;-----

      ;-----
      ; ALDL MODE 4, IAC CONTROL
      ;
      ;
      ; ANHT/4298 P/N 16134297
      ; ECM TYPE $8D, ECM P/N 1227727
      ;
      ; CNT'L WD $0198, b0
      ; CNT'L WD $0198, b1
      ; CNT'L WD $0199, NEW IAC VALUE
      ;
      ; ECM 8D, ECM P/N 1227730
      ;-----
BRCLR  L0048, #$01, LC293        ; BR IF NOT IN ALDL MODE 4

LDAA   L0198                    ; CNT'L WD.. ,b0
BITA   #$01                     ; b0,
BEQ    LC293                     ; BR IF NOT b0, MOD'ING
      ; ... elsec
BITA   #$02                     ; b1
BEQ    LC293                     ; BR IF NOT b1, MOD'ING
      ; ... else

```

```

LDAB    L0199                ; GET NEW IAC VAL, (RPM or STEPS)
;-----

;-----
LC293:  STAB    L0112                ; CNT'L WD.. ,IAC CNT'L SPEED
        CMPB    L0059                ; RPM/12.5
        BCC     LC29D
        BSET    L00FD,#$02           ; ... else
        ; SET MOTOR RESET STARTED BIT

LC29D:  LDX     #$8628                ; 0.188 COEF RPM/12.5 FILTER
        LDD     L010F                ; RPM

        SUBD    L0059                ; RPM/12.5
        RORA
        RORB
        ASRA
        RORB
        ASRA
        RORB
        ASRA
        RORB
        BSR     LC311

        STAB    L010F                ; RPM
        CLRA
        CLR    CLR
        STD     0,Y

        LDAB    L0059                ; RPM/12.5
        STD     4,Y

        LDAB    $0112                ; IAC CNT'L;L SPEED
        SUBD    4,Y
        BSR     LC311

        STAB    4,Y
        TBA
        BPL     LC2CB
        ; ... else
        NEGA

LC2CB:  STAA    2,Y
        CMPA    $1D,X
        BLS     LC2D5
        ; ... else

        CLRA
        BRA     LC2DB

LC2D5:  LDAA    L0116                ; DEAD BAND TMR
        INCA
        BEQ     LC2DE                ; BUMP UP TMR
        ; BR IF Z
        ; ... else

LC2DB:  STAA    L0116                ; DEAD BAND TMR

LC2DE:  LDAA    2,Y
        CMPA    $8646                ; COMAND SPEED DEAD BAND TLL
        BLS     LC2E9

        CLRA
        BRA     LC2EF

```



```

LC2E9: LDAA    L0117                ; RPM ERR TMR
      INCA
      BEQ     LC2F2

      ; ... else
LC2EF: STAA    L0117                ; RPM ERR TMR

LC2F2: LDAA    L011B                ; DISABLE CLSD LP PID TMR
      BEQ     LC2FB

      ; ... else
      DECA    ; DECR TMR
      STAA    L011B                ; DISABLE CLSD LP PID TMR

LC2FB: LDAA    L0097                ; TF TPS T/F Ld Axies Var
      CMPA    $1F,X
      BCC     LC307

      ; ... else
      LDAA    L0083                ; VSS TABLE INPUT
      CMPA    $21,X
      BLS     LC321

      ; ... else
LC307: LDAA    L00FD                ; IAC Cnt'l Word
      ANDA    #$FE                ; CLR b0
      STAA    L00FD                ; IAC Cnt'l Word

      TBA
      JMP     LC391

LC311: ASLD
      BCC     LC31A                ; MILT * 2
                                      ; IF NO OVERFLOW
                                      ; ... else

      INCA
      BEQ     LC31F                ; BR IF Z
                                      ; ... else

      CLRB
      BRA     LC31F

LC31A: TSTA
      BEQ     LC31F                ; BR IF Z
                                      ; ... else

      LDAB    #256

LC31F: RORB

      RTS
      ;-----

      ;-----
      ;
      ;
      ;
      ;-----

LC321: TST     L0083                ; VSS TABLE INPUT
      BNE     LC382

      ; .... else
      LDAA    L011B                ; DISABLE CLSD LP PID TMR
      BNE     LC390

      PSHB

      LDAA    L00FE                ; IAC Md Wd #2
      BPL    LC339

      ; ... else
      BITA    #$10                ; b4,
      BNE    LC34A                ; BR IF b4

```

```

; .... else
CLR B CLR
ORAA # $10 ; b4, OLD PK/NEUT IN NEUT, LAST PID
BRA LC341

LC339: BITA # $10 ; b4
BEQ LC34A ; BR IF NOT b4
; ... else

LDAB $3C,X
ANDA # $EF

LC341: STAB L011B ; DISABLE CLSD LP PID TMR
; ... else
CLR L0116 ; DEAD BAND TMR
CLR L0117 ; RPM ERR TMR

LC34A: STAA L00FE ; IAC Md Wd #2
BMI LC381
; ... else
LDAA L0117 ; RPM ERR TMR
CMPA $838C ; .04 Msec MIN RPM ERR
BLS LC381
; ... else
LDAA L00FE ; IAC Md Wd #2
ASRA
BCS LC381
; .... else
BRSET L003A,#8,LC381 ; BR IF HOT RE-STRART REATRD ON

LDAA $0000 ; NINOR LP CNTR
CMPA # $30 ;
BNE LC381 ; BR IF NOT LP # $30
; .... else
LDAA L0119 ;
LDAB L0120 ;
BEQ LC381 ;
; ... else
BMI LC37A
INCA
CMPA L011D ; INIT IAC WARM PARK POSIT
BLS LC37E
; ... else
LDAA L011D ; INIT IAC WARM PARK POSIT
BRA LC37E

LC37A: DECA
BNE LC37E ; .... else

LC37E: STAA L0119 ;

LC381: PULB

LC382: LDAA L00FD ; IAC MODE WD
BITA # $04 ; b2, MOTOR RESET WORKING
BEQ LC394
; ... else
BITA # $18 ; b3 & b4
BNE LC390 ; .... else

```

```

BRCLR    L003B,#$08,LC3D2          ; BR IF NOT b3, HI PWR STEER PRESS
                                           ; (CPU INPUT STATUS WD)
                                           ; ... else
LC390:   TBA
LC391:   TSTA
        BPL      LC3C0
LC394:   CLR      L0116              ; DEAD BAND TMR
        LDAB     L0083              ; VSS TABLE INPUT
        BEQ      LC3BD
                                           ; ... else
        LDAB     $0120
        BLE     LC3BD
                                           ; ... else
        LDAA     $0000              ; NINOR LP CNTR
        ANDA     #$18               ; MASK 0001 1000
        CMPA     #$18
        BNE     LC3BD              ; BR IF b3 & b4
                                           ; ... else
        LDAA     L00FD              ; IAC MODE WD
        BITA     #8                 ; B3 1 = START UP DELAY DONE
        BNE     LC3BD
                                           ; ... else
        ANDA     #$EF               ; CLEAR b4
        STAA     L00FD              ; IAC MODE WD
        LDAB     #129
        STAB     L0111              ; REQUEST MOTOR STEPS
;
; SET PID CMD STEP ACTIVE
;
        LDAA     L00FE              ; IAC Md Wd #2
        ORAA     #$20               ; SET b5, PID CMD STEP ACTIVE
        STAA     L00FE              ; IAC Md Wd #2
LC3BD:   JMP      LC4ED
;-----
;-----
LC3C0:   LDAB     $49,X
        BEQ      LC3C8
                                           ; ... else
        LDAB     L0083              ; VSS TABLE INPUT
        BNE     LC394
                                           ; .... else
LC3C8:   CMPA     $22,X
        BLS     LC394
        LDAA     L00FD              ; IAC MODE WD
        ANDA     #$E7               ; CLR b3 & b4
        STAA     L00FD              ; IAC MODE WD
LC3D2:   LDAA     L00FE              ; IAC Md Wd #2
        BMI     LC3D7
                                           ; BR IF
                                           ; ... else
        INX
LC3D7:   LDAA     $1B,X
        STAA     5,Y
        LDAA     $2E,X
        LDAB     $2C,X
        STD     6,Y

```

```

LDAB      4,Y
LDX       #$8628      ; IDLE AIR CONTROL CONSTANTS &
                    ; TABLES FOR PID F.B.

LDAA      2,Y
CMPA      5,Y
BLS       LC431

LC3F1:    TSTB
          BMI      LC419
          LDAA     $23,X
          TST      4,Y
          BMI      LC422

          PSHA
          LDAA     2,Y
          CMPA     $8653      ; 50 RPM Min UNDER SPD ERROR
                    ; TO ENABLE HI PROP & DERIV
                    ; GAIN ADDITIONS, (RPM/12.5)

          PULA
          BLS      LC422

          CMPB     $25,X
          BLS      LC422

          ADDA     $26,X
          BCC      LC422

                    ; ... else

          PSHB
          MUL
          ADCA     #$00
          PULB
          ABA
          BPL      LC425

          LDAA     #$7F
          BRA      LC425

LC419:    LDAA     $24,X
          NEGB
          MUL
          ADCA     #0      ; ROUND OFF
          NEGA
          BRA      LC425

LC422:    MUL
          ADCA     #0      ; ROUND OFF

LC425:    TAB
          CLRA
          TSTB
          BPL      LC42B
                    ;
                    ; ... else

LC42B:    COMA
          ADDD     0,Y
          STD      0,Y
                    ;
                    ;

LC431:    CPX      #$862C
          BEQ      LC44A
                    ;
                    ; ... else
          LDX      #$862C
                    ; STP'S ADDED FOR FAN 2
          LDAB     L010F
                    ; RPM
          TBA
                    ;
          BPL      LC440
                    ;
                    ; ... else

```

```

LC440:  NEGA                ;
        STAA             $03,Y ;
        CMPA            $3F,X  ;
        BCC             LC3F1  ;
        ; ... else
        LDD             0,Y    ;
LC44A:  JSR              LC311 ;
        STAB            0,Y    ;
        BPL             LC453  ;
        ; ... else
LC453:  NEGB                ;
        TBA                ;
        LDAB            L00FE  ; IAC Md Wd #2
        BPL             LC45D  ; BR IF
        ; ... else
        LDAB            $2C,X  ;
        MUL                ;
LC45D:  ADCA             #0    ; Round off
        CMPA            $35,X  ;
        BCS             LC46C  ;
        ; .... else
        CLR             L0113  ;
        LDAB            0,Y    ;
        STAB            $4,Y   ;
        BRA             LC4AE  ;
LC46C:  LDAB             3,Y    ;
        CMPB            6,Y    ;
        BLS             LC477  ;
        ;
        JMP             LC4E7  ;
        ;-----
LC477:  LDAB             7,Y    ;
        LDAA            2,Y    ;
        CMPA            5,Y    ;
        BHI             LC48F  ;
        ; ... else
        CMPA            $3D,X  ;
        BLS             LC4E7  ;
        ; ... else
        LDAB            $3C,X  ;
        ;
        TST             L00FE  ; IAC Md Wd #2
        BPL             LC48F  ; BR IF
        ; ... else
        LDAB            $3B,X  ;
LC48F:  MUL                ;
        ASLD            ;
        LDAB            4,Y    ;
        BPL             LC497  ;
        ; ... el;se
LC497:  NEGA                ;
        ADDA            L0113  ;
        BVC             LC4A1  ;
        ; .... el;se
        LDAA            #$7F   ;
        BCC             LC4A1  ;
        ; ... else
        NEGA                ;
LC4A1:  STAA            4,Y

```

```

        STAA    L0113
        BPL     LC4AA
; ..... else
LC4AA:  NEGA
        CMPA   $35,X
        BCS    LC4EA
; ..... else
LC4AE:  LDAB   $34,X
        MUL
        ADCA   #$00
        STD    6,Y
; ..... else
        LDAA   $35,X
        MUL
        ADCA   #0
        BPL    LC4BF
; Round off
        LDAA   #127
; ... else
LC4BF:  STAA   5,Y
        LDAB   6,Y
        LDAA   4,Y
        ROLA
        LDAA   5,Y
        BCC    LC4D1
; ... else
        NEGA
        ORAB   #$80
; b7
LC4D1:  STAB   L0111
        ASLB
        BEQ    LC4DD
; REQUEST MOTOR STEPS
; ... else
;
; PID CMD STEP ACTIVE
;
        LDAB   L00FE
        ORAB   #$20
        STAB   L00FE
; IAC Md Wd #2
; SET b5, PID CMD STEP ACTIVE
; IAC Md Wd #2
LC4DD:  LDAB   L0113
        BEQ    LC4ED
; ... else
        STAA   L0113
        BRA    LC4ED
LC4E7:  CLR    L0113
LC4EA:  CLR    L0111
; REQUEST MOTOR STEPS
LC4ED:  PULX
        PULX
        PULX
        PULX
;-----
;-----
;
;
;-----
LC4F1:  LDD    L0059
        STD    L010F
; RPM/12.5
; RPM
LC4F6   JMP    LB6A7
; EXIT TO MJR LP

```

* Air Fuel 12.5 Msec LOOP, \$8D ECM

*

*

```
LC4F9:  LDX      L00B1                ; Ign Off tmr
        BRCLR   L0036,#$10,LC545    ; BR IF IGN ON

        CPX     #$08                ;
        BCS     LC548                ;
                                           ; .... else
        BCLR    L0001,#$FF          ; CLR ALL Non Vol MD WD 1 BITS
        BCLR    L0002,#$EF          ; CLR ALL EST MD BITS
                                           ; EXCEPT FAILED THIS TEST BIT, (B8)

        BCLR    L0003,#$FF          ; CLR ALL NV MD WD 2
        CLR     L00D1                ; VATS ERR CNT'R
        CPX     $8012               ; SEC, IF IGN OFF THIS TIME TURN OFF ECM
                                           ; (3.2 SEC)

        BCS     LC529                ;
                                           ; .... else
        LDAA    L0083                ; VSS TABLE INPUT
        BNE     LC526                ;
                                           ; .... else
        BRCLR   L0038,#$C0,LC51F    ; BR IF NOT b6 & b7,
                                           ; HI BAT, SHIFT LIGHT ON

        BRA     LC526                ; GO WAIT FOR SWI INT

LC51F:  LDAA    #$30                 ; Set A/D Ch for MAP read, (ch 3)
        JSR     LF0D2                ; Go read A/D

        STAA    L002E                ; Non Vol A/D BARO

        ;-----
        ; WAIT HERE FOR INTERUPT
        ; SWI VECTOR TO $6000
        ;-----

LC526:  SWI     ; SOFTWARE INTERUPT
        BRA     LC526                ; LOOP HERE
        ;-----

LC529:  INX
        STX     L00B1                ; Ign Off tmr

        CLRA
        CLR    B
        CLR    B
        STD     L0019                ; SAVE Eng run time

        LDAA    L0037                ; MINOR LOOP MD WD 1
        BPL    LC537

        ; ... else
        JSR     LE548                ; Go init BLM cells

LC537:  BCLR    L0037,#$80           ; CLR B7, ENG RUNNING BIT

        CLR     L01A4                ; INJECTS SINCE START
        CLR     L0148                ; SMALL REF PERIOD CNTR FOR ENG WARM UP

        BCLR    L0047,#$05           ; CLR b0 & b2
                                           ; b0 1 = IN SINGLE FIRE MODE
                                           ; b2 1 = DELIVER 0 FUEL, SINGLE FIRE

        BRA     LC54B
        ;-----
```

```

;-----
LC545: LDX      #$FFFF
LC548: INX
      STX      L00B1          ; Ign Off tmr

LC54B: LDAB     L0038          ; MODE WD 2
      ANDB     #$DF          ; CLR b5, IN ALDL MODE

      BRCLR    L0037,$80,LC568 ; BR IF NOT b7, ENGINE RUNNING
      ; ... else
      LDAA     $8249          ; ERROR WD 5 MASK, (0000 0000b)
      BITA     #02           ; CK BIT 1, (CODE 22 LO TPS)
      BEQ      LC570          ;
      ; ... else
      LDAA     L0094          ;
      CMPA     $8265          ; IF TPS A/D > = 12 DISABLE ERR 22
      BCC      LC570          ;
      ; ... else
      BSET     L004D,#2       ; SET 22, TPS LOW BIT
      ORAB     #$20           ; SET b5
      BRA      LC570

LC568: LDAA     $83E5          ; 19% TPS IS LOW TPS
      STAA     L0099          ; SAVE LOW TPS THRESH

      CLR      L009A

LC570: STAB     L0038          ; MODE WD 2

      BRCLR    L004A,#1,LC595 ; BR IF TCC UNLOCKED

      LDAA     L0095          ; TPS Ld val
      SUBA     L0096
      BCS      LC595          ; BR IF
      ; .... else
      BRCLR    L003B,$04,LC587 ; BR IF NOT b2, IN 4th GEAR
      ; (CPU INPUT STATUS WD)
      ; ... else
      CMPA     $8311          ; UNLOCK TCC IF DIFF %TPS GT 2.3%
      ; G.T. IN 12.5 Msec

      BHI      LC58C

      BRA      LC595

LC587: CMPA     $8312          ; UNLOCK TCC IF %TPS G.T 3.5%
      BLS      LC595          ;

LC58C: BCLR     L004A,$01      ; CLR b0, TCC LOCKED

      LDAB     $8313          ; DIFF TPS UNLOCK TIME (12 SEC'S)
      STAB     L0144          ; TCC LOCK TIMER

LC595: LDAA     L0095          ; TPS Ld val
      STAA     L0096          ; 12.5 Msec OLD TPS Ld val

      BRCLR    L0037,$80,LC5A0 ; BR IF NOT b7, ENGINE NOT RUNNING
      ; TO CRANK FUEL ROUTINE

      JMP      LC5EA          ; TO IDLE TEST
;-----

```

* CRANK FUEL ROUTINE

*

```
LC5A0: LDAA    L005D                ; FLT COOLANT TEMP
      BRCLR   L0046, #80, LC5A8    ; BR IF NOT b7
                                          ; B7 1 = USE MAT FOR FUEL CALC
                                          ; ... else

      LDAA    L0063                ; INV A/D MAT
LC5A8: CMPA    #208                 ; 116c
      BLS     LC5AE                ; IF MAT L.T. 92 DEG C
                                          ; ... else

      ;-----
      ; LK UP CRANK PW vs COOLANT TEMP
      ;
      ;-----

LC5AE: LDAA    #208                 ; 116c
      LDX     #$83AF              ; CRANK FUEL PW vs COOLANT TABLE
      JSR     LE3D0               ; 2D LOOK UP, NO OFF SET

      LDX     #$83AD              ; CRANK PW vs COOLANT TEMP SCALAR
      JSR     LE33F              ; 8 x 16 MULT

      PSHB
      PSHA
      JSR     LE604               ; DO SCALED BARO FOR TBL LK UP'S

      ;-----
      ; LK UP CRANK PW MULT vs BARO
      ;
      ;-----

      LDX     #$83DF              ; CRANK FUEL PW MULT vs BARO TABLE
      JSR     LE3CC              ; 2D LK UP

      TSX
      JSR     LE33F              ; 8 x 16 MULT

      PULX
      ASLD
      BCS     LC5E2
                                          ; .... else

      PSHB
      PSHA

      LDAA    L0095                ; TPS Ld val
      LDX     #$83BD              ; CRANK FUEL PW MULT vs TPS
      JSR     LE3C5              ; 2D LK UP W/LINE COUNT

      TSX
      JSR     LE33F              ; 8 x 16 MULT

      PULX
      ASLD
      BCS     LC5E2
                                          ; .... else

      ASLD
      BCC     LC5E5
                                          ; ... else

LC5E2: LDD     #$FFFF

LC5E5  STD     L00E2                ; Base Pulse width
      JMP     LCCD8
```

```

;-----
; IDLE TEST
; TYOE 8D ECM
;-----
LC5EA: BCLR      L003E,#$04      ; AIR MODE WD
      LDAB      L0095          ; TPS Ld val
      CMPB      $8494         ; 2.3% TPS MAX FOR IDLE
      BHI       LC5FE

      ; ... else
      LDAB      L0083          ; VSS TABLE INPUT
      CMPB      $8495         ; MAX SPD FOR IDLE, 15 MPH
      BHI       LC5FE

      ; ... else
      BSET      L003E,#$04     ; AIR MODE WD

;-----
; DECEL ENLEAN ROUTINE
;-----
LC5FE: BRSET     L0046,#$02,LC656 ; BR IF b1
      ; B1 1 = DECEL FUEL C/O ENABLED
      ; ... else

      LDAA      L00AB          ;
      SUBA      L0095          ; TPS Ld val
      BCS       LC656

      ; .... else
      CMPA      $85A6         ; 0.4% TPS
      BLS       LC656

      LDAA      L00AD          ;
      SUBA      L0074          ; RAW MAP
      BCS       LC656

      ; .... else
      CMPA      $85A7         ; DIFF A/D MAP VALUE
      BLS       LC656

;-----
; LK UP Diff enlean term
; mult vs Diff MAP, (A/D)
;-----
      LDX       #$85BF        ; DIFF ENLEAN TERM MULT vs Diff MAP, (A/D)
      JSR       LE3D0         ; 2D LOOK UP, NO OFF SET

      LDAB      L00AF          ; DECEL ENLEAN INJ CNT'R
      LDX       #$85AE        ; DIFF ENLEAN FACTOR N vs NUM OF
      ; INJ'S IN DECELL LEAN

      ABX
      LDAB      0,X
      ABA
      BCC       LC62B

      ; ... else
      LDAA      #255

;-----
; LK UP DIFF ENLEAN
; TERM MULT VS COOL
;-----
LC62B: TAB
      LDAA      L005D          ; FLT COOLANT TEMP
      LDX       #$85D0        ; DIFF ENLEAN TERM MULT VS COOL TBL
      JSR       LE3D0         ; 2D LOOK UP, NO OFF SET

      MUL
      ASLD
      BCS       LC63B

```

```

; .... else
ASLD
BCC LC63D
; ... else
LC63B: LDAA #$00FF
LC63D: BRCLR L004A,#1,LC647 ; Br if TCC not locked
LDAB $85A9 ; 0.50 DECEL MULT IF TCC LOCKED
MUL
ADCA #$00
LC647: COMA
BSET L003F,#$80 ; SET DECEL ENLEAN ACTIVE BIT
CLR $00A3
BCLR L0045,#$10 ; CLR USE LO ORDER NIBBLE OF
; FUEL LIMIT TBL BIT
BCLR L003E,#$01 ; AIR MODE WD
BRA LC65B
LC656: BCLR L003F,#$80 ; DECEL ENLEAN ACTIVE BIT
LDAA #255 ; MAX VAL
LC65B: STAA L00B0
LDAA L0074 ; RAW MAP
LDAB $85A8 ; 0.023 COEF FOR MAP FILTER
LDX L00AD
JSR LE31C ; LAG FILTER ROUTINE
STD L00AD
LDAA L0095 ; TPS Ld val
LDAB $85AA ; 0.027 COEF 12.5 MSEC FILTER
LDX L00AB
JSR LE31C ; LAG FILTER ROUTINE
STD L00AB
;-----
; DECEL CUT OFF ROUTINE
;
;-----
LDAA L005D ; FLT COOLANT TEMP
CMPA $85E8 ; 0.38 SEC STALL SV'R TPS FOLLOW TIME
BCS LC6FA ; BR IF ...
; .... else
BRSET L003B,#$01,LC6FA ; BR IF b0, PK/NEUT
; (CPU INPUT STATUS WD)
; .. else
BRSET L0047,#$10,LC68B ; BR IF b4 VSS FAIL
; .... else
;
; CK Vss QUAL
;
LDAA L0081 ; MPH/1
CMPA $85E9 ; 15 MPH MIN SPD FOR C/O
BLS LC6FA ; BR IF Vss G.T. THRESH
; .... else
LC68B: LDAA L0038 ; MODE WD 2
ANDA #$24 ; CLR ALL EXCEPT b2 & b6
BNE LC6AA
; .... else
;-----

```

```

; LK UP DECELL FUEL CUT OFF
; TPS THRESH vs RPM
;-----
LDAA    L0085                ; RPM/25
CMPA    #144                 ; 3600 RPM
BLS     LC699

LC699:  LDAA    #144          ; ... else
        LDX     #$85F5       ; USE 3600 RPM
        JSR     LE3D0        ; DECELL FUEL CUT OFF
        JSR     LE3D0        ; TPS THRESH vs RPM
        JSR     LE3D0        ; 2D LOOK UP, NO OFF SET

BRCLR   L0046,#$02,LC6A6    ; BR IF NOT b1
        ; B1 1 = DECEL FUEL C/O ENABLED
        ; ... else

ADDA    $85F4

LC6A6:  CMPA    L0095        ; TPS Ld val
        BCS     LC6FA       ; .... else

LC6AA:  LDAA    L005C        ; OLD RPM (FOR DIFF RPM)
        SUBA   L0059        ; RPM/12.5
        BCS     LC6BC

        ; .... else
        CMPA    $85E5       ; 50 RPM MAX INCR TO REMAIN IN C/O,
        ; (12.5 INTERVAL)
        BCS     LC6BC

        ; .... else
        LDAA    #$FF
        STAA   L013B        ; SPK ADV BLEND MULT
        BRA     LC6FA

LC6BC:  LDAB    $85E3        ; IF Kpa < 20, ENABLE C/O (low hyst)
        BRCLR  L0046,#$02,LC6C6 ; BR IF NOT b1
        ; B1 1 = DECEL FUEL C/O ENABLED
        ; .... else

        LDAB    $85E4        ; IF Kpa < 28, STAY IN C/O (upper Hyst)

LC6C6:  CMPB    L0079        ; MAP for Spd/Dens calc
        BCS     LC6FA

        ; .... else
        LDAB    $85E1        ; IF RPM > 1400, ENABLE C/O (upper hyst)

BRCLR   L0046,#2,LC6D4    ; BR IF DECEL FUEL C/O NOT ENABLED
        ; ... else
        LDAB    $85E2        ; IF RPM > 1100 STAY IN C/O, (low hyst)

LC6D4:  CMPB    L0085        ; RPM/25
        BCC     LC6FA

        ; ... else
        LDAB    L00F7
        BNE     LC6F7

BRSET   L0056,#$02,LC6E4    ; BR IF DECEL FUEL C/O ENABLED

        LDX     L009B        ; STALL SAVER TIMER
        BNE     LC6FA       ; BR IF N/Z
        ; .... else

LC6E4:  BSET    L0046,#$02    ; SET b1
        ; B1 1 = DECEL FUEL C/O ENABLED
        ; ... else

```

```

LDAA    $85F2    ; SA BLEND VAL IF IN DECEL FUEL C/O
STAA    L013B    ; SPK ADV BLEND MULT

BCLR    L003E,#$10    ; AIR MODE WD

LDX     $85EF    ; ____ SEC MIN BETWEEN CONSEC DECEL C/O's
STX     L009B    ; STALL SAVER TIMER

BRA     LC72E    ; GO CALC DIFF MAP FOR ACCEL

LC6F7:  DECB
BRA     LC6FD

;
;
;
LC6FA:  LDAB     $85E7    ; 0.25 SEC C/O QUAL TIME FOR ENABLE
LC6FD:  STAB     L00F7

BCLR    L0046,#$02    ; CLR b1,
; B1 1 = DECEL FUEL C/O ENABLED

LDAA    L003E    ; AIR MODE WD
LDX     L009B    ; STALL SAVER TIMER
BNE     LC70E

; .... else
ANDA    #$EF
STAA    L003E    ; AIR MODE WD
BRA     LC72E    ; GO CALC DIFF MAP FOR ACCEL

LC70E:  DEX      ; DEC STALL SAVER TIMER
STX     L009B    ; SAVE NEW STALL SAVER TIME
LDAB    L005C    ; OLD RPM (FOR DIFF RPM)
SUBB    L0059    ; RPM/12.5
BCS     LC72E    ; GO CALC DIFF MAP FOR ACCEL
; .... else
CMPB    $85E6    ; 87.5 RPM FOR STALL SAVER
BCS     LC72E    ; GO CALC DIFF MAP FOR ACCEL
; .... else
BITA    #$10     ; b4
BNE     LC72E    ; BR IF NOT b4,
; GO CALC DIFF MAP FOR ACCEL
; .... else
ORAA    #$10     ; SET b4
STAA    L003E    ; AIR MODE WD

LDAA    $85EE    ; ???
STAA    L009D    ; PULSE CNT'R

LDAA    $85EB    ; 0.38 SEC STALL SV'R TPS FOLLOW TIME
STAA    L00E7    ; STALL SVR TIMER

;
; CALC DIFF MAP, ACCEL
;
;
LC72E:  BCLR    L003E,#$20    ; CLR b5, AIR MODE WD

LDAA    L0071    ; NORMILIZED LOAD MAP VAL
SUBA    L00A6
BEQ     LC798
;
; ... else
STAA    L00A9    ; SAVE DIFF MAP

```

```

;-----
; LK UP DIFF MAP vs %TPS for Acell Enrich Enable
;
; (APROX 3 Kpa, 0 - 50%TPS)
;-----
LC741: LDAA    L0095          ; TPS Ld val
        CMPA    #128        ; CK FOR 50%
        BCS     LC741        ; BR IF LT 50%, (TBL MAX LMT)
        ; .... else
        LDAA    #128        ; USE 50%
        LSRA    ; DIV by 2
        PSHA    ; SAVE TO STX

        LDX     #$859A      ; DIFF MAP for Acell Enrich Enable TBL
        JSR     LE3D0       ; 2D LOOK UP, NO OFF SET

        BRCLR   L003E,#4,LC763 ; AIR MODE WD, BR IF NOT b2

;
; CALC ACELL RPM (CURRENT - OLD)
;
        LDAB    L0059        ; RPM/12.5
        SUBB    L013D        ; SUB OFF OLD RPM
        BCC     LC763        ; BR IF ACCEL
        ; ... else
        ;
        NEGB    ;
        CMPB    $857E        ; 75 RPM, NEG DECELL RPM THRESH
        ; FOR USING L857F
        BCS     LC763        ; BR IF NEG DECELL RPM GT 75 RPM
        ; .... else
        BSET    L003E,#$20   ; AIR MODE WD, SET b5

        LDAB    $857F        ; L859A THRESH MULT ON DECEL RPM
        MUL    ;
        ADCA    #0           ; round off

LC763: TAB          ; THRESH MULT TO B Reg
        LDAA    L00A9        ; DIFF MAP ACCEL VAL
        SBA     ; SUB OFF LK'ED UP THRESH
        BCC     LC76D        ; BR IF MAP ACCEL ENRICH OK TO DO,
        ; (SET FLG)
        ; ... else
        ;
        INS    ;
        CLRA    ; CLR Acell Enr MAP COOL TRIM FACTOR
        BRA     LC792        ; AND EXIT

LC76D: BSET    L0045,$08    ; SET MAP ACCEL ENRICH FLAG

;-----
; LK UP MAP Acell Enr vs
; vs DIFF MAP ABOVE MIN
;-----
LC776: CMPA    #160          ; 50kpa, (MAT TBL LU LMT)
        BLS     LC776        ; BR IF LT 50 Kpa
        ; ... else
        LDAA    #160        ; USE 50kpa, (MAT TBL LU VAL)
        LSRA    ; DIV BY 2
        LDX     #$8580      ; MAP ACEL ENRICH FACTOR
        JSR     LE3D0       ; 2d LK UP, NO OFF SET

        LDAB    L00AA        ; Acell Enr MAP COOL TRIM FACTOR
        MUL    ;
        ASLD   ;
        BCS     LC796        ; BR IF OVERFLOW

```

```

; .... else
TAB                                ; RESULT TO B Reg FOR MULT LATER
;-----
; LK UP MAP Acell Enr FACTOR MULT
; vs TPS LOAD
;-----
PULA
LDX    #$859F                      ; MAP Acell Enr FACTOR MULT vs TPS LOAD
JSR    LE3D0                        ; 2D LOOK UP, NO OFF SET

MUL                                ; APPLY FACTOR
ASLD                                       ; Mult * 2
BCS    LC796                        ; BR IF OVERFLOW
; .... else
ASLD                                       ; MULT * 2
BCS    LC796                        ; BR IF OVERFLOW
; .... else

;
; BR HERE IF ACCEL ENR BYPASSED
;
LC792: ADDA    L00A5                ; Acell Enr MAP COOL TRIM FACTOR
      BCC     LC798                ; BR IF NO OVER FLOW
; ... else
LC796: LDAA    #255                ; MAX LMT
LC798: STAA    L00A5                ; Acell Enr MAP COOL TRIM FACTOR
      BNE     LC7A3                ; BR IF N/Z
; .... else
LDX    L00F8                        ; ACCUM Acell Enr FUEL
BNE     LC7A3
; ... else
BCLR   L0045,#$0E                  ; CLR b1, 2 & 3, accel enrich
; B1 1 = MAP ACCEL ENRICH DONE 1ST TIME
; B2 1 = INJ IN LAST 6.25 Msec
; B3 1 = MAP ACCEL ENRICH ON

;-----
; READ o2 A/D FOR
; RESULT OF o2 COMP
;-----
LC7A3: LDAA    L008E                ; MNR LP FILTERED A/D o2
      STAA    L00EA                ; 12.5 Msec OLD FILT o2

LDAA    #$20                        ; SEL o2, A/D CH 2
JSR    LF0D2                        ; GO GEWT A/D RESULT

;-----
; FILTER o2
;-----
LDX    L008E                        ; MNR LP FILTERED A/D o2
LDAB   $83F1                        ; 0.937, 12.5 MSEC FILTER COEFF
JSR    LE31C                        ; LAG FILTER ROUTINE

STD    L008E                        ; MNR LP FILTERED A/D o2

*****
* PWR ENRICH ROUTINE
* (WOT)
*
*****
LDAB    L0046                        ; AFR MD WD

BCLR    L0046,#$20                  ; CLR b1, (1 = IN PWR ENR)
; B1 1 = DECEL FUEL C/O ENABLED

```

```

;-----
; ALDL MODE 4,  COMMAND AIR FUEL RATIO
;
;
; ANHT/4298 P/N 16134297
; ECM TYPE $8D, ECM P/N 1227727
;
; CNT'L WD $0198, b..
;
;
; ECM 8D, ECM P/N 1227730
;-----
BRCLR  L0048,#$01,LC7D5          ; BR IF NOT IN ALDL MODE 4
; ... else
LDAA   $0198                    ; ALDL FUNCT MOD ENABLE WD
BITA   #4                        ; b2
BEQ    LC7D5                      ;
; ... else
LDAB   $019A                    ; ALDL DS AF
CLRA
PSHB
PSHA

PULX
LDD    #$FFFF                    ;
IDIV
STX    L00F1                      ; SAVE FINAL TOTAL AF VAL

BRA    LC83A
;-----

```

```

;-----
LC7D5: LDAA   L0038                ; MODE WD 2
BITA   #$24                      ; b2 & b5, DRP HAPPENED, & IN ALDL MODE
BNE    LC802                      ; BR IF b2 & b5
; .... else
;-----

```

```

; LK UP PWR ENRICH (NORM COOL)
; TPS THRESHOLD vs RPM
;-----

```

```

LDX    #$8603                    ; PWR ENRICH TPS THRESHOLD vs RPM
LDAA   L005D                    ; FLT COOLANT TEMP
CMPA   $8602                    ; USE TBL L8608 IF COOL
; = or GT 116C (244F)

```

```

LC7E3: BCS    LC7E8                ; BR IF COOL
; .... else
;-----

```

```

; LK UP PWR ENRICH (HI COOL)
; TPS THRESHOLD vs RPM
;-----

```

```

LDX    #$8608                    ; PWR ENRICH TPS THRESHOLD vs RPM

```

```

LC7E8: LDAA   L0057                ; RPM/25
LSRA
LSRA
JSR    LE3D0                      ; 2D LOOK UP, NO OFF SET

```

```

BITB   #$20                      ; b5
BEQ    LC7F9                      ; BR IF NOT b5
; ... else
SUBA   $8601                    ; CK %TPS HYST 6.25%, PE off if
; TPS < (L8603 - L8601)
BCC    LC7F9

```



```

; ... else
CLRA
LC7F9:  CMPA    L0095    ; TPS Ld val
        BHI     LC83D    ; BR IF TPS LT TBL VAL
        CPX     #$8608   ; PWR ENRICH TPS THRESHOLD vs RPM TBL
        BEQ     LC810    ; HIGH COOLANT
LC802:  LDAA    $85FF    ; ... else
        BITB   #$20      ; APROX 30 Kpa VAC
        BEQ     LC80C    ; b5
        SUBA   $8600    ; BR IF b5
LC80C:  CMPA    L007F    ; ... else
        BHI     LC83D    ; VAC HYST FOR PWR ENRICH Kpa, (3.1Kpa)
LC810:  BSET    L0046,$20 ; ARG = Kpa * (256/80)
        ; Current manifold vacuum
LC810:  BSET    L0046,$20 ; SET b1, b1 1 = IN PWR ENR
;-----
; LK UP POWER ENRICH FUEL/AIR % CHG vs RPM
; *** WOT ***
;
; ECM $85
;-----
        LDX     #$8617   ; POWER ENRICH FUEL/AIR % CHG vs RPM
        LDAA    L0085    ; RPM/25
        JSR     LE3D0    ; 2D LOOK UP, NO OFF SET
;
; ADD 128 TO TBL VALUE
;
        TAB     ; RESULT TO B reg
        LDX     #128
        ABX     ; LK'ED UP CHG + 128
;
; ADD WOT ENR TO AFR, MUTL * 14.7 % DIV BY 256
; (AFR+WOT AFR+ 128)/256
;
        LDAB    L00F5    ; PWR ENRICH FUEL/AIR %CHG vs COOL RESULT
        ABX
        LDD     $841A    ; AFR CALIB, 445d, (14.7)
        JSR     LE424    ; 16 * 16 (RET W MIDDLE 2 BYTES IN D)
        ; (PROD/256)
        STD     L00F1    ; SAVE FINAL TOTAL AF VAL
BRCLR   L003D,$04,LC83A ; BR IF NOT IN OPN LP PRIOR TO
        ; GOING TO PWR ENRICH
        CPD     L01AF    ; OPEN LP AFR
        BHI     LC83A
        LDD     L01AF    ; OPEN LP AFR
        STD     L00F1    ; FINAL TOTAL AF VAL
LC83A:  JMP     LC8E7    ; TO FINAL AFR

```

```

LC83D: LDAB      L000D                ; TIME OUT AFR
      BCLR      L0037,#$02           ; CLR b1, HIWAY FUEL TMR RUNNING
      LDAA      #$01

      BRCLR     L0047,#$80,LC8AB      ; Br IF NOT b7, B7 1 = CLOSED LOOP
      ; ... else

      BCLR      L003D,#$04           ; CLR IN OPN LP PRIOR TO
      ; GOING TO PWR ENRICH
      ; THU FIELD SVC MODE BIT

*****
* HIGHWAY FUEL QUAL'S
*
*****
LDAA    L0079                ; MAP for Spd/Dens calc
CMPA    $8480                ; MAP DISABLE MAP VAL, 60 kpa
BHI     LC8A0                ; DISABLE HI WAY IF MAP GT 60 Kpa
      ; ... else

LDAA    L0081                ; DISABLE HIWAY Vss UPPER LMT
CMPA    $8484                ; DISABLE HIWAY IF SPD =< 255 MPH
BLS     LC8A0                ; ... else

LDAA    L005D                ; FLT INV COOLANT TEMP
CMPA    $8481                ; DISABLE HI WAY COOL THRESH
BLS     LC8A0                ; DISABLE HI WAY IF COOL =< 82c, (185f)
      ; ... else

*****
* CAN PURGE PARAMS
*****
LDAA    L0126                ; OLD CCP D.C.
CMPA    $82E6                ; 50 PCT CCP DC.
BLS     LC8A0                ; BR IF L.T. 50% CCP
      ; ... else

LDAA    L0080                ; HIWAY MODE FUEL TIMER
BSET    L0037,#$02           ; SET b1, HIWAY FUEL TMR RUNNING
BRCLR   L0039,#$80,LC878     ; IF NOT IN HIWAY FUEL

CMPA    $8483                ; 60 SEC MAX TIME IN HIWAY
BCC     LC8A0                ; BR IF LT 60 SEC IN HIWAY MODE FUEL
      ; ... else

BRA     LC883                ; LK UP HI WAY AFR vs MAP

LC878: CMPA    $8482                ; Ck 10 SEC HIWAY LOCK OUT
      BCS     LC8A7                ; BR IF TMR LT 10 SEC'S
      ; .... else

      CLR     L0080                ; CLR HIWAY MODE FUEL TIMER
      BSET    L0039,#$80           ; SET IN HIWAY FUEL BIT

*****
* LK UP HI WAY AFR vs MAP
*
*****
LC883: LDAA    L0071                ; NORMILIZED LOAD MAP VAL
      BPL     LC889

LC889: LDAA    #128                ;
      LSRA    ; DIV BY 2
      LDX     #$8485                ; HI WAY AFR vs MAP TABLE
      LDAB    #16
      JSR     LE3CC                ; 2d Lk up

      CLRB
      PSHA
      PSHB                ; SAVE AFR TO SRX

```



```

BCLR      L0002,#1                ; CLR ERR 43B EST TST IN WORK

;-----
; CK IF HEADS UP
; AIR FUEL MOD'S
;-----
LC8F5:    BRCLR    L0036,#$80,LC8FC ; BR IN NO HEADS UP
; ... ELSE
JSR      L5800                    ; HEADS UP
;-----

;-----
; GET BLM & ITS CELL
;-----
LC8FC:    BCLR     L0046,#$04       ; CLR b2
; B2 1 = BLK LRN ADDR CHANGE

LDAB     $8527                    ; IDLE CELL NUMBER, (A CELL), (4d)
LDAA     $8522                    ; FORCE IDLE CELL LRN IF CCP D.C. LT 51
CMPA     L0125                    ; PURGE D.C., (CNTS/16)
BLS      LC90F

CMPA     L0126                    ; OLD CCP D.C.
BHI      LC95D                    ; BR IF OLD D.C. LOW ENOUGH TO
; FORCE IDLE CELL LEARN

;-----
;
;-----
LC90F:    LDAB     L00DF            ; BLM CELL NUMS 0 - 7
ANDB     #3                       ; RPM INDEX ID BIT 0&1 OF
; CELL
LDX      #$850E+1                 ; BLM MEMORY CNT'L
ABX      ; ADD B to X
TSTB    ; IF RPM INDEX = 0 THEN THERE IS
; NO LOWER BOUNDRY TO CHECK
BEQ      LC929
;
; ... else
LDAA     0,X
SUBA     $8516                    ; SUB 100 RPM, EDGE TO EDGE HYSYERESIS
BCS     LC925
;
; ... else
CMPA     L0085                    ; RPM/25
BHI      LC95F                    ; BR IF RPM L.T. LIMIT

LC925:    CMPB     #3              ; IF RPM INDEX = 3 THERE IS NO UPPER
; BOUND TO CK
BEQ      LC934
;
; ... else

LC929:    LDAA     1,X
ADDA     $8516                    ; 100 RPM, EDGE TO EDGE HYSYERESIS
BCS     LC934
;
; .... else
CMPA     L0085                    ; RPM/25
BCS     LC95F

; .... else
LC934:    LDAB     L00DF            ; BLM CELL NUMS 0 - 7
ANDB     #$0C                    ; AIR INDEX IS BITS 2&3 OF CELL
LSRB    LSRB
LDX      #$8512                    ; 2000 RPM, HI BLM CELL EDGE
ABX      ; ADD B TO X
TSTB    ; IF AIR INDEX = 0, HTERE IS NO LOWER
; BOUNDRY TO CKECK
BEQ      LC950
;

```

```

; ... else
LDAA 0,X
SUBA $8517
BCS LC94C
; SUB 2 kPA, EDGE TO EDGE HYSYERESIS

; .... else
CMPA L0079
BHI LC95F
; MAP for Spd/Dens calc
; BR IF AIR L.T. LIMIT, CELL HAS CANGED

LC94C: CMPB #3
BEQ
; IF AIR INDEX = 3 THEN NO UPPER TO CK
; ... else LC95B ;

LC950: LDAA 1,X
ADDA $8517
BCS LC95B
; ADD 2 kPA, EDGE TO EDGE HYSYERESIS
;
; .... else
CMPA L0079
BCS LC95F
; MAP for Spd/Dens calc
; BR IF AIR G.T. LIMIT, CELL HAS CHANGED
; .... else
LC95B: LDAB L00DF
LC95D: BRA LC98E
; BLM CELL NUMS 0 - 7
; BRIDGE

LC95F: CLRB
; CLR CELL POINTER

BSET L0046,$0C
; b2 & 3
; B2 1 = BLK LRN ADDR CHANGE
; B3 1 = DELAY BLM UPDATE

;-----
; MODIFY BLM CELL LOC'N
; BASED ON RPM
;-----
LDAA L0085
CMPA $8510
BCS LC977
; RPM/25
; 700 RPM, LOW BLM CELL EDGE
; BR IF L.T. 700 RPM
; .... else
INCB
CMPA $8511
BCS LC977
; INCR CELL POINTER
; 1200 RPM, MID BLM CELL EDGE
; BR IF L.T. 1200 RPM
; .... else
INCB
CMPA $8512
BCS LC977
; INCR CELL POINTER
; 2000 RPM, HI BLM CELL EDGE
; BR IF L.T. 2000 RPM
; ... else
INCB
; INCR CELL POINTER

LC977: LDAA L0079
CMPA $8513
BCS LC98E
; MAP for Spd/Dens calc
; 25 kpa, LOW BLM CELL EDGE
; BR IF L.T. 25 Kpa
; .... else
ADDB #4
CMPA $8514
BCS LC98E
; ADJ CELL POINTER
; 50 kpa, MID BLM CELL EDGE
; BR IF L.T. 50 Kpa

ADDB #4
CMPA $8515
BCS LC98E
; ADJ CELL POINTER
; 80 kpa, HI BLM CELL EDGE
; BR IF L.T. 80 Kpa
; .... else
ADDB #4
; ADJ CELL POINTER

LC98E: LDX #L001B
ABX
LDAA 0,X
CMPA $851D
BHI LC99E
; BLM CELL ADDRESS BGN
; ADD B to X, POINT TO MEM CELL
; GET CURRENT BLM CELL VALUE
; CK IF Max BLM value, (160d)
; IF BLM OUT OF RANGE FLG Non Vol
; mem fail & re-init BLM

```

```

; ... else
; Ck if Min BLM value, (108d)
;
; ... else
; AIR MODE WD
; SET RAM FAIL
; AIR MODE WD

LC99E: LDAA    L003E
      ORAA    #$40
      STAA    L003E

      JSR     LE548
; Go Init BLM cells

;-----
; SAVE UPDATED CURRENT BLM VAL
;-----
LC9A7: STAB    L00DF
      STAA    L00E0
      BMI    LC9AF
; BLM CELL NUMS 0 - 7
; SAVE BLM
; LIMIT BLM TO 128

;
; SET BLM = 128 IF IN PWR ENR
;
LDAA    #128
; MID VALUE

LC9AF: LDAB    L0046
      BITB    #$20
      BEQ    LC9B7
; AFR MD WD
; B5 1 = IN PWR ENRICH
; BR IF not IN PWR ENRICH
; ... else

      STAA    L00E0
; IF YES, SET BLM = 128

*****
* CLOSED LOOP CODE
*****

;-----
; LK UP INTEGRATOR DELAY
; vs FLOW
;-----
LC9B7: LDAA    L0071
      LSRA
      LDX    #$84B4
      JSR    LE3D0
; NORMILIZED LOAD MAP VAL
; DIV BY 2
; INTEGRATOR DELAY vs FLOW
; 2D LOOK UP, NO OFF SET

      SUBA    $00F6
      BRCLR  L003E,#4,LC9C9
; COOL LEAN OFF SET FOR C/L

      LDAA    $8496
; 0.60 V FAST o2 R/L THRES AT IDLE

LC9C9: BRSET  L003F,#2,LC9D0
; BR IF AIR CONTROLLED, 0 = DIVERT

      SUBA    $849D
; 0.10 V DIFF R/L WHEN AIR DIVERTED

LC9D0: PSHA
      LDD    $848F
; IF o2 VOLTS < 0.699 THEN o2 IS READY
; A = UPPER LIM FOR CLS TO OPN LP
; B = LOWER LIM FOR CLS TO OPN LP
; MNR LP FILTERED A/D o22
;
; .... else
; o2

      CMPB   L008E
      BCS    LC9DC
;

      CMPB   L008E
      BLS    LC9E2

LC9DC: CLR    L00DE
      BSET   L0001,$01
; CLR CLSD o2 SENS NOT RDY TMR
; SET o2 SENSOR ACTIVE

LC9E2: PULA

```



```

;
; CK AFR OPT WD 4, 0000 1001
; b3, 1 = IF CCP CHG, FORCE INT TO 128
;
LDAA    $8019                ; AFR OPT WD 4
BITA    #$08                 ; b3,
BNE     LCA88                ; BR IF b3, CCP CHG,
                                ; ...FORCE INT TO 128
                                ; .... else
LCA67:  BRSET    L003F,$$80,LCA79 ; BR IF b7, DECEL ENLEAN ACTIVE
                                ; ... else
LDAB    L0079                ; MAP for Spd/Dens calc
CMPB    $85AB                ; ENAB DECELL INT IF < or = 0 Kpa
BHI     LCA97                ; BR
                                ; .... else
LDAB    L0085                ; RPM/25
CMPB    $85AC                ; ENAB DECELL INT IF G.T. 900 RPM
BLS     LCA97                ; .... else
LCA79:  LDAA    L0047         ; AFR MD WD
BITA    #$40                 ; b6, o2 RICH
BNE     LCA97                ; BR IF RICH
                                ; .... else
LDAA    L00EB                ; Clsd lp intrgrator
CMPA    #128                 ; INT CENTER
BCS     LCA97                ; BR IF INT L.T. 128
                                ; .... else
;-----
; FORCE INTIGRATOR TO 128
;-----
LCA85:  BSET    L0042,$$04    ; SET INTIGRATOR FORCED TO 128,
                                ; ... ERR44/45
LCA88:  BSET    L0046,$$01    ; SET b0
                                ; B0 1 = INTEGRATOR CHG
                                ; ... (USED ERR 32)
LDAA    #128                 ; MID POINT
STAA    L00EB                ; Clsd lp intrgrator
CLRA    #0                   ; A = 0
STAA    L00E6                ; CLR Clsd lp intrgrator TMR
STAA    L00EC                ; CLR o2 CLSD LOOP PORP CNT'S
JMP     LCB09
;-----
; CALC o2 ERROR
;-----
LCA97:  LDAA    $8497         ; 0.60 V UPPER ZERO ERR o2 AT IDLE
BRSET   L003E,$$04,LCAA9    ; BR IF IDLING
                                ; ... else
;-----
;LK UP UPPER ZERO ERROR REF FOR
; SLOW o2 R/L vs MAP TABLE
;-----
;
LDAA    L0071                ; NORMILIZED LOAD MAP VAL
LSRA
LDX     $$84A2               ; UPPER ZERO ERROR REF FOR
                                ; SLOW o2 R/L vs MAP TABLE
JSR     LE3D0                ; 2D LOOK UP, NO OFF SET
SUBA    $00F6                ; COOL LEAN OFF SET FOR C/L
LCAA9:  BRSET   L003F,#2,LCAB0 ; SET AIR CONTROLLED, 0 = DIVERT BIT
                                ; ... else

```

```

SUBA      $849D                ; 0.10 V DIFF R/L WHEN AIR DIVERTED
LCAB0:    CMPA      L0090                ; GET FILT o2 FOR GAIN SETTING
          BCS      LCADA
          ; .... else
          LDAA     $8498                ; 0.57 V LOWER ZERO ERROR o2 AT IDLE
          BRSET    L003E,#$04,LCAC6     ; BR IF IDLING
          ;-----
          ; LK UP FAST o2 R/L THRESHOLD vs MAP
          ;
          ;-----
          LDAA     L0071                ; NORMILIZED LOAD MAP VAL
          LSRA
          LDX      #$84AB                ; FAST o2 R/L THRESHOLD vs MAP TABLE
          JSR      LE3D0                ; 2D LOOK UP, NO OFF SET
          SUBA     $00F6                ; COOL LEAN OFF SET FOR C/L
LCAC6:    BRSET    L003F,#2,LCACD     ; BR IF AIR NOT DIVERTED, 0 = DIVERT
          SUBA     $849D                ; ... else
          ; SUB DIVERT OFFSET FM REF
LCACD:    SUBA     L0090                ;
          BHI     LCAD4
          CLRA
          BRA     LCAF8                ; ERROR = 0
LCAD4:    LDAB     L0047                ; AFR MD WD
          ANDB    #$DF                ; CLR b5
          ; b1 = SLOW o2 RICH
          BRA     LCAE6
LCADA:    TAB
          LDAA     L0090                ;
          SBA
          LDAB     $849F                ; 0.87 MULT GAIN FACTOR FOR POS ERRORS
          MUL
          LDAB     L0047                ; AFR MD WD
          ORAB    #$20                ; SET b5
          ; B5 1 = SLOW o2 RICH
LCAE6:    STAB     L0047                ;
          BRCLR   L003E,#$04,LCAF2     ; BR IF NOT IDLING
          ; ... else
          LDAB     $849B                ; 0.75 MULT TO CLS LP GAIN FACTOR
          MUL
          ADCA    #0                ; MULT ERRO BY SACALR IF IDLING
LCAF2:    CMPA     #88
          BLS     LCAF8                ; LIMIT ERR TO 88
          LDAA     #88
LCAF8:    PSHA
          STAA     $045C                ; ERR TO STX
          ;-----
          ; GET PORP TERM
          ;-----
          ASLA
          LDX      #$84EA                ; PORPORTIONAL STEP DURATION vs ERROR TBL

```

```

JSR      LE3D0                ; 2D LOOK UP, NO OFF SET

PSHA
LDAA     L0057                ; RPM/25
CMPA     #160
BLS      LCB0C

LDAA     #160                ;

LCB0C:  LSRA                  ; SET FOR 6 ENTRY TBL, 400-2400 RPM
LDX      #$84D8              ; PORPORTIONAL STEP SIZE GAIN vs RPM TBL
JSR      LE3D0                ; 2D LOOK UP, NO OFF SET

PULB
ABA
BCC      LCB19                ; a+b
                                ; IF NO OVERFLOW
                                ; ... else

LDAA     #255                ; MAX LIMIT

LCB19:  TAB
LDAA     L0047                ; AFR MD WD
ANDA     #$60                ; CLR ALL EXCEPT b5 & b6
                                ; B5 1 = SLOW o2 RICH
                                ; B6 1 = o2 RICH
BEQ      LCB24                ; BR IF Z
                                ; ... else

CMPA     #$60
                                ; B5 1 = SLOW o2 RICH
                                ; B6 1 = o2 RICH
BNE      LCB2B                ; BR IF NZ
                                ; .... else

LCB24:  PULA
PSHA
CMPA     $849E                ; 0.017 V MIN EEROR TO MAKE INTIG CORR
BHI      LCB35

LCB2B:  CLR      L00E6          ; CLR Clsd lp intrgrator TMR
LDAA     L00E9                ; Clsd lp PORP CNTR
CBA
BHI      LCBA3

BRA      LCBA7

;-----
; INTEGRATOR UP DATE LOGIC
;
;-----

LCB35:  PULA
PSHA
ASLA
LDX      #$8504                ; INT DELAY MULT vs ERROR TABLE
JSR      LE3D0                ; 2D LOOK UP, NO OFF SET

;
; REDUCE GAIN IF COOL
;
LDAB     L005D                ; FLT COOLANT TEMP
NEGB
SUBB     $84A1                ; 32C, (90f) OFF SET FOR COOL INTIG GAIN
BCS     LCB4B                ; If No overflow
                                ; .... else
ABA
BCC     LCB4B                ; ELSE ..
                                ;
                                ; ... else
LDAA     #255                ; LIMIT TO 255

```

```

LCB4B:  PSHA
        LDAA    L006B           ; LIMITED AIR FLOW FM IDEAL GAS LAW
        CMPA    #64             ; CK FOR LIMIT
        BLS     LCB54           ; IF LT 64, ELSE...

        LDAA    #64             ; LIMIT FOR LK UP
        ;-----
        ; LK UP  INTIGRATOR
        ; vs FLOW
        ;-----

LCB54:  ASLA
        LDX     #$84BD          ; INTIGRATOR vs FLOW TABLE
        JSR     LE3D0           ; 2D LOOK UP, NO OFF SET

        PULB
        MUL

;-----
; SLOW UP INITIGRATOR IF AT IDLE
;-----
        BRCLR   L003E,$#04,LCB68 ; BR IF NOT AT IDLE

        ADDA    $849A           ; 0 SEC'S ADDED TO  CORR INT DLY AT IDLE
        BCC     LCB68           ;
        ; ... else
        LDAA    #255            ; LIMIT TO 255

LCB68:  CMPA    L00E6           ; Clsd lp intrgrator TMR, (SEC * 40)
        BHI     LCB6F

        CLRA
        BRA     LCB72           ; TIMED OUT, CLR

LCB6F:  LDAA    L00E6           ; Clsd lp intrgrator TMR
        INCA
LCB72:  STAA    L00E6           ; Clsd lp intrgrator TMR

        BNE     LCBA7           ; IF TMR NO ZERO
        ; .... else

        LDAA    L00EB           ; Clsd lp intrgrator

        LDAB    L0047           ; AFR MD WD
        BITB    #$20            ; B5 1 = SLOW o2 RICH
        BNE     LCB86           ; BR IF RICH
        ; .... else

        CMPA    $8493           ; MAX CLSD LP CORR VAL, (180)
        BEQ     LCB9F

        ; ... else
        INCA
        BRA     LCB9F

LCB86:  CMPA    $8492           ; MIN CLSD LP CORR VAL, (80)
        BEQ     LCB9F

;-----
; IDLE SPK UNDERSPEED INTIGRATOR
; DECR INHIBIT
;-----
        BRSET   L0042,#2,LCB9E  ; BR IF ERR 44 or 45UP
        ; (DATE INT IF o2 ERROR)

        BRCLR   L0045,$#20,LCB9E ; BR IDLE SPK HI GAIN ENABLED CLR

```

```

; (IN NOT IN IDLE)
; ... else
; BR IF OVER SPEED
; ... else
; o2 Rich CNTR
; RICH TIME LMT
; .... else

LCB9E: DECA

LCB9F: STAA L00EB ; Clsd lp intrgrator
      BRA LCBA7 ; Clr Proport term

LCBA3: INS ; Clr stack error
      CLRA ; Clr Proport cnt'r
      BRA LCBCC ;

;-----
; LK UP PROPOROT STEP Sz vs Error
;
;-----

LCBA7: PULA
      ASLA
      LDX #84DE ; PROPOROT STEP Sz vs Error TABLE
      JSR LE3D0 ; 2D LOOK UP, NO OFF SET

      PSHA
      LDAA $8499 ; Proport Step Sz Gain (0d)
      BRSET L003E,#$04,LCBC6

      LDAA L006B ; LIMITED AIR FLOW FM IDEAL GAS LAW
      CMPA #64 ; CK UPPER LIMIT
      BLS LCBBF

      LDAA #$64 ; LIMIT TO 64

;-----
; LK UP PERORORTIONAL STEP Sz vs FLOW
;
;-----

LCBBF: ASLA
      LDX #84CF ; PROPOROT STEP SZ VS FLOW TBL
      JSR LE3D0 ; 2D LOOK UP, NO OFF SET

LCBC6: PULB
      MUL
      ADCA #$00
      STAA L00EC ; O2 CLSD LOOP PORP CNT'S

LCBCC: BRCLR L0047,$$40,LCBD9 ; BR IF NOT B6, O2 RICH
; ... ELSE

      TAB
      LDAA L00EB ; CLSD LP INTRGRATOR
      SBA
      BCC LCBDF ;
; ... ELSE

      CLRA
      BRA LCBDF

LCBD9: ADDA L00EB ; CLSD LP INTIGRATOR
      BCC LCBDF ; IF NO OVERFLOW
; ... ELSE
      LDAA #255 ; LIMIT TO 255

```

```

LCBDF: STAA L00E8 ; CLSD LP CORR

*****
* LK UP VE FROM MAP & RPM
* USING VE1 or VE2 TABLES
* ANHT, TYPE $8D ECM
*****
LDAB L0071 ; NORMILIZED LOAD MAP VAL
LDAA L0085 ; RPM/25
CMPA #64 ; 1600 RPM ?
BCS LCBF7 ; BR if LT 1600 RPM
; .... else

;-----
; LK UP VE vs RPM vs MAP
; (FOR RPM GT 1600)
;-----
LDX #$8797 ; VE TABLE FOR GT 1600 RPM
SUBA #$64 ; SUB OFF 166 RPM OFFSET
LSRB
CMPA #160 ; CK IF 1600 RPM
BLS LCC08 ; BR If less

LDAA #160 ; Else limit to 1600 RPM
BRA LCC08 ; TO LOOK UP ROUTINE

;-----
; LO RPM TABLE
;-----
LCBF7: LDX #$86EB ; VE TABLE FOR RPM LT 1600 RPM
SUBA #16 ; SUB OFF 400 RPM OFFSET
BCC LCBFF ;
; ... else
CLRA ; IF UNDER FLOW

LCBFF: ASLA
ASLA
CMPB #128 ; CK IF MAP LT EQ 60 Kpa
BLS LCC08 ; IF LT, GO TO LOOK UP
; ... else
LSRB ; Rescale for 10 Kpa/ Break point
ADDB #64

;-----
; DO 3D L.U. OF VE Val
; (FOR RPM LT 1600 RPM)
;-----
LCC08: JSR LE353 ; 3D Lk Up

PSHA ; SAVE UN-CORRECTED VE RESULT

;-----
; APPLY BARO MULT TO VE
;-----
JSR LE604 ; DO SCALED BARO FOR TBL LK UP'S

LDX #$881F ; Table
JSR LE3CC ; 2D LK UP

PULB ; GET UN-CORRECTED VE RESULT
MUL ; APPLY BARO CORR FM TABLE
ASLD ; SCALING
BCC LCC1C ;
; ... else

LDAA #$00FF

;-----

```

```

; APPLY CONDITIONAL
; FILTERING TO VE
;-----
LCC1C: LDAB    L0095      ; GET TPS LD VAL
      CMPB    $86E0     ; TP LIMIT,
      BCC     LCC34     ; IF TPS% TO HI
                          ; ... else
      LDAB    L0085     ; RPM/25
      CMPB    $86DF     ; 1200 RPM, VE filtering TPS LIMIT
      BHI     LCC34     ; EXIT & USE _____ VE VALUE

      LDX     $0067     ; OLD FILTERED V.E, (TRACKING)
      BEQ     LCC34     ; EXIT & USE OLD VE VALUE
                          ; ... else
      LDAB    $86E1     ; VE FILTER COEF,
      JSR     LE31C     ; LAG FILTER ROUTINE

LCC34: STD     L0067     ; SAVE FILTERED V.E, (TRACKING)
      *****

      *****
      * CALCULATE SPEED DENSITY
      *
      * ANHT, TYPE $8D ECM
      *****
;-----
; GMS of AIR per CYLINDER
; (CALC VE)
;-----
      LDX     L006D     ; GET INV MAT
      LDD     L0079     ; MAP FOR SP/DENS CALC
      SUBD    L0069     ; PARTIAL PRESS DUE TO EGR
                          ; SUB OFF EGR PART PRESSURE
      BCC     LCC40     ; IF NO UNDERFLOW
                          ; ... else
      CLRA                    ; LIMIT, (MAP-EGR) = 0
      CLRB

;
; VE CALC = ((MAP - EGR part press) * CYL VOL)/((MAT + 233) * 128)
;
LCC40: JSR     LE424     ; 16 * 16 )RET W MIDDLE 2 BYTES IN D)

      PSHB
      PSHA

      TSX                    ; Point to VE CAL ON STACK
      LDAA    L0067     ; FILTERED V.E, (TRACKING)
      JSR     LE33F     ; 8 x 16 MULT,

      STD     L006F     ; GRAMS AIR/CYL

      PULX                    ; CLR STX

;-----
; CALC AIR FLOW USING
; IDEAL GAS LAW
;-----
      LDX     L00B3     ; CURRENT MNR LOOP DRP PERIOD
      IDIV                    ; X = (((EGRO * INV MAT) * VE * 1/DRP PER
;
; REMAINDER TO STX
;

```

```

PSHB
PSHA
PSHX
;
; GET INTEGER PORTION
;
PULA
PULB
ASLD ; Mult * 2, FOR SCALE
STD L0064 ; SAVE INTG PART OF AIR FLOW

;
; REMAINDER FROM STX
;
PULA ;
PULB ;

LDX L00B3 ; CURRENT MNR LOOP DRP PERIOD
FDIV ; FRACT = REMAINDER
PSHX ; 16 bit FRACT TO STX

PULA
PULB
ASLD ; SCALE & ROUND FRACT
STAA L0066 ; SAVE FRACT PART OF AIR FLOW
BCC LCC6B ; BR IF NO FRACT OVERFLOW
; ... else
LDX L0064 ; GET INTEG AIR FLOW
INX ; INCR FRO FRACT
STX L0064 ; AIR FLOW

LCC6B: TAB ; FRACT TO B Reg
LDAA L0065 ; GET LO BYTE ON INTEG

BRCLR L0064, #$FF, LCC75 ; BR TO SPD DENS CALC IF
; AIR FLOW LT 256 G/SEC
; ... else
LDD #$FFFF ; LIMIT TO 255.996 Gms\
LCC75: STD L006B ; SAVE LIMITED AIR FLOW FM IDEAL GAS LAW

;-----
; SPEED DENS BPW CALCULATION
;-----
LDD L006F ; GRAMS AIR/CYL
LDX L00F1 ; FINAL TOTAL AF VAL
JSR LE424 ; 16 * 16 )RET W MIDDLE 2 BYTES IN D)

LDX $841C ; SEC/GRAM PROD OF INJ FLOW Rate,
; (0.359 SEC/GRAM, 2.86 g/Sec)
JSR LE3EE ; 16 x 16 (RET W/UPPER 2 BYTES IN D)

ROLB ; MULT X 2
ROLA
STD L00E2 ; Base Pulse width
;-----

;-----
; DO BLM CORRECTION
;
;-----
LDX #L00E2 ; BPW
LDAA L00E0 ; GET BLM
JSR LE586 ; Factor in a 8 X 16 Multiply
;-----

```



```

; DO CLSD LP CORR FACTOR
;
;-----
CLRA                ; A = 0
LDAB    L00E8       ; Clsd lp corr
SUBB    #128        ; Mid point
BMI     LCC9B       ; BR IF L.T. 128 TO SUB OFF CORR
; ... else
ADDD    L00E2       ; ADDCORR TO Base Pulse width
BRA     LCCA8

LCC9B:  NEGB
        PSHB
        PSHA
        TSX
        LDD    L00E2       ; Base Pulse width
        SUBD    0,X       ; SUB OFF CORR FM BPW
        PULX
        BCC    LCCA8       ; BR IF NO UNDERFLOW
; ... else
        CLRA                ; A = 0
        CLRB                ; B = 0

LCCA8:  STD    L00E2       ; Base Pulse width
        BEQ    LCCD2       ; BR TOFUEL OUT IF BPW = 0
; ... else

;-----
; DO FINE CORRECTION
; PORP TERM
;-----
LDAB    L00EC       ; o2 CLSD LOOP PORP CNT'S
LSRB                ; DIV by 2
BCC     LCCB6       ; BR IF NO UNDERFLOW
; ... else
LDAA    L003E       ; AIR MODE WD BOOK KEEP F/F'S EACH INJ
RORA
ADCB    #00         ; ROUND

LCCB6:  CLRA
;
; SAVE BPW TO STX FOR FINE CORRECTION
;
        PSHB
        PSHA
        TSX
        LDD    L00E2       ; Base Pulse width

        BRSET  L0047,#$40,LCCC4 ; BR IF b6, o2 RICH
; ... else

        SUBD    0,X       ; IF LEAN SUB OFF 1/2 FINE RES
        BRA     LCCC6

LCCC4:  ADDD    0,X       ; IF RICH ADD 1/2 FINE RES

LCCC6:  PULX                ; REMOVE OLD BPW FM STX

        PSHB
        PSHA
;-----

;-----
; DO DECEL ENLEAN MULT
;

```

```

;-----
TSX                ; POINT X TO A REG FOR DECEL MULT
LDAA    L00B0      ; GET DECEL EN-LEAN MULT
JSR     LE33F      ; 8 x 16 MULT

STD     L00E2      ; SAVE FINAL BPW
PULX

LCCD2: JSR     LE8FA      ; FUEL OUT

JSR     LCD6C      ; ASYNC OUT
*****

*****
* 50 Msec AIR FUEL MINOR LOOP
*
*****
LCCD8: LDAA    L0000      ; NINOR LP CNTR
      ANDA    #$06      ; LP'S 1 & 9
      BEQ     LCCE1      ;
      ; ... else
      JMP     LCD69      ; EXIT

;-----
; UPDATE BLM
;-----
LCCE1: LDD     L0046      ; AFR MD WD
      BITB    #02      ; B1 1 = DECEL FUEL C/O ENABLED
      BEQ     LCD52      ; BR IF NOT DECEL FUEL C/O ENABLED
      ; ... else

      BITA    #$08      ; b3 1 = DELAY BLM UPDATE
      BNE     LCD52      ; BR IF DELAY BLM UPDATE
      ; .... else

      LDAA    L00EB      ; GET Clsd lp intrgrator
      CMPA    #128      ; IF E.Q. 128, NO CORRECTION
      BEQ     LCD52      ; ... else

      LDAB    L00ED      ; BLM UPDATE TIMER
      INCB    ; INCR VALUE
      BNE     LCCF7      ; BR IF NON ZERO
      ; ... else
      DECB    ; DECR TMR VAL

LCCF7: STAB    L00ED      ; BLM UPDATE TIMER
      CMPB    $851B      ; 1.35 sec BLM update rate, (SEC*20)+1
      BCS     LCD55      ; .... else

      SUBA    #128      ; IF L.T. 128
      BCS     LCD0E      ; .... else
      CMPA    $851F      ; If INIT (Clsd Lp) GT 128+5,
      ; enable BLM update

      BLS     LCD55

      LDAA    L0047      ; AFR MD WD
      ASLA    ; MULT * 2
      BMI     LCD55      ; BR IF o2 IS RICH
      ; ... else
      BRA     LCD19      ;

LCD0E: NEGA

```

```

        CMPA    $8520                ; If INIT (Clsd Lp) LT 128-5,
        BLS     LCD55                ; enable BLM update
        LDA    L0047                ; BR IF o2 IS LEAN
        ASLA    L0047                ; AFR MD WD
        BPL     LCD55                ; MULT * 2
        ;
        LCD19: LDX     #L001B        ; ... else
        LDAB    L00DF                ; BLM ADDRESS BGN
        ABX     L00DF                ; GET CURRENT BLM CELL
        LDA    0,X                    ; ADD B TO X
        CMPB    $8527                ; GET CURRENT BLM CELL VALUE
        BNE     LCD2E                ; IDLE CELL ? NUMBER, (A CELL), 4d
        ;
        LDAB    L0125                ; BR IF NOT
        CMPB    $8521                ; .... else
        BHI     LCD4E                ; GET PURGE D.C., (CNTS/16)
        ;
        LCD2E: BRCLR   L0047,#$40,LCD41 ; DISABLE IDLE CELL LRN IF CCP DC G.T. 0
        ; BR IF D.C. TO HI FOR IDLE LEARN
        ; BR IF NOT b6, o2 RICH
        ; ... else
        ;
        SUBA    $851C                ; BLM UPDATE AMOUNT, (2)
        BCS     LCD3C                ; IF NO UNDERFLOW, CK FOR MIN VAL
        ; .... else
        CMPA    $851E                ; Min BLM value, 108d
        BCC     LCD4E                ; IF NO OVERFLOW
        ; ... else
        LCD3C: LDA    $851E          ; Min BLM value, 108d
        BRA     LCD4E                ; EXIT W/ MIN BLM
        ;
        LCD41: ADDA    $851C          ; 2, BLM update amount
        BCS     LCD4B                ; IF OVERFLOW, USE MAX BLM
        ; .... else
        CMPA    $851D                ; Max BLM value, 160d
        BLS     LCD4E                ; EXIT IF L.T. 160
        ;
        LCD4B: LDA    $851D          ; Max BLM value, 160d
        ;
        LCD4E: STAA    0,X            ; SAVE BLM
        STAA    L00E0                ; BLM
        ;
        LCD52: CLRB                    ; CLR TIMER
        STAB    L00ED                ; BLM UPDATE TIMER
        ;
        ;-----
        ;
        ;-----
        LCD55: LDAB    L00DF          ; BLM CELL NUMS 0 - 7
        CMPB    $8527                ; IDLE CELL NUMBER 4
        BEQ     LCD5F                ; BR IF CELL = #4
        ; ... else
        CLRB                    ; CLR UPDATE TMR
        STAB    $00E1
        ;
        LCD5F: LDA    L0046          ; AFR MD WD
        ANDA    #$F7                ; CLR b3
        ; B3 1 = DELAY BLM UPDATE
        STAA    L0046                ; AFR MD WD
        ;
        LDA    L0059                ; FILT RPM/12.5
        STAA    L005C                ; OLD RPM (FOR DIFF RPM)
        ;
        LCD69: JMP     LB6A7          ; EXIT TO MJR LP SUB, (INTERUPT)

```

```
* ASYNC FUEL OUT
*
* OUTPUT ASYNC FUEL FROM DIFF TPS and FUEL CUT OFF
* STALL SAVER
*
* AE BPW = (BPW * L8534) + (L853B) * DIFF TPS * 32 *
*           L8543 * L8554
```

ORG \$CD6C

LCD6C:

```
BRCLR L0037,$80,LCD74 ; BR IF NOT b7, 1 = ENGINE RUNNING
; ... else
BRSET L0045,$80,LCD77 ; BR IF TPS ACCEL ENRICH ON BIT SET
; ... else
LCD74: JMP LCDFA ; EXIT
```

```
;-----
; INDEXED LK UP ACELL ENRICH FACTOR PER BPW
; (0 - 7)
; (PUMP SHOT)
;-----
```

LCD77:

```
LDAB L00A2 ; Acell Enr COUNTER
INC L00A2 ; INCR NUM PW's SENT
CMPB #7 ; CK IF 7 PW's
BLS LCD82 ; BR IF L.T. 7
```

LCD82:

```
LDAB #7 ; MAX VAL
LDX #$853B ; ACELL ENRICH FACTOR PCR BPW (0 - 4)
ABX ; ADJ POINTER FOR LK UP
LDAA 0,X ; GET FACTOR FM TBL
PSHA ; SAVE TO STX
LDD L00E2 ; GET BPW
PSHB ; SAVE BPW TO STX
PSHA
TSX ; POINT TO VAL ON STX
LDAA $8534 ; 0.125 MULT FOR 12.5 Msec TPS Acell Enr
; SCALAR
JSR LE33F ; 8 x 16 MULT
```

```
PULX
PSHB
PSHA
TSX
LDAA $8531 ; 50% TPS DIFF FOR V6 ACELL ENR CALC
CMPA L009E ; GET DIFF TPS FOR ENRL
BCS LCDA1 ; BR IF GT 50% TPS
; .... else
LDAA L009E ; DIFF TPS VAL FOR ENR
```

LCDA1

```
CLRB ; B = 0
LSRD ; DIFF TPS/32
LSRD
LSRD
ADCB #0 ; ROUND
ADCA #0
PSHB ; SAVE DIFF TPS/32 TO STX
PSHA
TSX
LDAA $8535 ; 0.125 MULT FOR DIFF TPS * 32
```

```

JSR      LE33F          ; 8 x 16 MULT

PSHB
PSHA          ; SAVE DIFF TPS/32 * FACT TO STX

LDAA      L009E          ; DIFF TPS VAL
ASLA
BCC       LCDBB          ; IF NO OVER FLW
; ... else
LDAA      #255           ; MAX VAL

;-----
; LK UP  ASYNC FACTOR vs DIFF TPS
; TBL = FACTOR * 128
;-----
LCDBB:  LDX      #$8543          ; ASYNC FACTOR vs DIFF TPS
        JSR      LE3D0          ; 2D LOOK UP, NO OFF SET

        TSX
        JSR      LE33F          ; 8 x 16 MULT

        ASLD
        BCC      LCDCB          ; IF NO OVERFLOW
; ... else
        LDD      #$FFFF          ; USE MAX LIMIT

LCDCB:  PULX
        PULX
        TSX
        ADDD     0,X            ; ADD SCALED BPW TO SCALED DIFF TPS

        PULX
        PSHB
        PSHA
        TSX
        LDAA     2,X            ; GET ACELL ENRICH FACTOR PER BPW
; (FROM L853B LK UP)
        JSR      LE33F          ; 8 x 16 MULT

        PULX
        INS
        ASLD
        ASLD
        BCC      LCDE2          ; IF NO OVERFLOW
; ... else
        LDD      #$FFFF          ; USE MAX LIMIT

LCDE2:  STD       L00FA          ; CURRENT ACELL ENRICH FUEL PW

        LDAA     L009F          ; COOL FACTOR FOR ASYNC FUEL
;-----
; FACTOR IN
;-----
        LDX      #L00FA          ; CURRENT ACELL ENRICH FUEL PW
        JSR      LE586          ; Factor in a 8 X 16 Multiply

        LDX      $8532          ; 6.92 MSEC MAX ACELL ENR PW LMT
        CPX      L00FA          ; CURRENT ACELL ENRICH FUEL PW
        BCS      LCDFD          ; BR IF PW LT LMT
; .... else
        LDX      L00FA          ; CURRENT ACELL ENRICH FUEL PW
        CPX      $83EF          ; 0.85 Ms, MSEC MIN ASYNC PULSE
        BCC      LCDFD          ; BR IF NEW AE PW GT MIN
; ... else
LCDFD:  LDX      #$00           ; ZERO OUT
LCDFD:  STX      L00FA          ; CURRENT ACELL ENRICH FUEL PW

```

```

LCDFFF  BRCLR  L003E,#$10,LCE11      ; BR IF NOT IN STALL SAVER
                                           ; ... else
BRCLR   L009D,#$FF,LCE11            ; BR IF DONE
                                           ; ... else
DEC     L009D                        ; DECR PULSE CNT'R
LDD     $85EC                        ; DECEL FUEL C/O STALL SVR
                                           ; ACCEL ENRICH PW = 1.953 Msec
ADDD    L00FA                        ; CURRENT ACELL ENRICH FUEL PW
STD     L00FA                        ; CURRENT ACELL ENRICH FUEL PW

LCE11   BRCLR  L0031,#$80,LCE25      ; BR IF NOT b7,  A/C CLUTCH OFF
                                           ; ... else
BRSET   L003A,#$04,LCE28            ; BR IF A/C ON, LAST PASS THRU ASYNC OUT
                                           ; ... else
BSET    L003A,#4                    ; SET A/C ON LAST PASS FLG
                                           ; THRU ASYNC OUT BIT

LDD     $85A4                        ; 1.95 SEC A/C ON > OFF XISSITION PW
ADDD    L00FA                        ; CURRENT ACELL ENRICH FUEL PW
STD     L00FA                        ; CURRENT ACELL ENRICH FUEL PW

BRA     LCE28                        ;

LCE25:  BCLR   L003A,#4              ; CLR A/C ON, LAST PASS
                                           ; THRU ASYNC OUT BIT

LCE28:  LDD    L00FA                  ; CURRENT ACELL ENRICH FUEL PW
ADDD    L00F8                        ; ACCUM Acell Enr FUEL
BCC     LCE31                        ; IF NO OVERFLOW
                                           ; ... else
LDD     #$FFFF                       ; USE MAX LIMIT

LCE31   STD    L00F8                  ; SAVE CCUM Acell Enr FUEL THIS INJ
LDX     L00FA                        ; CURRENT ACELL ENRICH FUEL PW
BEQ     LCE4E                        ; BR IF Z
                                           ; ... else
LDAB    L00EE                        ; BATT BPW CORRECTION
ABX                                           ; ADD TO BPW (twice)
ABX                                           ;
STX     L3FF2                        ; BPW PULSE TO CPU ***** <----
*****

JSR     LE4B7                        ; SHORT DELAY

LDAA    #4                          ; bit 2
JSR     LE5D7                        ; TRIG ASYNC FUEL OUT

LDAA    #$FB                         ; not B2
JSR     LE4B7                        ; SHORT DELAY

JSR     LE5E2                        ; RESET ASYNC TRIG
;-----

LCE4E   LDD    L012E+1                ; get accum fuel
ADDD    L00FA                        ; ADD NEW fuel pw
BCC     LCE58                        ; BR IF NO OVERFLOW
                                           ; ... else
INC     L012D                        ; OVERFLW TO 3rd BYTE OF CUM FUEL

LCE58   STD    L012E+1                ; SAVE AS CUM FUEL

```

RTS

* BARO.SRC

*

* >>>> SEGMENT 0 OF MAJOR LOOP EXEC <<<<

*

* Checks if conditions are ok to update BARO.

* If ok an offset based on RPM & TPS is added to the

* MAP value to form BARO. If the offset is too hi

* the BARO update is stopped.

*

* Returns with RESULT IN RAM ADDR L007B

*

* ORG \$CE5C

```
LCE5C: BRSET    L0036,$$10,LCEBC      ; EXIT if Ign Off, B4 1 = IGN OFF
      BRCLR   L0037,$$80,LCEBC      ; ... else
      BRCLR   L0040,$$C0,LCE6A     ; BR IF NOT b7, 1 = ENGINE RUNNING
      BRA     LCEBC                 ; .... else
      ; Br IF NOT b7 & b6, ...
      ; .... 1 = ERR 33 or 34
      ; ... else
      BRA     LCEBC                 ; EXIT & DON'T UP DATE BARO
```

```
;-----
; Check TPS diff limits -0 + 1.17%
; and min TPS of 37%, exit if exceeded
;-----
```

```
LCE6A: LDD     L0095                 ; CURRENT & OLD TPS
      SBA     ; CALC DIFFERENTIAL TPS
      BCS     LCEBC                 ; EXIT & DON'T UP DATE BARO
      ; .... else
      CMPA    #3                    ; DELTA TPS <= 1.17% ?
      BHI     LCEBC                 ; EXIT & DON'T UP DATE BARO
      ;
      LDAA   L0095                 ; Low TPS limit
      CMPA   $$86C3                ; MIN TPS FOR BARO UPDATE, 37.5 %
      BCS    LCEBC                 ; EXIT & DON'T UP DATE BARO
      ; .... else
```

```
;-----
; Ck for MAP Delta
; If diff MAP g.t. 0.98 exit
; or if MAP EQ or GT 0
;-----
```

```
LDAA   L0071                 ; NORMILIZED LOAD MAP VAL
SUBA   $$0073                ; Sub old MAP, (25 msec)
BCC    LCE81                 ; Br if delta MAP >= 0
      ; ... else
      NEGA   ; Else use ABS val of Delta MAP
```

```
LCE81: CMPA   $$86C4                ; Upper delta limit, 3d, 0.975d
      BCC   LCEBC                 ; Exit & don't up date BARO
      ; ... else
```

```
;-----
; Ck for Max of 3600 RPM
; exit if RPM arg GT 3600 else scale RPM
; Also limit RPM arg to 1200 RPM
;-----
```

```
LDAA   L0085                 ; RPM/25
CMPA   $$86C5                ; Upper RPM limit, 141, 3600 RPM
BHI    LCEBC                 ; Exit & don't up date BARO
```

```

LDAB    #85                ; Scale RPM (in A reg) for Lk Up
MUL
ADCA    #0                 ; Rounding
CMPA    #48                ; Limit RPM arg to 1200
BLS     LCE98              ; If RPM arg < 1200 RPM

LDAA    #48                ; Else RPM arg = 1200

;-----
; Look up BARO adder from table
; & Check for max value
;-----
LCE98:  LDAB    L0095        ; CURRENT TPS Value
        LSRB
        LDX     $86C7        ; BARO vs TPS & RPM TABLE
        JSR    LE353        ; 3D LOOK UP

        CMPA    $86C6        ; MAX OFFSET, = 11 Kpa
        BCC    LCEBC        ; Exit
        ; ... else

;-----
; Sum in BARO to MAP & limit BARO adder
; to Kpa
;-----
ADDA    L0074                ; A/D MAP Value
BCC     LCEAC                ; IF NO OVERFLOW
        ; ... else
LDAA    #255                ; MAX VALUE

;-----
; DO MATH ....
;
; BARO Kpa = (((AD_BARO/694)*65536)+2647)/256
;-----
LCEAC:  STAA    L002E        ; CALC'ED A/D BARO VAL
        TAB
        CLRA
        LDX     #694        ;
        FDIV
        PSHX

        PULA
        PULB
        ADDD    #2647        ;
        STD     L007B        ; save BARO

LCEBC:  RTS                ; RETURN TO MAJOR LOOP EXE
*****

*****
* DESCRETE & PWM OUTPUT ROUTINE
*
*
* >>>> SEGMENT 1 OF MAJOR LOOP EXEC <<<<
*
*
*****
LCEBD:  LDAA    L0103        ; A/D BATTERY VOLTAGE
        CMPA    #171        ; 17.1 VDC
        BCS    LCED2        ; BR IF SAFE Vbatt
        ; .... else
        BRCLR   L0038,#$40,LCECD ; BR IF NOT b6, HI BATTERY Err SET
        ; .... else
        BSET    L0050,#$40   ; SET b6, HI BATTERY Err
        BRA     LCEE4        ;

```



```

LCECD  BSET    L0038,#$40          ; SET b6, HI BATTERY Err 53 FLAG
      BRA     LCED5                ;
      ;-----

LCED2  BCLR    L0038,#$40          ; CLR b6, HI BATTERY Err 53 FLAG
      ;
LCED5  LDX     L0037                ; GET MINOR LOOP MODE WD 1
      BMI     LCEDD                ; IF ENG RUNNING
      ; .... else

      ;-----
      ; ALDL MODE 4,  ???
      ;
      ;
      ; ECM 8D, ECM P/N 1227730
      ;-----
      BRCLR   L0048,#$01,LC EE0    ; BR IF NOT ALDL MODE 4
      ; .... else
LCEDD  JMP     LCF60                ; EXIT MODE 4 TO AIR CNT'L
      ;-----

LCEE0  LDX     L0042                ; DIAG MD WD 3
      BMI     LCF05                ; BR IF IN DIAGNOSTICS

      ;-----
      ; ENGINE NOT RUNNING
      ; NOT IN ALDL
      ; NOT IN DIAGNOSTICS
      ;-----
LCEE4  LDX     #$D000                ; 0 D.C. IF UNSAFE BATTERY (HI)
      LDY     #$DFFF                ;
      ;
      BCLR    L0032,#8              ; UNLOCK TCC/CARS

      LDAA    $4002                ; CPU Data latch
      ANDA    #$EF                  ; CLR b4, TURN OFF Eng Air Cnt'l
      STAA    $4002                ; CPU Data latch
      ;
      LDAA    $4004                ; Baud I/O Reg, (SCI)
      ANDA    #$FD                  ; CLR b1, TURN OFF FAN 2
      STAA    $4004                ; Baud I/O Reg, (SCI)
      ;
      LDD     L3FFC                 ; ECU CR
      ANDB   #$FB                  ; CLR b2, TURN OFF A/C CLUTCH
      BRA     LCF24                ; BR to WR to OUTPUT PWM's & ECU CR

      ;-----
      ; ENGINE NOT RUNNING NOT IN ALDL
      ; IN DIAGNOSTICS <<
      ;
      ;-----
LCF05  LDX     #$DFFF                ;
      LDY     #$D000                ;
      ;
      BSET    L0032,#$08            ; LOCK TCC/CARS
      ;
      LDAA    $4002                ; CPU Data latch
      ORAA   #$10                  ; SET b4, TURN ON EAC

```

```

    STAA    $4002                ; CPU Data latch
                                ;
    LDAA    $4004                ; Baud I/O Reg, (SCI)
    ORAA    #$02                ; SET b1, TURN ON FAN 2
    STAA    $4004                ; Baud I/O Reg, (SCI)

    LDD     L3FFC                ;
    ORAB    #$04                ; SET b2, TURN ON A/C CLUTCH
                                ;
LCF24:   STX     L3FD6            ; EGR
    JSR     LE4B7                ; SHORT DELAY
                                ;
    STX     L3FDA                ; Eng Air Sys, AIR DUTY CYCLE
    JSR     LE4B7                ; SHORT DELAY
                                ;
    STX     L3FEA                ; FAN, PWM7 REG, (PWMI)
    STD     L3FFC                ; STORE A/C CLUTCH TO ECU CR

                                ;
                                ; CK AFR OPTION WD 1 0001 0100
                                ; b1, 1 = 16 Hz CCP SOLENIOD IN USE
                                ;
    LDAA    $8016                ; OPT WD 1
    BITA    #$02                ; b1, 16 Hz CCP SOLONOID ??
    BEQ     LCF4F                ; BR IF NOT b1
                                ; ... else
    PSHX                    ; PUSH X TO USE FOR 16 Hz STUFF
    PULA                    ; GET UPPER BYTE
    ANDA    #$0F                ; MASK FOR LOWER NIBBLE
    ORAA    #$B0                ; PWM MASK FOR 16 Hz
    PSHA                    ;
    PULX                    ;
    PSHY                    ;
    PULA                    ;
    ANDA    #$0F                ; MASK FOR LO NIBBLE
    ORAA    #$B0                ; PWM MASK FOR 16 Hz
    PSHA                    ;
    PULY                    ;

                                ;
                                ; CK AFR OPT WD 3, 0101 0010
                                ;
LCF4F:   TST     $8018            ; OPT WD 3, CK FOR NORM CLSD CCP SOLENIOD
    BMI     LCF5A                ; BR IF CCP PURGE WHEN VALVE ON
                                ; ... else
    STY     L3FD4                ; ELSE PURGE BY USING 0% DC
                                ;
    BRA     LCF5D                ; EXIT

LCF5A:   STX     L3FD4            ; CCP OUT
                                ;
LCF5D:   JMP     LD0CE            ; BR TO E LIGNT & CARS

                                ;-----
                                ; AIR CONTROL
                                ;
                                ;-----
LCF60:   LDAA    $4002            ; CPU Data latch
    ANDA    #$EF                ; ASSUME AIR CNT'L OFF

                                ;-----
                                ; ALDL MODE 4, COMMAND
                                ;
                                ; CNT'L WD $0192, b4, 1 = AIR PUMP CONTROLLED

```

```

; CNT'L WD $0193, b4, 1 = AIR PUMP ON
;
; ECM 8D, ECM P/N 1227730
;-----
BRCLR   L0048, #$01, LCF79           ; BR IF NOT IN ALDL MODE 4
; ... else
LDAB    L0192                       ; DISCRETE ENABLES, ALDL
BITB    #$10                         ; b4, 1 = AIR PUMP CONTROLLED
BEQ     LCF79                         ; BR IF NOT CNT'LING AIR
; ... else
LDAB    L0193                       ; ALDL DISCRETE STATES
BITB    #$10                         ; b4 = 1 = AIR PUMP ON
BEQ     LCF83                       ; BR IF AIR NOT REQUESTED
; ... else
BRA     LCF81                       ; TURN ON AIR VALVE
;-----

;-----
;
;
;-----
LCF79:  BRCLR   L0037, #$80, LCF83     ; BR IF NOT b7, 1 =ENGINE RUNNING
; .. else
BRCLR   L003F, #$02, LCF83           ; BR IF NOT b1, {AIR DIVERTED}
; .. else
LCF81:  ORAA    #$10                   ; SET b4, {TURN ON AIR VALVE}
LCF83:  STAA    $4002                  ; CPU Data latch

;-----
; AIR SWITCH
;-----
LDX     #$D000                       ; AIR SW OFF CODE

;-----
; ALDL MODE 4,  COMMAND
;
; CNT'L WD $0192, b5, 1 = AIR SW CONTROLLED
; CNT'L WD $0193, b5, 1 = AIR SW ON
;
; ECM 8D, ECM P/N 1227730
;-----
BRCLR   L0048, #$01, LCF9D           ; BR IF NOT IN ALDL MODE 4

LDAB    L0192                       ; DISCRETE ENABLES, ALDL
BITB    #$20                         ; B5, 1 = AIR SW CONTROLLED
BEQ     LCF9D                       ; BR IF NOT CNT'LING AIR SW
; ... else
LDAB    $0193                       ; ALDL DISCRETE STATES
BITB    #$20                         ; b5, 1 = AIR SW ON
BEQ     LCF9D                       ; BR IF AIR NOT REQUESTED
; ... else
BRA     LCF9D                       ; ELSE TURN ON AIR SW
;-----

;-----
;
;
;-----
LCF9D:  BRCLR   L0037, #$80, LCF9D     ; BR IF NOT b7, 1 = ENGINE RUNNING
; ... else

;
; CK AFR OPT WD 2, 0000 0100

```

```

; b2, 1 = DUAL AIR VALVE/ 0 = SINGLE
;
LDAB    $8017                ; AFR OPT WD 2
BITB    #$04                ; b2,
BEQ     LCFAF                ; BR IF NOT b2
; ... else
BRCLR   L003F,$$04,LCFAF    ; BR IF NOT b2,
; {AIR SW'ING NOT REQUESTED}

LCFAC:  LDX    #$DFFF        ; LD X TO TURN ON AIR SW
LCFAF:  STX    L3FDA         ; ARS DUTY CYCLE

;-----
; A/C CLUTCH
;
;-----
LDD     L3FFC                ; ECU CR
ANDB    #$FB                ; A/C CLUTCH OFF CODE

;-----
; ALDL MODE 4,  COMMAND
;
; CNT'L WD $0192, b7, 1 = CONT'LING A/C CLUTCH
; CNT'L WD $0193, b7, 1 = CLUTCH ENGAGED
;
; ECM 8D, ECM P/N 1227730
;-----
BRCLR   L0048,$$01,LCFC8    ; BR IF NOT IN ALDL MODE 4
; ... else
LDX     #L0192              ; CNT'L WD
BRCLR   0,X,$$80,LCFC8     ; BR if NOT b7, NOT CMD'ING CLUTCH
; ... else
BRCLR   1,X,$$80,LCFD2     ; BR if NOT b7
; ... else
BRA     LCFD0               ; TURN ON A/C CLUTCH
;-----

;-----
LCFC8:  BRCLR   L0037,$$80,LCFD2 ; BR IF NOT b7, 1 = ENGINE RUNNING
; ... else
BRSET   L0037,$$20,LCFD2    ; BR IF b5, A/C CLUCTH OFF
; ... else
LCFD0:  ORAB    #$04        ; TURN ON A/C CLUTCH
LCFD2:  STD     L3FFC       ; WR NEW A/C CLUTCH STATE

;-----
; XMISSION TCC or CARS
;
;-----
BCLR    L0032,#8           ; TCC/CARS UNLOCK CODE

;-----
; ALDL MODE 4,  COMMAND
;
; CNT'L WD $0192, b2, 1 = ALTER TCC
; CNT'L WD $0193, b2, 1 = SHFT/TCC ON
;
; ECM 8D, ECM P/N 1227730
;-----
BRCLR   L0048,$$01,LCFEC    ; BR IF NOT IN ALDL MODE 4
; ... else

```

```

LDAA    L0192                ; CNT'L WD ..., b2
BITA    #$04                ; b2, 1 = SKIP SHFT/TCC CONTROLLED
BEQ     LCFEC                ; BR IF NOT CNT'LING TCC/CARS
                        ; ... else
LDAA    $0193                ; ALDL DESCRETE STATES
BITA    #$04                ; b2, 1 = SK SHFT/TCC ON
BEQ     LD004                ; BR IF CMD'ING TCC/CARS OFF
                        ; ... else
BRA     LD001                ; TURN ON TCC/CARS & EXIT
;-----

;-----
;
;
;
;-----
LCFEC:  BRCLR    L0037,$#80,LD004        ; BR IF NOT b 7, 1 = ENGINE RUNNING
;
; CK AFR OPT WD 3, 0101 0010
;   b2, 1 = CPU Auto Ratio Sel, 0 = TCC
;
LDAB    $8018                ; OPT WD 3
BITB    #04                  ; b2
BNE     LCFFD                ; BR b2
                        ; .... else
BRCLR   L004A,#1,LD004        ; BR IF TCC UN-LOCKED
                        ; ... else
BRA     LD001                ; TURN ON TCC/CARS & EXIT

LCFFD:  BRCLR   L004B,#2,LD004        ; IF TCC/CARS NOT ACTIVE
                        ; ... else
LD001:  BSET    $0032,#8          ; TURN ON TCC/CARS
;-----

;-----
; Carbon Can Purge
;
;-----

LD004:  CLR    CLR    B
;-----
; ALDL MODE 4,  COMMAND
;
; CNT'L WD $0192, b3, 1 = CCP CNTL'ED
; CNT'L WD $0193, b3, 1 = CCP ON
;
; CNT'L WD $0196, b3, 1 = CCP PWM MAD ENABLE
; CNT'L WD $0197, PWM DUTY CYCLE
;
; ECM 8D, ECM P/N 1227730
;-----
BRCLR   L0048,$#01,LD027        ; BR IF NOT IN ALDL MODE 4
                        ; ... else
LDAA    L0192                ; CNT'L WD .. b3
BITA    #$08                ; b3, 1 = CCP CNTL'ED
BEQ     LD01B                ;
                        ; ... else
LDAA    L0193                ; ALDL DESCRETE STATES
BITA    #$08                ; b3, 1 = CCP ON
BEQ     LD02E                ; CCP CMD'ED OFF
                        ; ... else

```

```

LDAB    #$FF                ; CCP OFF CODE

BRA     LD02E

LD01B:  LDAA    L0196        ; PWM ENABLE
        BITA    #$08        ; b3, 1 = CCP PWM CNT'L
        BEQ     LD027        ; BR IF NOT b3
        ; ... else
        LDAB    L0197        ; PWM DUTY CYCLE

BRA     LD02E
;-----

;-----
;
;
;
;-----

LD027:  BRCLR   L0037,$$80,LD02E ; BR IF NOT b7, 1 = ENGINE RUNNING
        LDAB    L0125        ; PURGE D.C., (CNTS/16)

;
; CK AFR OPT WD 3, 0101 0010
; b7, 1 = CCP WHEN OUTPUT ENRGIZED
;

LD02E:  TST     $8018        ; OPT WD 3
        BMI     LD034        ; IF PURGE WHEN VALVE IS OFF
        ; ... else

COMB

;
; CK AFR OPT WD 1, 0001 0100
; b1, 1 = 16 Hz CCP SOLENIOD IN USE
;

LD034:  LDAA    $8016        ; OPT WD 1
        BITA    #$02        ; b1, 16 Hz Valve ??
        BNE     LD041        ; BR IF b1, 16 Hz
        ; .... else

;-----
; 32 Hz OPERATION
;-----
LDAA    $$34                ;
ASLD                    ; MULT $34 * 4
ASLD
BRA     LD046                ; WR CCP P.W. & EXIT

;-----
; 16 Hz OPERATION
;-----

LD041:  LDAA    $$16        ;
        ASLD                    ; MULT $16 * 8 = $B0
        ASLD                    ; (176d)
        ASLD

LD046:  STD     L3FD4        ; WR CCP P.W.

;-----
; COOLING FAN 1
;
;
;-----
LDX     $$D000                ; FAN OFF CODE

```

```

;-----
; ALDL MODE 4,  COMMAND FAN 1
;
; CNT'L WD $0192, b1, 1 = FAN 1 CONTROLLED
; CNT'L WD $0193, b1, 1 = FAN 1 ON
;
; ECM 8D, ECM P/N 1227730
;-----
BRCLR  L0048,#$01,LD060          ; BR IF NOT IN ALDL MODE 4

LDAA   L0192                    ; DISCRETE ENABLES, ALDL
BITA   #$02                    ; b1,1 = FAN 1 CONTROLLED
BEQ    LD060                    ; BR IF NOT CNT'ING FAN
; ... else

LDAA   L0193                    ; ALDL DISCRETE STATES
BITA   #$02                    ; b1, 1 = FAN 1 ON
BEQ    LD06B                    ; BR IF FAN CMD'ED OFF
; ... else

BRA    LD068                    ;
;-----

;-----
;
;
;-----
LD060: BRCLR  L0037,#$80,LD06B    ; BR IF NOT b7, 1 = ENGINE RUNNING
; ... else
BRCLR  L0037,#$08,LD06B        ; BR IF NOT b3, ALL PID FAN STP'S ..
; ... NOT ADDED
; ... else

LD068: LDX    #$DFFF            ; FAN MAX DUTY CYCLE
LD06B: STX    L3FEA            ; SAVE NEW FAN STATE
;-----

;-----
; COOLING FAN 2
;
;-----
LDAA   $4004                    ; Baud I/O Reg, (SCI)
ANDA   #$FD                    ; FAN OFF CODE
;-----

;-----
; ALDL MODE 4,  COMMAND
;
; CNT'L WD $0192, b5, 1 = FAN 2 CONTROLLED
; CNT'L WD $0193, b5, 1 = FAN 2 ON
;
; ECM 8D, ECM P/N 1227730
;-----
BRCLR  L0048,#$01,LD087        ; BR IF NOT IN ALDL MODE 4
; ... else

LDAB   $0195                    ; ALDL MODE STATES
BITB   #$20                    ; b5, 1 = FAN 2 CNTL'ED
BEQ    LD087                    ; BR IF NOT CNT'ING FAN
; ... else

LDAB   $0196                    ; PWM ENABLES
BITB   #$20                    ; b5, 1 = FAN 2 ON
BEQ    LD091                    ; BR IF FAN CMD'ED OFF
; ... else

BRA    LD08F                    ;
;-----

```

```

;-----
;
;
;-----
LD087: BRCLR   L0037,#$80,LD091           ; BR IF NOT b7, 1 = ENGINE RUNNING
; ... else
BRCLR   L003F,#$40,LD091           ; BR IF ALL PID FAN STP'S NOT ADDED

LD08F: ORAA    #$02                   ; TURN ON FAN 2
LD091: STAA    $4004                  ;
;-----

```

```

;-----
; EGR
;
;-----

```

CLRB

```

;-----
; ALDL MODE 4,  COMMAND
;
; CNT'L WD $0192, b6, 1 = EGR CONTROLLED
; CNT'L WD $0193, b6, 1 = EGR ON
;
; CNT'L WD $0196, b0, 1 = EGR PWM ENABLED
; CNT'L WD $0197,  PWM DUTY CYCLE
;
; ECM 8D, ECM P/N 1227730
;-----

```

```

BRCLR   L0048,#$01,LD0B7           ; BR IF NOT IN ALDL MODE 4
; ... else
LDAA    L0192                       ; CNT'L WD
BITA    #$40                         ; b6, 1 = EGR CONTROLLED
BEQ     LD0AB                         ;
; ... else
LDAA    L0193                       ; CNT'L WD
BITA    #$40                         ; b6, 1 = EGR ON
BNE     LD0C5                       ; EGR CMD'ED ON
; .... else
LDAB    #$FF                          ;
BRA     LD0C5                       ; TURN EGR OFF

```

```

LD0AB: LDAA    L0196                   ; PWM ENABLES
BITA    #$01                         ; b0, 1 = EGR PWM ENABLED
BEQ     LD0B7                       ; BR IF NOT b0
; ... else
LDAB    L0197                       ; PWM DUTY CYCLE
BRA     LD0C5                       ;
;-----

```

```

;-----
;
;
;-----

```

```

LD0B7: BRCLR   L0037,#$80,LD0C5           ; BR IF NOT b7, 1 = ENGINE RUNNING

LDAB    #$FF                          ; TURN OFF EGR CODE
BRSET   L0043,#4,LD0C5                ; EGR DIAG IN WORK
; (IF Err 32 IN WORK, OFF EGR)
; ... else
LDAB    L0123                          ; OUTPUT EGR D.C.

```


COMB

```
LD0C5: LDAA    #52                ;  
      ASLD                ; MULT 52 * 4 = 208 = 32 Hz  
      ASLD  
      ORAB    #03          ; SET LOWER 2 BITS OF D.C.  
      STD     L3FD6       ; WR EGR TO ___
```

```
;-----  
; CK ENG LIGHT  
;  
;-----
```

```
LD0CE: LDX     #$D000          ; Eng LITE OFF CODE
```

```
;-----  
; ALDL MODE 4,  COMMAND SES LIGHT  
;  
; CNT'L WD $0195, b2, 1 = ENG LITE CNTL'ED  
; CNT'L WD $0196, b2, 1 = LIGHT ON  
;  
; ECM 8D, ECM P/N 1227730  
;-----
```

```
BRCLR  L0048,#$01,LD0E5      ; BR IF NOT IN ALDL MODE 4  
; ... else  
LDAA   $0195                ; ALDL MODE STATES  
BITA   #$04                 ; b2, 1 = ENG LITE CNTL'ED  
BEQ    LD0E5                ; BR IF NOT CMD'ING E LITE  
; ... else  
LDAA   $0196                ; PWM ENABLES  
BITA   #$04                 ; b2, 1 = LIGHT ON  
BEQ    LD0F4                ; BR IF CARS LITE CMD'ED OFF  
; ... else  
BRA    LD0F1  
;-----
```

```
;  
;  
;  
;-----
```

```
LD0E5: BRSET   L0042,#$08,LD0F1 ; BR IF b3, {Enb LIGHT BULB TST IN WORK}  
; (TURN Eng LITE ON IF IN BULB TST)  
; ... else  
BRCLR  L0037,#$80,LD0F4      ; BR IF NOT b7, {1 = ENGINE RUNNING}  
; ... else  
BRCLR  L0038,#$80,LD0F4      ; BR IF NOT b7, {Eng LITE}  
; ... else
```

```
LD0F1: LDX     #$DFFF
```

```
LD0F4: STX     L3FD8
```

```
;-----  
; Comp Aided Ratio Selection LIGHT  
;  
;-----
```

```
LDX     #$D000              ; CARS LITE OFF CODE
```

```
;-----  
; ALDL MODE 4,  COMMAND CARS LITE  
;  
; CNT'L WD $0195, b4, 1 = CARS LITE CONT'ED  
; CNT'L WD $0196, b4, 1 = CARS LITE ON  
;  
;
```

```

; ECM 8D, ECM P/N 1227730
;-----
BRCLR   L0048,#$01,LD10E      ; BR IF NOT IN ALDL MODE 4
                                ; ... else
LDAA    L0195                  ; ALDL MODE STATES
BITA    #$10                   ; b4, 1 = CARS LITE CNT'ED
BEQ     LD10E                  ; BR IF NOT CMD'ING CARS LITE
                                ; ... else
LDAA    L0196                  ; PWM ENABLES
BITA    #$10                   ; b4, 1 = CARS LITE ON
BEQ     LD11D                  ; BR IF CARS LITE CMD'ED OFF
                                ; ... else
BRA     LD11A                  ;
;-----

;-----
;
;
;-----
LD10E:  BRSET   L0042,$$10,LD11A  ; BR IF CARS LIGHT TST IN WORK
                                ; (TURN CARS LITE ON IF IN BULB TST)
                                ; ... else
BRCLR   L0037,$$80,LD11D        ; BR IF NOT b7, 1 = ENGINE NOT RUNNING
                                ; ... else
BRCLR   L004B,#2,LD11D         ; BR IF CARS NOT ACTIVE
                                ; ... else
LD11A:  LDX     $$DFFF           ; CARS MAX D.C.
LD11D:  STX     L3FCC           ; WR NEW CARS STATE

RTS                                ; RETURN TO MAJOR LOOP EXE

*****

*****
* OIL TEMPERATURE
*
* >>>> SEGMENT 2 OF MAJOR LOOP EXE <<<<<
*
*
*****
LD121:  BCLR    L0043,$$80        ; CLR ERR 52/62 BIT

LDAA    $$01                    ; SEL A/D CH 1, (OIL TEMP)
JSR     LE2D4                    ; A/D MUX READ

STAA    L01BC                    ; A/D OIL TEMP RESULT
LDX     $$F116                  ; OIL TEMP LIN TABLE
JSR     LE3D0                    ; 2D LOOK UP, NO OFF SET

LDAB    $83E8                    ; OIL TEMP FILT COEF
LDX     L01BD                    ; LINEAR OIL TEMP
JSR     LE31C                    ; LAG FILTER ROUTINE

STAA    L01BD                    ; SAVE LINEAR FILT OIL TEMP

;-----
; ERR 52 OIL TEMP LOW
;-----
LDAA    L01BC                    ; A/D OIL TEMP RESULT
LDAB    $$80                    ; BIT CODE FOR ERR 52
CMPA    $82A8                    ; TEMP VAL THRESH, 245d
BHI     LD152                    ; IF OIL TEMP LO, (A/D HI)
                                ; ... else
;-----

```

```

; ERR 62 OIL TEMP HI
;-----
LDAA    $01BD                ; FILT A/D OIL TEMP
LDAB    #2                   ; BIT CODE FOR ERR 62
CMPA    $82AD                ; TEMP VAL THRESH
BCS     LD160                ; BR IF TEMP OK, (EXIT)
; .... else
LD152:  LDX     L0019         ; Eng run time
CPX     $82A6                ; RUN TIME TRESH FOR VATS ERR 52/62
; 1800 SEC'S
BLS     LD160                ; BR IF NOT TIME YET, (EXIT)
; ... else
BSET    L0043,#$80          ; SET ERR 52/62 BIT
ORAB    $0050
STAB    $0050

LD160:  RTS                    ; RETURN TO MAJOR LOOP EXE
*****

*****
* MISC 100 MSEC ROUTINE
* TYPE $8D ECM, 1227730
*
* >>> SEGMENT 4 OF MAJOR LOOP EXEC <<<<
*****

;-----
; CK IF DIAG SW in DIAG MODE
;
; SET FLAG L0038 = 1 FOR BLINK OUT DIAGNOSTICS
; REQUIRES ALDL LINE TO BE WITHIN 780 mV of gnd
;-----
LD161:  BCLR    L0038,#$10    ; CLR b4, DIAGNOSTIC SW MODE

LDAA    #$70                 ; SEL A/D CH 7, (ALDL)
JSR     LF0D2                ; Go read A/D

CMPA    #40                  ; 780 mvdc
BCC     LD170                ; BR IF NORM SELECTED
; ... else
BSET    L0038,#$10          ; SET b4, DIAG SW in DIAG MODE
; (Blink out diagnostics)
;-----

;-----
; MISC 100 MSEC ROUTINE
;-----
LD170:  LDAA    L01C2         ; STACK OVER WRITE TEST ADDR
; CK FOR STACK OVERFLOW
BEQ     LD178                ; BR OF 'OK'
; ... else
BSET    L0036,#4            ; SET STACK OVERFLOW FLAG

LD178:  LDD     #0             ; MAKE SURE DLY'ED SART OF INJ --
STD     L3FCE                ; REG'S IS CLR

*****
* MAKE CENTRAL COMPUER MODULE
* (CPU) OUTPUT MODE WORD
*****
CLRA

BRCLR   L0003,#$20,LD185    ; BR IF NOT b5, VATS
; ... else

```

```

        ORAA    #$01                ; VATS DISABLING FUEL, (BIT 0)
LD185: BRCLR   L0031,#$80,LD18B    ; BR IF NOT b7, A/C CLUTCH ON
        ; ... else
        ORAA    #$02                ; A/C CLUTCH ON, (BIT 1)
LD18B: BRCLR   L0038,#$24,LD191    ; BR IF NOT b5 & b2, ERR 21 or 22, (TPS)
        ; ... else
        ORAA    #$04                ; TPS FAIL, (BIT)
LD191: BRCLR   L0038,#$C0,LD197    ; BR IF NOT b..., ERR 33 or 34, (MAP)
        ; ... else
        ORAA    #$08                ; MAP FAIL (BIT 3)
LD197: BRSET   L0046,#$10,LD19F    ; BR IF NOT b4, 1 = ERR 23 THIS PASS,
        ; (MAT)
        ; ... else
        BRCLR   L0041,#$02,LD1A1    ; BR IF NOT b1, ERR 23, (MAT)
        ; ... else
LD19F: ORAA    #$10                ; MAT FAIL (BIT 4)
LD1A1: BRCLR   L0043,#$80,LD1A7    ; BR IF NOT b7, ERR 52 or 62, (OIL T)
        ; ... else
        ORAA    #$20                ; OIL T FAIL, (BIT 5)
LD1A7: BRCLR   L0038,#2,LD1AD      ; BR IF NOT b1, COOL ERR THIS START UP
        ; ... else
        ORAA    #$40                ; SET b6, COOL SENSOR FAIL
LD1AD: BRCLR   L0038,#$10,LD1B3    ; BR IF NOT b4, IN DIAG MODE
        ; (Blink out diagnostics)
        ; ... else
        ORAA    #$80                ; SET b7, ECM IN DIAG MODE
LD1B3: STAA    L0049                ; SAVE UP DATED CCM OUTPUT MODE WD
        *****
BRSET   L0037,#$80,LD1E0          ; BR IF NOT b7, 1 = ENGINE RUNNING
        ; ... else

;-----
; ENGINE MOT RUNNING
;
;-----
LDAA    L0046                ; CLR DECEL F C/O & P/E FLAG
ANDA    #$D5                ; CLEAR b0,2, 4, 6 & 7
        ; B0 1 = INTEGRATOR CHG (USED ERR 32)
        ; B2 1 = BLK LRN ADDR CHANGE
        ; B6 1 = o2 RICH
        ; B7 1 = USE MAT FOR FUEL CALC
STAA    L0046                ; SAVE AFR MD WD

LDAA    L005D                ; FLT COOLANT TEMP
STAA    L0060                ; USE AS START UP COOLANT

BRSET   L0001,#$40,LD1CF        ; BR IF BAD SHUTDOWN

```

```

        CMPA    $8031                ; WARM TEMP THRES
        BCS    LD1CF                ;
        ; .... else
        BSET    L0001,#$40          ; SET WARM KICK DN ENAB FLAG

LD1CF  CLRB                    ;
        LDAA    $83F3                ; o2 DEFAULT FILT VALUE
        STD     L008E                ; MNR LP FILTERED A/D o2
        ;
        STAA    $0092                ;
        ;
        LDAA    $8629                ; Cold IAC park posit
        STAA    L0119                ;
        ;
        JMP     LD261                ; EXIT

;-----
; ENGINE RUNNING
;-----
LD1E0  BCLR     L0002,#$10          ; CLR SKP IAC MOTOR RESET BIT

        LDX     $0092                ; OLD DIAG o2 FILT VAL
        LDAA    L008E                ; NEW MR LP o2 FILT VAL
        LDAB    $83F2                ; 0.195, 100 MSEC FILTER COEF
        JSR     LE31C                ; LAG FILTER ROUTINE

        STD     L0092                ; NEW DIAG o2 FILT VAL

        LDAA    L0095                ; TPS LD
        BNE     LD25B                ;
        ; .... else
        LDAA    L0083                ; VSS TABLE INPUT
        BNE     LD25B                ; ONLY LRN AT IDLE
        ; .... else
        LDAA    L005D                ; FLT COOLANT TEMP
        CMPA    $869A                ; 87 DEG c TEMP THRESH
        BLS     LD261                ; DONT LRN IF TO HOT

        CMPA    $869B                ; LEARN WHN COOL < 103.3 C, (216 F)
        BCC     LD261                ; DONT LRN IF TO COLD
        ; ... else
        LDAA    L01B1                ; Idle spd Time out steps
        BNE     LD261                ; DONT LRN IF TO SOON
        ; .... else
        BRSET   L003A,#$08,LD261    ; BR HOT RE-STRART REATRD ON
        ; (DON'T LRN IF IN HOT RE-START RETARD)

        LDAA    L01B5                ; DIFF DESIRED RPM FOR LOW BATTERY
        BNE     LD261                ; DONT LRN IF LOW BAT ADJ
        ; .... else
        LDAA    L01B4                ; IAC K/A MOTOR TMR
        CMPA    $86A0                ; 50 SEC TO LEARN MIN MOTOR K/A STEPS
        BCC     LD221                ; BR IF TIME TO UP DATE LRN VALUE
        ; ... else

        INCA                    ;
        BEQ     LD261                ;
        ; ... else
        STAA    L01B4                ; IAC K/A MOTOR TMR

        BRA     LD261                ; EXIT

;-----
; UPDATE LEARN
;-----
LD221: CLR     L01B4                ; IAC K/A MOTOR TMR
        LDAA    L002D                ; Keep alive IAC posit

```

```

        CMPA    L0119                ;
        BEQ     LD261                ; DONT LRN IF NO CHANGE
                                           ; ... else
        BCS     LD238                ; BR IF NEED TO LRN UP
                                           ; .... else

        DECA
        CMPA    $869C                ; MIN KEEP ALIVE MOTOR STEPS, 10 STP'S
        BCC     LD241                ; ... else
        LDAA    $869C                ; MAX KEEP ALIVE MOTOR STEPS, 70 STP'S

        BRA     LD241

LD238:  INCA
        CMPA    $869D                ; MAX KEEP ALIVE MOTOR STEPS
        BLS     LD241                ;
                                           ; ... else
        LDAA    $869D                ;

LD241:  PSHA
        SUBA    L002D                ; KEEP ALIVE IAC POSIT
        BCC     LD247                ;
                                           ; ... else

LD247:  ADDA    L01B3
        BCC     LD24E                ; BR IF NO OVERFLOW
                                           ; ... else
                                           ; MAX VALUE

LD24E:  LDAA    #255
        STAA    L01B3                ; MAX DELTA STEPS/IDLE FOR KA LEARN, 1d
        CMPA    $869F                ;
        PULA
        BHI     LD261                ;
                                           ; ... else
        STAA    L002D                ; Keep alive IAC posit
        BRA     LD261

LD25B:  CLR     L01B3
        CLR     L01B4                ; IAC K/A MOTOR TMR

LD261:  LDAA    #$FF
        STAA    L4003                ; Set for output
                                           ; CPU DDR

        LDD     L012D                ; RUN TOTAL DELIVERED FUEL
        ASLD
                                           ; DIV BY 8
        ASLD
        ASLD
        STD     L0130                ; RUN TOTAL DELIVERED FUEL TO INST PNL

        RTS                          ; RETURN TO MAJOR LOOP EXE
*****

*****
* LOG RAM TO I^U ROUTINE
*
* >> SEGMENT 3 OF MAJOR LOOP EXEC <<
*
*****

;-----
; CK HEADS UP
; DISPLAYS ETC
;-----

LD270:  BRCLR   L0036,#$80,LD277    ; BR IF NO HEADS UP

        JSR     L5812                ; TO I^2U CODE

```

```

LD277: LDAA    #$F7
      STAA    $5000                ; 12U Bit 3, LOGGING RAM CNT'L

RTS                                ; RETURN TO MAJOR LOOP EXE
*****

*****
* A/C, CLS LP, TEST OF FAN
*
*   >>> SEGMENT 5 OF MAJOR LOOP EXE <<<
*
* ORG $D27D
*****

;-----
; CLS LOOP FUEL DETERMINATION
;
;-----
LD27D: LDAA    $0000                ; MINOR LP CNTR
      BITA    #$10                ; FAN DONE IF Bit 4 = 0
      BNE     LD289                ;
      ; .... else
      JSR     LD324                ; FAN ROUTINE, (FAN & A/C)
      JMP     LD401                ; TO A/C CLUTCH ROUTINE

LD289  BRSET   L0037,$#80,LD291    ; BR IF B7, 1 = ENGINE RUNNING
      ; .... else
      BRCLR  L0001,#8,LD2F4        ; BR IF 'OK' SHUT DOWN
      ; .... else
LD291  BRCLR  L0042,#2,LD29A        ; BR IF ERR 44 or 45
      ; ( CMD'ING OPEN LP)
      ; .... else
      CLR    L00DE                ; CLR CLSD o2 SENS NOT RDY TMR
      ; & FORSE OPN LP

      BRA    LD2F4                ;

;-----
; ALDL MODE 4,  COMMAND  OPN/CLSD LOOP
;
; CNT'L WD $0194, b0, 1 = CONTROLLING OPN/CLSD LOOP
; CNT'L WD $0195, b0, 1 = CLOSED LOOP
;
; ECM 8D, ECM P/N 1227730
;-----
LD29A  BRCLR  L0048,$#01,LD2AC      ; BR IF NOT IN 8192 Baud MODE 4

      LDX    #L0194                ; MODE ENABLE CNT'L WD, (ALDL)
      BRCLR  0,X,$#01,LD2AC        ; BR IF NOT b0,

      BRCLR  1,X,$#01,LD2F4        ; BR IF b0 = 0, 1 = CLSD LOOP

      JMP    LD2EF                ; BR TO SET CLSD LP FLG IF
      ; CMD'ING CLSD LP

;-----

;-----
; ENGINE RUN TIME BIT ENABLE TEST
;
;-----
LD2AC  BRSET   L0001,#2,LD2CB        ; BR IF CLS LOOP TIMER DONE

```

```

LDD      L0019                ; Eng run time
LSRD
LDX      #$848C              ; SEL TIME THRES BASED ON Start Up COOL

LDAA    L0060                ; START UP COOLANT
CMPA    $848B                ; COOL THESH FOR COLD CLS LP TMR, (14c)
BLS     LD2C4                ;

INX     ; INCR ADDR
CMPA    $848A                ; COOL THRES FOR HOT CLSD LP, (40c)
BLS     LD2C4                ;

INX     ; INCR ADDR

LD2C4:  CMPB    0,X           ;
BCS     LD2F4                ; BR IF TIME TOO SHORT FOR CLS LP
; .... else
BSET    L0001,#2            ; SET BIT 1, CLS LP TIMER DONE

;
; COOLANT ENABLE TEST
;

LD2CB:  LDAA    L005D         ; FLT COOLANT TEMP
CMPA    $8489                ; MIN COOL FOR CLSD LP, (40c/104f)
BLS     LD2F4                ; BR IF TOO COLD FOR CLSD LP

;-----
; A/F HU SLEW CLS LP DISABLE TEST
; (MOD AFR)
;-----
BRCLR   L0036,#$80,LD2DC    ; BR IF NO HEADS UP
LDAA    L004C                ; HEADS UP FLG FOR SLEW
ANDA    #$60                 ; Mask for mod'ing AFR
BNE     LD2F4                ; BR If cnt'ling AFR
; .... else

;
; o2 SENSOR READY TEST
;

LD2DC:  LDAA    L00DE         ; CLR CLSD o2 SENS NOT RDY TMR
CMPA    $8491                ; 5 SECONDS MAX TIME IN WINDOW
BCC     LD2E8                ; BR IF L.T. 5 SEC
; .... else
INCA    ; incr CLSD o2 SENS NOT RDY TMR
STAA    L00DE                ; CLR CLSD o2 SENS NOT RDY TMR

BRA     LD2EB                ;

LD2E8   BCLR    L0001,#1      ; CLR o2 SENS RDY FLG
LD2EB   BRCLR   L0001,#01,LD2F4 ; BR IF o2 SENS NOT RDY
; .... else

LD2EF   BSET    L0047,#$80    ; SET CLS b7,
; B7 1 = CLOSED LOOP

BRA     LD2F7                ; CL LEARN QUAL'S

LD2F4   BCLR    L0047,#$80    ; CLR b7,
; B7 1 = CLOSED LOOP

*****
* LEARN ENABLE QUAL'S
*
* TYPE $8D ECM
*****
LD2F7   BRCLR   L0047,#$80,LD31A ; EXIT IF NOT IN CLSD LOOP,

```



```

;
; CK COOLANT LMT'S
;
LDAA    L005D                ; FLT COOLANT TEMP
CMPA    $8518                ; LOW COOL THRES FOR BLM UPDATE, (50 C)
BLS     LD31A                ; BR IF TOO COLD FOR LEARN
; ... else
CMPA    $8519                ; HI COOL THRES FOR BLM UPDATE, (140 C)
BHI     LD31A                ; BR IF TOO HOT FOR LEARN
; ... else

;
; CK AFR FOR STOCH
;
LDX     L00F1                ; FINAL TOTAL AFR VAL
CPX     $841A                ; AFR CALIB, (14.7)
BNE     LD31A                ; BR IF AFR N.E. STOCH
; .... else

;
; CK MAP LOW LMT FOR LEARN
;
LDAA    L0079                ; MAP for Spd/Dens calc
CMPA    $851A                ; IF MAP L.T. 0 then disable BLM update
BCS     LD31A                ; EXIT VIA DISABLE LRN
; .... else

;
; SET LEARN ENABLE FLG
;
BSET    L0047,$02            ; SET B1
; B1 1 = LEARN ENABLED

BRA     LD320                ; EXIT

;
; CLR LEARN ENABLE FLG
;
LD31A: BCLR    L0047,$02      ; clr b1
; DISABLE LEARN

CLR     L00E1                ; CLEAR CELL 0 TMR

LD320: JSR     LD401          ; TO A/C CLUTCH ROUTINE

RTS     ; RETURN TO MAJOR LOOP EXE
*****

;-----
; FAN ROUTINE
; P/O SEGMENT 5 OF MAJOR LOOP EXE
;-----
LD324: BCLR    L003F,$18      ; CLR FAN 1 & 2 REQUEST FLGS
BRSET   L0031,$80,LD333     ; BR IF A/C CLUTCH IS ON
; ... else
LDAA    L0102                ; A/C OFF TIMER
BEQ     LD339                ;
; ... else
DECA    ; DEC A/C OFF TMR

BRA     LD336                ; GO STORE NEW TMR VALUE

LD333  LDAA    $8373          ; 45 SEC A/C CLUTCH OF

```

```

LD336 STAA L0102 ; ... TIME FOR DISABLE
; NEW A/C OFF TMR VAL

;-----
; COOL ERR ACTIVE TEST
;-----
LD339 BRCLR L0038, #02, LD342 ; BR IF NOT ERR 14 or 15

BSET L003F, #18 ; SET FAN 1 & 2 REQ FLAGS IF ERR
BRA LD398 ; BR TO TMR & OUTPUT ROUTINE

;-----
; HI A/C HEAD PRESS &
; Vss ENABLE TEST
;-----
LD342 LDAA L0102 ; A/C OFF TMR
BEQ LD376 ; BR IF = 0, (A/C OFF LONG ENOUGH)
; ... else
LDAA L0031 ; FMD 1 MODE WD
LDX #8019 ; AFR OPT WD 4

BRSET 0, X, #20, LD352 ; 0 = N.C. FAN REQ INPUT
; BR IF N.O. SW

LD352 EORA #40 ; INV HI HEAD PRESS BIT IF N.C.
BITA #40 ; b6, HEAD PRESS HI ?
BEQ LD362 ; BR IF HIGH
; ... else

;-----
;
;-----
LDAA L0081 ; Vss
CMPA $8374 ; 35 MPH FOR FAN 1 ENAB, A/C ON
BHI LD376 ; BR IF Vss TO HIGH FOR FAN 1 With A/C

BSET L003F, #08 ; SET FAN 1 REQUEST
BRA LD376 ;

LD362: BSET L003F, #08 ; SET FAN 1 REQ FLG IF PRESS HI &
; A/C OFF TIMER NON ZERO BITS
LDAA $8376 ; FAN 2 DISABLE, 55 MPH W/HI HEAD PRESS
BRSET L003F, #40, LD36F ; BR IF FAN 2 ON

LDAA $8375 ; FAN 2 50 MPH Vss THRESH
LD36F: CMPA L0081 ; Vss
BCS LD376 ; Vss LO ENOUGH FOR FAN 2 ON
; .... else
BSET L003F, #10 ; SET FAN 2 REQUEST

;-----
; FAN 1 HI COOL ENAB TEST
;-----
LD376: LDAA $8377 ; FAN 1, 103C DEG FOR DISABLE
BRSET L0037, #08, LD380 ; BR IF NOT b3, FAN 1 PID STPS DONE
; ... else

LDAA $8378 ; FAN 1, 106 Deg C FOR ENABLE
LD380: CMPA L005D ; FLT COOLANT TEMP
BCC LD387

BSET L003F, #8 ; SET DECEL ENLEAN ACTIVE BIT

;-----
; FAN 2 HI COOL ENAB TEST
;-----

```

```

LD387: LDAA    $8379          ; FAN 2, 106 Deg C FOR DISABLE
      BRSET   L003F,#$40,LD391 ; BR IF ALL FAN PID STP'S ADDED

      LDAA    $837A          ; FAN 2, 111 Deg C FOR ENABLE
LD391: CMPA    L005D          ; FLT COOLANT TEMP
      BCC     LD398

      BSET    L003F,#$10     ; SET FAN 2 REQ'ED BIT

;-----
; FAN TIMER & OLD FAN STATE
;-----
LD398 LDAA    L0101          ; FAN TMR VAL
      BRCLR   L003F,#$28,LD3A6 ; BR IF NOT ....
      ; AIR CONTROLLED, 0 = DIVERT
      ; DECEL ENLEAN ACTIVE
      ; ... else
      BRSET   L003F,#$28,LD3A6 ; BR IF ....
      ; AIR CONTROLLED, 0 = DIVERT
      ; DECEL ENLEAN ACTIVE
      ; ... else
      CLRA
      BRA     LD3AA          ; CLR FAN TMR

LD3A6 INCA
      BNE     LD3AA

LD3AA DECA
      STAA   L0101          ; NEW FAN TMR VAL

      BCLR    L003F,#$20     ; CLR OLD FAN 1, WAS ON bit
      BRCLR   L003F,#8,LD3B7 ; BR IF NOT DECEL ENLEAN ACTIVE
      ; ... else
      BSET    L003F,#$20     ; SET OLD FAN 1, WAS ON BIT

;-----
; FAN 1 OUTPUT FLAG SET UP
;-----
LD3B7 BRSET   L003F,#$08,LD3C6 ; BR IF FAN 1 REQ FLAG SET
      CMPA   $837C          ; 20 SEC, FAN TIME ON THRESH
      BCC    LD3CF          ; IF ON LONG ENOUGH TO TURN OFF
      ; ... else
      BRCLR   L0037,#$08,LD3CF ; BR IF NOT b3, FAN 1 PID STPS DONE
      ; FAN 1 NOW OFF
      ; ... else
      BRA     LD3DA          ; TURN FAN 1 ON

LD3C6 BRSET   L0037,#$08,LD3DA ; BR IF b3, FAN 1 PID STPS DONE
      ; {NO DLY IF FAN 1 ON NOW}
      ; ... else
      CMPA   $837B          ; 1 SEC FAN DELAY TMR
      BHI    LD3DA          ; IF OFF LONG ENOUGH TO TURN ON
      ; ... else

LD3CF CLR     $0102          ; NO FAN IN VENT OR HEAT REQUEST
      BCLR   L0037,#$08     ; CLR b3, FAN 1 PID STPS DONE FLG
      BCLR   L0039,#$40     ; CLR b6, FAN 1 OUTPUT REG FLG

      BRA     LD3DD          ; EXIT FAN 1 va FAN 2

LD3DA BSET    L0039,#$40     ; SET FAN 1 OUT REQUEST FLG

;-----
; FAN 2 OUTPUT FLAG SET UP
;-----

```



```

; A/C CLUTCH HI MPH DISABLE TEST
;-----
LDAA    L0081                ; Vss
CMPA    $8383                ; HIGH SPD DSABLE A/C, (255 MPH)
BCS     LD45F                ; BE IF Vss L.T. THRESH
; ... else
BRSET   L0043,#$20,LD448    ; IF HI MPH 1ST PASS DONE
; ... else
BSET    L0043,#$20          ; SET 1ST PASS HI MPH FLG

LDD     L0019                ; Eng run time
ADDD    $8384                ; 180 SEC'S A/C HI MPH TIME B4 DISABLE
STD     $01B9                ; SAVE A/C DISABLE TIME

LD448   LDX     L0019                ; Eng run time
        CPX     L01B9                ; HI MPH LONG ENOUGH ?
        BCS     LD45A                ; BR IF NOT LONG ENOUGH TO OFF A/C
; ... else
BSET    L0043,#$40          ; SET EGR DIAG IN WORK BIT
; (IF LONG ENOUGH SET FLG)

LDAA    $8386                ; 30 SEC'S A/C HI MPH TIME B4 RE-ENABLE
LD458   STAA   L01BB                ; A/C HI MPH RE ENGAGE TIMER
        BRA     LD483                ; GO DISABLE A/C CLUTCH

LD45A   BCLR   L0043,#$40          ; CLR A/C DISABLED BY HI MPH BIT
        BRA     LD470                ; TO CK HI RPM DISABLE

;-----
; HI MPH DISABLE TO RE-ENABLE TEST
;-----
LD45F   BRCLR  L0043,#$40,LD46D    ; BR IF A/C NOT DISABLED BY HI MPH
LDAA    $01BB                ; LO MPH RE-ENABLE TMR
BEQ     LD46D                ; IF TIME UP GO
; ... else
DEC     $01BB                ; DEC TMR
BRA     LD483                ; OFF A/C CLUTCH

LD46D   BCLR   L0043,#$20          ; CLR 1ST PASS WITH HI MPH BIT

;-----
; HI RPM ENABLE PREVENTION TEST
;-----
LD470:  BRCLR  L0037,#$20,LD47B    ; BR IF NOT b5, 1 = A/C OFF
; {BY HI RPM DISABLE IF A/C ON}
; ... else
LDAA    L0085                ; RPM/25
CMPA    $8387                ; 5K RPM FOR NO A/C CLUTCH ENGAGE
BCC     LD483                ; BR TO DIABLE A/C IF RPM G.T. THRESH

;-----
; ENABLE A/C CLUTCH
;-----
LD47B:  BCLR   L0043,#$40          ; CLR A/C HI MPH DISABLE BIT
        BCLR   L0037,#$20          ; CLR b5, 1 = A/C OFF
; {TURN ON A/C CLUTCH ENABLE}

BRA     LD486                ; TO A/C LRN

;-----
; DISNABLE A/C CLUTCH
;-----
LD483:  BSET   L0037,#$20          ; SET b5, 1 = A/C OFF
; {TURN OFF A/C CLUTCH}

```

```

;-----
; A/C LEARN ROUTINE
;-----
LD486: BRSET    L0031, #80, LD496      ; BR IF A/C CLUTCH ON
                                           ; ... else
      BCLR     L00FD, #1              ; CLR MOTOR RESET STARTED BIT
                                           ; DISABLE A/C LRN

      BRSET    L00FD, #80, LD501      ; BR IF A/C ON
                                           ; ... else
      BSET     L00FD, #20            ; SET 'A/C CNT'ED' FLAG

      BRA      LD4B5                  ; EXIT

LD496: BCLR     L00FD, #20            ; CLR PID CMD'ED STP'S HIGHER BIT
      BSET     L00FD, #40            ; SET START UP DELAY IN WORK BIT

      BRCLR   L00FD, #81, LD501      ; BR IF A/C OFF
                                           ; ... else
      LDAA    L0116                  ; DEAD BAND TMR
      CMPA   $838C                  ; .04 Msec MIN RPM ERR
      BCS    LD501                  ; BR IF NOT IN DB LONG ENOUGH
                                           ; .... else

;-----
; TOGGLE LRN FLAG
;-----
      LDAA    L00FD                  ; AC MODE  WD
      EORA   #1                      ;
      STAA   L00FD

      BPL    LD4BA                    ; BR IF A/C ON, (LEARNED)
                                           ; ... else
      LDAA   L002B                  ; IAC CURRENT POSITION
      STAA   $0118                  ; SAVE BEGINNING IAC POSIT
LD4B5: CLR     L0116                  ; RESET DEADBAND CNT'R

      BRA    LD501                    ; EXIT

LD4BA: LDAA    L002B                  ; IAC CURRENT POSITION
      SUBA   $0118                  ; COMP LEARNED VALUE FOR ANTICIPATE
      BCC   LD4C2                    ; BR IF LRANED VAL E.Q. or G.T. 0
                                           ; ... else
      CLRA                                ; CLR IF NEGITIVE

LD4C2: BRCLR   L00FE, #80, LD4CB      ; BR IF NOT b7, PK/NEUT, 1 = NEU
                                           ; ... else
      LDAB   $838D                  ; (TF), P/N GAIN, A/C LRN, (1)
      MUL                                ; SCALE NEUT STEPS TO DRIVE EQUIV
      ASLD

LD4CB: LDAB    $8389                  ; MAX A/C LRN, 32 STEPS
      CBA                                ; PRESENT vs MAX LMT
      BLS   LD4D2                    ; BR IF PRESENT L.T. MAX
                                           ; ... else
                                           ; LIMIT TO MAX VAL
LD4D2: LDAB    $838A                  ; A STEP MIN A/C LRN STEPS
      CBA                                ; PRESENT vs MIN VALUE
      BHI   LD4D9                    ; BR IF G.T. LIMIT
                                           ; ... else
      TBA

LD4D9: TAB                                ; LIMIT TO MIN VAL
      ;
      ; CALC  DIFF CHANGE IN LEARN
      ;

```

```

SUBA    L002C                ; IDLE SPD A/C ANTICIPATE
BCS     LD4E9

; .... else
CMPA    $838B              ; 5 STEPS MAX+ A/C LEARN EACH
; ON/OFF CYCLE
BLS     LD4E9
;
; ... else
LDAB    $838B              ; 5 STPS, MAX+A/C LEN STPS
; PER ON/OFF CYC777
TBA     ; SAVE TF DIFF COMP
ADDB    L002C              ; IDLE SPD A/C ANTICIPATE
; ADD TO PRIOR LEARNED VALUE

LD4E9: STAB    L002C                ; IDLE SPD A/C ANTICIPATE
; REP OLD A/C COMP WITH NEW

BRCLR   L00FE, #80, LD4FB    ; BR IF NOT b7, PK/NEUT, 1 = NEUT
; ... else
LDAB    $865F              ; PK/Neut TF SCALAR
TSTA    ; CHANGE NEGITIVE ?
BPL     LD4FA              ; BR IF NOT NEG
; ... else
NEGA    ; ABS VALUE OF CHANGE
MUL     ; SCALE POS CHANGE
NEGA    ; RESTORE SIGN

BRA     LD4FB              ;

LD4FA:  MUL              ; SCALE DN POS CHANGE

LD4FB:  ADDA    L0115        ; ADJ THROT FOLOWER DIF
STAA    L0115              ; SAVE ADJ THROT FOLOWER DIF

LD501:  BSET    L00FD, #80    ; A/C STATE OFF

BRCLR   L0031, #80, LD50B    ; BR IF CLUTCH OFF
; ... else
BCLR    L00FD, #80          ; SET NEW A/C STATE ON

LD50B:  RTS              ; RETURN TO MJR LP, SEGMENT 5
*****

*****
* LOOK UP COOLANT VAR ROUTINE
*
* >>> SEGMENT 7 OF MAJOR LOOP EXE <<<<<
*
*****

LD50C:  LDAA    L005D        ; FLT COOLANT TEMP
CMPA    #224              ; 208c
BLS     LD514              ;
; ... else
LDAA    #224              ; DEFAULT VAL, 208C

;-----
; GET ASYNC FACTOR
;-----

LD514:  LDX     #$8554        ; ASYNC FACTOR, (COOLANT) TABLE
JSR     LE3D0              ; 2D LOOK UP, NO OFF SET

STAA    $009F              ; COOL FACTOR FOR ASYNC FUEL
;-----

```

```

;-----
; CK FOR VALID LIMITS
;-----
LDAA    L005D                ; FLT COOLANT TEMP
CMPA    #208                 ; 116 Deg C
BLS     LD524

LDAA    #208                 ; ... else
;-----

;-----
; LK UP R/L OFFSET vs COOLANT
; 14 LINES
;
; TBL = COUNTS
;-----
LD524: LDX     #$84F6         ; R/L OFFSET Vs COOL TBL
      JSR     LE3D0          ; 2D LOOK UP, NO OFF SET
      ;
      STAA   L00F6          ; COOL LEAN OFF SET FOR C/L
;-----

;-----
; LK UP PWR ENRICH F/A %CHG vs COOLANT
; *** WOT ***
;
; TBL = (LAMBDA * 256) + 128
;-----
LDAA    L005D                ; FLT COOLANT TEMP
LDX     #$860D              ; PWR ENRICH FUEL/AIR %CHG vs COOL TBL
JSR     LE3C5               ; 2D LK UP W/LINE COUNT
;
STAA    L00F5               ; PWR ENRICH FUEL/AIR %CHG vs COOL RESULT
;
BRSET   L00FE,#$08,LD54B    ; BR IF b3, START UP DELAY OVER
; ... else

;
; CK AFR OPT WD 3, 0101 0010
; b0, 1 = COOL FOR IDLE SPD TIME OUT
;
LDAA    $8018                ; AFR OPT WD 3
RORA    ;
BCC     LD54B                ; DON'T DO LK UP BASED OM MAT TBL
; ... else

;-----
; IDLE SPEED TIME OUT IAC STEPS vs COOLANT or MAT
;
;
; TBL = STEPS
;-----
LDAA    L005D                ; FLT COOLANT TEMP
LDX     #$86A1              ; INDEX IDLE SPD TIME OUT STPS vs COOL TBL
JSR     LE3D0               ; 2D LOOK UP, NO OFF SET
;
STAA    L01B1               ; IDLE SPD TIME OUT STEPS
;
LD54B: BRCLR   L003C,#$02,LD55B ; BR IF ENG START TME NOT ENABLED
; ... else
BRSET   L0036,#$35,LD55B    ; BR IF ALL SCREWED UP
; ... else
INC     L045E                ; TIME FM 1st DRP TILL ENG RUN
BNE     LD55B                ; EXIT IF NO OVERFLOW
; .... else
DEC     L045E                ; TIME FM 1st DRP TILL ENG RUN
;

```



```

LD55B:  RTS                                ; RETURN TO MAJOR LOOP EXE
*****

*****
* KNOCK, LOW OCTAINE SPK MOD LOGIC
*
* >>> SEGMENT 8 OF MAJOR LOOP EXE <<<<<
*
*****

LD55C:  LDAA      $0000                    ; MINOR LP CNTR
        CMPA      #$08                    ; CK FOR 1 SECOND
        BNE       LD5C5                    ; EXIT IF L.T. 1 SECOND
                                           ; .... else

;-----
; LOW OCTAINE ENABLE/DISABLE TEST
;
;-----

;-----
; KNOCK SENSOR FAIL TEST
;-----

LDAB      L01BF                            ; LO OCTANE RETARD KNK CNT'R
;
BRSET     $003C, #$20, LD5AE                ; BR IF Err 43A
; ... else
BRSET     L0002, #$80, LD5AE                ; BR IF EST Err 43B
; ... else
LDAA      $0103                            ; A/D BATTERY VOLTAGE
CMPA      #93                             ; 9.3 VDC
BCS       LD5AE                            ; BR IF Vbat L.T. 9.3 Volts
; .... else

;-----
; ABS MAP ENABLE TEST
;-----

LDAA      L0071                            ; NORMILIZED LOAD MAP VAL
CMPA      $8229                            ; LO MAP THRESH FOR LO OCT MOD, (70 Kpa)
BCS       LD5C2                            ; IF MAP TOO LOW, CLR FLG'S
; .... else

;-----
; DIFF MAP ENABLE TEST
;-----

SUBA      $0073                            ; 25 msec DIFF MAP
BCS       LD584                            ; BR IF DIFF MAP L.T. 0
; .... else
CMPA      $822A                            ; DIFF MAP RHRESH, (0.6 Kpa)
BHI       LD5C2                            ; IF DIFF MAP TO HIGH, CLR FLG'S

;-----
; LO OCTANE SPK MOD ENABLED
; CK FOR HI KNK OR NO KNK
;-----

LD584:  LDAA      L00C4                    ; KNK RETARD
        BEQ       LD5A2                    ; BR IF NO KNOCK
; ... else
CMPA      $822B                            ; KNK RETARD THRESH, (3.8 Deg)
BLS       LD5C2                            ; IF LO KNOCK, CLR FLG'S

;-----
; HIGH KNOCK, INCR RETARD
;
;-----

BRSET     L003A, #1, LD599                  ; BR IF HI KNK LAST PASS
; ... else
BSET      L003A, #$01                       ; SET HI KNOCK FLAG

```

```

BCLR    L003A,#$02          ; CLR NO KNK FLAG
;
BRA     LD5B4                ; DO SCALING TO ACTIVITY CNT'R

LD599:  ADDB    $822C        ; RETARD INCR VALUE, (18 DEG)
BCC     LD5B4                ; IF NO OVERFLOW
; ... else
LDAB    #255                ; FORCE MAX LIMIT
;
BRA     LD5B4                ; DO SCALING TO ACTIVITY CNT'R

;-----
; NO KNOCK, DECREASE RETARD
;
;-----
LD5A2:  BRSET   L003A,#2,LD5AE ; BR IF NO KNK PRIOR PASS
; ... else
BSET    L003A,#2            ; SET NO KNK FLAG
BCLR    L003A,#1            ; CLR KNK ACTIVE FLG
;
BRA     LD5B4                ; GO SAVED ACTIVITY CNTR

LD5AE:  SUBB    $822D        ; DEC LO OCT RETARD, (2 CTS)
BCC     LD5B4                ;
; ... else
CLR     CLR     LD5B4        ;

;-----
; USE LO OCT KNK COUNTER TO BASE
; SPK MODIFICATION
;-----
LD5B4:  STAB    L01BF         ; LO OCTANE RETARD KNK CNT'R
LDAA    $822E                ; DEG S.A.RETARD, (5.8 DEG)
MUL     ; KNK Cnt * $822E VAL
ADCA    #0                   ; ROUND
STAA    L01C0                ; LO OCTANE RETARD MOD'ER, TEMPORARY

BRA     LD5C5                ; EXIT

;-----
; LO OCT SA RETARD MOD DISABLED
; CLR FLAGS
;-----
LD5C2:  BCLR    L003A,#3      ; CLR HI & NO KNOCK FLAGS

LD5C5:  RTS                ; RETURN TO MAJOR LOOP EXE
*****

*****
* AIR MANAGMENT ROUTINE
*
* >>> SEGMENT 9 OF MAJOR LOOP EXE <<<
*
* AIR PUMP CNT'ED WITH 3 WAY VALVE, AIR IS DIRECTED
* TO PORTS, CAT or DIVERTED BACK TO PUMP.
*****
LD5C6:  LDX     #$838E        ; INDEX OF CALIB'S

;
; IF ENG IN CLSD LP, SET FLG FOR ENG IN CLS LP
; SINCE RESTART
;
LDAB    L0047                ; AFR MD WD

```

```

ANDB    #$80                ; CLR b7. 1 = CLOSED LOOP
ORAB    L003E               ; AIR MODE WD
STAB    L003E               ; AIR MODE WD

;-----
; CK COOLANT FOR DIVERT
;-----
LDAA    L005D               ; FLT COOLANT TEMP
CMPA    $0C,X               ; COOLANT THRESH
BCS     LD5DD               ; DIVERT AIR
; .... else
LDAA    L003C               ; CL FOR ERR'S
BITA    #$04                ; b2
BEQ     LD5E0
; ... else

LD5DD:  JMP     LD6B5

;-----
;CLS LP 1ST TIME TEST
;-----
LD5E0:  LDAA    L0047         ; AFR MD WD
BPL     LD5EE               ;
; ... else
CLR     $010C               ; CLE DECEL TMR
LDAA    $0D,X               ;
STAA    $010D               ;
;
BRA     LD618               ; SKIP DE-CELL ROUTINE

LD5EE:  TSTB
BPL     LD5FA               ; TO AIR DECEL ROUTINE
; ... else
LDAA    L010D               ; GET OLD TMR
BEQ     LD5DD               ; BR IF TMR = 0
; ... else
DECA
STAA    L010D               ; SAVE NEW TMR VAL

;-----
; AIR DECEL ROUTINE
;-----
LD5FA:  BRCLR   L0037,$$80,LD618 ; BR IF NOT b7, 1 = ENG RUNNING
; ... else
LDAB    L010C               ; DECEL TMR
LDAA    L010B               ; OLD MAP
SUBA    L0079               ; MAP for Spd/Dens calc
BLS     LD60E               ; BR IF NEW MAP LT OLD MAP

;
CMPA    $0B,X               ;
BLS     LD60E               ;

LD60E:  LDAB    7,X          ;
TSTB
BEQ     LD618               ; CK DECEL TMR
; ... else
DECB
STAB    $010C               ; DEC TMR EVERY 100 MSEC
; DECEL TMR

JMP     LD6A6

;-----
; FUEL ENRICH MODE
;-----
LD618:  LDAB    L0057         ; RPM/25
CMPB    6,X                 ;

```

```

BLS      LD62A      ; BR IF
;
LDAB     L0106     ; HI RPM DIVERT TMR
CMPB     1,X       ;
BHI      LD666     ; DIVERT, IN HI RPM TOO LONG
;
INCB     LD62B     ; INC HI RPM TMR
BNE      LD62E     ;
; .... else
BRA      LD62E     ;

LD62A:   CLR      ;
LD62B:   STAB     L0106 ; SAVE NEW HI RPM DIVERT TMR
;
LD62E:   BRCLR    L0046, #20, LD63F ; BR IF NOT b5
; B5 1 = IN PWR ENRICH
; ... else
LDAB     L010E     ; GET PORT AIR TMR
CMPB     0,X       ;
BEQ      LD6B5     ; DIVERT IF IN P.E. TOO LONG
; ... else
INCB     LD6A0     ; INCR PORT AIR TIMER
STAB     L010E     ; SAVE NEW PORT AIR TMR
;
BRA      LD6A0     ;

LD63F   CLR      L010E ; PORT AIR TMR
BRSET    L0039, #80, LD6B5 ; BR IF IN HIWAY FUEL
; ... else
;-----
; Rich O2 mode
;-----
LDX      L0107     ; RICH o2 TMR
LDAB     L0092     ; A/D Flt o2
CMPB     $8390     ; IF FILT o2 GT 782 mv FOR
; TIME = LB392, DIVERT
BLS      LD658     ;
; ... else
CPX      $8392     ; 0d
BCC      LD663     ;
; ... else
INX
BRA      LD65E

LD658   LDX      L0107 ; RICH o2 TMR
BEQ      LD668     ;
; ... else
;
LD65E   DEX      ; DECR TMR
STX      L0107     ; SAVE RICH o2 TMR
BRA      LD66B

LD663   BSET     L0042, #20 ; SET Err 44 or 45 BIT
;
LD666   BRA      LD6B5 ;

LD668   BCLR     L0042, #20 ; CLR Err 44 or 45 BIT

```

```

;-----
; Lean O2 mode
;-----
LD66B  BRSET    L0042,#$20,LD6B5      ; BR IF Err 44 or 45 Bit = 1
;
CMPB    $8391      ; IF FILT o2 LT 0.248V for
; TIME = LB392, DIVERT
BCC     LD678
;
LDAA    L0047      ; AFR MD WD
BMI     LD67D
;
; ... else

LD678  LDX     #0
;
BRA     LD686
;

LD67D  LDX     L0109      ; R/L DIVERT TIMER
CPX     $8392      ; 20 SEC R/L DIVERT TIMER
BHI     LD6B5
;
;
LD686  INX
STX     L0109      ; INCR R/L DIVERT TIMER
; SAVE NEW R/L DIVERT TIMER

;-----
; Low Load mode
;-----
LDAA    L0079      ; MAP for Spd/Dens calc
CMPA    $8396      ; 20 Kpa MAP DIVERT IF MAP LT
BCS     LD6B5
;
; .... else
CMPA    $8397      ; 60 Kpa, DIVERT AIR IF MAP GT
BLS     LD69C
;
; ... else
LDAA    L0081
CMPA    $8398      ; 60 MPH, DIVERT AIR IF MPH GT 60 Kpa
; & MAP L8397
BHI     LD6B5
; DIVERT

;-----
; OPN/CLS LOOP
;-----
LD69C  LDAB    L003E      ; AIR MODE WD
;
BMI     LD6A6
;
;-----
; PORTS
;-----
LD6A0  LDAB    L003F      ; MAJOR MD WD
ORAB    #$06
; SET
; B1 1 = AIR CONTROLLED
; B2 1 = AIR TO PORT
BRA     LD6B9
; EXIT

;-----
; CAT
; Air when on directs AIR to CAT
;-----
LD6A6  LDAB    L003F      ; MAJOR LP MD WD
;
; CK AFR OPT WD 2, 0000 0100

```

```

;      b2, 1 = 4TH GR SW N.O.
;
LDAA    $8017          ; AFR OPT WD 2
BITA    #$04          ; b2
BEQ     LD6B5         ; BR IF NOT b2
; ... else
ANDB    #$FB          ; CLR b2, AIR TO PORT BIT
ORAB    #$02          ; SET b1, AIR CONTROLLED BIT

BRA     LD6B9         ; EXIT

;-----
; DIVERT
; (When air cnt'l sol de-ener air is diverted)
;-----
LD6B5   LDAB    L003F          ; MAJOR MD WD
        ANDB    #$F9          ; CLR 1111 1001
;      b1 1 = AIR CONTROLLED, 0 = DIVERT
;      b2 1 = AIR TO PORT
LD6B9   STAB    L003F          ; MAJOR MODE WD
;
        LDAA    L0079          ; MAP FOR SPD/DENS CALC
        STAA    L010B          ; SAVE FOR NEXT LOOP
;
        RTS                ; RET TO MAJOR LOOP EXE
*****

*****
*   KNOCK, MAT IAC VAR'S, ETC...
*
* >>> SEGMENT A (10), OF MAJOR LOOP EXE <<<
*
*****
LD6C1:  LDAB    L0000          ; NINOR LP CNTR
        BITB    #$10          ; b4
        BNE    LD6E2         ; BR IF b4
; ... else

;-----
; LK UP KNOCK PCT RECOVERY RATE vs RPM
;
; TBL = (DEG/MSEC)/.0225
;-----
        LDAA    L0057          ; RPM/25
        LSRA          ; SPACE FOR LK UP
        LSRA          ;
;
        LDX    #$8213          ; INDEX KNOCK PCT RECOVERY RATE TBL
        JSR    LE3D0          ; 2d LOOK UP, NO OFF SET

        LDAB    L00C4          ; KNOCK RETARD
        MUL          ; APPLY RECOVERY RATE
        ADCA    #0            ; ROUND
        NEGA          ; INVERT
        BNE    LD6DB         ; IF NO MATH ERROR
; .... else
        LDAA    #255          ; FORCE MAX VALUE
;
LD6DB:  ADDA    L00C4          ; KNOCK RETARD RECOVERY
        BCS    LD6E0         ; BR IF OVERFLOW
; .... else
        CLRA          ;
LD6E0:  STAA    L00C4          ; KNOCK RETARD

LD6E2:  LDAA    $824B          ; Error WD 3 MASK

```

```

BITA    #$10          ; b4 = 1 Err 43, KNOCK
BEQ     LD74C         ; BR IF NOT b4
                    ; ... else
LDAB    L0000        ; NINOR LP CNTR
CMPB    #26          ;
BNE     LD726        ;
                    ; .... else
BRSET   L0002,#$40,LD726 ; BR IF Err 43B SPK TEST DONE
                    ; THIS START UP
                    ; ... else
BRSET   L0002,#1,LD718 ; BR IF Err 43B SPK TST IN WORK
                    ; ... else
BRSET   L0036,#$10,LD726 ; BR IF IGN OFF
                    ; ... else
BRCLR   L0037,#$80,LD726 ; BR IF NOT b7, 1 = ENGINE RUNNING
                    ; ... else
LDAB    L0079        ; MAP for Spd/Dens calc
CMPB    $829D       ; If MAP LT 75 THEN DISABLE Err 43B
BCS     LD726        ;
                    ; .... else
LDAB    L013B        ; SPK ADV BLEND MULT
INCB    LD726        ; INCR SPK ADV BLEND MULT
BNE     LD726        ; ... else
                    ; .... else
LDAB    L005D        ; FILT COOLANT TEMP
CMPB    $829C       ; 95c, Min cool for for Err 43B
BLS     LD726        ;
                    ; ... else
BSET    L0002,#1    ; SET Err 43B SPK TEST IN WORK
BRA     LD726        ;

LD718:  LDAB    L01A7    ; SPK FOR ERR 43B TST
        CMPB    $829E    ; 24.9 Deg MAX S.A. for ERR 43B TEST
        BNE     LD726    ; BR IF SPK NOT 24.9 Deg
                    ; .... else
        BCLR   L0002,#1  ; CLR Err 43B EST TEST IN WORK
        BSET   L0002,#$C0 ; SET b6 & b7
                    ; Err 43B SPK TEST DONE THIS START UP
                    ; Err 43B SPK TEST FAILED THIS START UP
;-----
;
;-----
; CK FOR KNOCK ERROR ERR 44
;
; CK SENSOR INTERNAL RESISITOR
; SENSOR WINDOW 1.44 --> 3.62 VDC
;-----
LD726:  LDAA    #03      ; SEL A/D CH
        JSR     LE2D4    ; MUX READ
                    ;
        CMPA    $8299    ; 3.62V UPPER VOTLAGE THRESH FOR Err 44
        BHI     LD735    ; CK Err TIMER
                    ;
        CMPA    $829A    ; 1.44V LOWER VOTLAGE THRESH FOR Err 44
        BCC     LD746    ; EXIT via CLR Err
                    ;
LD735:  LDAA    L00C3    ; KNK SENS VDC CK TMR
        CMPA    $8298    ; 10 Sec TIME REQUIRED
        BHI     LD741    ; BR IF GT 10 SEC'S
                    ; ... else
        INC     L00C3    ; BUMP UP KNOCK Err TMR
                    ;
        BRA     LD74C

```

```

LD741  BSET    L003C,#$20      ; SET KNOCK ERROR BIT
      BRA     LD74C          ; EXIT

      ;
      ; CLEAR ERROR
      ;
LD746  BCLR    L003C,#$20      ; CLR b5, KNOCK ERROR BIT
      CLR     L00C3          ; CLR KNOCK Err TMR
;-----

;-----
; READ MAT SENSOR AND CK ERROR'S
;
;
;
;-----

LD74C: LDAA    #$80           ; Set A/D ch FOR MAT, (Ch 8)
      JSR     LF0D2          ; Go read A/D
      ;
      COMA           ; INVERT RESULT
      STAA    L013C         ; MAT A/D RESULT
      STAA    L0061         ; MAT A/D VAL

      ;-----
      ; CK ERR 23, MAT LO
      ;-----

      LDX     L0019         ; Eng run time
      CPX     $8268         ; 240 SEC, ENG RUN TIME <= DISABLE 23/25
      BLS     LD77B         ; BR IF RUN TIME LT 240 SEC'S
      ; ... else
      LDAB    L0081         ; Vss/1
      ;
      LDAA    L0061         ; A/D MAT VALUE
      CMPA    $8266         ; IF MAT < 2 ENABLE ERR 23
      BCC     LD778         ; BR IF
      ; ... else
      LDAA    L00CB         ; GET MAT ERR QUAL TMR
      CMPA    $8267         ; 12 SEC QUALIFICATION TIME
      BHI     LD780         ; BR IF ERR GT 12 SEC
      ; ... else
      CMPB    $826A         ; 1 MPH, IF > DO ERR 25, else 23
      BHI     LD77B         ; BR IF Vss GT 1 MPH
      ; ... else
      INC     L00CB         ; INCR QUALIFICATION TMR
      ;
      BRA     LD783         ;

      ;
      ; CLR Err 23
      ;
LD778  BCLR    L0046,#$10      ; CLR b4, 1 = Err 23 THIS PASS, (MAT)
LD77B  CLR     L00CB          ; CLR MAT ERR QUAL TIMER
      ;
      BRA     LD783         ;

      ;
      ; SET Err 23
      ;
LD780  BSET    L0046,#$10      ; SET b4, 1 = Err 23 THIS PASS, (MAT)
      ;

```



```

;
; CK Err 25 QUAL'S
; 12 SEC, 1 MPH
;
LD783 LDAA    L0061      ; A/D MAT VALUE
      CMPA    $8272     ; 139 Deg c, IF MAT > ENABLE 25
      BCS    LD79B     ; BR IF MAT LT 129c
      ; .... else
      LDAA    L00CC     ; ERR 25, HI MAP TMR,
      CMPA    $8273     ; 12 SEC, MAT Err QUAL TIME
      BHI    LD7A3     ;
      ; ... else
      CMPB    $826A     ; 1 MPH, IF > DO 25, ELSE Err 23
      BLS    LD79E     ;
      ; ... else
      INC     L00CC     ; INCR HI MAP TMR,
      ;
      BRA    LD7A6     ;

;
; CLR Err 25
;
LD79B BCLR    L0041,#$02 ; CLR Err 25 THIS PASS BIT
LD79E CLR     L00CC     ; ERR 25, HI MAP TMR,
      ;
      BRA    LD7A6     ;

;
; SET Err 25
;
LD7A3 BSET    L0041,#$02 ; SET Err 25 THIS PASS BIT
LD7A6 BRSET   L0046,#$10,LD7B3 ; BR IF b4, 1 = Err 23 THIS PASS, (MAT)
      ; ... else
      BRCLR  L0041,#2,LD7BB ; BR IF NOT Err 25 THIS PASS
      ; ... else
      BSET   L004E,#$40     ; SET Err 33, MAP HIGH BIT
      ;
      BRA    LD7B6     ; EXIT W/DEFAULT MAT VAL

LD7B3: BSET    L004D,#1   ; SET Err 23, MAT LOW
      ;
LD7B6: LDAA    $826B     ; DEG C MAT DEFAULT IF Err 25, (29d)
      STAA   L0061     ; A/D MAT VALUE

;-----
; LINEARIZE A/D MAT, MAT TABLE FOR 1K PU
; 17 LINE TBL
;-----
LD7BB: LDAA    L0061      ; A/D MAT VALUE
      LDX    #$F12B     ; INDEX MAT LINEARITY TABLE
      JSR    LE3D0     ; 2d LOOK UP, NO OFF SET
      ;
      STAA   L0063     ; INV MAT VALUE
      ;
      ;
      BRSET  L00FE,#$08,LD7DA ; BR IF b3, START UP DELAY OVER
      ; (PID Md Wd #2)
      ; ... else

;
; CK AFR OPT WD 3, 0101 0010
; b0, 1 = COOL FOR IDLE SPD TIME OUT
;
LDAA    $8018      ; AFR OPT WD 3

```

```

RORA                                ;
BCS      LD7DA                       ; BR IF
                                           ; .... else
;-----
; IDLE SPEED TIME OUT IAC STEPS vs COOLANT or MAT
;
;
; TBL = STEPS
;-----
LDAA     L0063                        ; INV MAT
LDX      #$86A1                       ; INDEX IDLE SPEED TIME OUT STEPS
JSR      LE3D0                        ; 2D LOOK UP, NO OFF SET

STAA     L01B1                        ; IDLE SPD TIME OUT STEPS
;
LD7DA:   LDAA     L0063                ; INV MAT

;
; CK AFR OPT WD 3, 0101 0010
;   b4, 1 = OPT INV MAT Lk Up
;
LDX      #$8018                       ; AFR OPT WD 3
BRCLR    0,X,$10,LD7FC               ; BR IF NOT b4

;-----
; LK UP (COOLANT - MAT) DELTA MULT
; FOR INV TERM L.U. vs AIR FLOW
;
; TBL = MULT * 128
;-----
LDAA     L006B                        ; gms/sec, LIMITED AIR FLOW FM IDEAL GAS LAW
LDX      #$880E                       ; (COOLANT - MAT) DELTA MULT
                                           ; FOR INV TERM L.U. vs AIR FLOW
JSR      LE3D0                        ; 2D LOOK UP, NO OFF SET

LDAB     L005D                        ; FLT COOLANT TEMP
SUBB     L0063                        ; INV MAT
BCC      LD7F2                        ;
                                           ; .... else
LD7F2:   CLRB
MUL      ; APPLY MULT
ASLD
BCS      LD7FA                        ; IF NO OVERFLOW
                                           ; ... else
ADDA     L0063                        ; INV MAT
BCC      LD7FC                        ;
                                           ; ... else
LD7FA:   LDAA     #255                 ; FORCE MAX VAL

;-----
; MAT COMP Vs MAT
;
; TABLE is MAT COMP COUNTS
;-----
LD7FC:   LDX      #$87FD               ; MAT COMP VS MAT TBL (^ .5)
JSR      LE3D0                        ; 2d LOOK UP, NO OFF SET
;
LDAB     #128                          ;
MUL      ; * 128
ADDD     #$7480                        ; MAT = ((MAT^.5)+233)*128
PSHB
PSHA
;
;
PULX
LDD      $86DD                         ; ((MAT^.5)+233)*128, (Denom)
FDIV    ; CYL VOL & UNIT CONV, 159d, (LITRE)
;

```

```

STX      L006D                                ; SAVE INV MAT, CYL VOL & SCALING TERM
                                                ; CYL VOL/((MAT^.5)+233)*128
                                                ;
RTS      ; RETURN TO MAJOR LOOP EXE
*****

*****
*   EGR ROUTINE of MAJOR LOOP
*
*   >>> SEGMENT B (11), OF MAJOR LOOP EXE <<<
*
*****
LD812:   BRSET   L003B,#$01,LD864              ; BR IF b0, IN PK/NEUT
                                                ; (CPU INPUT STATUS WD)
                                                ; CLR EGR & EXIT
                                                ; ... else
LDAA     L0061                                ; A/D MAT VALUE
CMPA     $82B0                                ; IF MAT < 30 DEG C DISABLE EGR
BCS      LD864                                ; CLR EGR & EXIT
                                                ; .... else

;
; L82AE, ENABLE EGR IF TPS > 3.25%, LOWER HYST
; L82AF, ENABLE EGR IF TPS > 3.90% UPPER HYST
;
LDD      $82AE                                ; LOAD EGR ENABLE %TPS PAIR
;
TST      L0123                                ; EGR LK'ED UP D.C.
BNE      LD826                                ; BR IF D.C. NZ
; .... else
LD826:   TBA      ; %TPS THRESH TO A Reg
CMPA     L0095                                ; TPS Ld val
BHI      LD864                                ; BR IF %TPS LD VAL LT THRESH
                                                ; (CLR EGR & EXIT)
; ... else
BRSET    L0046,#$20,LD864                    ; BR IF b5, 1 = IN PWR ENRICH
                                                ; (CLR EGR & EXIT)
                                                ; .... else

;-----
; EGR DC Vs PRM Vs VAC LOAD TABLE
;
;
; TBL = %EGR * 2.56
;-----
LDX      #$82B2                                ; INDEX EGR DC TABLE, (3D)
;
LDAB     L007F                                ; Current manifold vacuum
LSRB     ; VAC/2
;
LDAA     L0085                                ; RPM/25
CMPA     #80                                  ; CK 2000 RPM UPPER LIMIT
BLS      LD83C                                ; IF L.T. 2K RPM
; .... else
LDAA     #80                                  ; FORCE 2000 RPM (MAX TBL LIMIT)
;
LD83C:   JSR      LE353                        ; 3d LK UP
;
STAA     L0123                                ; SAVE LK'ED UP EGR D.C.

;-----
; EGR D.C MULT (0 - 2) vs COOLANT
; tbl = 7 lines
;
; TBL = MULT * 128

```

```

;-----
LDAA L005D ; FLT COOLANT TEMP
CMPA #160 ; 80c, CK IF MAX VAL
BLS LD84A ; BR IF COOL LT 80c
; ... else
LDAA #160 ; FORCE 80c, MAX VAL

LD84A: LDAB #64 ; 8 Deg c MIN FOR LK UP
LDX #$82D9 ; INDEX EGR D.C MULT
JSR LE3CC ; 2d LK UP
;
LDAB L0123 ; EGR LK'ED UP D.C.
MUL ; APPLY MULT
ASLD ;
BCC LD85B ;
; ... else
LDAA #255 ; FORCE MAX VALUE

;-----
; CK HEADS UP FOR EGR SLEW
;
;-----
LD85B: LDAB L0036 ; CK HEADS UP FLAG
BPL LD865 ; BR IF NO HEADS UP ON LINE
; ... else
JSR L580C ; TO HEADS UP CODE
;
BRA LD865 ; GO SAVE HEADS UP VALUE

;
; CLR EGR & EXIT
;
LD864 CLRA ;
;
LD865 STAA L0123 ; LK'ED UP EGR D.C.
BEQ LD879 ; BR IF Z
; ... else
LDAA L0124 ; EGR D.C.
ADDA $82B1 ; INC EGR ?? BY 16 EVERY 100 MSEC
BCC LD874 ; IF NO OVERFLOW
; ... else
LDAA #255 ; USE MAX VALUE
LD874: STAA L0124 ; EGR D.C.
;
BRA LD87C ; EXIT VIA RTS

LD879: CLR L0124 ; ZERO EGR D.C.
;
LD87C: RTS ; RETURN TO MAJOR LOOP EXE
*****

*****
* CAN PURGE, (CCP) ROUTINE
*
* >>> SEGMENT C (12), OF MAJOR LOOP EXE <<<
*
*****
LD87D: LDAA L003C ; CCP MODE WORD
LSRD ;
;
LDAA L003F ; MAJOR MODE WD
RORA ;

```

```

ASLD                                ;
STAA    L003F                        ; MAJOR MD WD
;
BRSET   L0056,#$02,LD8D9            ; BR IF DECEL FUEL C/O ENABLED
; ... else
BRSET   L0036,#$10,LD8D3            ; BR IF IGN OFF
; ... else
LDAA    L005D                        ; GET FLT COOLANT TEMP
CMPA    $82EA                        ; 60 Deg c, PURGE COOL ENAB THRESH
BLS     LD8D9                        ;
; ... else
LDX     #$82EB                        ; ENABLE IF > 1.8 MPH
BRCLR   $3C,$$01,LD89F              ; BR IF NOT b0
; ... else
LDX     #$82EE                        ; DISABLE IF <= 0.9 MPH
;
LD89F: LDAB    L0083                  ; Vss TABLE INPUT
LDAA    L0125                        ; PURGE D.C., (Cnts/16)
CMPB    0,X                          ; 0.9 MPH
BLS     LD8D3                        ;
; ... else
LDAB    L0095                        ; TPS Ld val
CMPB    2,X                          ; 2.3% TPS, DISABLE IF <=
BCC     LD8CD                        ; BR IF
;
; CK AFR OPTION WD 1 0001 0100
;   b3, 1 = LET LOW TPS DISABLE CCP
;
LDAB    $8016                        ; AFR OPTION WD 1
BITB    #$08                        ; b3 = LOW TPS DISABLE CCP
BNE     LD8D3                        ; BR IF b3
; .... else
BRSET   L0047,$$10,LD8D3            ; BR if b4, Vss FAIL
; ... else
LDAB    L0083                        ; VSS TABLE INPUT
CMPB    $82E7                        ; SAVE PURGE IF > 20 MPH
BLS     LD8D3                        ;
; ... else
LDAB    L00E0                        ; Blk learn mult
CMPB    $82E9                        ; DISABLE PURGE IF <= 107 BLM
BLS     LD8D3                        ; BR IF BLM LT 107
; ... else
BRSET   L003D,#2,LD8D9              ; BR IF b1, CLEAN CANISTER
; ... else
BRA     LD8D3                        ;

LD8CD: LDAB    L0079                  ; MAP FOR SPD/DENS CALC
CMPB    1,X                          ;
BCC     LD8DC                        ;
; ... else
LD8D3: BCLR   L003D,#2              ; CLR b1, CLEAN CANISTER
;
CLR     L0126                        ; OLD CCP D.C.
;
LD8D9: JMP     LD97B                  ;

LD8DC: BRCLR  L003D,#2,LD8E6        ; BR IF NOT IN CLEAN CANISTER
; ... else
LDAA    L0126                        ; OLD CCP D.C.
STAA    L0125                        ; PURGE D.C., (CNTS/16)
;

```

```

LD8E6: LDAB    L007F          ; Current manifold vacuum
        CMPB    $82E0        ; IF VAC > __ kpa dont change
        BHI     LD923        ; can purge D.C.
        BRCLR   L0045, #88, LD8F3 ; BR IF NOT b3 & b7
        BRA     LD923        ; B3 1 = MAP ACCEL ENRICH ON
        ; B7 1 = TPS ACCEL ENRICH ON
        ; ... else
        ;

LD8F3: LDAB    L0127          ; PURGE TMR
        CMPB    $82E1        ; 0.1 SEC BETWEEN PURGE D.C. UPDATES
        BCC     LD900        ; BR IF TMR GT THRESH
        ; ... else
        INC     L0127        ; PURGE TMR
        BRA     LD972        ;

LD900: LDAB    L00E6          ; Clsd lp intrgrator
        CMPB    $82E2        ; 125 INT
        BCC     LD91C        ; IF INTIGRATOR G.T. E.Q. 125 INRC CCP D.C.
        ; ... else
        CMPB    $82E3        ; 90 INT
        BCC     LD923        ; IF INTIGRATOR LT EQ 90 DECR CCP D.C.
        ; ... else
        SUBA    $82E5        ; 0.8 PCT CCP D.C.
        BCC     LD912        ; BR IF CCP D.C. GT 0.8%
        ; ... else
        CLRA    ; A = 0
LD912: CMPA    $8522        ; D.C. = 51
        BHI     LD923        ; BR IF D.C. GT 51%
        ; ... else
        LDAA    $8522        ; FORCE IDLE CELL LRN IF CCP DC LT 51%
        BRA     LD923

LD91C  ADDA    $82E4        ; 1.9% CCP DC,
        BCC     LD923        ; IF NO OVER FLOW
        ; ... else
LD923  LDAA    #255        ; LIMIT TO MAX
        CLRB    ; B = 0
        STAB    L0127        ; PURGE TMR

        BRSET   L0047, #80, LD935 ; BR IF b7
        ; B7 1 = CLOSED LOOP
        ; ... else

        BCLR    L003D, #2    ; CLR CLEAN CANISTER BIT
        ;
        ; AFR OPTION WD 1 0001 0100
        ; b5, 1 = OPEN LOOP PURGE
        ;
LD935  LDAB    $8016        ; AFR OPTION WD 1
        BITB    #$20        ; b5, 1 = OPEN LOOP PURGE (IS 0)
        BEQ     LD97B        ; BR IF NTO b5
        ; ... else

LD935  PSHA
        LDAA    L0125        ; PURGE D.C., (CNTS/16)
        CMPA    $82E6        ; 50 PCT DC.

```

```

BLS      LD955      ; BR IF D.C. LT 50%
          ; ... else
LDAB     L0083     ; VSS TABLE INPUT
CMPB     $82E7     ; 20 MPH
BLS      LD955     ; SAVE PURGE IF MPH > 20 MPH
          ; ... else
LDAB     L00E0     ; Blk learn mult
CMPB     $82E8     ; BLM = 115
BLS      LD955     ; SAVE PURGE IF > 115 BLM
          ; ... else
BSET     L003D,#2  ; SET CLEAN CANISTER BIT
PULA
STAA     L0126     ; OLD CCP D.C.

BRA      LD972

LD955    BCLR      L003D,#2  ; CLR CLEAN CANISTER BIT
          CLR      L0126     ; OLD CCP D.C.
          LDD      L006B     ; LIMITED AIR FLOW FM IDEAL GAS LAW
          CMPA     #32       ; LMT AIR FLOW
          BCS      LD964     ;
          ; .... else
          LDD      #$FFFF    ; USE MAX VAL

;-----
; PURGE DUTY CYC Vs AIR FLOW (MAF)
;   TBL = 9 LINES
;
; Table = D.C. * 2.56
;-----
LD964    ASLD
          ASLD
          ASLD
          LDX      #$82F1    ; INDEX PURGE DUTY CYC vs AIR FLOW TBL
          JSR      LE3C5     ; 2d LK UP W/LINE COUNT
          ;
          PULB
          CBA
          BCS      LD972     ;
          ; .... else
          TBA
          ;-----
          ; CK HEADS UP
          ; FOR CCP MOD
          ;-----
LD972    BRCLR     L0036,#$80,LD97F ; BR IF NO HEADS UP
          JSR      L580F     ; HEADS UP

          BRA      LD97F     ; SAVE SLEW'ED CCP D.C.

LD97B    CLRA
          CLR      L0127     ; PURGE TMR
          ;-----

          ;
          ;   SAVE SLEW'ED CCP D.C.
          ;
LD97F    BCLR      L003C,#$01  ; CLR b0
          STAA     L0125     ; PURGE D.C., (CNTS/16)
          BEQ      LD98A     ; BR IF D.C. = Z
          ; ... else
          BSET     L003C,#$01  ; SET b0

          ;-----
          ; GET BATTERY VOLTAGE

```

```

;-----
LD98A  LDAA    #$60      ; SEL A/D CH 6, FUEL PUMP
      JSR     LF0D2     ; Go read A/D

      STAA   L0104     ; PUMP VOLTAGE

      ;
      ; CK DIAG MOD WD 2
      ;   b4, 1 = Err 54 SET
      ;   b3, 1 =
      ;   b6, 1 =
      ;
      LDAA   L0041     ; DIAG MOD WD 2
      BITA   #$10     ; b4, 1 = Err 54 SET
      BNE   LD9D4     ; BR IF b4
      ;     .... else
      BITA   #$08     ; b3
      BEQ   LD9D9     ; BR IF NOT b3
      ;     ... else
      LDAB   L0103     ; A/D BATTERY VOLTAGE
      CMPB   $82AC     ; 2.0 VDC, IF IGN VOLTAGE < 2V SKIP ERR 54 CHK
      BCS   LD9D9     ; br if Vbat LT 2.0 VDC
      ;     .... else
      BITA   #$40     ; b6
      BEQ   LD9B6     ; BR IF NOT b6,
      ;     ... else
      LDX   L0037     ; MINOR LOOP MD WD
      BPL   LD9D9     ; BR IF
      ;     .... else
      CMPB   $82AB     ; IF PUMP VOLTAGE < 2.0 VDC SET Err 54
      BCC   LD9CE     ;
      ;     ... else

      BRA   LD9D4

LD9B6  LDAB   L0105     ; PUMP LOW VOLTS TMR
      INCB   INCB     ; INCR PUMP LOW VOLTS TIMER
      BEQ   LD9BF     ; BE IF TIMER = 0
      ;     ... else

LD9BF  STAB   L0105     ; PUMP LOW VOLTS TMR
      CMPB   $82A9     ; IF VOLTAGE LO FOR 1.5 SEC SET ERR
      BCC   LD9D2     ; BR IF LOW VDC TIMER LT 1.5 THRESH
      ;     ... else

      ;
      ; IF PUMP VOLTAGE < 2.0 VDC SET ERR 54
      ;
      LDAB   L0104     ; GET PUMP VOLTAGE
      CMPB   $82AA     ; 2.0 VDC, PUMP LOW VDC ERR THRESH
      BCS   LD9D9     ; BR IF PUMP VOLTAGE LT 2.0 VDC
      ;     .... else

LD9CE  ORAA   #$40     ; SET b6
      ANDA   #$DF     ; CLEAR b5
      ;
      BRA   LD9D9     ;

LD9D2  ORAA   #$10     ; SET BIT 4
LD9D4  BSET   L0050,$#20 ; SET b4, ERR 54 PUMP VOLTAGE BIT

      ORAA   #$20     ; SET BIT 5

```



```

;
; SET ERR 54
;
LD9D9  TAB
      ANDB    #$04                ; CLEAR b2

      ANDA    #$F3                ; CLEAR b2 & b3, (DRP'S)

      ASLB
      ABA
      STAA    L0041                ; DIAG MOD WD 2

      LDAA    L0103                ; GET A/D BATTERY VOLTAGE

;
; CK AFR OPTION WD 1, 0001 0100
;   b6, 1 = USE A/D BATT FOR INJ Lk Up, (else use PUMP VDC)
;
      LDX     #$8016                ; AFR OPTION WD 1
      BRSET  0,X,$$40,LD9F3        ; BR IF b6
                                      ; ... else
      BRSET  L0041,$$20,LD9F3      ; BR IF b5

;-----
; LK UP BPW BATTERY CORRECTION
; (USE PUMP or BATTERY VDC)
;
; TBL = usec * 0.032768
;-----
      LDAA    L0104                ; PUMP VOLTAGE
LD9F3  LDX     $$83FA                ; INJ OFF SET vs BAT VOLT TABLE
      JSR     LE3D0                ; 2d Lk Up, (NO OFF SET)

      STAA    L00EE                ; BATT BPW CORRECTION

      RTS
*****

*****
* DIAGNOSTICS
*
* >>> SEGMENT D (13), OF MAJOR LOOP EXEC
*
*****
      ORG    $D9FC

LD9FC: BCLR    L0038,$$18            ; CLR b3 & b4, IN CIM MODE, & ALDL DIAG
                                      ; (Blink out diagnostics)
                                      ;
      LDAB    $0000                ; NINOR LP CNTR
      ANDB    #$F0                ; MASK FOR HI NIBBLE
      BNE    LDA0F                ; BR IF NZ
                                      ; .... else
      BSET    L0038,$$08            ; SET b3, IN CCM MODE

      BRCLR  L001A,$$01,LDA0F      ; BR IF NOT b0
                                      ; ... else
      BSET    L0038,$$10            ; SET b4, ALDL DIAG,(Blink out diag)

LDA0F  BRSET  L0038,$$02,LDA35      ; BR IF B1
                                      ; ... else
      BSET    L0038,$$02            ; SET b1,

*****

```

* CK IF MORE THAN 50 START UP W/O ERROR
 * IF SO, CLEAR THE NON VOL ERR WD'S 1 - 5

```

INC      L0016                ; INCR NO ERR COUNTER
LDAA     L0016                ; NO ERR COUNTER
CMPA     $824E                ; SUCCESSION PWR UP'A W/O ERR'S, (50d)
BLS      LDA35                ; IF L.T 50 SUCCESSION PWR UP'A W/O ERR'S

```

```

CLRA     ; CLR ERR'S IN ERR WD'S 1 TH 5
LDX      #L0004              ; ERR WD 1
LDAB     #5                  ; 5 ERROR WORDS
LDA26:  STAA     0,X          ;
INX      ; INC ERR WD POINTER
DECB     ; DECR WORD COUNTER
BNE      LDA26              ; TILL DONE
; ..... else

```

```

;-----
; CLEAR ADDITIONAL ERR WORDS
;-----

```

```

STAA     L0016                ; ERR COUNTER
STAA     L0044                ; FLG WD 4, DIAG READ OUT
STAA     L00DB                ; CURRENT ERR BLNK OUT CNT'R
INCB
STD      L0017                ; Non Vol ERR WD CK SUM

```

```

;-----
; CONTUNUE ERROR OR NO
;-----

```

```

LDA35  BRCLR    L0038,#$10,LDA52

BSET    L0042,#$80            ; SET IN DIAG MODE BIT
BRCLR   L0037,#$80,LDA4A     ; BR IF NOT b7, 1 = ENGINE RUNNING

```

```

JSR     LDB39

```

```

CLRA
STAA    L0044                ; FLG WD 4, DIAG READ OUT
STAA    L00DB                ; CURRENT ERR BLNK OUT CNT'R

```

```

BRA     LDA8D

```

```

LDA4A  BCLR     L003D,#$18     ; CLR BIT 0 & 7
; B0 1 = NOT USED
; B7 1 = P/U FOR COOL = 4K

```

```

JSR     LDABA

```

```

BRA     LDA8D

```

```

;-----
;
;-----

```

```

LDA52  BCLR     L0042,#$80     ; SET IN DIAG MODE BIT
CLRA
STAA    L0044                ; CLR FLG WD 4, DIAG READ OUT
STAA    L00DB                ; CLR CURRENT ERR BLNK OUT CNT'R

```

```

BCLR    L003D,#$18           ; CLR ALCL MODE, CLR B0 & B7
; B0 1 = CLEAN CANISTER
; B7 1 = P/U FOR COOL = 4K

```

```

BRCLR   L004F,#1,LDA72       ; BR IF NOT b0, ERR 51
; ... AND

```

```

BRCLR   L0054,#1,LDA72           ; BR IF NOT b1, PRIOR ERR 51

BSET    L0006,#1                 ; SET b0

;-----
; DO Non Vol RAM CK SUM
;-----
JSR     LE4A9                     ; GO DO Non Vol RAM CKSUM
STD     L0017                     ; Non Vol ERR CK SUMM
;-----
BSET    L0042,#1                 ; SET CK ENG LIGHT ON BIT

BRA     LDA8D

LDA72   BRCLR   L0037,#$80,LDA7E   ; BR IF NOT b7, 1 = ENGINE RUNNING
; ... else

JSR     LDC05

JSR     LDB76

BRA     LDA8D

LDA7E   BSET    L0042,#1           ; SET CK ENG LIGHT ON BIT

LDAB    #11
CLRA
LDX     #$00C6
LDA87   STAA   0,X
INX
DECB
BNE     LDA87

; .... else

;-----
; CLR CURRENT ERR FLG WORDS
;-----
LDA8D   LDAB    #5                 ; NUM OF ERR WD'S
CLRA    ; A = 0
LDX     #L004D                    ; START OF CURRENT ERR FLAG WD'S
LDA93   STAA   0,X                ; CLR ERR WD
INX     ; BUMP TO NEXT ADDR
DECB    ; DECR WD CNT'R
BNE     LDA93                     ; TILL DONE
; .... else

;-----

;-----
; ALDL MODE 4,  COMMAND CK ENG LIGHT
;
; CNT'L WD $0192, b1, 1 = CK ENG LIGHT CONTROLLED
; CNT'L WD $0193, b1, 1 = CK ENG LIGHT ON
;
; ECM 8D, ECM P/N 1227730
;-----
BRCLR   L0048,#$01,LDAAD         ; BR IF NOT b0, ALDL MODE 4

LDAA    L0192                     ; CNT'L WD
BITA    #$01                      ; b0, 1 = CK ENG LIGHT CONTROLLED
BEQ     LDAAD

; ... else
LDAA    L0193                     ; CNT'L WD
BITA    #$01                      ; b1, 1 = CK ENG LIGHT ON
BEQ     LDAB1

; ... else
BRA     LDAB6
;-----

```

```

;-----
;
;
;-----
LDAAD: BRSET    L0042,#1,LDAB6           ; BR IF b0, CK ENG LIGHT ON BIT SET
LDAB1: JSR     LE5BA                     ; TURN OFF CK ENG LAMP
      BRA     LDAB9
LDAB6: JSR     LE5AC                     ; SET BITS IN ECU CNT REG
LDAB9  RTS

;-----
LDABA  BRCLR   L0044,#$40,LDB1A         ; BR IF NOT b6, 1ST TIME FLG
      DEC    L00DD                       ; DECR BLINK ON/OF PERIOD CNT'R
      BNE    LDB38
      ; ... else
      LDAB   #L0004                       ; ERR WD 1
      BRSET  L0042,#1,LDB33              ; BR IF CK ENG LIGHT ON
      LDX   #L00D9                       ; 1ST DIGIT ERR BLINK CNT'R
      BRCLR  L0044,#$20,LDAD1           ; BR IF NOT 2ND PASS
      INX
LDAD1: DEC    0,X
      BMI    LDADE
      BSET   L0042,#1                   ; SET CK ENG LIGHT ON BIT
      LDAB   #04
      STAB  L00DD                       ; BLINK ON/OF PERIOD CNT'R
      BRA   LDB38
LDADE  BRSET  L0044,#$20,LDAE9          ; BR IF b5, 2nd PASS
      ; ... else
      BSET   L0044,#$20                 ; SET b5, 2nd PASS BIT
      LDAB   #8
      BRA   LDB33
LDAE9  BCLR   L0044,#$20                 ; CLR b5, PASS BIT
      BRCLR  L0044,#3,LDAF5            ; BR IF NIT b1 & b2, CNT'R = 0
      ; ... else
      DEC    L0044                       ; DCR FLG WD 4, DIAG READ OUT REP CNT'R
      BRA   LDB1D

;-----
;
;-----
LDAF5  LDAB   L00DB                       ; CURRENT ERR BLNK OUT CNT'R
      CMPB  #39
      BLS   LDB03                       ; BR IF L.T. 39
      ; ... else
      CLR   L00DB                       ; CLR CURRENT ERR BLNK OUT CNT'R
      BCLR  L0044,#$40                 ; CLR b5, 1st TIME FLG BIT
      BRA   LDB35
LDB03  BITB   #7

```

```

BNE      LDB12
; ... else
LSRB
LSRB
LSRB
; DIV BY 8

LDX      #L0004
ABX
LDAB     0,X
STAB     L00DC
LDB12   INC     L00DB
ASL      L00DC
BCC      LDAF5
; ERR WD 1
; ADD B TO X
; FOR ROTATING ERR'S IN BLINK OUT
; BUMP UP CURRENT ERR BLNK OUT CNT'R
; FOR ROTATING ERR'S IN BLINK OUT

LDB1A   BSET    L0044,#2
LDB1D   LDAB    L00DB
LDX      #LF13C
ABX
LDAA     0,X
TAB
ANDB     #$0F
LSRA
LSRA
LSRA
LSRA
STD      L00D9
; ... else
; SET b1, REP CNTR 2ND BIT
; CURRENT ERR BLNK OUT CNT'R
; ERR NUMBER TABLE FOR BLINK OUT
; ADD B TO X
; GET BLNK NUMBER
; MASK FOR LSB
; SHT TO UPPER NIBBLE
; 1ST DIGIT ERR BLINK CNT'R

LDAB     #$001C
BSET     L0044,$$40
; SET 1ST TIME FLG

LDB33:  STAB    L00DD
; BLINK ON/OF PERIOD CNT'R

LDB35:  BCLR    L0042,#1
; CLR CK ENG LIGHT ON BIT

LDB38   RTS

;-----
;
;-----
LDB39   BRCLR   L0047,$$80,LDB45
; BR IF NOT b7
; B7 1 = CLOSED LOOP
; ... else

BRSET    L003D,$$08,LDB53
; BR IF b3,
; B3 1 = 1st Cls Lp PASS
; Thu field svc mode
; ... else

BRSET    L0038,$$08,LDB60
; BR IF b3
; B3 1 = IN CCM MODE
; ... else

LDB45   LDAA    L003D
ANDA     #$F7
; ALCL MODE WD
; CLR BIT 3 1st Cls Lp PASS THUR
; FIELD SVC MODE

EORA     #$40
STAA     L003D
; 200 Msec TGL FOR 2 1/2Hz FLASH
; ALCL MODE WD

BRCLR    L003D,$$40,LDB69
; BR IF NOT 200 Msec TGL
; FOR 2 1/2Hz FLASH

BRA      LDB75

LDB53   BRCLR   L0038,#8,LDB75
; br if not b3

```

```

BRCLR    L003D,#$10,LDB63          ; ... else
                                           ; BR IF NOT b4,o2 R/L XMISSION
                                           ; (FLD SVC MODE)
BCLR     L003D,#$10                ; ... else
                                           ; CLR b4, o2 R/L XMISSION (FLD SVC MODE)
BRA      LDB69

LDB60    BSET    L003D,#$08          ; SET b3,
                                           ; B3 1 = 1st Cls Lp PASS THU
                                           ;         FIELD SVC MODE

LDB63    BRCLR   L0047,#$40,LDB6D    ; BR IF NOT b6
                                           ; B6 1 = o2 RICH
                                           ; ... else

BRA      LDB72

LDB69    BRCLR   L0042,#1,LDB72      ; BR if not b0, CK ENG LIGHT OFF
                                           ; ... else
LDB6D    BCLR    L0042,#1            ; CLR b0, CK ENG LIGHT ON BIT

BRA      LDB75                        ; EXIT

LDB72    BSET    L0042,#1            ; SET b0, CK ENG LIGHT ON BIT

LDB75    RTS
;-----

LDB76    LDY     #$824D                ; ERROR WD 5 MASK, 0000 0000
                                           ; ERR 65 & 66

LDB7E    CLRA
LDX      #$0005
ORAA    $51,X
DEX
BNE     LDB7E                        ; BR IF N/Z
                                           ; .... else

TSTA
BNE     LDBA6                        ; BR IF N/Z
                                           ; .... else

LDB89    LDX     #$0005
LDAA    $4C,X
ANDA    0,Y
STAA    $51,X
DEX
BNE     LDB89

                                           ; .... else
LDAA    $00D4                        ; LOG ERR TIME FILTER
BEQ     LDB9E

                                           ; ... else
DEC     $00D4                        ; LOG ERR TIME FILTER

BRA     LDC04

LDB9E    BCLR    L0042,#1            ; CLR B0, CK ENG LIGHT ON BIT
BCLR    L003C,#4
BRA     LDC04

LDBA6    LDX     #$0005
CLRA

LDBAA    LDAB    $4C,X
ANDB    $51,X
ANDB    0,Y
STAB    $51,X

```

```

ORAA    $51,X
DEY
DEX
BNE     LDBAA
; .... else
INC     L00D4
LDX     #$824F
; LOG ERR TIME FILTER
; 1 SEC'S LOG FILTER CONST #1

BRCLR   L0042,#1,LDBC6
; BR IF CK ENG LIGHT OFF

LDBC6   INX
        INX
        TSTA
        BEQ     LDBCA
; ... else

LDBCA   INX
        LDAB    0,X
        CMPB   L00D4
        BCC    LDC04
; LOG ERR TIME FILTER
; ... else
        STAB   L00D4
        TSTA
        BEQ     LDC04
; LOG ERR TIME FILTER
; ... else
        LDAA   $8251
        STAA   L00D4
; 10 SEC'S LOG FILTER CONST #3
; LOG ERR TIME FILTER

        CLRB
; B = 0
        LDY    #$01AC
        LDX    #$0005
LDBE2   LDAA   3,X
        ORAA   $51,X
        STAA   3,X
        LDAA   0,Y
        ORAA   $51,X
        STAA   0,Y
        STAB   $51,X
        DEY
        DEX
        BNE     LDBE2
; .... else
        STAB   L0016
; NO ERR CNT'R

;-----
; DO Non Vol RAM CK SUM
;-----
        JSR    LE4A9
        STD    L0017
; GO DO Non Vol RAM CKSUM
; Non Vol ERR CK SUM
;-----
        BSET   L0042,#1
        BSET   L003C,#4
; SET CK ENG LIGHT ON BIT

LDC04   RTS

;-----
LDC05:  LDAB    L0038
        BITB   #$24
        BNE     LDC4E
; b2 & b5, ERR14/15 & IN ALDL
; BR IF b2 & b5
; .... else
        BRSET  L0042,$#40,LDC1A
; BR IF INTIGRATOR
; FORCED TO 128, ERR44/45
; ... else

        LDD    L0019
        LSRD
; Eng run time
; DIV BY 2

```

```

CMPB      $8256                ; 30 SEC, IF RUN TIME < DISABLE ERR 13
BCS      LDC4E

; .... else
BSET     L0042, # $40         ; SET INTIGRATOR FORCED
; TO 128, ERR44/45 BIT

LDC1A:  LDAB      L008E                ; MNR LP FILTERED A/D o2
CMPB     $8258                ; 0.549 VDC, IF o2 > DIABLE ERR 13
BHI      LDC4E

; ... else
CMPB     $8257                ; 0.035 VDC, IF o2 =< DIABLE
BLS      LDC4E

LDAB     L00C9                ; o2 HEAVY LD TMR, 2 SEC
CMPB     $825A                ; 15 SEC MIN COND'S PERIOD FOR ERR 13
BHI      LDC53

; ... else
BRCLR   L0038, # $10, LDC56   ; BR IF NOT b4, (ALDL blink out mode)
; ... else
LDAB     L005D                ; FLT COOLANT TEMP
CMPB     $8255                ; IF COOL <= 69.5 C, (157F) DISABLE ERROR
BLS      LDC56

; ... else
LDAA     L0095                ; TPS Ld val
CMPA     $8259                ; 5% TPS, MIN FOR ERR 13
BHI      LDC49

; ... else
TST     L00C9                ; o2 HEAVY LD TMR, 2 SEC
BEQ     LDC56

; ... else
DEC     L00C9                ; o2 HEAVY LD TMR, 2 SEC

BRA     LDC56

LDC49   INC     L00C9                ; o2 HEAVY LD TMR, 2 SEC

BRA     LDC56

LDC4E   CLR     L00C9                ; o2 HEAVY LD TMR, 2 SEC
BRA     LDC56

LDC53   BSET   L004D, # $40         ; SET b6, ERR 21, TPS HIGH BIT

LDC56   LDAB   L0038

LDAA     L0094                ;
CMPA     $8261                ; IF TPS A/D VAL < = DISABLE ERR 21A
BLS     LDC72                ; BR IF TPS GT THRESH
; ... else
LDAA     L00C6                ; TPS ERR CNT'R
CMPA     $8262                ; CONDITION MUST EXIST FOR 3 SEC
; TO QUALIFY
BHI     LDC79

; ... else
LDAA     L006B                ; LIMITED AIR FLOW FM IDEAL GAS LAW
CMPA     $8263                ; IF AIR FLOW >= 15 gm/sec
; DISABLE ERR 21A
BCC     LDC74                ; EXIT via CLR ERR
; ... else

```



```

        INC      L00C6                ; INCR TPS ERR CNT'R
        BRA      LDC7E                ; EXIT
LDC72:  ANDB    #$FB
LDC74:  CLR     L00C6                ; TPS ERR CNT'R
        BRA      LDC7E
LDC79:  BSET    L004D,#4            ; SET ERR 21, TPS HIGH BIT
LDC7E:  ORAB    #$04                ; SET b2, DRP HAPPENED THIS PERIOD
        STAB    L0038
        LDAB    L0038
        BITB    #$E4
        BNE     LDCB2
                                ; .... else
                                ; Vss
                                ; 3 MPH, IF > DISABLE 24
        LDAA    L0081
        CMPA    $826C
        BHI     LDCB2
                                ; ... else
        LDAA    $00CA
        CMPA    $8271
        BHI     LDCC0
        LDAA    L0079                ; MAP for Spd/Dens calc
        CMPA    $8270                ; 22 Kpa, IF MAP > DISABLE 24
        BCC     LDCB2
                                ; ... else
                                ; TPS Ld val
                                ; 1.9% TPS, IF >= DISABLE ERR 24
        LDAA    L0095
        CMPA    $826F
        BCC     LDCB2
                                ; ... else
                                ; BR IF b0, IN PK/NEUT
                                ; (CPU INPUT STATUS WD)
                                ; ... else
                                ; RPM/25
                                ; IF <= 1000 RPM DISABLE Vss ERR 24
        BRSET   L003B,#$01,LDCB2
                                ; ... else
                                ; IF > 4600 RPM DISABLE Vss ERR 24
                                ; ... else
LDCB2:  CLR     $00CA
        BRA      LDCC6
LDCB7:  BITB    #$08
        BEQ     LDCC6
                                ; ... else
        INC     $00CA
        BRA      LDCC6
LDCC0:  BSET    L004E,$$80          ; SET ERR 24, Vss BIT
        BSET    L0047,$$10          ; b4,
                                ; B4 1 = VSS FAIL
LDCC6:  LDX     $$824A
        BRCLR   0,X,$$08,LDD35     ; BR IF NOT
        BRCLR   L0047,$$02,LDD35   ; BR IF NOT b1
                                ; B1 1 = LEARN ENABLED
                                ; ... else

```

```

BRSET    L0043,#$08,LDD35          ; BR IF OPT FOR 1 PASS
; ... else

BRSET    L0043,#$10,LDCE6          ; BR TST CYCLE TIME BIT SET
; ... else

LDAA     L00CD                      ; EGR ERR TIMER
CMPA     $8275                      ; BIN 2700 TO QUALIFY FOR ERR
BCS      LDD3B                      ; .... else

BSET     L0043,#$10                ; SET TST CYCLE TIME FLG BIT
CLR      L00CD                      ; EGR ERR TMR

;-----
; ALDL MODE 4,  COMMAND
;
;
;
; ECM 8D, ECM P/N 1227730
;-----
LDCE6:  BRSET    L0048,#$01,LDD35    ; BR IF IN IN ALDL MODE 4

LDAA     L0123                      ; LK'ED UP EGR D.C.
CMPA     $8274                      ; 75% DC, SKIP ERR 32 IF EGR < 75%
BCS      LDD35                      ; BR IF L.T. 75%
; .... else

LDAA     L007F                      ; CURRENT MANIFOLD VACUUM
CMPA     $8277                      ; DISABLE IF LD < 90 Kpa (ERR 32)
BLS      LDD35                      ; BR IF L.T. 90 Kpa

CMPA     $8278                      ; DISABLE IF LD > 1 Kpa
BHI      LDD35

LDAA     L0095                      ; TPS LD VAL
CMPA     $8279                      ; DISABLE IF TPS < 7%
BLS      LDD35                      ; BR IF TPS L.T. 7%

CMPA     $827A                      ; DISABLE IF TPS > 30%
BHI      LDD35                      ; BR IF TPS G.T. 30%

SUBA     150
BCC      LDD0F

; ... else

LDD0F:  NEGA
CMPA     $827C                      ; TPS % CHANGES 0.39%
BHI      LDD35                      ; BR IF TPS % CHANGES 0.39 DISALLOW TEST

LDAA     L0081                      ; 30 MPH, ENABLE TST
CMPA     $827F                      ; br if vSS lt 30 MPH
BCS      LDD35                      ; .... else

BRCLR    L0047,#$80,LDD35          ; BR IF NOT b7, 1 = CLOSED LOOP
; .. else

BRSET    L0046,#1,LDD35            ; BR IF b0, 1 = INTEGRATOR CHG
; (USED ERR 32)
; ... else

LDAA     L00EB                      ; Clsd lp intrgrator

BRSET    L0043,#4,LDD50            ; BR IF EGR DIAG IN WORK BIT SET
STAA     $00D2
SUBA     #128
BCC      LDD30

```

```

; ... else
LDD30  NEGA
      CMPA  $827B      ; CLS LP INT WINDOW VALUE , 6d
      BLS   LDD44
; ... else
;-----
;
;
;
;-----
LDD35: BRCLR  L0043,#4,LDD8A      ; BR IF EGR DIAG NOT IN WORK
      BRA   LDD87
LDD3B  BRCLR  L0038,#$08,LDD90    ; BR IF NOT b3, IN CCM MODE
; ... else
LDD3F  INC    L00CD              ; BUMP EGR ERR TMR
      BRA   LDD90
LDD44  LDAA   L00CD              ; EGR ERR TIMER
      CMPA  $8281              ; DELAY 5 SEC
      BLS   LDD3F
      BSET  L0043,#4            ; SET EGR DIAG IN WORK
      BRA   LDD8D
LDD50  SUBA   L00D2              ; OLD EGR DIAG VAL
      BCS   LDD59
; ... else
      CMPA  $827E              ; 5 IS INT or VAC CHANGE
; FOR EGR FAULT
      BCC   LDD70
; ... else
LDD59  LDAA   L00CD              ; EGR ERR TIMER
      CMPA  $827D              ; RUN EGR INT TST 2.5 SEC'S
      BCS   LDD3F
; .... else
      LDAA  L00D3              ; EGR FAIL COUNTER
      INCA  ; BUMP EGR FAIL COUNT
      CMPA  $8282              ; FAIL COUNTER = 3
      BLS   LDD6C
      BSET  L0043,#1            ; SET ERR TST 32 ACTIVE BIT
; DEC EGR FAIL COUNTER
LDD6C  DECA  ; DEC EGR FAIL COUNTER
      STAA  L00D3              ; EGR FAIL COUNTER
      BRA   LDD87
LDD70  LDAA   L00D3              ; EGR FAIL COUNTER
      SUBA  $8280              ; Dec FAIL COUNTER BY 1
      BCC   LDD6C              ; IF NO UNDERFLOW
; ... else
      CLRA  ;
      BCLR  L0043,#1            ; CLR ERR TST 32 ACTIVE BIT
      STAA  L00D3              ; EGR FAIL COUNTER
;

```

```

; CK AFR OPT WD 4, 0000 1001
; b0, 1 = ERROR 32 DISABLED IF PASSES 1 TIME
;
LDAA    $8019                ; AFR OPT WD 4
BITA    #$01                ; b0
BEQ     LDD87                ; BR IF NOT b1
; ... else
BSET    L0043,#8            ; SET OPT FOR 1 PASS BIT

LDD87:  BCLR    L0043,#$10   ; CLR TST CYCLE TIME FLG BIT

LDD8A:  BCLR    L0043,#4     ; CLR EGR DIAG IN WORK BIT

LDD8D:  CLR     L00CD        ; EGR ERR TIMER

LDD90:  BRCLR   L0043,#1,LDD97 ; BR ERR TST 32 NOT ACTIVE

        BSET    L004E,#$08   ; SET ERR 32, EGR DIAG BIT

LDD97:  BCLR    L0046,#$01   ; CLR b0
; B0 1 = INTEGRATOR CHG (USED ERR 32)

        LDAA    L0041        ; DIAG MOD WD 2
        LSRA
        BCC     LDDA2
; ... else
        BSET    L004F,#$40   ; SET ERR 45, o2 RICH BIT

LDDA2:  ASLA
        STAA    L0041        ; DIAG MOD WD 2

        BRSET   L0039,#1,LDDD1 ; BR IF IN SKIP ERR Stall
; Saver SPK BYPASS

        BRSET   $0000,#$10,LDDD1 ; NINOR LP CNTR

        BRCLR   L0037,#$40,LDDD1 ; BR IF NOT b6, 1 = SPK BYPASS CK ENABLED
; ... else
        LDAB    L00D0        ; NUM CONSEC PASSES W/ ERR 42
        LDAA    L0057        ; RPM/25
        CMPA    $8294        ; IF RPM < 100 THEN DISABLE ERR 42
        BLS     LDDCB
; ... else
        LDAA    $00D7        ; EST FB CNT'R
        BNE     LDDCB
; .... else
        CMPB    $8297        ; P/O EST ERR 42, 3d
        BCC     LDDC6
; ... else

        INCB

        BRA     LDDCC

LDDC6:  BSET    L0001,#$80   ; SET EST ERR, (42)

        BRA     LDDCE

LDDCB:  CLRB

LDDCC:  STAB    L00D0        ; NUM CONSEC PASSES W/ ERR 42

LDDCE:  CLR     $00D7        ; EST FB CNT'R

LDDD1:  BRSET   $003C,#$20,LDDD9

        BRCLR   L0002,#$80,LDDDC ; BR IF ERR 43B EST TEST

```

```

; FAILED THIS START UP

LDDD9: BSET    L004F,#$10      ; SET ERR 43, KNOCK BIT
LDDDC: BCLR    L0042,#2        ; CLR ERR 44 or 45 BIT

LDAA    L0038
BITA    #$C0                  ; b6 & b7
BNE     LDE45                 ; BR IF b6 & b7, SHFT LIGHT ON & HI BATT
; .... else

LDAA    $829F                 ; If o2 VOLTS >= 0.199 DISABLE o2 ERR 44
CMPA    L0092                 ; o2
BLS     LDE04

LDAA    L00C7                 ; o2 LEAN CNTR, 2 SEC
CMPA    $82A0                 ; R/L ERR TIME REQUISITE
BHI     LDE09                 ; ERR TIME G.T. 50 SEC

BRCLR   L0047,#$80,LDE04     ; BR IF NOT b7
; B7 1 = CLOSED LOOP
; ... else

BRSET   L0042,#4,LDE04       ; BR IF INTIGRATOR FORCED
; TO 128, ERR44/45
; ... else

BRCLR   L0038,#$08,LDE0F     ; BR IF NOT b3
; B3 1 = IN CCM MODE
; ... else

INC     L00C7                 ; BUMP UP o2 LEAN ERR TMR

BRA     LDE0F

LDE04: CLR     L00C7           ; CLR o2 LEAN ERR TMR

BRA     LDE0F

LDE09: BSET    L004F,#8        ; SET ERR 44 o2 LEAN
BSET    L0042,#2             ; SET ERR 44 or 45

LDE0F: LDAA    L0092           ; o2
CMPA    $82A1                 ; If o2 VOLTS >= 0.699 DISABLE
; ERR 45, LEAN
BLS     LDE3A                 ; BR IF o2 G.T. E.Q 0.699

LDAA    L00C8                 ; o2 RICH CNTR, 2 SEC
CMPA    $82A2                 ; 50 SEC; o2 LEAN TIME REQUISITE
BHI     LDE3F

BRCLR   L0047,#$80,LDE3A     ; BR IF NOT b7,
; B7 1 = CLOSED LOOPv

BRSET   L0042,#4,LDE3A       ; BR IF INTIGRATOR FORCED
; TO 128, ERR44/45

LDAA    L0095                 ; TPS Ld val
CMPA    $82A3                 ; ENABLE o2 LEAN ERR 45 IF TPS > 1.9%
BHI     LDE31

CMPA    $82A4                 ; ENABLE o2 LEAN ERR 45 IF TPS < 0%
BCC     LDE3A

; ... else

LDE31: BRCLR   L0040,#$08,LDE45 ; BR IF NOT b3,
INC     L00C8                 ; BUMP UP o2 RICH CNTR

```

```

        BRA        LDE45
LDE3A:  CLR        L00C8                ; CLEAR o2 RICH TIMER
        BRA        LDE45
        ;-----

LDE3F:  BSET       L004F,#4            ; SET ERR 45, o2 RICH BIT
        BSET       L0042,#2            ; SET ERR 44 or 45 BIT

LDE45:  BCLR       L0042,#4            ; CLR INTIGRATOR FORCED TO 128, ERR44/45

RTS
*****

*****
* BATTERY VDC, TCC, CARS, & E LIGHT ROUTINE
*
* >>> SEGMENT E (14), OF MAJOR LOOP EXE <<<
*
* TYPE $8D ECM P/N 1227730
*****
LDE49:  LDAA        #$10                ; SEL A/D CH 1, BATTERY
        JSR         LF0D2                ; Go read A/D

        STAA       $0103                ; A/D BATTERY VOLTAGE, (0.0968/V)
        CMPA       #90                  ; 8.7 VDC
        BCC        LDE5E                ;
        ; ... else

        CMPA       #40
        BCC        LDE70                ;
        ; ... else

        BSET       L0036,#$10           ; SET IGN OFF

        BRA        LDE70

LDE5E:  BRCLR      L0036,#$10,LDE6A     ; BR IF IGN ON

        LDX        L3FC8                ; PA2 CNT'R
        STX        $00D5                ; OLD PA2 CNT'R

        CLR        L00D7                ; EST FB CNT'R

LDE6A:  BCLR       L0036,#$10           ; CLR IGN OFF BIT
        JSR         LDF0F

LDE70:  BRSET      L0003,#$20,LDE91     ; BR IF VATS OK

        LDAA       L00D1                ; VATS ERR CNT'R
        CMPA       $82A5                ; IF NO VATS IN X SEC SET ERR 46, (50d)
        BHI        LDE82                ;
        ; ... else

        INCA
        BEQ        LDE91                ; INCR VATS ERR TMR
        ; BR IF TMR = 0
        ; ... else

        STAA       L00D1                ; VATS ERR CNT'R

        BRA        LDE91

        ;-----
        ; CK IF VATS ERR
        ; ALLOWED
        ;-----

LDE82  LDX         #$824B                ; ERROR WD 3 MASK, 0111 1111
        BRCLR      0,X,#2,LDE91         ; BR IF BIT 1 = 0 (NO CODE 46 VATS)

```

```

; ... esle
BSET      L0006,#2      ; SET VATS ERR
;-----

;-----
; DO Non Vol RAM CK SUM
;-----
JSR      LE4A9          ; GO DO Non Vol RAM CK SUM CKSUM
STD      L0017          ; ERR CK SUM
;-----

LDE91:   BRCLR      L00FD,#4,LDEFD      ; br if not MOTOR RESET WORKING
; ... else
LDAA     $0103          ; A/D BATTERY VOLTAGE, (0.0968/V)
CMPA     $8693          ; ADD OFF SET TO DESIRED RPM IF
; BATT =< 11.7 V
;
BHI      LDED0
;
CLR      L01B7          ; BAT TIME HI FOR DESIRED RPM OFFSET
;
LDAA     L0000          ; NINOR LP CNTR
CMPA     #$0E          ; 14d
BNE      LDEFD
; .... else
LDAA     L01B6          ; BAT TIM LO FOR DESIRED RPM OFFSET
CMPA     $8695          ; ADD OFF SET IF BATT LO 5 SEC'S
BHI      LDEB6          ; BR IF G.T. 5 SEC
;
INCA
BEQ      LDEFD
; ... else
STAA     L01B6          ; BAT TIM LO FOR DESIRED RPM OFFSET
;
BRA      LDEFD

LDEB6   CLR      L01B6      ; BAT TIM LO FOR DESIRED RPM OFFSET
LDAA     L01B5          ; DIFF DESIRED RPM FOR LOW BATTERY
ADDA     $8696          ; ADD 12.5 RPM TO OFFSET IF BATT LO
BCC      LDEC3          ; if no overflow
; ... else
LDAA     #255          ; MAX VAL
LDEC3   CMPA     $8698      ; LIMIT IAC OFFSET FOR LO BAT TO 75 RPM.
;
BLS      LDECB

LDECB   LDAA     $8698      ; LIMIT IAC OFFSET FOR LO BAT TO 75 RPM.
STAA     L01B5          ; DIFF DESIRED RPM FOR LOW BATTERY
BRA      LDEFD

LDED0   CMPA     $8694      ; SUB OFF SET TO DESIRED RPM IF
; BATT > 13.5 V
;
BLS      LDEFD
;
CLR      L01B6          ; BAT TIM LO FOR DESIRED RPM OFFSET
;
LDAA     $0000          ; NINOR LP CNTR
CMPA     #$000E
BNE      LDEFD
; .... else
LDAA     L01B7          ; BAT TIM HI FOR DESIRED RPM OFFSET
CMPA     $8695          ; ADD OFFSET IF BATT LO 5 SEC'S
BHI      LDEEE
;
INCA
BEQ      LDEFD

```

```

; ... else
STAA    L01B7    ; BAT TIME HI FOR DESIRED RPM OFFSET

BRA     LDEFD

LDEEEE  CLR      L01B7    ; BAT TIM HI FOR DESIRED RPM OFFSET
        LDAA     L01B5    ; DIFF DESIRED RPM FOR LOW BATTERY
        SUBA    $8696    ; ADD 12.5 RPM TO OFFSET IF BATT LO
        BCC     LDEFA

; ... else
LDEFA   STAA     L01B5    ; DIFF DESIRED RPM FOR LOW BATTERY
LDEFD   LDAA     L0081
        CMPA    $8697    ; CLR BATT IAC WHEN Vss > 20 MPH
        BLS     LDF07

        CLR     L01B5    ; DIFF DESIRED RPM FOR LOW BATTERY

LDF07   BRCLR   L0036, #10, LDF0E    ; BR IF IGN ON

        CLR     L01B5    ; DIFF DESIRED RPM FOR LOW BATTERY

LDF0E   RTS

; -----
LDF0F   LDX     L0081

        LDAA    L0085    ; RPM/25
        LDAB    #25     ; 675 RPM
        MUL
        FDIV
        PSHX
        PULA
        PULB
        STAA    L0140    ; GEAR RATIO
        CMPA    $82FC    ; 255d, 1st GEAR RATIO
        BHI     LDF44

        BCLR   L004B, #20    ; CLR b5, IN 1ST GEAR BIT
        CMPA    $82FC    ; 255d, 1st GEAR RATIO

        BCS     LDF47

; .... else
        CMPA    $82FD    ; 36d 4th GR MAX RATIO
        BHI     LDF47    ; BR IF

        LDAB    $0141    ; 4th GEAR TIMER
        CMPB    $82FF    ; 0.8 Msec FOR 4th GEAR DETERMINE
        BHI     LDF3F    ; BR IF TIME LT 800 Msec

        INCB
        BEQ     LDF4A

; ... else
        STAB    $0141
        BRA     LDF4A

LDF3F:  BSET    L004B, #10    ; SET IN 4TH GEAR BIT

        BRA     LDF4D

LDF44:  BSET    L004B, #20    ; SET IN 1ST GEAR BIT

LDF47:  CLR     L0141    ; 4TH GEAR TIMER

LDF4A:  BCLR   L004B, #10    ; CLR IN 4TH GEAR BIT
LDF4D:  BCLR   L0042, #8     ; CLR E LIGHT BULB TST IN WORK BIT

```



```

BRSET    L0037, #80, LDF69      ; BR IF NOT b7, 1 = ENGINE RUNNING
; ... else
BRSET    L0001, #8, LDF69      ; BR IF BAD SHUTDOWN
; ... else
LDAA     $0146
CMPA     $8253
BCC      LDF69
; ... else
INC      $0146
BSET     L0042, #8             ; SET E LIGHT BULB TST IN WORK BIT

JMP      LDF60

LDF69:  LDAA     L005D          ; FLT COOLANT TEMP
        CMPA     $8357
; ...
        BLS      LDFEA

        LDAA     L0081
        CMPA     $8358          ; 0 MPH THERSH FOR ENG LAMP ENABLE
        BLS      LDFEA

        LDAA     L0095          ; TPS Ld val
        CMPA     $8359
        BCS      LDFEA
; .... else
        LDAA     L0079          ; MAP for Spd/Dens calc
        CMPA     $835A
        BLS      LDFEA

        LDAA     L0140          ; GEAR RATIO
        CMPA     $82FC
        BLS      LDFEA

        LDAA     L0085          ; RPM/25
        LDX      #$835B
; ...
        BRCLR   L0038, #80, LDF99 ; BR IF NOT b7, SHIFT LIGHT ON

        LDX      #$8364          ; 1300 RPM, THRESHOLD FOR ENG LIGHT ON

LDF99:  CMPA     0, X
        BLS      LDFEA

        CMPA     1, X
        BLS      LDFAF

        LDX      #0
        STX      $0142
        BSET     L0038, #80      ; SET b7, SHIFT LIGHT ON
        BSET     L0039, #10      ; SET SHFT LIGHT DELAY STARTED
        BRA      LDF60

LDFAF   CMPA     #$00E0
        BLS      LDFB5
LDFB5   LDAA     #$00E0
        LSRA
        LDAB     #$0010
        INX
        INX
        JSR      LE3CC          ; 2D LK UP

        CMPA     L0095          ; TPS Ld val
        BCS      LDFEA

```

```

; .... else
BRSET   L0039, #10, LDFDC      ; BR IF SHFT LIGHT DELAY STARTED

BSET    L0039, #10            ; SET SHFT LIGHT DELAY STARTED

LDAA    L0095                ; TPS Ld val
BPL     LDFCE
LDAA    #128
LDFCE   LSRA                 ; DIV BY 2
LDX     #8336E               ; E LITE ON DELAY MODIFER TBL
JSR     LE3D0                ; 2D LOOK UP, NO OFF SET

LDAB    $836D                ; E LITE ON DELAY MODIFER TBL, (SCALER)
MUL
STD     $0142

LDFDC   BSET   L0038, #80     ; SET b7,  SHFT LIGHT ON
LDX     $0142
BEQ     LDFFO
; ... else
DEX
STX     $0142
BRA     LDFED

LDFEA   BCLR   L0039, #10     ; CLR b4,  SHFT LIGHT DELAY STARTED BIT

LDFED   BCLR   L0038, #80     ; CLR b7,  SHFT LIGHT ON

;
; CK AFR OPT WD 3,  0101 0010
;  b2, 1 = CPU Auto Ratio Sel, 0 = TCC
;
LDFFO   LDAA    $8018         ; AFR OPT WD 3, 0101 0010
BITA    #504                ; b2
BNE     LDFFA               ; BR IF b2
; ... else
JMP     LE080               ; IF TCC JP TO TCC FUNCTION
;-----

*****
* COMPUTER AIDED RATIO SELECTION
*
*****

LDFFA   BCLR   L0042, #10     ; CLR CARS LIGHT TST IN WORK BIT

BRSET   L0037, #80, LE015    ; BR IF b7, 1 = ENGINE RUNNING
; .. else
BRSET   L0001, #8, LE015     ; BR IF BAD SHUTDOWN
; ... else
LDAA    $0147               ; CARS TMR
CMPA    $8254               ; 2 Sec, CARS LITE TST TIME
BCC     LE015               ; BR IF TIME GT 2 Sec's
; ... else
INC     $0147
BSET   L0042, #10           ; SET CARS LIGHT TST IN WORK BIT
BRA     LE07F

LE015   LDAA    $8306         ; 0d
LDAB    L004B               ; MODE WD FOR C.A.R.S
BCLR   L004B, #1           ; CLR CARS OFF DUE TO LOW BARO BIT

BITB    #1
BEQ     LE024
; ... else
LDAA    $8307
LE024   CMPA    L007B        ; BARO

```

```

BLS      LE02D

BSET     L004B,#$41      ; SET CARS OFF DUE TO LOW BARO BIT, AND
                        ;   IN 4TH GEAR BIT

BRA      LE07C

LE02D    LDAA     L0081
          CMPA     $8303      ; CARS DISABLE SPEED, 0d
          BCC     LE07C
                        ; ... else
          LDAB     L0085      ; RPM/25
          CMPB     $8304      ; RPM CARS DISABLE, (0d)
          BLS     LE047

          LDAB     L0095      ; TPS Ld val
          CMPB     $8305      ; RPM CARS DISABLE, 0
          BLS     LE047

BSET     L004B,#$40      ; SET WAIT FOR Vss RESET

BRA      LE07C

LE047    CMPA     $8301      ; CARS RESET SPEED, 0
          BLS     LE079

          BRSET    L004B,#$40,LE068 ; BR IF WAIT FOR Vss RESET SET
          CMPA     $8302      ; CARS ENABLE SPEED, 0
          BCS     LE07F
                        ; .... else
          LDAB     L005D      ; FLT COOLANT TEMP
          CMPB     $8300      ; IF COOL GT 151 C CK FOR CARS ACTIVE
          BLS     LE07F

          BSET     L004B,#$40      ; SET WAIT FOR Vss RESET BIT

          BRCLR   L004B,#$20,LE07F ; BR IF NOT IN 1ST GEAR
                        ; ... else
          BSET     L004B,#2      ; SET IN 1ST GEAR BIT
          BRA      LE07F

LE068    ADDA     #2
          BCC     LE06E
                        ; ... else

LE06E    LDAA     #$00FF
          CMPA     $8302      ; CARS ENABLE SPEED, 0d
          BCS     LE07C
                        ; .... else
          BRSET    L004B,#$10,LE07C ; BR IF IN 4TH GEAR

          BRA      LE07F

LE079    BCLR    L004B,#$40      ; CLR WAIT FOR Vss RESET BIT
LE07C    BCLR    L004B,#2      ; CLR CARS ACTIVE BIT

LE07F    RTS
          *****

          *****
          * TCC FUNCTION
          * P/O SEGMENT E OF MAJOR LOOP EXE
          *****

LE080:   LDX      L0139      ; SOFTWARE TIMER VAL, UP SHIFT TMR
          BNE     LE0B8      ; BR IF TMR NE 0, TCC IS LOCKED
                        ; .... else
          LDAA     L0095      ; TPS Ld val
          CMPA     $8308      ; CARS PASS BY UNLOCK CALIB

```

```

BCS      LE0B0                ; (disabled),
                                ; IF TPS LT BR
                                ; .... else
LDAA     L0081                ; GET FILT MPH
CMPA     $8309                ; 28d
BLS      LE0B0                ; IF MPH GT, BR
                                ; ... else
CMPA     $830A                ; 32 MPH
BHI      LE0B0                ; IF MPH LT or EQ, BR

LDAA     L0085                ; RPM/25
CMPA     $830B                ; IF GT 6375 RPM THEN BR
BCS      LE0B0                ;
                                ; .... else
LDX      L0137                ;
CPX      $830C                ; TIMER > 255 , BR & LOCK TCC
BHI      LE0AA                ;

INX
BRA      LE0B3

;
; INIT UP SHIFT TMR
;
LE0AA   LDX      $830E        ; 0d
        STX      L0139        ; UP SHIFT TMR

;
; CLR TMR
;
LE0B0   LDX      #0           ; X = 0
LE0B3   STX      L0137        ; TCC RDY TMR

BRA      LE0C4

LE0B8   DEX                ; DEC UP SHFT TIMER
        STX      L0139        ; UP SHIFT TMR
        BEQ      LE0C4        ; BR IF TIME UP
                                ; ... else
        BSET     L004A,#$11    ; SET TCC LOCKED & TCC PASS BY FLGS
        JMP      LE1BB

LE0C4:  BCLR     L004A,#$10    ; CLR PASS BY FLG, TCC MD WD

;-----
; COOLANT LIMITS
;
;-----
LDAA     L005D                ; FILT COOLANT TEMP
CMPA     $8314                ; TCC CALIB
BCS      LE0D9                ; BR IF L.T.
                                ; .... else

;
; AFR OPT WD 3, 0101 0010
; b3, 1 = UNLOCK TCC IN DECEL FUEL C/O, (TO COAST)
;
LDAA     $8018                ; AFR OPT WD 3
BITA     #$08                ; b3,
BEQ      LE0DC                ;
                                ; ... else
BRCLR    $0056,#$02,LE0DC    ; BR IF NOT b1, IN FCO

LE0D9:  JMP      LE1AD

;-----
; COAST RELEASE

```

```

;
;-----

;-----
; TPS COAST RELEASE vs MPH
;
;-----
LE0DC: LDAB      $8315                ; Coast release, 32 MPH thresh
      BRSET     L004A,#2,LE0E7      ; BR if coast
      ; ... else
      ADDB      #5                  ; 5 MPH HYST
      BCS       LE0EB              ; BR IF OVERFLOW
      ; .... else
LE0E7: CMPB     L0081              ; FILT MPH
      BLS       LE0F3              ; TO COAST 2 IF MPH HIGH

;-----
; COAST 1 IN USE , LOW Vss
;
;-----
LE0EB: BCLR     L004A,#2            ; CLR TCC COAST BIT
      LDAB      $8316              ; COAST 1, (8d)
      BRA       LE0F9              ; BR TO COAST 3

;-----
; COAST 1 IN USE , HIGK Vss
;
;-----
LE0F3: BSET     L004A,#2            ; SET TCC COAST BIT
      LDAB      $8317              ; COAST 2, 0

;-----
; CK FOR TCC LOCKED HYST
; (USE IF LOCKED)
;
;-----
LE0F9: BRCLR    L004A,#1,LE103      ; TCC MD WD, BR IF TCC UNLOCKED
      SUBB      $8318              ; Else Sub hysy, (3d) fm Coast
      ; rel TPS Thresh
      BCC       LE103              ; if no math error
      ; ... else
      CLRB      ; if undrflow

;-----
; CK CURRENT TPS WITH COAST
; RELEASE THRESH
;
;-----
LE103: CMPB     L0095              ; TPS Ld val
      BLS       LE10A
      JMP       LE19A              ; GO UN-LOCK TCC

;-----
;
;-----
LE10A: LDX      $833B              ; MOVE TO NXT GEAR
      BRCLR    L003B,#$04,LE114    ; BR IF NOT b2, IN 4TH GR
      ; (CPU INPUT STATUS WD)
      ; ... else
      LDX      #$831F              ; TCC CALIB, 40
      ; MOVE TO LOW GEAR TBLs
LE114: LDAB     #15
      ;
      BRCLR    L004A,#1,LE11E      ; BR IF TCC NOT LOCKED
      ; ... else

```

```

        INX                ; IF LOCKED
        INX
        LDAB    #1
LE11E:  LDAA     L0081      ; FILT MPH

        BRSET   L0031,#$02,LE12A ; BR IF b1, IN 3rd or 4th GR
        ; ... else
        SUBA    $831E     ; TCC CALIB, 0 MPH
        BCC     LE12A
        ; ... else
        CLRA
        ; A = 0

LE12A  CMPA    0,X        ;
        BCS     LE1AD
        ; ... else
        INX
        CMPA    0,X
        BCC     LE1AD
        ; INC TBL POINTER
        ;
        ; ... else
        ABX
        BRCLR  L004A,#1,LE14B ; ADD B TO X
        ; BR IF TCC UN-LOCKED
        ; ... else

        ;
        ; CK AFR OPTION WD 1 0001 0100
        ; b0, 1 = RPM FOR TCC vs MPH
        ;
        LDAB    $8016     ; AFR OPTION WD 1, 0001 0100b
        BITB    #$01
        BEQ     LE14B     ; b0
        ; BR IF NOT b0, RPM FOR TCC
        ; ... else

        ;-----
        ; FOR RPM FOR TCC
        ;
        ;-----
        LDAB    #16      ; 400 RPM
        ;
        LDAA    L0085     ; RPM/25
        CMPA    #192     ; 4800 RPM LIMIT
        BCS     LE156
        ;
        ; ... else
        LDAA    #192     ; USE 4800 RPM
        BRA     LE156
        ;
        ;-----

        ;-----
        ; FOR MPH FOR TCC
        ;
        ;-----

LE14B  LDAB    #24      ;
        ;
        LDAA    L0081     ; FILT MPH
        CMPA    #100     ; 100 MPH
        BCS     LE155     ; BR IF MPH LT 100 MPH
        ; ... else
        LDAA    #100     ; USE 100 MPH

LE155  ASLA
        ; MULT BY 2

LE156  JSR     LE3CC     ; 2d Lk Up
        ;
        BEQ     LE1AD
        ; ... else
        CMPA    L0095     ; TPS Ld val
        BCS     LE1AD
        ;

```

```

;-----
; 4-3 / 4-2 DN SHIFT RELEASE
;-----
BRCLR    L004A,#4,LE16D      ; IF TTC NOT IN COAST

DEC      L0145                ; DN SHFT TIMER
BNE      LE18D

; .... else
BCLR     L004A,#$04          ; TCC MD WD, CLR COAST
; CLR DN SHFT IN PROGRESS

BRA      LE18D

LE16D:  BRCLR    L003B,#$04,LE18D  ; BR IF NOT b2, IN 4TH GR
; (CPU INPUT STATUS WD)
; ... else
BRSET    L004A,#$08,LE18D  ; BR IF NOT 4TH LAST PASS
; ... else
BSET     L004A,#4          ; SET TCC COAST FLG
; 4-3 / 4-2 DN SHFT FLG

LDAA     $8319                ; TCC CALIB, 2d
BNE      LE182

BCLR     L004A,#4
BRA      LE18D

; .... else
; CLR TCC COAST BIT

LE182   STAA     $0145          ; 4 --> 3 SHIFT TIMER
CMPA     $0144                ; TCC LOCK TIMER
BLS      LE18D                ; BR IF TCC TMR LT 4 -> 3

STAA     $0144                ; TCC LOCK TIMER

;-----
; LOCK or UNLOCK
;-----
LE18D   LDAA     $0144          ; TCC LOCK TIMER
BEQ      LE195

; ... else
DECA
BRA      LE1AE

LE195   BSET     L004A,#1      ; SET TCC LOCKED BIT
BRA      LE1BB

LE19A   LDAA     L0081          ; FILT MPH
CMPA     $831C                ; 0 MPH, TCC CALIB
BCS      LE1AD

; .... else
LDAA     $831B                ; TCC CALIB, 0d
BRSET    L003B,#$04,LE1AE    ; BR IF b2, IN 4TH GR
; (CPU INPUT STATUS WD)
; ... else
LDAA     $831A                ; TCC CALIB, 0d
BRA      LE1AE

LE1AD   CLRA                  ; A = 0

LE1AE:  STAA     $0144          ; DELAY TMR FOR TCC LOCK

LDAA     L0081                ; FILT MPH
CMPA     $831D                ; TCC CALIB, 76 MPH
BCC      LE195

; ... else

```

```

BCLR      L004A,#$01                ; CLR TCC LOCKED BIT

;-----
; OLD 4TH GEAR STATE FLG SET UP
;-----
LE1BB:    BCLR      L004A,#8          ; CLRR TCC MD WD BIT 3

BRCLR     L003B,#$04,LE1C5          ; BR IF NOT b2, IN 4TH, (EXIT)
; (CPU INPUT STATUS WD)
; ... else

BSET      L004A,#8                  ; ELSE SET 4TH GR LAST PASS FLG

LE1C5:    RTS                          ;
*****

*****
* Air/Fuel MAJOR LOOP
*
* >>> SEGMENT F OF MAJR LOOP EXE <<<
*
*****
LE1C6:    LDAA      L0000              ; MNR LOOP CNTR
BITA      #$10                       ; b4,
BNE       LE21D                       ;
; .... else

;-----
; 200 MSEC LOOP
;-----
LDX       #L001B                      ; BLM BASE ADDR FOR 16 CELLS
BRCLR     L0047,#$02,LE21C           ; BR IF NOT b1, LEARN ENABLED
; ... else

;
; CK COOLANT LMT'S
;
LDAB      L005D                      ; FLT COOLANT TEMP
CMPB      $852A                      ; MIN COOL THRES FOR KEEP ALIVE UPDATE
BCS       LE21C                      ; BR IF COOL LT THRESH, (EXIT)
; .... else

CMPB      $852B                      ; HI TEMP THRESH
BCC       LE21C                      ; BR IF TEMP GT THRESN, (EXIT)
; ... else

;
; GET APPROPRAITE BLM CELL CONTENTS
; & CK IF IDLE CELL
;
LDAB      L00DF                      ; BLM CELL, 0-7
ABX                          ; POINT to BLM FOR KEEP ALIVE 'B'
LDAA      0,X                       ; GET CURRENT BLM
CMPB      $8527                      ; CELL A NUM (IDLE)
BEQ       LE1FA                      ; IF CELL = 0, DO CELL 0
; ... else

;-----
; KEEP ALIVE MEM B CELL
; LRN LOGIC
;-----
CMPB      $8528                      ; HI LIMIT FOR OTHER CELLS)
BNE       LE21C                      ; IF WRONG CELL, EXIT
; .... else

LDX       L000B                      ; KEEP ALIVE MEM CELL B
LDAB      $852D                      ; CELL B FILTER COEF
JSR       LE31C                      ; LAG FILTER ROUTINE

```



```

STD      L000B          ; SAVE SAM B FILTERED
BRA      LE21C         ; EXIT

;-----
; KEEP ALIVE MEM A CELL
; LRN LOGIC
;-----
LE1FA:  LDAB      $00E1      ; CELL 0 LRN ENAB TMR
        CMPB      $8529      ; CELL 0 LRN TMR THRESH
        BCC       LE21C      ; EXIT
        ; ... else
        LDAB      L0095      ; TPS Ld val
        CMPB      #6         ;
        BHI       LE21C      ; BR IF TO HI FOR CELL 0 LEARN

        LDAB      L0081      ; FILT MPH, (Vss)
        BNE       LE21C      ; BR IF Vss GT 0, (EXIT)
        ; .... else
        LDAB      $00E1      ; CELL 0 LRN ENAB TMR
        INCB      ; INCR CELL 0 LRN ENAB TMR
        BEQ       LE212      ;
        ; ... else
        STAB      $00E1      ; CELL 0 LRN ENAB TMR

LE212:  LDX       L0009      ; KEEP ALIVE MEM CELL A
        LDAB      $852C      ; KEEP ALIVE MEM CELL A FILT COEF
        JSR       LE31C      ; LAG FILTER ROUTINE

        STD       L0009      ; KEEP ALIVE MEM CELL A

LE21C:  RTS            ; RETURN TO MAJOR LOOP EXE

;-----
; LK UP ACELL ENRICH
; COOLANT MULT
;-----
LE21D:  LDAA      L005D      ; FLT COOLANT TEMP
        LDX       #$8586     ; MAP Acell Enr COOL MULT TABLE
        JSR       LE3C5      ; LK UP W/LINE COUNT

        STAA      $00AA      ; Acell Enr MAP COOL TRIM FACTOR

        LDAA      L005D      ; FLT COOLANT TEMP

        BRCLR     $0046,$80,LE22F ; BR IF NOT b7
        ; B7 1 = USE MAT FOR FUEL CALC
        ; ... else

        LDAA      L0063      ; INV MAT

LE22F:  PSHA      ; SAVE MAT TO STX
        CMPA      #208      ; 120c
        BLS       LE236      ; BR IF LT 120c

        LDAA      #208      ; ELSE USE 120c FOR LK UP

LE236:  PSHA      ; SAVE TO STX
        LDAB      L0037      ; MINOR LP MODE WD 1, RUNNING ?
        BMI       LE28A      ;

        BRSET     L0001,#8,LE28A ; BR IF BAD SHUTDOWN
        LDX       #$8431     ; INIT VALUE, START UP ENRICH
        ; vs COOLANT TABLE
        JSR       LE3D0      ; 2D LOOK UP, NO OFF SET

        PSHA

```

```

;-----
; LK UP START UP
; ENRICH vs MAT
;-----
LDAA    L0063          ; INV MAT
LDX     #$846F        ; START UP ENRICH vs MAT TABLE
JSR     LE3D0         ; 2D LOOK UP, NO OFF SET

PULB
ABA
LDAB    #0
BCC     LE257

LE257   LDD     #$FFFF  ; ... else
STD     L000D         ; INIT SOFTWARE CNT'R
                          ; TIME OUT A/F RATIO

;-----
; LK UP SPK vs START
; UP COOL
;-----
LDX     #$81B9        ; START UP SPK vs START UP COOL
LDAA    L0060         ; START UP COOLANT
CMPA    #208          ; 114 DEG C
BCS     LE264         ; IF G.T. 114 DEG C
                          ; ... else
LE264: LDAA    #208    ; USE 114 DEG C
PSHA
JSR     LE3D0         ; 2D LOOK UP, NO OFF SET

CLRB
STD     L0012         ; START UP vs COOLANT SPK ADV

PULA
LDX     #$81C9        ; START UP SPK ADV DECAY vs
                          ; START UP COOLANT
JSR     LE3D0         ; 2D LOOK UP, NO OFF SET

;-----
; LK UP SPK vs
; SPK DECAY DELAY
;-----
LDX     #$81C7        ; MULT FOR $81C9 TBL
JSR     LE33F         ; 8 x 16 MULT

STD     L0014         ; SPK DECAY DELAY

LDX     #$8441        ; START UP ENRICH DECAY DLY vs
                          ; START UP COOLANT TABLE

PULA
PSHA
JSR     LE3D0         ; 2D LOOK UP, NO OFF SET

LDX     #$843F        ; SCALAR FOR $8441 TABLE
                          ; (START UP ENRICHMENT DECAY DELAY
                          ; vs COOLANT TABLE)
JSR     LE33F         ; 8 x 16 MULT

STD     L000F         ; INITAL CHOKE DELAY

;-----
; LK UP OPEN LP
; FUEL/AIR %CHG vs COOL
;-----
LE28A: LDX     #$845E  ; OPEN LP FUEL/AIR %CHG vs COOL
PULA
PULA

```

```

JSR      LE3D0                ; 2D LOOK UP, NO OFF SET

STAA     $00F0                ; COOL FUEL/AIR RATIO

;-----
; LK UP  Acell Enr
; DECAY vs COOL
;-----
LDX      #$8590                ; Acell Enr DECAY vs COOL
LDAA     L005D                ; FLT COOLANT TEMP
JSR      LE3C5                ; 2D LK UP W/LINE COUNT

STAA     $00A8                ; DIFF MAP DECAY

;-----
; LK UP  IAC WARM
; PK vs COOL
;-----
LDX      #$8689                ; IAC WARM PK vs COOL TBL
LDAA     L005D                ; FLT COOLANT TEMP
JSR      LE3c5                ; 2D LK UP W/LINE COUNT

PSHA
JSR      LE604                ; DO SCALED BARO FOR TBL LK UP'S

;-----
; LK UP  IAC MULT
; vs BARO
;-----
LDX      #$8684                ; IAC MULT vs BARO TBL
JSR      LE3CC                ; 2D LK UP

PULB
MUL
ASLD
BCS      LE2BE

; .... else
ADDA     L002D                ; KEEP ALIVE IAC POSIT
BCS      LE2BE                ; IF OVER FLOW
; .... else
ADDA     L01B1                ; IDLE SPD Time OUT STEPS
BCC      LE2C0

; .... else
LE2BE   LDAA     #255          ; MAX VALUE
LE2C0   STAA     L011D        ; IAC WARM PARK INIT POSIT, STEPS

BRSET    L0036, #$10, LE2D3    ; EXIT IF IGN OFF
; .... else
BRSET    L00FE, #$04, LE2D3    ; BR IF b2, IAC RESET IN WORK
; (IAC MD WD 2)
; .... else
CMPA     L0119                ; IAC MIN POSIT
BCC      LE2D3

; ... else
STAA     L0119                ; IAC MIN POSIT

LE2D3   RTS                    ; RETURN TO MAJOR LOOP EXE

```

* >>> GENERAL UTILITIES <<<<

* >>> 2/9/93 ANHT.BIN (\$8D) <<<
*

```
* MUX_READ.SRC
*   $E2D4
*
* COP2TOG.SRC
*   $E2F4
*
* GM P4 fuel modeling device read
*   $E2FA
*
* LAG_FIL.SRC
*   $E31C
*
* 8X16MULT.SRC
*   LE33F
*
* LKUP_3D.SRC
*   $E353
*
* LKUP_2DA.SRC
*   E3C5           ; 2D LK UP W/LINE COUNT
*
* HANDLE SPURIOUS INTERRUPTS
*   $E3ED
*
* P4M1616B.SRC
*   $E3EE
*
* 16 * 16 FIXED POINT MULTIPLY
*   $E424
*
* TPSLOAD.SRC
*   $E45C
*
* CHKSUM.SRC of Non Volatile RAM FLAG WORDS
*   $E4A9
*
* SHORT DELAY W/RTS
*   $E4B7
*
* SHORT DELAY W/RTS
*   $E4B7
*
* LIMITS TEST
*   $E579
*
* FACTOR, Factor in a 8 X 16 Multiply
*   $E586
*
* BIT ON/OFF IN ECU
*   $E5AC
*
* BIT SET, TOGGLE ECU BYTE 1
*   $E5D7
*
* MAKE PRESSURE LOAD VAR'S
*   $E5F3
*
* SCALED BARO FOR TBL LK UP'S
*   $E604
*
*****
*****
```

```

* MUX_READ.SRC
* TYPE $8D, ECM P/N 1227730
*
* CALL WITH:      A Reg = Mux ch num
* RETURNS WITH:   A Reg = A/D result
*
* CALLS:          LF0D2, A/D READ
* STACK REQ:      2 BYTES
*
*****

```

```

ORG $E2D4

```

```

LE2D4:  PSHX                ; SAVE FOR CALLER
        LDX      #$4002    ; CPU DATA LATCH

        BCLR     2,X,$0C   ; CLR b1, INITIALLY ASSUME CH #0
        BCLR     0,X,$20   ; CLR b5

        BITA     #4        ; BIT 2, CK CH #C SELECTED
        BEQ      LE2E5    ; BR IF NOT CH# C (CK CH A/B)
                        ; ... ELSE
        BSET     0,X,$20   ; SELECT CH# C

LE2E5:  ASLA
        ASLA
        ANDA     #$0C      ; PREVENT CH C FM BEING OR'ED
                        ; W/BIO CR

        ORAA     2,X       ; OR CH A/B & SELECT DATA W/BIOCR
        STAA     2,X

        LDAA     #$90      ; SET UP MUX READ
        JSR      LF0D2    ; GO DO A/D

        PULX                ; RESTORE X

        RTS

```

```

*****

```

```

*****

```

```

* COP2TOG.SRC
*
* Toggle COP2 to FMD
*
*

```

```

* ORG$ $E2F4
*****

```

```

LE2F4:  LDAA     L0033      ; Access serial data byte
        EORA     #2        ; FMD COP2, Toggle bit 1
        STAA     L0033
*****

```

```

*****

```

```

* GM P4 fuel modeling device read via SPI
* (AT $4002)
*

```

```

* Exec time:
* LE2FF 86 Cyc
* LE302 83 Cyc
*

```

```

* ORG $E2FA

```

```

*****

```

```

LE2FA:  BCLR    L0033,#$80          ; SEL FMD BYTE 1, BIT 7
        BRA     LE302              ;
                                           ;
LE2FF:  BSET    L0033,#$80          ; SEL FMD BYTE 2, BIT 7
                                           ;
LE302:  LDX     #$4000              ; INDEX CHIP'S, (I/O D)
        SEI     ; INT'S OFF
        BSET    2,X,#$04           ; SET b2, SELECT FMD CHIP ($4002)

        LDAA   L0033              ; SET UP FMD DESC OUT'S
        STAA   0,X                ; STORE DATA IN SERIAL SR, (SRR)

        BCLR   1,X,#$80           ; START XMISSION

        ;
        ; Wait min 8 sclk's, (39 clocks worst case)
        ;
        PSHB
        MUL
        MUL
        MUL
        PULB

        ;
        ; Get data returned fm SSR, disable FMD
        ;
        LDAA   0,X
        BCLR   2,X,#$04           ; CLR b2 ($4002)

        CLI    ; CLEAR & TURN ON INTERUPTS

        RTS

```

```

* LAG_FIL.SRC
*
* First order lag filter
*
* Filter out unwnated transients from 8 bit unsigned
* intigers.
*
* Computes NEWVALUE = OLDVALUE + Q * (INPUT - OLDVALUE)
*   where Q is a 8 bit fraction (00.xx)
*
* Call with:
*   X Reg = old value   (prior filtered value)
*   A Reg = New value  (to be filtered)
*   B Reg = Q          (Filter coef)
*
* Exec time:   104 Cycles
* Stack req:   6   Bytes
* Code length: 35   Bytes
*
* ALGORITHM:
*   NEW = OLD + Q * (IN - OLD)
*         = OLD HIGH + OLD LOW + Q * (IN - OLD HIGH)
*           - Q * OLD LOW
*         = OLD HIGH + Q * (IN - OLD HIGH) HIGH2
*           + Q * (IN - OLD HIGH) HIGH
*           + Q * (IN - OLD HIGH) LOW
*           +
*           + (1-Q) * OLD LOW
*
*   ORG $E31C

```

```

*****
LE31C:  PSHB                ; SAVE Q TO STACK
        PSHX                ;
        PSHX                ; SAVE OLD VAL
        ;
        TSX                ;
        SUBA    0,X         ; RESULT = INPUT - OLD VALUE
        BCC     LE32B      ;
        ;                ; ... ELSE
        NEGB                ;
        ADDB    0,X         ;
        STAB   0,X         ; CALC OLD VALUE HIGH + Q * DELTA HIGH2
        LDAB   4,X         ;
        ;
LE32B:  MUL                ;
        STAB   1,X         ;
        PSHA                ;
        LDD    3,X         ; CALC (1 - Q) * (OLDVALUE LOW)
        NEGB                ;
        BEQ    LE337      ;
        ;                ; ... else
        MUL                ;
        ADCA   #0          ;
        ;
LE337:  TAB                ;
        PULA                ;
        ADDD   0,X         ; ADD Q * DELTA
        PULX                ;
        PULX                ;
        INS                ;
        ;
        RTS
*****

```

* 8X16MULT.SRC

*

* 8 x 16 Multiply with 16 bit result result rounded
 * to the upper 16 bits.

*

* CALL WITH:

* A Reg = 8 BIT Multiplier
 * X Reg = Address of 16 bit Multiplicand

*

* RETURN WITH:

* A Reg = MSB of 16 bit result
 * B Reg = LSM of 16 bit result

*

* Exec time: 66 Cycles
 * Code length: 20 Bytes
 * Stack req: 3 Bytes

*

* ORG \$E33F

```

LE33F:  PSHA                ; SAVE MULTIPLIER
        LDAB   1,X         ; GET LSB OF MULTIPLICAND

        MUL                ; MSB PARTIAL PRODUCT
        ADCA   #0          ; ROUND

        PULB                ;
        PSHA                ; SAVE PARTIAL PRODUCT

        LDAA   0,X         ; GET MSB OF MULTIPLICAND

```

```

    MUL                ; MSB PARTIAL PRODUCT
    PSHX              ; SAVE

    TSX
    ADDB    2,X        ; ADD IN LSB PARTIAL PROD
    ADCA    #0         ; ROUND
    PULX
    INS

    RTS                ; RETURN TO CALLER

```

```

* LKUP_3D.SRC
*
* 3D Look up routine. Returns with the table value,
* (interpolated), based on 2 independent inputs.
*
* INPUTS:
* 1. R min value, (Rows)
* 2. Q min value, (Col's)
* 3. RNUM, number of Q Vals, (col's)
*    in each R table
* 4. Thr 1st R table, R num entries in length
*
*      4 + RNUM secont R table
*      4 + (N-1) Nth R table
*
* CALL WITH:
* A Reg = R input, (Row arg)
* B Reg = Q input, (Coll arg)
* X Reg = Table start address
*
* RETURNS WITH:
* A Reg = F(Q,R)
*
* EXEC TIME:    257 - 264 Cycles
* SRACK REG:    9 Bytes
* CODE LENGTH:  114 BYTES
*
*
* ORG$ E353
*****

```

```

LE353:  PSHY
        PSHB
        PSHX
        SUBA    0,X        ; CALC ROW ARG OFFSET, (LIMITED TO 0)
        BCC    LE35C      ; ... ELSE

        CLRA

LE35C:  SUBB    1,X        ; CALC COL ARG OFFSET, (LIMITED TO 0)
        BCC    LE361      ; ... ELSE

        CLRB

LE361:  PSHX                ; XFER TABLE ADDR TO Y REG
        PULY
        PSHA                ; SAVE ROW ADDRESS
        LDAA    #16        ; SPLIT COL ARG INTO TABLE OFFSET
        MUL                ; & INTERP PORTION
        PSHB                ; SAVE INTERP PORTION TO STACK

        TAB
        ABX
        PULA

```



```

PULB          ; GET ROW ARG
PSHA

LDAA    #16          ; SPLIT ROW ARG INTO R
                        ; TABLE NUMBER
MUL
PSHB          ; (MAX RESULT IS 255D)
LDAB    2,Y
MUL
ABX
PSHX
LDAB    2,Y
ABX

TSY
LDD     3,X
SBA

LDAB    3,Y
BCC     LE38C

                        ; ... else

NEGA
MUL
ADCA    3,X
BRA     LE392

;-----
; INTERP Y2, RND DOWN
;-----
LE38C: MUL
ADCA    #0
NEGA
ADDA    3,X

LE392:
PULX
PSHA
LDD     3,X
SBA
LDAB    3,Y
BCC     LE3A8          ;
                        ; ... else

NEGA
MUL
ADCA    3,X
BRA     LE3A8

;-----
; Interp Y & round
; down if necessary
;-----
LE3A2: MUL
ADCA    #0
NEGA
ADDA    3,X

LE3A8: PULB
PSHA
SBA          ; Y1 - Y2
LDAB    2,Y  ; FRACTION MULT
BCC     LE3B7  ; BR IF D >= 0
                        ; ... else
NEGA          ; ELSE INTERP Y (RND UP IF Req)
MUL
ADCA    1,Y
BRA     LE3BE          ; TO EXIT ROUTINE

```

```

;-----
; INTERP Y, & ROUND DOWN
; IF NECESSARY
;-----
LE3B7:  MUL
        ADCA    #0
        NEGA
        ADDA    1,Y

LE3BE:  INS
        PULX
        PULX
        PULB
        PULY

        RTS                ; Retun to caller
*****

*****
* LKUP_2DA.SRC
*
* LKUP_2D:  2D table lookup w/offset
*           spaced from table
*
* LE3C5:   2D table look up w/no offset
*           spacing from table
*
* P4LKUPQ: 2D table lookup
*           spaced 16
*
* Returns a value from a table based on the value of an
* independent variable. Only certin values of indepedednt
* variable are represented in the table.
*
* Call Arg:
* A Reg = independent var
* X Reg = Address of table
*
* LKUP_2D:
* B Reg = Value of indepedednt var
*
* LE3C5 & LKU_PQ:
* B Reg = Ignmored & 1st tabulated val is 0
*
* LE3C5:
* 1st byt of table = 256/H (subsequent bytes
* are the tabulated values)
* EXAMPLE:
* 2 for H = 128, (3 table entries max)
* 16 for H = 16, (17 table entries max)
* 25 for H = 10.4, (26 table entries max)
*
* LKUP_2D & LKU_PQ
* This byte omittedm H = 16
*
* RETURNS WITH: A Reg = table value
*
* EXEC TIME:  LE3C5 = 81 - 82 cyc
*             LKUP_2D = 78 - 81
*             LKUP_Q  = 73 - 74
*
* STACK REQ:  4 Bytes
*
*
* ORG $E3C5

```

```

*****
LE3C5:  PSHX                ; 2D LK UP W/LINE COUNT
        PSHB

        LDAB      0,X        * Get 256/spacing value
        INX

        BRA      LE3D4      ;

;-----
; LKUP_2D: ENTRY
;-----
LE3CC:  SBA                * Subtract offset
        BCC      LE3D0      * Br if D > 0
                                * 2D LOOK UP, NO OFF SET
                                ; ... else
        CLRA                *

;-----
; P4LKUPQ: ENTRY, LD3D0:
; 2D LOOK UP, NO OFF SET
;-----
LE3D0:  PSHX
        PSHB
        LDAB      #16        * Set 256/spacing, H to 16

;-----
; Seperate val into table offset
; & interp fraction
;-----
LE3D4:  MUL
        PSHB
        TAB
        ABX

;-----
;
;-----

        LDD      0,X        * Pick up Y1 & Y2
        SBA                * Get delta Y1 - Y2,
        PULB         * Ck if table is increasing ?
        BCC      LE3E4      * Br if not inc
                                ; ... else
        NEGA         * Interp & round up if req
        MUL
        ADCA      0,X
        BRA      LE3EA      ;

;-----
; Interp & round
; down if req'ed
;-----

LE3E4:  MUL
        ADCA      #0
        NEGA
        ADDA      0,X
LE3EA:  PULB
        PULX

        RTS

```

* HANDEL SPURIOUS INTERRUPTS

PULX
INS

RTS

* 16 * 16 FIXED POINT MULTIPLY
*
* ROUNDING IS FORM LOW ORDER BYTE OF RESULT
* THE MIDDLE 2 BYTES RETURNED IN A & B REG'S
*

* Calling:
* X = multiplicand
* A = MSB of Multiplier
* B = LSB of Multiplier
*

* Returning:
* X+5 = Partial prod
* +4 = Partial prod
* 3 = LSB of Multipicand
* 2 = MSB of Multipicand
* 1 = LSB of Multiplier
* 0 = MSB of Multiplier
*

* Execution time: 155 Cycles
*

```
LE424: PSHX          ; RESERVE STAKC SPACE FOR PARTIAL'S
      PSHX          ; MULT'END TO STACK
      PSHB          ; LSB MULT'PLR TO STACK
      PSHA          ; MSB MULT'PLR TO STACK
      TSX           ;
      LDAA          3,X          ; GET LSB OF MULT'END
      MUL           ; LSB MULT'END * LSM MULT'PLR
      ADCA          #0          ; ROUND PART PROD IN A REG
      STAA          5,X          ; SAVE ASL LSB OF RESULT, (TEMP)

      LDD           1,X          ; GET LSB OF MULT'PLT, (A) & MSB OF MULT'END
      MUL           ;
      ADDB          5,X          ; ADD RESULT LSB TEMP LSB OF RESULT
      ADCA          #0          ; ROUND TO MSB OF FINAL RESULT
      STD           4,X          ; SAVE AS TEMP FINAL RESULT

      LDAA          0,X          ; GET MSB OF MULT'PLR
      LDAB          3,X          ; GET LSB OF MULT'END
      MUL           ;
      ADDD          4,X          ; AD TO FINAL RESULT
      BCS           LE452       ; BR IF OVERFLOW
                                   ; .... ELSE
      STD           4,X          ; SAVE AS TEMP FINAL RESULT

      LDAA          0,X          ;
      LDAB          2,X          ; GET MSB OF MULT'PLR
      MUL           ; GET MSB OF MULT'END
      TSTA          ;
      BNE           LE452       ; BR IF OVERFLOW

      ADDB          4,X          ; ADD LSB TO MSB OF FINAL RESULT
      STAB          4,X          ; SAVE MSB OF FINSL RESULT, (PRIOR TO OVERFLOW tst
      BCC           LE457       ; EXIT WITH OUT OVERFLOW
                                   ; ... ELSE

LE452: LDD          #$FFFF      ; LIMIT TO $FFFF
      STD           4,X
```

```

LE457:  PULX          ; RESTORE STACK
        PULX          ; RESTORE STACK
        PULA          ; MIDDLE RESULTS IN A & B
        PULB

```

```

RTS

```

```

*****

```

```

*****

```

```

* TPSLOAD.SRC

```

```

*

```

```

* EXIT WITH 8 BIT TPS LD VAL AT L0095, TPS Ld val

```

```

*

```

```

*****

```

```

LE45C:  LDAB          L0038          ; DIAG MD WD 1
        LDAA          $8249         ; ERROR WD 1 MASK, 1111 0111
        BITA          #4            ; BIT 2
        BEQ           LE471         ;
        ; ... else
        LDAA          L0094         ; GET A/D TPS
        CMPA          $8260         ; 246 TPS A/D VAL
        BLS           LE471         ; IF TPS A/D VAL > SET ERR 21A
        ORAB          #4            ;
        BSET          L004D,#4      ; SET ERR 21, TPS HIGH BIT

```

```

LE471:  STAB          L0038          ; DIAG MD WD 1
        LDAA          L0094         ; A/D TPS
        BITB          #$24         ; IF ERR 21 & 22 SET
        BEQ           LE47C         ;
        ; ... else

```

```

LE47C:  LDAA          $8264         ; 0.96V A/D IS DEFAULT TPS IF ERR 21
        PSHA          ; SAVE TPS FOR LD CALC
        LDAB          L0099         ;
        CBA           ;
        BHI           LE48C         ; IF TPS > OFF SKIP FILTER

```

```

;-----
; FILTER
;-----

```

```

LDX          L0099          ;
LDAB          $83E6         ; 3.9% LOW TPS FILTER COEF
JSR          LE31C         ; LAG FILTER ROUTINE

```

```

STD          L0099          ; SAVE FLT'ED RESULT
;-----

```

```

LE48C:  LDD          L0099          ; FLT'ED RESULT
        ADDD         #128          ;
        TAB

```

```

        PULA          ; GET TPS BACK
        SBA
        BCC          LE497
        ; ... else
        CLRA

```

```

LE497:  LDAB          $83E4         ; 105d, %TPS FOR TPS HI to TPS LOW
        MUL
        ADDD         #$20         ; ROUND OFF
        ASLD         ; SCALE BY 2
        BCS          LE4A4
        ; .... else
        ASLD

```

```

        BCC      LE4A6
LE4A4:  LDAA     #255
LE4A6:  STAA     L0095

```

```

        RTS

```

```

*****

```

```

*****

```

```

* CHKSUM.SRC

```

```

*

```

```

*   CHECKSUM of Non Volatile RAM FLAG WORDS

```

```

*

```

```

* RETURNS WITH:

```

```

*   A Reg = MSB of check sum

```

```

*   B Reg = LSB of check sum

```

```

*

```

```

* EXEC TIME:

```

```

*   71 Cyc

```

```

*

```

```

*   ORG $E4A9

```

```

*****

```

```

LE4A9:  LDX      #5           ; SET INDEX FOR 5 NV BYTES

```

```

        LDD     #1           ; INC TO PREVENT 0 RESULT

```

```

LE4AF:  ADDB     3,X          ; SUM EACH FLK WORD INTO

```

```

        ADCA    #0          ; PARTIAL CKSUM

```

```

        DEX

```

```

        BNE     LE4AF       ; DEC INDEX & LOOP TILL DONE

```

```

        BNE     LE4AF       ; WITH ALL 5 WORDS

```

```

        BNE     LE4AF       ;

```

```

        BNE     LE4AF       ; .... ELSE

```

```

        RTS

```

```

*****

```

```

*****

```

```

* SHORT DELAY W/RTS

```

```

*****

```

```

LE4B7:  RTS

```

```

*****

```

```

*****

```

```

* LE4B8.SRC

```

```

*

```

```

*   >>>> SEGMENT 6 OF MAJOR LOOP EXE <<<<

```

```

*

```

```

*   READ COOLANT SENSOR & USE DEFAULT IF FAILED

```

```

*

```

```

*

```

```

*   Get A/D temp sensor reading:

```

```

*   If current flag shows 384 P.U. use 384 OHM

```

```

*   else 4K then use 4K OHM

```

```

*

```

```

*   If results are not within limits after some

```

```

*   engine run time, flag error & use default.

```

```

*

```

```

*   Exec time:      Cyc

```

```

*   Codel length:  115 Bytes

```

```

*

```

```

*

```

```

*   ORG $E4B8

```

```

*****

```

```

;-----
; CK IF HEADS UP CONNECTED
;-----
LE4B8: LDX      L5800          ; $5800
        CPX      #$7E58       ; CK IF HEADS UP ON LINE,
        BNE      LE4C3        ; (1ST CODE VAL)
        JSR      L5B32        ; BR IF NOT ON LINE
        JSR      L5B32        ; ELSE XFER THE LOGGING RAM Back
        JSR      L5B32        ; TO ECM
        JSR      L5B32        ; IM PRESENCE CK
;-----

;-----
; READ COOLANT MAKE INV COOLANT Var's
; TYPE $8D, ECM P/N 1227730
;-----
LE4C3: LDAA     #$40          ; SET A/D CH #4
        JSR      LF0D2        ; GET A/D RESULT
        PSHA     PSHA         ; SAVE A/D RESULT TO STX
;-----

;
; LK UP INV COOLANT VAL's
;
        LDX     #$F105        ; POINT TO 4K INV COOL TABLE
        BRSET   L003D,$$80,LE4D9 ; BR IF 4K PULL UP FOR COOL
        BRCLR   L0033,$$01,LE4D9 ; ... ELSE
        BRCLR   L0033,$$01,LE4D9 ; BR IF 4K IN IN USE
        LDX     #LF105        ; .... else
        BRA     LE4DF         ; POINT TO 348 OHM INV COOL TABLE
        BRA     LE4DF         ; LOOK UP VALUE

LE4D9: ADDA     #10           ; IS VAL GOING TO RAP AROUND
        BCC     LE4DF         ; BR IF NOT GOING TO RAP
        LDAA     #255         ; ... else
        JSR      LE3D0        ; LIMIT
LE4DF: JSR      LE3D0        ; 2D LOOK UP, NO OFF SET
        STAA    L005F        ; COOLANT TEMP, (DEG C +40) * (256/192)

        CMPA    #120          ; 50c
        BHI     LE4F1        ; BR IF COOL > 50c
        CMPA    #106         ; ... else
        BHI     LE4F4        ; 40 C
        BHI     LE4F4        ; BR IF COOL > 40c
        BCLR   L0033,$$01    ; ... else
        BCLR   L0033,$$01    ; CLR B0, 0 = COOL P/U = 4K OHM,
        BRA     LE4F4

LE4F1: BSET     L0033,$$01    ; SET COOL PU, 348 OHM PULL UP,
LE4F4: BCLR     L003D,$$80    ; CLR 4K PULL UP FOR COOL
;-----
; ERR 14, High temp error
;-----
        PULA     PULA         ; GET A/D COOL VAL FM STX
        BRSET   L003C,$$08,LE523 ; BR IF MODE 4 FORCING B.P. FUEL
        PSHA     PSHA
        LDAA    L005F        ; Get INV COOL VAL

```



```

LDAB    #$20                ; bit 5, assum ERR 14

CMPA    $825B              ; IF COOL 210c, DISABLE ERROR

PULA
BCC     LE50E              ; GET ERR TMR FM STX
                          ; Ck timer, BR IF TMR NOT OVERFLOW
                          ; ... else

;-----
; ERR 15, Low temp error
;-----

LDAB    #$10                ; Bit 4, assum ERR 15
CMPA    $825F              ; IF A/D VAL <= 250 DISABLE ERR 15
BLS     LE52E              ; Filter cool ic > cal val
                          ; (A/D <= cal)

LE50E:  LDX     L0019        ; Eng run time
        CPX     L0019        ; Ck eng run time vs cal value
        BLS     LE51C        ;
                          ; ... else
        BSET    L0038,$$02    ; Set b1, ERR14/15 THIS START UP
                          ; "this startup" flag

        ORAB    L004D        ; Signal ERR 14 or 15
        STAB    L004D        ; CURRENT ERR FLG WD 1

LE51C:  BCLR    L0046,$$80    ; CLR B7, B7 1 = USE MAT FOR FUEL CALC
        BRSET   L0038,$$02,LE526 ; Br if b1, ERR 14/15
                          ; ... else
        BSET    L0046,$$80    ; SET b7, B7 1 = USE MAT FOR FUEL CALC

;-----
; Implement default Coolant
;-----

LE526:  LDAA    $825C        ; DEFAULT COOLANT, 61c, 135d
        CLRB
        STD     L005D        ; Save default coolant

        BRA     LE523        ;

;-----
; Lag filter routine
;-----

LE52E:  LDAA    L005F        ; NEW COOLANT TEMP, (DEG C +40) * (256/192)
        BRCLR   L003C,$$10,LE53F ; SKIP FILT IF NOT 2ND PASS
                          ; ... ELSE
        LDAB    $83E7        ; FILTER COEF, (6.3%)
        LDX     L005D        ; COOLANT TEMP
        JSR     LE31C        ; LAG FILTER ROUTINE

        STD     L005D        ; FILT INV COOLANT TEMP

LE523:  RTS

;-----
; Init L005D, (COOLANT TEMP)
; L0060
;-----

LE53F:  CLRB
        STD     L005D        ; COOLANT TEMP

```

```

    STAA    L0060                ; START UP COOLANT

    BSET    L003C,#$10          ; SET 2ND PASS COOLANT FLAG

    RTS                          ; RET TO MAJOR LOOP EXE
*****

*****
* BLM_INIT.SRC
*
*
*   ORG $E548          ENDS $E578
*****
LE548: CLRB                ; ZERO B REG
        LDAA    #128      ; DEFAULT MID POINT VALUE
        STD     L0009     ; STAY ALIVE MEM CELL A
        STD     L000B     ; STAY ALIVE MEM CELL B

;-----
; Init 16 cells
;-----
LE5F4: LDX      #16        ; SET TO INIT 16 BLM CELLS

LE552: DEX                ; DEC LOOP COUNT
        STAA    $1B,X     ; SET MID PT VAL (128)
        CPX    #0         ; CK FOR DONE
        BNE    LE552     ; TILL DONE
        RTS              ; .... ELSE

LE55B: LDAA    L000B      ; VAL TO LIMIT
        LDX    #$8525    ; LO LIMIT FOR OTHER CELLS
        BSR    LE579     ; GO DO LIMIT OF VALUES

        STAA    L000B    ; SAVE LIMITED VALUE TO  S.A. CELL B

        BSR    LE5F4    ;

        LDAA    L0009    ; S.A. CELL A, VAL TO LIMIT
        LDX    #$8523    ; LO LIMIT FOR CELL 0, (IDLE)
        BSR    LE579     ; GO DO LIMIT OF VALUES

        STAA    L0009    ; SAVE LIMITED VALUE TO  S.A. CELL A

        LDX    #L001B    ; SAVE STAY ALIVE MEM CELL A IN IDLE CELL
        LDAB   $8527     ; NUM (REL ADDR) OF IDLE CELL, 4D
        ABX                ; POINT TO CELL A
        STAA    0,X      ; GET CURRENT CELL A VALUE

    RTS

*****

*****
* LE579.SRC
* LIMITS TEST
*
* Upper & lower bounds check on a 8 bit value
* If bounds exceeded apply cabilb table limits
* upper and lower are mutually exclusive
*
* CALL WITH:
*   A Reg = input value to limmit
*   X Reg = Lower bound in MSB

```

* = upper bound in LSB

* RETURNS:
* A Reg = limited result

* EXEC TIME: 16 - 24 Cyc
* COLDE LENGTH: 13 Bytes

* ORG \$E579 ENDS \$E585

LE579: CMPA 0,X ; IS A REG > LOW LIMIT ?
BHI LE5791 ; BR IF YES

LDAA 0,X ; ELSE APPLY LOW LIMIT

LE5791: CMPA 1,X ; IS A REG < HIGH LIMIT ?
BLS LE5792 ; BR TO EXIT IF YES

LDAA 1,X ; ELSE APPLY HIGH LIMIT

LE5792: RTS

* LE586.SRC
* FACTOR
*
* Factor in a 8 X 16 Multiply
*
* CALL WITH:
* A Reg = multiplicand
* X Reg = Multiplier address

* RETURNS:
* A Reg = Factored result, (Limited to FFFF)
* CY = TEMP + 2

* EXEC TIME: Cyc
* CODE LENGTH: Bytes

* ORG \$E586 ENDS \$E5A8

LE586: PSHA
LDAB 1,X ; MULT LO BYTE OF 16 BIT NUM
MUL ;
PSHB ; SAVE BOTH BYTES OF RESULT
PSHA

TSY

LDAA 2,Y ; MULTIPLY HI BYTE OF 16 BIT NUM
LDAB 0,X
MUL
ADDB 0,Y
ADCA #0 ; ADD LOW BYTE OF RESULT TO
; HIGH BYTE OF PRIOR RESULT
ROL 1,Y ; MULT BY 2, SHT HI BYTE
; OF PRIOR RESULT LOW BYTE
ROLB ; PRIOR RESULT'S LOW BYTE
ROLA
BCC LE5A2 ; REMAING 7 BITS ARE IN TEMP+2
; ... ELSE

```

        LDD      #$FFFF                ; LIMIT TO MAX
LE5A2:  STD      0,X
        ROL     1,Y                    ; SET/CLR CY
        PULY
        INS

        RTS
*****

*****

*   BIT ON/OFF IN ECU
*
*   Multiple entry subroutine used to tuen on
*   or off a bit in the lower byte of the ECU CR,
*   ECUCR.
*
*   ENTRY:
*       LE5AE: is used to turn off a bit
*
*       LE5BC: is used to turn on a bit
*
*   CALL WITH:
*       A Reg = Bit pattern to be stored
*
*   RETURNS:
*       None
*
*   EXEC TIME:          Cyc
*   CODE LENGTH:       42  Bytes
*   STACK REQ:         4   Bytes
*
*   ORG $E5AC
*****
LE5AC:  LDAA    #$F7                    ; BIT3*, TURN ON CK ENG LAMP
        ;
        ; used to turn off a bit
        ;
LE5AE:  PSHX                    ; SAVE ENTRY WD
        LDX     L3FFC            ; TURN OFF ECU CR BIT
        PSHX
        TSY
        ANDA   1,Y
        BRA    LE5C6

LE5BA:  LDAA    #8                  ; TURN OFF CK ENG LIGHT
        ;
        ; used to turn off a bit
        ;

LE5BC:  PSHX
        LDX     L3FFC            ; TURN OFF ECU CR BIT
        PSHX
        TSY
        ORAA   1,Y

LE5C6:  ANDA   #$FE                ; ECU TEST NORMAL
        ORAA   #2                  ; ENABLE MASTER OUTPUT
        STAA   1,Y
        PULA
        ORAA   #$FB                ; ECU REFRESH

        PSHA

```

```
PULX
STX    L3FFC          ; ECU CR
PULX
```

```
RTS
*****
```

```
*****
* BIT SET
*
*
* TOGGLE ECU BYTE 1
*****
```

```
LE5D7: LDX    L3FFC          ; ECU CR
       PSHX
```

```
TSY
ORAA   0,Y            ; TOGGLE
```

```
BRA    LE5EB
```

```
LE5E2: LDX    L3FFC          ; ECU CR
       PSHX
```

```
TSY
ANDA   0,Y            ; GET ECU ADDR
                          ; RESET A BIT
```

```
LE5EB: STAA   0,Y
       PULX
       STX    L3FFC          ; ECU CR
```

```
RTS
*****
```

```
*****
* MAKE PRESSURE LOAD VAR'S
*
* VAR'S = K(A/D-C)
*   K SCALED TO 1
*
* Call with:   A reg = (A/D-C)
*             B reg = K
*
* Return wuth  A REG = A VAL LIMITED TO 255
*
*****
```

```
LE5F3: BCC    LE5F7          ; MAKE PRESS LD VAR'S
       CLRA          ; ... ELSE
                          ; LOAD = 0

RTS          ; TO CALLER
```

```
LE5F7: MUL
ASLD          ; MULT * 2
BCS    LE601   ; EXIT WITH 255 IN A
          ; .... ELSE
TSTB          ; CK IF Z
BPL    LE603   ; EXIT WITH RESULT IN A

INCA
BNE    LE603   ; IF NO VERFLOW, EXIT WITH RESULT IN A
          ; ... ELSE
```

```

LE601: LDAA    #255                ; LIMIT TO 255

LE603: RTS
*****

*****
* DO SCALED BARO FOR TBL LK UP'S
*
* RETURN WITH RESULT IN A Reg
*     $50 = 65 Kpa
*     $60 = 75
*     $70 = 85
*     $80 = 95
*     $90 = 105 Kpa
*
*   B Reg = $50 Main for Lk Up, ( 65 Kpa)
*
*****

LE604: LDAA    L002E                ; GET A/D BARO COUNTS
      SUBA    #13                    ; SUB OFF OFFSET
      LDAB    #151
      MUL
      ADCA    #0                      ; ROUND
      CMPA    #143                    ; CK IF 104 Kpa
      BLS     LE613                    ; IF LT
                                           ; ..ELSE
      LDAA    #143                    ; LIMIT TO 104 Kpa

LE613: LDAB    #80                    ; MAX FOR LOOK UP 65 Kpa

RTS
*****

*****
*   >>>> END OF UTILITIES <<<<<<
*****

*****
* IAC  PID (Minor loop)
*
* No in's or Outs
*
*****

LE616: LDX     #$8628                ; SET POINTER TO CONTANTS ADDR'S
      ASL     L00FE                    ; PID MD WD #2

;
; Ck PK/NEUT STATUS, (bit 0)
;
      LDAB    L003B                    ; CPU INPUT STATUS WD
      RORB
      ROR     L00FE                    ; PID MD WD #2

      BRSET   L004F,#1,LE632           ; CK FOR ERR 51, BR IF ERR 51
                                           ; ... ELSE
      LDAB    L0036                    ; CK IGN ON STATUS
      BITB    #$10                    ; IGN OFF = 1
      BNE     LE641                    ; IF IGN OFF BYPASS OPN LP & THROT FOLLER
                                           ; ... else
      LDAB    L00FE                    ; PID MD WD #2

```

```

        BITB    #$04                ; B2, MOTOR RESET STARTED
        BNE     LE641              ; BR IF B2
                                   ; IF NO, BYPASS OPN LP & THROT FOLLER
                                   ; ... ELSE
LE632:  LDAB    L0037              ; CK ENGINE STATUS W/MINOR LP MODE WD
        BMI     LE644              ; IF RUNNING GO TO OPN LP & TF MODE

        LDAB    L0038              ; CK ECU MODE
        BITB    #$10              ; B4, IN ADLD (BLINK OUT MODE)
        BEQ     LE641              ; BR IF NOT
                                   ; ... ELSE
        LDAB    #$81              ; IF IN DIAG SET UP FOR MOTOR EXTEND
        STAB    L0111             ; SAVE AS CMD STEPS

LE641:  JMP     LE6FF              ; GO OPERATE MOTER

        ;-----
        ; OPEN LOOP IDLE SPEED
        ;-----
LE644:  LDAB    L0103              ; A/D BATTERY VOLTAGE
        CMPB    #171              ; 17.1 V
        BHI     LE64F              ; BR IF G.T. 17.1 VDC

        CMPB    #90               ; CK 9 VDC
        BHI     LE652              ; IF G.T. 9VDC BR

LE64F:  JMP     LE788              ; DONT MOVE MTR IF VDC OUT OF LIMITs

        ;-----
        ; START UP DELAY
        ;-----
LE652:  CLRA
        BRSET   L00FE, #$08, LE69F ; BR IF B3, START UP DELAY OVER
                                   ; (IAC MODE WD 2)
                                   ; .... else
        LDAB    L0059              ; RPM/12.5
        CMPB    L0112             ; IAC CNT'L SPEED
        BCS     LE664              ; BR IG RPM DEC REQUESTED
                                   ; .... else
        BRSET   L0037, #$80, LE66E ; BR IF B7, 1 = ENGINE RUNNING

        BRA     LE66B

LE664:  BRCLR   L00FE, #$40, LE66B ; BR IF NOT B6, START UP DELAY IN WORK
                                   ; (PID MD WD #2)
        BSET    L00FE, #$08        ; SET B3, START UP DELAY OVER
                                   ; (PID MD WD #2)

LE66B:  JMP     LE6FC

LE66E:  LDAB    L011D              ;
                                   ;
        BSET    L00FE, #$40        ; SET B6, START UP DELAY IN WOR
                                   ; (PID MD WD #2)

;
; If A/C Req, add start up offset
;
        BRSET   L003B, #$80, LE67F ; BR IF B7, A/C REQUEST
                                   ; (CPU INPUT STATUS WD)
                                   ; ...ELSE
        ADDB    $862A             ; STARTUP OFFSET CALIB

```

```

        BCC      LE67F          ; IF NO OVERFLOW FN ADDER
                                ; .... else
        LDAB     #255          ; LIMIT TO 255
LE67F: JSR      LC01F          ; GO DO PARK DN STEPS
                                ;
        CLRA                      ; A = 0
        CMPB    L0119          ; B HAS DESIRED VALUE
        BCS     LE690          ; GO IF NOT YET AT PARK POSIT
                                ; .... ELSE
        LDAA    $0114          ; GET DELAY CNTR VALUE
        BRCLR   $0000,#1,LE690 ; BR IF NOT 12.5 MSEC
                                ;
LE690: INCA                      ; UP DATE DELAY
        CMPA    7,X            ; CURRENT DELAY TO MAX DELAY
        BLS     LE6FC          ; IF DELAY NOT LONG ENOUGH,
                                ; BY PASS OPEN LOOP
                                ;
        LDAB    L00FE          ; PID MD WD #2
        ORAB    #$08           ; B3, START UP DELAY OVER
        STAB    L00FE          ; PID MD WD #2
                                ;
        CLR     L0114          ; RESET DELAY CNT'R AFTER START
                                ; UP DELAY
                                ;
        BRA     LE6C2          ;
                                ;
;-----
LE69F: BRSET    L00FE,$$20,LE6FF ; BR IF B5, PID CMD STEP ACTIVE
                                ; (PID MD WD #2)
                                ; ... ELSE
        LDAB    L00FD          ; PID MD WD #1
        BITB    #$10           ; CK FOR THROT FOLLOW MODE, (BIT 4)
        BNE     LE6B1          ; BR IF IN TF MODE & DO STEP
                                ; ... ELSE
        BITB    #$08           ; B3, START UP DELAY OVER
        BNE     LE6FF          ; BR IF B3
                                ; ... ELSE
        BITB    #$04           ; B2, MOTOR RESET STARTED
        BNE     LE6C2          ; BR IF B2
                                ; ... ELSE
LE6B1: LDAA    L0111          ; ARE STEPS REQUESTED ?
        BPL     LE6C2          ; IF YES, BR TO TRAJ DECAY
                                ; ... ELSE
                                ;
        ; TRAJECTORY DELAY
        ;
        BITB    $$20           ; B5,PID CMD STEP ACTIVE
                                ; (A/C STEPS BEING REMOVED ?)
        BNE     LE6C2          ; BR IF B5
                                ; IF SO, BYPASS DECAY
                                ; ... ELSE
        DEC     L0114          ; DECR DECAY STEP TIMER
        BEQ     LE6C2          ; IF DECAY DONE, STEP MOTOR
                                ; ... ELSE
        JMP     LE7BC          ; ELSE RETURN WITH NO STEP
                                ;
LE6C2: LDAA    $31,X          ; GET NEUT DELAY
                                ;
        LDAB    L00FE          ; PID MD WD #2,
        BMI     LE6DA          ; BR IF NOT IN DRIVE
                                ;
        LDAA    L0081          ; GET FILT MPH
        CMPA    #32           ; LIMIT TO 32

```



```

BCS      LE6D0                ;
; .... ELSE
LDAA     #255                ; LIMIT TO 255

LE6D0:   ASLA                  ; MULT * 8
ASLA
ASLA
LDAB     $33,X              ; GET DRIVE MULT
MUL
;
ADDA     L0032,X           ; ADD TO DRIVE DELAY
BCS      LE6F7              ;
; .... ELSE
LE6DA:   LDAB     L00FD      ; PID MD WD #1
BITB     #$04              ; CK FOR OPN LOOP MODE, BIT 2
BNE      LE6FC              ; BR IF NOT AROUND OPN LP CALC'S
; ... ELSE
;
ASLA
BCS      LE6F7              ;
; .... ELSE
PSHA
LDAB     L0059              ; GET RPM/12.5
SUBB     L0112              ; SUB DESIRED IAC CNT'L SPEED
CLRA
ASLD
ASLD
TSTA
BNE      LE6FB              ;
; BY PASS INCR OF DLY CNT'R IF RPM > Max
; ... ELSE
;-----
; INCR DELAY CNT'R
;-----
COMB
LDAA     $34,X              ; MAX DIFF COUNT'S
MUL
;
PULB
; GET BASE COUNT
ABA
; ADD DIFF TO BASE COUNT
BCC      LE6FC              ; IF NO OVERFLOW, EXIT W/SAVE
; .... ELSE
LE6F7:   LDAA     #255        ; LIMIT TO MAX NUM MNR LPS/STEP
BRA      LE6FC              ; EXIT W/SAVE

LE6FB:   PULA                  ; RESTORE BASE DECAY

LE6FC:   STAA     L0114        ; SAVE

;-----
; MOVE MOTOR
;
; A Req has MCU I/O
; B Reg use as GP reg
;-----
LE6FF:   BCLR     L00FE,#$01   ; CLR B0, SLEW OVERIDED LEARN, DONT LRN
; (PID MD WD #2)

;-----
; CK FOR HEADS UP IAC MOD'ER
;
;-----
LDAA     L0036              ; CK IF HUD ON LINE
BPL      LE709              ; IF NO, SKIP HUD, TRY ALDL MD 4
. ... else
JSR      L5809              ; IAC FM HUD
;-----
;-----

```

```

; ALDL MODE , CONTROLING
;
; CNT'L WD L0198, b0, 1 = CONT'LING IAC
;
; CNT'L WD L0198, b1, 1 = RPM or STEPS
; CNT'L WD L0199, IAC MOD VALUE
;
;
;-----
LE709: BRCLR   L0048,#$01,LE72E      ; BR  IF NOT IN ALDL MODE 4

      LDAA    L0198                  ; CNT'L WD
      BITA    #$01                   ; B0, 1 = CONT'LING IAC
      BEQ     LE72E                   ; BR IF NOT b0
      ; ... else
      BITA    #$02                   ; B0,
      BNE     LE72E                   ; BR IF B0
      ; ... else
      LDAA    L002B                   ; IAC CURRENT POSITION
      SUBA    L0199                   ; CNT'L WD IAC DESIRED POSIT / RPM
      BCC     LE720
      ; .... else

      NEGA

LE720: BPL     LE724

      LDAA    #L007F

LE724: BCS     LE728
      ; .... else
      ORAA    #$80                   ; SET B7,
LE728: STAA    L0111                 ; CMD STEPS

      BSET    L00FE,#$01             ; B0, SLEW OVERIDED LEARN, DONT LRN
      ; (PID MD WD #2)
;-----

;-----
;
;
;-----
LE72E: LDAB    L0111                 ; GET NUM OF CMD STEPS
      ASLB
      BEQ     LE761
      ; ... ELSe
      DEC     L0111                   ; CMD STEPS
      LDAB    L002B                   ; IAC CURRENT POSITION
      BCC     LE743
      ; .... ELSe

;-----
; EXTEND PINTEL
; less air to engine
;-----
      BEQ     LE73E                   ; IF Z, SAVE CURRENT STATE
      ; ... ELSe
      DECB
      ; ADJ CURRENT POSIt

LE73E: DEC     L011A                 ; DEC RUNNING CNT'r
      BRA     LE74C                   ; & UP DATE A/C & T.F. FLAGS

;-----
; RETRACT PINTEL
; more air to engine
;-----

```

```

LE743:  CMPB    $42,X          ; LIMIT POSIT TO MAX
        BCC     LE761         ;
                                     ; .... ELSe
        INCB                    ; IF LT MAX, UP DATE POSIT
        INC     L011A         ; INCR RUNNING CNT'r
        CLC                                     ;

;-----
; UPDATE A/C LEARN &
; THROT FOLLOWER
;-----

LE74C:  STAB    L002B         ; NEW PRESENT MOTOR POSIt
        LDAB    L00FD         ; PID MD WD #1
        BITB    #$10         ; CK THROT FOLLOWER ACTIVE, (BIT 4)
        BEQ     LE761         ; IN NOT, CK WARMUP & FAn
                                     ; ... ELSe
        LDAB    L0115         ; GET THROT FOLLOW LEARN VALUe
        INCB                    ; MAKE CURRENT STEP IS RETRACT
        BCC     LE75E         ; IF RETRACT, BYPASS DEc
                                     ; .... ELSe
        DECB                    ; ORIG TF DIFF VALUe
        BEQ     LE75E         ; IF ORIG IS = 0, BYPASS DEc
                                     ; ... ELSe
        DECB                    ; DEC ACTUAL TF DIFF

LE75E:  STAB    L0115         ; UPDATE THROT FOLLOW LEARN VALUe

LE761:  LDAB    L00FD         ; PID MD WD #1
        BITB    #8           ; BIT 3,
        BEQ     LE76A         ;
                                     ; ... ELSe
        INC     $011F         ;

LE76A:  LDAB    L002B         ; IAC CURRENT POSITION
        CMPB    $42,X         ;
        BLS                    ;

        LDAB    $0111         ;
        BPL                    ;

;-----
; MOVE MOTOR 1 STEP
;-----

LE775:  LDAB    L011A         ; GET RUNNING CNT'R
        LDX     #$F127        ; STEP FORMAT TABLE

        ANDB    #$03         ; MASK FOR 2 LSB's

        LDAA    $4002         ; CPU DATA LATCH
        ANDA    #$FC         ; MASK OFF A & B MOTOR POS BITs

        ABX                    ; ADDED TO NEW BASE ADDR

        ORAA    0,X          ; INSERT NEW ONE's
        STAA    $4002        ; CPU DATA LATCH

;-----
; BASE BIAS
;-----

LE788:  LDAB    L0119         ;
        ADDB    L0115         ; TPS FOLLOWER DIFF
        BCC     LE792         ;
                                     ; .... ELSe
        LDAB    #$FF         ; LIMIT TO 255

LE792:  PSHB                    ;

```

```

LDAB    $011F          ;
LDAA    L003C          ;
ASLA                    ;
PULA                    ;
BPL     LE79D          ;

NEGB                    ;

LE79D:  ABA            ;
        TAB            ;
LDAA    L002B          ; IAC CURRENT POSITION
SBA                    ;
STAA    $0120          ;
LDAB    L002B          ; IAC CURRENT POSITION
SUBB    $0120          ; CK SIGN OF UNK
LDAB    #128           ; LAST BIAS TO +/-128
BCC     LE7B4          ;
        ; .... ELSe
CBA                    ; SIGN IS NEG, (CY SET)
BHI     LE7B9          ; BR IF BIAS IS IN LIMITs

INCB                    ; ELSE LIMIT TO 128

BRA     LE7B8          ;

LE7B4:  CBA            ; SIGN IS POS, (CY CLR)
        BCS     LE7B9  ; BR IF BIAS IN LIMITs
        ; .... ELSe
        DECB         ; ELSE LIMIT TO 128

LE7B8:  TBA            ;
LE7B9:  STAA    L0120  ; ASJUST
*****

*****
* INJECT.SRC
*
* THESE OP'S DONE EACH INJECT:
*
*     1. CK CYL SEL (AND ERR CODE)
*     2. DECAY'S IAC TIME OUT
*     3. DECAY AFR IF RUNNING
*     3. TOGGLE SINGLE FIRE F/F
*     4. TOGGLE BOOKKEEP F/F
*     5. UPDATE QUM FUEL
*     6. SET MAP ACCEL ENR HAPPENED FLAG
*     8. SET TPS ACELL ERN LIMIT
*     9. SET ALLOW SINGLE FIRE DISABLE FLAG
*    10. INCREMENT DECEL INJ WHEN IN DECEL ENLEAN
*    11. MAKE UP CCM ENGINE REV WD FOR CCM USE
*
*****

LE7BC:  BRCLR    L00C0,#$40,LE7E3  ; BR IF NOT b6, INJ HAPENED
        ; (OLD CPU STATUS)
        ; ... else
BRSET   L0048,#$02,LE7D0          ; BR IF b1, IN ALDL MODE 4
        ; ... else
LDAA    L01B8          ; GET OLD CIM RPM
INCA                    ; INCR EVERY OTHER INJECT
STAA    L01B8          ; UPDATE GET OLD CIM RPM

BSET    L0048,#$02      ; SET CCM ENG REV WD FLP/FLP'ER BIT

BRA     LE7D3

```

```

LE7D0:  BCLR      L0048,#$02          ; CLR  b1, CCM ENG REV WD FLP/FLP'ER BIT
;
; CK FOR CYC SEL ERROR
;
LE7D3:  LDAB      L0032          ; FMD BYTE 2, FROM RES PACK
        ANDB     #$18          ; MASK FOR BIT'S 3 & 4
        CMPB     $8293        ; 0d, ERROR 41, cyl select
        BEQ      LE7DF        ; BR IF MODE BITS OK
; ... else
        BSET     L0041,#1      ; SET ERR 41 BIT

LE7DF:  BRSET    L0037,#$80,LE7E6    ; BR IF b7, ENG RUNNING
;
LE7E3:  JMP      LE8F9          ; BR IF ENG RUNNING
;
; DECEL ENLEAN COUNTER
;
LE7E6:  CLR      CLR      CLR      CLR      CLR      CLR      CLR      CLR
;
        BRCLR   L003F,#$80,LE7F2    ; BR IF NOT IN DECEL ENLEAN
; ... else
        LDAB    L00AF          ; GET DECEL ENLEAN INJ CNT'R
        INCB    INCB          ; INCR DECEL ENLEAN INJ CNT'R
        CMPB    #16          ; LIMIT COUNT TO 17
        BHI     LE7F4          ;
;
LE7F2:  STAB     L00AF          ; DECEL ENLEAN INJ CNT'R
;
; DECAY'S IAC TIME OUT
;
LE7F4:  BRCLR   L00FE,#$08,LE815    ; BR IF NOT b3, START UP DELAY OVER
; (IAC MD WD 2)
; ... else
        TST     L01B1          ; Idle spd Time out steps
        BEQ     LE815          ; BR IF TIME UP
; ... else
        LDAA    L01B2          ; DELAY BETWEEN DECAYS
        BEQ     LE808          ;
; ... else
;
        DECA    DECA          ;
        STAA    L01B2          ; DELAY BETWEEN DECAYS
;
        BRA     LE815          ;

LE808:  LDAA    L01B1          ; Idle spd Time out steps
        DECA    DECA          ; DEC STEPS
        STAA    L01B1          ; Idle spd Time out steps
;
        LDAA    $869E          ; 4 INJ'S, IDLE SPEED INJECT
; TIME OUT DECAY RATE
        STAA    L01B2          ; DELAY BETWEEN DECAYS

LE815:  BRCLR   L000D,#$FF,LE850    ; BR IF NOT b0->7, TIME OUT AFR
;
; CK INITAL CHOKE DELAY
;
        LDD     L000F          ; INITAL CHOKE DELAY
        BEQ     LE824          ; BR IF Z

```

```

; ... else
SUBD    #01
STD     L000F      ; INITIAL CHOKE DELAY

BRA     LE850

;
; CK TIME OUT A/F CNT'R
;
LE824:  LDAA    L00EF      ; TIME OUT AF CNT'R
        BEQ     LE82D      ; BR IF Z
        ; ... else
        DECA    ; DECR CNT'R
        STAA    L00EF      ; TIME OUT AF CNT'R

        BRA     LE850

;
; LK UP START UP ENRICH %CHG vs TEMP
;
LE82D:  LDAA    L0060      ; START UP COOLANT
        CMPA    #208      ; 116 Deg C
        BLS     LE835      ; BR IF COOL LT 116c
        ; ... else
        LDAA    #208      ; 116 Deg C LIMIT

LE835:  LDAB    $841E      ; 4 INJ'S, START UP INJ DECAY RATE
        STAB    L00EF      ; TIME OUT AF CNT'R

;-----
; START UP ENRICHMENT DECAY DECAY AMOUNT vs COOLANT
;
; TBL = 6.5536 * %CHG
;-----
        LDX     #$844F      ; INDEX START UP ENRICH, %CHG vs TEMP
        JSR     LE3D0      ; 2d LOOK UP, NO OFF SET

        PSHA
        LDD     L000D      ; TIME OUT AFR
        TSX
        SUBB    0,X
        SBCA    #0
        BCC     LE84D
        ; ... else
        LDD     #0         ; ROUND

LE84D:  STD     L000D      ; TIME OUT AFR
        PULA

LE850:  BRSET   L0039,#$20,LE8AE ; BR IF b5, IN Stall Saver SPK BYPASS
        LDD     L0014
        BEQ     LE85F
        ; ... else
        SUBD    #1
        STD     $0014

        BRA     LE8AE

LE85F:  BRCLR   L003C,#$80,LE86C ; BR IF NOT b7
        LDAA    $0011
        BEQ     LE86C
        ; ... else
        DECA
        STAA    $0011

        BRA     LE8AE

```

```

LE86C:  BSET    L003C, #\$80      ; SET b7
        LDAA    \$0060          ; START UP COOLANT
        CMPA    #208           ; 116 Deg C
        BCS     LE877
        LDAA    #208           ; .... else
        LDAA    #208           ; 116 Deg C LIMIT

LE877:  LDAB    \$81B8
        STAB    \$0011

;-----
; LK UP START UP SA DECAY vs START UP COOLANT
; 14 LINES
;-----
        LDX     #\$81D7        ; INDEX START UP SA vs COOL
        JSR     LE3D0          ; 2D LOOK UP, NO OFF SET

        PSHA
        TSX
        LDD     L0012          ; START UP vs COOLANT SPK ADV
        SUBB    0,X
        SBCA    #\$0000
        BCC     LE88F
        LDD     #0
        LDD     #0
        LE88F: STD     L0012          ; START UP vs COOLANT SPK ADV
        PULA
        BNE     LE8AE
        BRCLR   L0002, #1, LE8AA ; BR IF NIT b0, ERR 43B TST NOT IN WORK
        LDAA    L01A7          ; ... else
        ADDA    \$829B          ; SPK ADV FOR DIAG, ESC ERR 43 TST
        BCS     LE8A5          ; Add 0.3 deg SA for test
        CMPA    \$829E          ; .... else
        BLS     LE8AB          ; 24.9 deg max S.A. for ERR 43B TST
        LE8A5: LDAA    \$829E          ; ... else
        BRA     LE8AB
        LE8AA: CLRA
        LE8AB: STAA    L01A7          ; SPK ADV FO DIAG, ERR 43 TST

;-----
; TOGGLE SINGLE FIRE F/F
;-----
LE8AE:  LDAA    L0047          ; AIR FUEL MODE WD
        EORA    #\$04          ; b2, B2 1 = DELIVER 0 FUEL, SINGLE FIRE
        STAA    L0047
        BRCLR   L0047, #4, LE8BB ; BR NOT b2,
        BSET    L0047, #\$08    ; B2 1 = DELIVER 0 FUEL, SINGLE FIRE
        BSET    L0047, #\$08    ; ... else
        BSET    L0047, #\$08    ; SET b3,
        BSET    L0047, #\$08    ; B3 1 = ALLOW SINGLE FINE DISABLE

;-----
; TOGGLE BOOKKEEP F/F
;-----
LE8BB:  LDAA    L003E          ; AIR MODE WD

```

```

EORA    #1                ; BIT 0
STAA    L003E            ; AIR MODE WD
;-----

BSET    L0045,#$14       ; SET b4 & b2
                        ; B2 1 = INJ IN LAST 6.25 Msec
                        ; B4 1 = USE LO ORDER NIBBLE OF
                        ;           FUEL LIMIT TBL

LDX     L00E2            ; Base Pulse width
STX     L00E4            ; OLD Base Pulse width

LDD     L012E+1          ;
ADDD    $0132            ;
BCC     LE8D3            ;
                        ; .... else
LE8D3:  INC     L012D      ; RUN TOTAL DELIVERED FUEL, (COUNTS)
        STD     L012E+1    ;

BRCLR   L0037,#$80,LE8E2 ; BR IF NOT b7, 1 = ENGINE RUNNING

INC     L01A4            ; INC INJ's CTR SINCE START UP
BNE     LE8E2            ; BR IF NO OVERFLOW
                        ; ... else
DEC     L01A4            ; DEC INJ's CTR SINCE START UP

LE8E2:  BRCLR   L0045,#$40,LE8EE ; BR IF NOT b6
                        ; B6 1 = TPS ACCEL ENRICH LIMITING EXT ON

LDAB    L00A4            ; EXTENDED A/F LIMITING
INCB    ;
CMPB    $852F            ; 1 INJ IN FUEL LIMITING
BCS     LE8F7            ;
                        ; .... else
LE8EE:  BCLR    L0045,#$40 ; B6 1 = TPS ACCEL ENRICH LIMITING EXT ON

CLRB    ;                ; CLR Acell Enr COUNTER
STAB    L00A2            ; Acell Enr COUNTER

CLRA    ;                ; CLR
STD     L00F8            ; ACCUM Acell Enr FUEL

LE8F7:  STAB    L00A4      ; EXTENDED A/F LIMITING

LE8F9   RTS

*****
* FUEL OUTPUT
*
*
*****

;
; MAP Acell Enr FUEL DELIVERY
;
LE8FA:  LDAA    L00A5      ; MAP Acell Enr SYNC FUEL MULT
                        ;
BRCLR   L0045,#$02,LE90E ; BR IF NOT b1, MAP ACCEL ENRICH
                        ; ... DONE 1ST TIME
                        ; .... else
BRCLR   L0045,#$04,LE90E ; BR IF NOT b2, INJ IN LAST 6.25 Msec
                        ; ... else

```



```

SUBA      L00A8      ;
BCC       LE90C     ; BR IF EQ LT 0
           ; .... else
CLRA      ; LIMIT TO 0
BCLR     L0045,#$08 ; CLR b3, MAP ACCEL ENRICH ON
           ;
LE90C    STAA      L00A5 ; MAP Acell Enr SYNC FUEL MULT
           ;
LE90E    BCLR     L0045,$$04 ; CLR b2, INJ IN LAST 6.25 Msec
BSET     L0045,$$02 ; SET b1, MAP ACCEL ENRICH DONE
           ; .... 1ST TIME
           ;
LDX      L00E2     ; Base Pulse width
JSR     LE33F     ; MULT 8 * 16,
           ;
ADDD     L00E2     ; ADD TO Base Pulse width
BCC     LE921     ; IF NO OVER FLOW
           ; .... else
LDD     #$FFFF    ; MAX LIMIT
LE921:   STD      L00E2 ; Base Pulse width
           ;
;
; OUTPUT BLOCK
;
BRCLR    L0003,$$20,LE94C ; BR IF MOT b5, VATS PASS
           ; .... else
BRSET    L0036,$$10,LE94C ; BR IF b4, IGN OFF
           ; ... else
BRSET    L0046,$$02,LE94C ; BR IF b1, DECEL FUEL C/O ENABLED
           ; ... else
;
; FUEL CUT OFF
;
LDX     #$83F4    ; MPH FOR FUEL CUT OFF
LDAA    L0081     ; FILT MPH
           ;
BRCLR    L0046,$$40,LE93B ; BR IF NOT b6, o2 RICH
           ; ... else
LDX     #$83F7    ; MPH FOR FUEL BACK ON
           ;
LE93B:   BCLR     L0046,$$40 ; CLR b6, o2 RICH
           ;
CMPA    0,X      ; MPH GT THRESH
BHI     LE949    ; BR IF YES
           ; ... else
LDD     L00B3    ; CURRENT MNR LOOP DRP PERIOD
CPD     1,X      ; RPM GT THRESH
BCC     LE953    ; BR IF NOT
           ; .... else
LE949:   BSET     L0046,$$40 ; SET b6, o2 RICH
           ;
;-----
; ZERO BPW IF VATS FAIL
; ZERO BPW IF IGN OFF
; ZERO BPW IF DECEL C/O
;-----
LE94C:   LDX      #0 ; ZERO PW
STX     L00F8     ; SET Acell Enr FUEL = 0
STX     L00E2     ; SET BPW FUEL = 0
           ;
LE953    BRCLR    L0045,$$C0,LE9AA ; BR IF NOT b7 & b6
           ; B7 1 = TPS ACCEL ENRICH ON
           ; B6 1 = TPS ACCEL ENRICH LIMITING EXT ON

```

```

; ... else
LDX      #$8563      ; FUEL LIMIT FACTOR vs COOL TBL
;
LDAB     L005D      ; FLT COOLANT TEMP
LSRB     LSRB      ; SCALE COOLANT DIV by 32
LSRB     LSRB
LSRB     LSRB
LSRB     LSRB
ABX      ; ADD COOL FOR INDEX LK UP
LDAB     0,X        ; GET SELECTED FUEL LIMIT FACTOR
ASLD     ; SWAP NIBBLES
ASLD
ASLD
ASLD
ANDB     #$F0      ; MASK FOR UPPER NIBBLE
LDAA     0,X        ; FUEL LIMIT FACTOR
ANDA     #$F0      ; MASK FOR UPPER NIBBLE
;
BRCLR    L0045,$$10,LE973 ; BR IF NOT b4
; B4 1 = USE LO ORDER NIBBLE OF
; FUEL LIMIT TBL
; ... else
;
LE973    TBA
LDX      #$L00E2   ; Base Pulse width
JSR      LE933F    ; 8 * 16 MUL,
; FUEL LIMIT = FUEL FACT * BPW
;
ASLD     ;
BCC      LE97F     ; IF NO OVER FLOW
; .... else
LDD      #$FFFF   ; USE MAX LIMIT
;
LE97F    PSHB
PSHA
;
LDD      L00E4     ; OLD BASE PULSE WIDTH
ADDD     L00F8     ; ACCUM Acell Enr FUEL
BCC      LE98A     ; IF NO OVER FLOW
; .... else
LDD      #$FFFF   ; MAX LIMIT
;
LE98A    TSX
SUBD     0,X
PULX
BCS      LE9AA
; .... else
;
PSHB
PSHA
TSX
LDD      L00E2     ; Base Pulse width
SUBD     0,X
PULX
BCS      LE9A0
; .... else
; 0 MSEC MIN BPW
CPD      $83E9
BCC      LE9A2
; .... else
;
LE9A0    CLRA
CLRB
;
LE9A2    PSHB
PSHA
PULX
STX      L0465

```

```

        BRA        LE9C0

LE9AA   LDX        L00E2           ; Base Pulse width
        STX        L0465

        ;-----
        ; SINGLE FIRE ENABLE-DISABLE
        ; & FUEL CONTROL
        ;-----
        LDD        L009B
        BNE        LE9C0

        BRCLR     L0047,#$01,LE9DA ; ... else
                                           ; BR IF NOT b0, IN SINGLE FIRE MODE
        ; ... else
        CPX        $83EB           ; 1.1 Ms SINGLE FIRE BPW DISABLE THRESH
        BLS        LE9C5           ; BR IF PW =< 1.1 MSEC
        BRCLR     L0047,#$08,LE9C5 ; ... else
                                           ; BR IF NOT b3, ALLOW SINGLE FINE DISABLE
        ; ... else
LE9C0   BCLR      L0047,#$01       ; clr b0, IN SINGLE FIRE MODE

        BRA        LE9E9

LE9C5   PSHX
        PULA
        PULB
        ASLD
        BCC        LE9CE           ; .... else

        LDD        #$FFFF

LE9CE   PSHB
        PSHA
        PULX
        BRCLR     L0047,#$04,LE9E9 ; BR IF NOT b2, DELIVER 0 FUEL,
                                           ; .. SINGLE FIRE
        ; ... else

        LDX        #$0000
        BRA        LE9E9

LE9DA   BRSET     L003F,#$80,LE9E9 ; BR IF b7, DECEL ENLEAN ACTIVE
        ; ... else
        CPX        $83ED           ; 0.85 Ms, SINGLE FIRE BPW
        ; ENABLE THRESH

        BCC        LE9E9

        ; .... else
        BSET      L0047,#$01       ; SET b0, IN SINGLE FIRE MODE

        BCLR     L0047,#$04       ; CLR b2
        ; B2 1 = DELIVER 0 FUEL, SINGLE FIRE
        ;
LE9E9   BCLR      L0047,#$08       ; CLR b3, ALLOW SINGLE FIRE DISABLE
        ;
        STX        L0132           ; SAVE BPW
        BEQ        LEA0A           ; BR IF Z
        ; ... else
        CPX        #256            ; 3.90 Msec, (Low BPW bypass thresh)
        BCC        LEA06           ; BR IF BPW GT 4.90 Msec, (SKIP LOW PW CORR)
        ; ... else

        PSHX
        PULA
        ;

```

```

        PULA                                ;

;-----
; LK UP LOW PULSE WIDTH INJECTOR OFFSET
;   15 line tbl
;
;   TBL = MSEC * 65.536
;-----
        LDAB    #32                        ; MIN VAL FOR TBL LOOKUP, (0.488 Msec)
        LDX     #$840B                      ; INDEX LOW PW INJ OFFSET vs PW
        JSR     LE3CC                       ; 2d LK UP
;
        TAB     ; LK UP RESULTS TO B Reg
;
        LDX     L0132                       ; GET BPW
        ABX     ; ADD IN LOW PW BPW OFFSET
;
LEA06:  LDAB    L00EE                       ; BATT BPW CORRECTION
        ABX     ; ADD TO BPW (twice)
        ABX     ;
;
;   SIMULTANIOUS FUEL OUT
;
LEA0A:  CPX     #$7FFF                      ; 499.9 Msec MAX BPW
        BCS     LEA12                      ; BR IF GT MIN
;   .... else
        LDX     #$7FFF                      ; FORCE 499 Msec
;
LEA12:  STX     L3FD0                      ; OUT INJ PW TO ECU <----<<<
;
        BRSET   L0047,$#05,LEA1C          ; BR IF b0 & b2
;           b0 IN SINGLE FIRE MODE
;           b2 DELIVER 0 FUEL, SINGLE FIRE
;   ... else
;
        STX     L0463                      ; BPW CORRECTED HU USR RAM
;
LEA1C:  RTS                                ;

*****
; END OF FUEL OUT

*****
* 8192 BAUD SERIAL DATA CODE
*
*
*****
LEA1D:  LDAB    L4008                      ; CPU TX/RX STAT REG, (8192 BAUD SCI)
        LDAA   L4009                      ; CPU RX Data Reg, (8192 BAUD SCI)
;
        LDX     L014C                      ; SCI DATA MSG TBL INDEX
        CLR    L019E
;
; CK RX STAT REG FOR ERRORS
; EXIT IF ERROR
;
        BITB   #$0E                        ; 1110 0000
        BNE   LEAA5                       ; BR IF SCI ERROR, Frame, Ovr Run, ETC.
;   ... else
        TAB     ; RX DATA TO B Reg
        ADDB   L014B                       ; RUNNING CK SUM ON SPI RX DATA
        STAB   L014B                       ; SAVE NEW SCI DATA CHECKSUM

```

```

LDAB    L014A           ; SCI DATA BYTE COUNTER
BNE     LEA57           ; BR IF NZ
; ... else

;
; CIM MESSAGE, ID $40
;
LDX     #$8882         ; SERIAL DATA RX CIM DEVICE ID 40

LEA3C:  CMPA    2,X     ; $40
BEQ     LEA46         ; IF MATCH
; ... else

LDX     0,X           ; GET NEXT ENTRY ADDRESS FM ROM
BNE     LEA3C         ; BR IF NEXT ADDRESS IS NZ
; ... else

BRA     LEAA5         ; ERROR, DEVICE ID NOT FOUND, WAIT FOR NEXT

LEA46:  CMPA    #$F4    ; CK IF ALDL MSG HEADER
BNE     LEA4D         ; BR IF NOT
; ... else

BSET    L0038,$$20    ; Set b5, IN ALDL MODE

LEA4D:  STX     L014C   ; SCI DATA MSG TBL

LDAA    #$25         ; SCI MODE
STAA    L4007        ; CPU TX/RX CNT'L REG, (8192 BAUD SCI)
BRA     LEA9A

LEA57:  DECB           ; DECR
BNE     LEA63         ; CK FOR ALDL MODE NUMBER
; ... else

SUBA    #$55         ; SUB OFF MSG LENGTH BIAS
BCS     LEAA5         ; ERROR, NEGITIVE MSG LENGTH
; .... else

STAA    L019D        ; SERIAL DATA MSG LENGTH

BRA     LEA9A

;-----
;
;-----
; CHECK OUT RX'ED ALDL MODE COMMAND
; MODE 0 th 4, 7 & 10
;
;-----
LEA63:  DECB           ; DECR BYTE CNT
CMPB    L019D        ; SERIAL DATA MSG LENGTH
BCC     LEAA0         ; BR IF LENGTH NZ
; .... else

TSTB
BNE     LEA95         ; ... else

BRCLR   L0038,$$20,LEA95 ; BR IF NOT b5, (IN ALDL MODE)

;-----
; MODE BYTE IN ALDL BYTE 3,
; CALC ADDRES OF CORRECT MODE TBL
;-----
LDX     #$8978       ; INDEX TO DEVICE $F4 TBL POINTER
; MESSAGE ID $4F MODE 0 th MODE 10
;
CMPA    #10          ; MODE 10 ?
BHI     LEAA5         ; BR IF GT MODE 10, ERR <--**
; ... else

BEQ     LEA90         ; BR IF MODE 10 ALDL

```

```

        CMPA    #7
        BHI    LEAA5
        BEQ    LEA8C
        CMPA    #4
        BHI    LEAA5
        PSHB
        TAB
        ASLB
        ABX
        PULB
        LDX    0,X
        BRA    LEA92

;
; MODE 7 INDEX
;
LEA8C:  LDX    10,X
        BRA    LEA92

;
; MODE 10 INDEX
;
LEA90:  LDX    12,X

LEA92:  STX    L014C

LEA95:  LDX    7,X
        ABX
        STAA   0,X

LEA9A:  INC    L014A
        JMP    LEB92

;
; CKSUM BYTE
;
LEAA0:  LDAB   L014B
        BEQ    LEAA8

LEAA5:  JMP    LEB84

;-----
; HOUSE KEEPING CODE
;
;-----

;-----
; CHECK CCM MODE
; MSG $4x
;-----
LEAA8:  BCLR   L0038,$08
        LDAA  2,X
        STAA  L019C

```

```

; ... else
; MODE 7 ALDL ?
; BR IF GT MODE 7, ERR <--**
; ... else
; BR IF MODE 7
; ... else
; MODE 4 ?
; BR IF MODE 5 or 6, ERR <--**
; ... else
; SAVE BYTE COUNT TO STX

; MODE NUM TO B Reg
;
; ADD MODE NUM TO ADDR FOR ADDR LK UP

; GET BYTE COUNT FROM STX
; GET RX'ED DATA MSG ADDRESS

; GO SAVE ADDRESS

```

```

; GET MODE 7 MESSAGE TBL POINTER
; GO SAVE ADDRESS

```

```

; GET MODE 10 MESSAGE TBL POINTER

```

```

; SCI DATA MSG TBL INDEX

```

```

; GET BUFFER ADDRESS
; CALC INDEX
; SAVE DATA IN INPUT BUFFER

```

```

; SCI DATA BYTE COUNTER

```

```

; SCI DATA CHECKSUM
; BR IF Z, CK SUM OK
; ... else

```

```

; CLR b3, IN CCM MODE

```

```

; GET DEVICE ID CODE
; SERIAL DATA DEVICE CODE

```

```

        ANDA    #$F0                ; Mask for LO NIBBLE
        CMPA    #$40                ; CK FOR CIM DEVICE ID
        BNE     LEAB9              ; BR IF NOT $40
                                   ; ... else
        BSET    L0038,$$08         ; SET b3, IN CCM MODE
;-----

```

```

;-----
; XFER RX'ED MESSAGE TO AL SO ICB BUFFER IF GOOD MSG
;
; XFER DATA FROM L014E th L016E to L0170 th L0190, (32 WD'S)
;
; $014E = TOP OF DATA OCB/ICB
;-----

```

```

LEAB9:  LDAA    L019D                ; SERIAL DATA MSG LENGTH
        BEQ    LEB24              ; BR IF Z
                                   ; ... else

        LDD    L014E              ; SERIAL DATA OCB
        STD    L0170              ; SERIAL DATA ALDL MODE WD (0-10)

        LDD    L0150
        STD    L0172

        LDD    L0152
        STD    L0174

        LDD    L0154
        STD    L0176

        LDD    L0156
        STD    L0178

        LDD    L0158
        STD    L017A

        LDD    L015A
        STD    L017C

        LDD    L015C
        STD    L017E

        LDD    L015E
        STD    L0180

        LDD    L0160
        STD    L0182

        LDD    L0162
        STD    L0184

        LDD    L0164
        STD    L0186

        LDD    L0166
        STD    L0188

        LDD    L0168
        STD    L018A

        LDD    L016A
        STD    L018C

        LDD    L016C
        STD    L018E

```



```

LEB5C  STAA    L400A           ; CPU TX DR, (SCI) TX CKSUM VAL
      STAA    L014B           ; SCI DATA CHECKSUM

      LDAB    #1
      STAB    L014A           ; SCI DATA BYTE COUNTER

;-----
; ENABLE SXR XMITTER
;
;-----
      LDX     #$4000           ; INDEX CPU REG'S

;
; SELECT
;
      BCLR    2,X,#$80         ; CLR b7 $4002,
      BSET    $32,X,#$01      ; SET b0 $4032
                                ; SCI STATUS, 1 = TX ENAB)
                                ;
      LDAA    L0032           ; ENABLE SXR TX
      JSR     LFOC0           ; XMIT ON SERIAL LINE

;
; DESELECT
;
      BSET    2,X,#$80         ; SER b7 $4002,

;
; FORCE IDLE BYTE TX
;
      LDAA    #$81            ; 1000 0001,
      STAA    L4007           ; CPU TX/RX CNT'L REG, (8192 BAUD SCI)

      LDAA    #$89            ; 1000 1001,
      STAA    L4007           ; CPU TX/RX CNT'L REG, (8192 BAUD SCI) ; CPU TX/R
      BRA     LEB92

;-----
; BAD MESSAGE
;
;-----
LEB84: LDD     #$0000
      STD     L014C           ; CLR SCI DATA MSG TBL INDEX
      STD     L014A           ; CLR SCI DATA BYTE COUNTER

;
; Enabel 8192 SCI Rx interupts, $4007
; & WAIT FOR NEXT MESSAGE
;
; b0 CNT'R INTERUPT
; b1 RX WAKE UP
: b2 RX ENABLE
: b5 RX INTERUPT ENABLE
;

      LDAA    #$27
      STAA    L4007           ; CPU TX/RX CNT'L REG, (8192 BAUD SCI)

LEB92: RTS

LEB93: LDX     L014C           ; INDEX SCI DATA MSG TBL

      LDAB    L014A           ; SCI DATA BYTE COUNTER

```

```

;
; MSG LEN BYTE 2
;
    DECB                ; DECR CNT'R
    BNE    LEBBF        ; BR IF CNT'R NZ
                    ; ... else
    LDAA    4,X        ; L0150,
                    ;
    BRCLR   L0038,#$20,LEBB7 ; CLR b5, IN ALDL MODE
                    ;
    LDAB    L0170      ; SERIAL DATA MODE WD, (MODE 0 - 10)
    SUBB    #$02      ; CK FOR MODE 2
    BLS    LEBB7      ; BR IF MODE GT 2
                    ; ... else
    LDAA    L019D      ; SERIAL DATA MSG LENGTH
    CMPB    #05        ; BR IF MODE 0 or MODE 1
    BEQ    LEBB7
                    ; ... else
;-----
; IN MODE 3 or 4
;
; OUTPUT MSG LENGTH DEPENDS ON INPUT LENGTH
;
;-----
    DECA                ; CALC OUTPUT MSG LENGTH
                    ;
    DECB
    BEQ    LEBB5        ; BR IF Z,, (BR IF MODE 3)
                    ; ... else
    CLRA                ;
LEBB5:  ASRA                ;
    INCA                ; ADD 1 TO TX SERIAL DATA MSG LENGTH
LEBB7:  STAA    L019D      ; SERIAL DATA MSG LENGTH
    ADDA    #$55        ; SERIAL BYTE CNT BIAS
                    ;
    JMP    LEC40        ; TX DATA & CALC SCI CHKSUM
                    ;
;-----
; DATA BYTE, BYTE 3 to LEN -1
;
;-----
LEBBF:  DECB                ; DECR SCI DATA BYTE COUNTER
    BNE    LEBCD        ; BR IF NZ, (NOT 1st DATA BYTE)
                    ; ... else
    BRCLR   L0038,#$20,LEBCD ; BR IF NOT b5, IN ALDL MODE
                    ; ... else
    LDAA    L0170      ; SERIAL DATA MODE WD, (MODE 0 - 10)
    CMPA    #7        ; MODE 7
    BNE    LEC40      ; BR IF NOT MODE 7,
                    ; TX DATA & CALC SCI CHKSUM
                    ; ... else

LEBCD:  CMPB    L019D      ; SERIAL DATA MSG LENGTH
    BCC    LEC3A      ; BR IF CNT GT SERIAL DATA MSG LENGTH
                    ; .... else
    BRSET   L0048,#$80,LEC2E ; BR IF b7, 2nd BYT XMIT WAITING
                    ; .... else
    BRSET   3,X,$$80,LEBE5 ; BR IF b7, OUTPUT IS ROM TBL
                    ; .... else
    BRSET   3,X,$$40,LEBF0 ; BR IF b6, OUTPUT IS RAM TBL
                    ; .... else
    LDX    5,X        ; OUTPUT IS RAM BUFFER, GET ADDR OF BUFF

```

```

        ABX                ;
        LDAA              0,X          ; GET DATA BYTE

        BRA              LEC40        ; TX DATA & CALC SCI CHKSUM <---***

;
;
;
LEBE5:  BRSET           L0038,#$08,LEBEA ; BR IF b3, IN CCM MODE, (NO MODE BYTE)
        DECB                ; .... else
        DECB                ; DECR BYTE CNT TO ACCOUNT FOR MODE BYTE

LEBEA:  ASLB                ; SCALE FOR DBL BYTE ADDR
        ABX                ; ADJ INDEX
        LDX              9,X          ; GET ADDRESS
        BRA              LEC14        ; TO CK FOR DBL BYTE LD/ILLEGAKL ADDR

;
;
;
LEBF0:  BRSET           L0038,#$08,LEBF5 ; BR IF b3, IN CCM MODE, (NO MODE BYTE)
        DECB                ; .... else
        DECB                ; DECR BYTE CNT TO ACCOUNT FOR MODE BYTE

LEBF5:  LDX              5,X          ; GET ADDRESS OF RAM BUFFER
        LDAA             L0170        ; SERIAL DATA I/O CNT'L BUFFER
        BRSET           L0038,#$20,LEBFF ; BR IF b5, IN ALDL MODE
        CLRA                ;
        CLRA                ;

LEBFF:  CMPA             #2           ; MODE 2 ?
        BEQ              LEC10        ; BR IF MODE 2
        CLRA                ; ... else
        CLRA                ;
        ASLB                ;
        INCB                ;
        CLRA                ;
        CMPA             #4           ; MODE 4
        BNE              LEC0B        ; BR IF MODES 0,1,3, OR 7
        CLRA                ; .... else
        CLRA                ;
        ADDB             #10          ;
        CLRA                ;

LEC0B:  ABX                ;
        LDX              0,X          ; LOAD ADDRESS
        BRA              LEC14        ;
        CLRA                ;

LEC10:  INX                ; MODE 2 IS SPECIAL CASE
        LDX              0,X          ; GET BASE ADDRESS
        ABX                ; ADD OFFSET

LEC14:  CPX              #$3000       ; CK ADDR WINDOW
        BCS              LEC36        ; BR IF ADDR BELOW WINDOW
        CLRA                ; .... else
        CPX              #$6FFF       ; CK ADDR WINDOW
        BHI              LEC36        ; BR IF ADDR ABOVE WINDOW
        CLRA                ; .... else

        PSHX

        PULA
        ANDA             #$01        ; 0000 0001
        PSHA

```

```

PULX

LDD      0,X          ; DO DBL BYTE LOAD
STAB    L019F

BSET    L0048,#$80   ; SET b7

BRA     LEC40        ; TX DATA & CALC SCI CHKSUM

LEC2E:  BCLR    L0048,#$80 ; clr B7

        LDAA    L019F
        BRA     LEC40        ; TX DATA & CALC SCI CHKSUM

LEC36:  LDAA    0,X
        BRA     LEC40        ; TX DATA & CALC SCI CHKSUM <---***

;
; CHECK SUM BYTE
;
LEC3A:  BNE     LEC4E        ; BR IF NZ, BR IF NOT CKSUM BYTE
        ; .... else
        LDAA    L014B        ; SCI DATA CHECKSUM
        NEGA
        ; INVERT

;-----
; TX DATA & CALC SCI CHKSUM
;
;-----
LEC40:  STAA    L400A        ; TX DR, (SCI) <----*****

        ADDA    L014B        ; SCI DATA CHECKSUM
        STAA    L014B        ; SAVE NEW SCI DATA CHECKSUM
        INC     L014A        ; SCI DATA BYTE COUNTER

        BRA     LEC64        ; EXIT VIA RTS

;-----
; LAST BYTE BEING TX'ED, CLEAN UP
;
;-----
LEC4E:  CLRA
        CLRB
        BCLR    L0048,#$80   ; CLR b7
        STD     L014A        ; SCI DATA BYTE COUNTER & CK SUM
        STD     L014C        ; SCI DATA MSG TBL INDEX

        LDAA    L4008        ; CLR CPU TX/RX STAT REG, (8192 BAUD SCI)
        LDAA    L4009        ; CLR CPU RX DR, (8192 BAUD SCI)

        LDAA    #$41         ; b6, b0
        STAA    L4007        ; CPU TX/RX CNT'L REG, (8192 BAUD SCI)

LEC64:  RTS
*****

INCLUDE 8D_FTEST.SRC ; FACTOR TEST & A/D READ

*****

*****
* 8D_FTEST.SRC

```

```

*
* FACTORY TEST ROUTINES, 8/31/93
*
* ASSEMBLED ERROR FREE 8/31/93 WAS
*****

```

```

; LIST E,F,4
; INCLUDE 8D_EQU.SRC

```

```

*****
* FACTORY TEST
*
* CALLER PRIOR TO NORMAL INIT.
*
* IF F.T. IS INDICATED BY _____ THE TEST MODULE
* WILL TAKE CONTROL.
*
*
*
*****

```

```

ORG $EC65

```

```

;-----
;FROM INIT AT PGM START
;
;-----

```

```

;
; INIT P4 ECM
;

```

```

LEC65:

```

```

LDAA    #$8C                ; BIT 2,3 & 7, SEL SI PIN, PRESCALER = 4
STAA    L4001               ; SER I/O CSR
                          ; SET PRESCALER PORT VALUE
LDAA    #$88                ; IAC COILS OFF
STAA    $4002               ; CPU DATA LATCH

LDAA    #$FF                ; SET PRP REG FOR OUTPUT'S
STAA    L4003               ; CPU DDR

LDAA    #$90                ; BITS 4 & 7, SET BAUD RATE, 8192
STAA    $4004               ; BAUD I/O REG, (SCI)

```

```

;-----
; CLEAR CPU REG'S, 3FC0 - 3FFA
;-----

```

```

LDX     L3FC0                ; POINT TO 1ST ECM ADDR

CLRA                    ; A = 0
CLRB                    ; B = 0
LEC7E  STD     0,X           ; CLEAR ECM REG'S

INX
INX
CPX     #L3FFA             ; ECU STAT REG

BNE     LEC7E              ; TILL DONE
;-----

```

```

*****
* SUM ALL FM 8008 - TO END OR EPROM
* TYPE 8D ECM
*
*****

```

```

LEC87: LDX      #$8008                ; START OF EPROM
      JSR      LF0A8                ; CK SUM SUBROUTINE

      LDX      #L0100              ; FACTORY TEST MODE WD
      CLR      0,X                 ; CLR FACTORY TEST MODE WD
LEC92: CPY      $8006                ; CK CALC CK SUM vs STORED CK SUM
      BEQ      LEC9B                ; IF OK BRANCH

LEC98: BSET     0,X,#$10             ; IF NOT SET ERR 51 FLAG
      *****

      ;-----
      ; INIT FMD 1 CHIP
      ; AND FMD1 & 2's BUFFERS
      ;-----
LEC9B  LDAA     #$4                  ; INT DATA FOR FMD 1
      TAB
      STD      $0113                ; SAVE IN FMD DATA BUFFERS

      JSR      LF083                ; SEND & RX FMD 1 DATA

      ANDA     #$03                 ; MASF FOR F.T. MODE SELECT BITS
      STAA     $0110                ; SAVE MODE
      ;-----

      ;-----
      ; INIT SAD
      ;-----
      CLRA
      JSR      LF0D2                ; SEL VALID SAD CHANNEL
      ; Go read A/D

      ;-----
      ; FACTORY TEST DETERMINATION
      ; TYPE $8D ECM P/N 1227730
      ;-----
      LDAA     #$60                 ; SEL A/D CH 6, (PUMP VOLTAGE)
      JSR      LF0D2                ; GO READ A/D

      CMPA     #160                 ; CK IF 16 VOLTS
      BLS      LECC4                ; IF VDC L.T. OR E.Q. F.T MODE NOT REQUESTED

      LDAA     #$10                 ; SEL A/D CH 1, (BATTERY VOLTAGE)
      JSR      LF0D2                ; GO READ A/D

      CMPA     #100                 ; CK IF 10 VOLTS
      BCC      LECC4                ; IF G.T. OR E.Q. 10 V, THEN F.T. NOT REQUESTED

      JSR      LF09B                ; CK IF 3.9K TO GND ON ALDL PIN
      BCS      LECC5                ; IF SO, F.T. MODE IS REQUESTED
      ; ... else
LEC94  RTS
      ;-----

      ;-----
      ; FACTORY TEST
      ; MAIN INIT
      ;-----
LEC95  LDS      #$01FF              ; CLR STACK & RE-INIT

      LDX      #$8000              ; SUM ALL EPROM BYTES
      JSR      LF0A8                ; RET W/ CKSUM IN Y

      LDX      #L0100              ; FACTORY TEST MODE WD
      STY      $2A,X               ; SAVE IN XMIT DATA TABLE

      LDD      $8000               ; READ PRM ID FROM EPROM

```

```

STD      $24,X          ; SAVE IN XMIT DATA TABLE

LDD      $8002          ; GET DATE CODE FROM EPROM
STD      $26,X          ; SAVE IN XMIT DATA TABLE

LDD      $8004          ; GET SEQ NUM FROM EPROM
STD      $28,X          ; SAVE IN XMIT DATA TABLE

LDAA     0,X            ; GET FACTORY TEST MODE WD
ORAA     #$80           ; SET BIT 7, (F.T. MODE)
STAA     0,X            ; UP DATE FACTORY TEST MODE WD
STAA     $43,X          ; SAVE IN XMIT DATA TABLE
CLR      $19,X          ; CLR SCI MODE WD

LDAA     #$8C           ; SET LEN BYTE OF XMIT TABLE
STAA     $23,X          ;

LDAA     #$CC           ;
STAA     L0012          ; INIT IAC MOTOR STATUS WD
STAA     $11,X          ; INIT IGN OFF TMR TO N.Z.

LDAA     #$0001         ; INIT FLAGE FOR MODE 2
STAA     $18,X          ; DESCRETE OUTPUT CYCLING
CLR      $15,X          ; INIT SC1 OUTPUT BUFF

LDD      #$F90A         ; GET ECM INIT DATA
STD      L3FFC          ; TO CPU CONT REG
;-----

;-----
; WRITE $AA TO NV RAM IF
; PWR UP IN MODE 1
;-----

TST      $10,X          ; CK IF MODE 1
BNE      LED14          ; BR IF NOT
; ... else
;
LDAA     #$AA           ;
LDX      #$00           ; WR TO ALL NV RAM
LED0C:  STAA     0,X          ; WR $AA
INX      ; NEXT ADDR
CPX      #$0100         ; CK FOR DONE
BNE      LED0C          ; TILL DONE
;-----

;-----
; SET UP REAL TIME INTERUPT
;-----

LED14:  LDX      #L4005          ; CPU UP CNT'R
LDAA     0,X            ; LD FM TMR REG
ADDA     #2             ; SET FOR 1ST INT
STAA     1,X            ; SAVE TO TMR CAPTURE REG
BSET     2,X,#1         ; ENABLE TMR INT'S

LED20:  CLI              ; TURN ON INIT'S

BRA      LED20          ; LP TILL 1ST INIT
;-----

*****
* MAIN TIMER INTERUPT HANDLER FOR FACTOR TEST MODE
*
*
*****

LED23:  LDAA     L4008          ; CPU TX/RX STAT REG, (8192 BAUD SCI)
BITA     #$01           ; TIMER INT, 6.25 Msec ??
BNE      LED38          ; IF SO BR & handle tme int's

```

```

; .. else
BITA    #$40    ; CK IF SCI INT'S
BEQ     LED31   ; IF NO CK OTHER INT'S
; ... else
JMP     LF06D   ; HANDLE SCI TC INT

LED31:  ASLA                    ; SCI TDRE INT ??
BCS     LED35   ; IF SO BR & HANDEL TRDE INT
; .... else
RTI                    ; IGNORE OTHER INT'S
;-----

LED35:  JMP     LF03B           ; TO TD READY HANDLER

LED38:  LDAA    L4006           ; GET CURRENT TIMER MATCH VALUE
ADDA    #204                ; FOR NXT INT IN 6.25 Msec
STAA    L4006                ; NEW TMER CAPTURE

LDS     #$01FF              ; CLR STACK OF INT ADDR'S
CLI                    ; ENABLE INT'S FOR SCI XMIT'S

;-----
; CK DIAG/ALDL LINE
;-----
JSR     LF09B                ; STILL 3.9K TO GND ??
;
BCS     LED4A                ; IF SO BR TO CONTINUSE FACTORY TEST
; .... else
SWI                    ; RESET SYSTEM
;-----

;-----
; Send buffer data & read bytes 1 & 2 of FMD 1
;
;-----
LED4A:  LDAA    $0113           ; LD FM FMD 1 BUFFER
ORAA    #$80                ; SEL byte 2
PSHA                    ; SAVE BUFFER DATA
JSR     LF083                ; SEND DATA & READ BYTE 2 TO A REG

TAB                    ; BYTE 2 TO B REG
PULA                    ; GET BUFFER DATA TO A
ANDA    #$7F                ; CLEAR SELECT BIT, (BIT 7)
JSR     LF083                ; SEND DATA & READ BYTE 1 TO A REG

STD     L0156                ; SAVE BOTH BYTES IN XMIT TABLE
;-----

;-----
; SEND BUFFER DATA & READ
; BYTES 1 & 2 OF FMD 2
;-----
LDAA    $0114                ; LD FM FMD 2 BUFFER
ORAA    #$80                ; SEL byte 2
PSHA                    ; SAVE BUFFER DATA
JSR     LF08F                ; SEND DATA & READ BYTE 2 TO A REG

TAB                    ; BYTE 2 TO B REG
PULA                    ; GET BUFFER DATA TO A
ANDA    #$7F                ; CLEAR SELECT BIT, (BIT 7)
JSR     LF08F                ; SEND DATA & READ BYTE 1 TO A REG

STD     $0158                ; SAVE BOTH BYTES IN XMIT TABLE

```



```

;-----
;
;-----
; TX SCI DAT FM BUFFER
; TO CHIP
;-----
LDAA    L0115          ; LD FM SCI BUFFER
JSR     LF077          ; SEND OUTPUTS'S RX INPUTS'S
;
LDX     #$0100        ; POINT TO RAM START ADDR
STAA    $5A,X         ; SAVE INPUT'S IN XMIT TABLE

;-----
; UPDATE SCI TABLE WITH
; DDR & PRP DATA
;-----
LDAA    L4003          ; CPU DDR REG
STAA    $55,X         ; SAVE IN SCI DATA TABLE
;
LDAA    $4002          ; CPU Data latch REG
STAA    $54,X         ; SAVE IN SCI DATA TABLE

;-----
; IGN VOLTAGE STATUS
;
;-----
LDAA    #$10           ; Ch $10, BATTERY A/D CH SEL
JSR     LF0D2          ; Go read A/D

LDAB    $11,X         ; GET IGN OFF FLAG/CNT'R
CMPA    #40           ; CK IF IGN ON
BHI     LED96         ; IF NOT BR
; ...else
INCB    ; BUMP UP IGN OFF FLG/CNT'R
CMPB    #160          ; TUEN OF DLY DONE ?
BLS     LEDA4         ; If not done, BR & keep going
; ... else
SWI     ; IF DONE ALL PWR DOWN

LED96   CMPA    #90    ; Ign on ??
BCS     LEDA6         ; IF NOT BR,
; ... else
TSTB    ; CK OF TO ON XISSITION ??
BEQ     LEDA6         ; IF NO BR
; ... else
CLR     ; CLR IGN OFF FLG/CNTR
LDAA    $56,X         ; LD FMD 1 DESCRETE I/P'S
ANDA    #3           ; MASK MODE REQUESTS
STAA    $10,X        ; SAVE MODE REQUESTED
LEDA4   STAB    $11,X ; SAVE IGN OFF FLG/CNTR

;-----
; UPDATE MINOR LOOP CNTR
;-----
LEDA6   INC     $16,X

;-----
; RESTART COP1
;-----
LDD     #$FF00        ; SET TO ARM COP
STD     L400B
;-----

```

```

; MODE DETERMINATION
;-----
LDAB    $10,X          ; CK IF OPERATING IN MODE 1
BEQ     LEDF7          ; IF SO BR TO MODE 1
; ... else
JSR     LF00B          ; TOGGLE COP2 OF FMD 1 & 2

LDAA    $19,X          ; LD SCI MODE WD
ASLA                    ; CK IF XMISSION IN WORK NOW
BCS     LEDBD          ; BR IF SO
; ... else
JSR     LF018          ; BGN SCI XMISSION
;
LEDBD   LSRB           ; CK IF OPERATING IN MODE 2
BCC     LEDC3          ; IF NO, BR TO MODE 3
; ... else
JMP     LEE4B          ; TO MODE 2

*****
* MODE 3 FACTORY TEST
* MISC TEST MODE
*****
LEDC3   LDD            #196          ; SET FOR 3 msec PULSE
STD     L3FF2          ; WR ASYNC PULSE TO CPU
;
MUL                    ; DELAT PRIOR TO ACCESS CPU
NOP                    ; (11 CYCLES)
;
LDD     L3FFC          ; GET ECU CR
ORAA    #4             ; SET ASYNC FUEL BIT
BSR     LEE4A          ; DELAY PRIOR TO CPU ACCESS
;
STD     L3FFC          ; INITIATE ASYNC PULSE
; W/WRITE TO ECU CR

;-----
; CALC Non Vol RAM CKSUM
;-----
LDX     #0             ; START ADDR FOR CK SUM
LDY     #0             ; CLR CKSUM ACCUM
LEDDC: LDAB            0,X          ; BYTE FM MEMORY
ABY                    ; ADD B TO Y
INX                    ; TO NEXT ADDR
CPX     #$0100         ; END OF NV RAM
BNE     LEDDC          ; TILL DONE
; . . . else
STY     L012C          ; SAVE NV CK SUM IN XMIT TABLE

;-----
; TURN OFF PRP4 & DO1
; DESCRETES
;-----
LDX     #$4002         ; CPU DATA LATCH REG
BCLR    0,X,#$10       ; TURN OFF PRP4 DESCRETE OUTPUT
BCLR    2,X,#2         ; TURN OFF DO1 OUTPUT

;-----
; TURN OFF ALL SC1 OUTPUTS
; EXCEPT SXR ENABLE
;-----
LDAA    #1             ; SXR ENABLE, BIT 0
BRA     LEE0E          ; CONTIUE MODE 1
;-----

```

```

*****
* MODE 1 FACTORY TEST
* ALL OFF MODE
*****

```

```

;-----
; COP 2 TOGGLE
; DETERMINE
;-----
LEDF7:  TST      $11,X          ; CK IF IGN ON
        BNE     LEE01         ; BR IF NOT
                                ; ... else
        LDAA   $56,X          ; IF IGN ON,
        BITA   #$03           ; NEW MODE COMMING ??
        BEQ    LEE04         ; IF NOT DONT TOGGLE COP
                                ; ... else
LEE01   JSR     LF00B         ; IF SO GO TOGGLE COP
;-----

;-----
; TURN ON PRP4 & DO1
; DESCRETES
; CK QDM DISABLE IN BYPASS
;-----
LEE04   LDX     #$4002        ; CPU Data latch, (PR REG)
        BSET   0,X,$$10      ; TURN ON PRP DISCRETE OUTPUTS
        BSET   2,X,$2        ; TURN ON DO1 DISCRETE OUTPUTS
;-----

;-----
; TURN OFF SC1 OUTPUTS
;-----
LEE0E   CLRA                    ; DATA FOR SC1 BUFFER
        STAA  L0115          ; SAVE IN BUFFER FOR SC1 OUTPUT
;-----

;-----
; SET CPU REG'S FOR
; MOD 1/3
;-----
LDD     L3FFC                ; ECU Cnt REG
ANDA   #$FB                  ; CLEAR ASYNC FUEL BITS
ANDB   #$EB                  ; CLEAR DIS5 & EST, (BOTH OFF)
ORAB   #$08                  ; OFF ERR LIGHT
BSR    LEE4A                 ; DELAY

STD     L3FFC                ; ECU Cnt Reg
CLRA                    ; CLEAR MSB OF FUEL
CLRB                    ; CLEAR LSB OF FUEL
BSR    LEE4A                 ; DELAY

STD     L3FD0                ; PW TIMER = 0
;-----

;-----
; TURN OFF ALL PWM'S
; 3FD4 - 3FDC
;-----
LDX     L3FD4                ; POINT TO 1ST DUTY CYC REG
;
LDD     #$7000              ; FOR 0% DUTY CYCLE
LEE2C   STD     0,X          ;
;
INX                    ; BUMP ADDR POINTER
INX                    ;

```

```

CPX      #L3FDC          ; LAST ADDR ?
BNE      LEE2C          ; BR TILL DONE
          ; ... else
STD      L3FCC          ; PWM 6 REG
          ;
BSR      LEE4A          ; DELAY
          ;
STD      L3FEA          ; PWM7 REG, (PWMI)
;-----
;-----
; SET FOR FMD'S 1/3
;-----
LDD      $0113          ; GET FM FMD BUFFERS
ANDA    #$E7           ; SEL PWMI, (PW7) FOR OPRT 1
ORAB    #$18           ; SET FMD 2 TO SELECT EFIB
STD     L0113          ; SAVE NEW DATA IN FMD BUFFERS

;-----
; WAIT FOR NEXT INTERRUPT
;-----
LEE47    CLI           ; TUNE ON INTERUPTS
          ;
BRA     LEE47          ; TILL INTERUPT
          ;
LEE4A    RTS           ;
;-----

*****
* FACTORY TEST
* MODE 2 (I/O CHECK)
*****
LEE4B    LDY          #$4002          ; CPU Data latch, PRP REG
BCLR    0,Y,$20        ; CLR SAD MUX SEL C
LDAA   $4004          ; GET Baud I/O Reg, (SCI)
ANDA   #$F3           ; CLR SAD MUX SEL'S A & B
PSHA   ; SAVE INITAL BIO CR DATA

LDX     #$012E+1      ; ADDR OF 1st SAD CH IN TABLE
CLRB   ; START W/ ch 0
LEE5D   STAA         $4004          ; Baud I/O Reg, (SCI) W/MUX SEL'S

PSHA   ; SAVE RUNNING BIO REG

LEE63   SUBB         #$10          ; COMPENSATE FOR LOOP ADDITION
ADDB   #$10          ; ADD FOR NEXT LOOP
CMPB   #$C0          ; CK IF ALL CH'S DONE
BEQ    LEE87         ; BR OUT IF DONE

TBA    ; IF NOT DONE LOAD SAD CH NUM
JSR    LF0D2        ; Go read A/D

STAA   0,X          ; SAVE RESULT IN XMIT TABLE
INX    ; INCT TABLE POINTER
CMPB   #$90          ; CH 9 ??
BNE    LEE63        ; BR IF NOT, (TILL DONE)
          ; ... else
PULA   ; GET BIO DAYA
ADDA   #4           ; SET SEL'S A & B FOR NEXT MUX
BITA   #$0C          ; DONE W/ 4 MUX CH'S ?
BNE    LEE5D        ; BR IF NOT, (TILL DONE)

BRSET  0,Y,$20,LEE63 ; IF SO, CONT W/ CH 10
          ; ... else
BSET   0,Y,$20      ; SET FOR MUX 4 - 7
PULA   ; GET BIO CR INIT DATA

```

```

BRA      LEE5D                ; LOOP AGAIN

;-----
; COOLANT PULL UP SW
; 25 MSEC
;-----
LEE87   LDAB    L0116          ; GE MINOR LP COUNTER
        PSHB                    ; TO STACK
        BITB    #$03          ; START OF 25 MS PERIOD ??
        BNE     LEEA1         ; BR IF NOT
        ; .. else
        LDAB    $0113         ; GET DATA FM FMD 1 BUFFER
        BITB    #$01          ; CK IF PULL UP 4K NOW
        BNE     LEE97         ; BR IF NOT

        INX                    ; IF SO, SET NEW ADDR
LEE97   LDAA    L0132         ; GET SAD COOLANT VALUE
        STAA    0,X           ;
        EORB    #$01          ; TOGGLE PULL UP BIT
        STAB    L0113         ; SAVE DATA IN FMD 1 BUFFER
;-----

;-----
; 100 Msec CK INTERVAL
;
;-----
LEEAA1  PULB                    ;
        BITB    #$0F          ; CK IF TIME FOR 100 Msec ??
        BEQ     LEEA9         ; BR IF TIME, TO 100 Msec
        ; ... else
        JMP     LEF3E         ; JMP OVER 100 Msec
;----- --

;----- -
; 100 MS, SET PEM DUTY CYCLES
;----- -

LEEAA9  LDX     #L3FD4        ; POINT TO 1ST DUTY CYCLE REG

LEEAF   LDD     #$D134        ; FOR 30% DUTY CYCLE
        STD     0,X           ; SAVE TO 1ST DUTY CYCLE REG
        ADDD    #$66          ; INCT D.C. 10%
        INX                    ; NEXT REG ADDR
        INX                    ;
        CPX     #L3FDC        ; CK IF AT PRM REG 5
        BNE     LEEAF         ; IF NOT LOOP TILL DONE

        STD     L3FCC         ; IF DONE WR TO PWM 6 REG
;-----

;-----
; CYCLE EACH OF 7 DISCRETE OUTPUTS
; ON EVERY 100 Msec's.
; DATA IN RAM LOC 0118 INDICATES
; WHICH IN ON NEXT.
;
; WHEN BIT 1 = 1, SC1 - 02 IS ON
;     BIT 2 = 1, SC1 - 03 IS ON
;     ETC THRU BIT 6, (7 NOT USED)
;
;-----
LDX     #$0118                ; POINT TO DISPLAY FLAG WD
LDAB    0,X                   ; GET DISCRETE CYCLE FLAG WD
ASLB                    ; ROTATE FOR NEXT OPTPUT CYCLE
BCC     LEEC8                 ; BR IF NOT THRU 7th CYCLE

LDAB    #$02                  ; IF TH 7th, RESET FOR NEXT

```

```

LEEC8  STAB    0,X          ; SAVE NEW FLG WD

      LDAA    L0115        ; GET FM SC1 BUFFER
      ANDA    #$F1        ; CLR SC1,02,03 & 04 OUTPUT BITS
      ANDB   #$0E        ; MASK OFF NON SC1 FLAG BITS
      ABA     ; SET STATUS OF SC1 OUTPUT BITS
      STAA   L0115        ; SAVE NEW SC1 DATA BUFFER

      LDD    L3FFC        ; GET CPU CR
      ANDB   #$FB        ; TURN OF DIS5
      ORAB   #8          ; TURN OFF ERR LIGHT
      PSHA   ; SAVE MSB OF CPU CR REG
      LDAA   0,X         ; RELOAD FLAG WD
      BITA   #$10        ; BIT 4, SHOULD ERR BE ON ??
      BEQ    LEEE5       ; BR IF NOT

LEEE5  ANDB   #$F7        ; ELSE TURN ON ERR LIGHT
      BITA   #$20        ; BIT 5, SHOULD DIS 5 BE ON ?
      BEQ    LEEEB       ; BR IF NOT
                        ; ... else

LEEEB  ORAB   #4          ; BIT 2, TURN DIS 5
      PULA   ; RESTORE MSB OF ECM DATA
      STD    L3FFC        ; ECU CR

      LDAA   0,X         ; RELOAD FLAG WD
      LDX    #$4002       ; CPU Data latch, (PRP REG)
      BCLR   0,X,$$10     ; TURN OFF PRP4
      BCLR   2,X,#2       ; TURN OFF DO1
      BITA   #$40        ; BIT 5, SHOULD PRP4 BE ON
      BEQ    LEF01       ; NR IF NOT

LEF01  BSET   0,X,$$10    ;
      ASLA   ; SHOULD DO1 BE ON ??
      BCC    LEF07       ; BR IF NOT
                        ; ... else
      BSET   2,X,#2      ; TURN ON DO1
      ;-----

      ;-----
      ; 100 MS LOGIC, CYCLE VSS
      ; SPEED BUFFER OPTION
      ;-----

LEF07  LDX    #L0100      ; FACTORY TEST MODE WD
      INC    $17,X       ; INC MOD 8 CNTR

      LDAB   $17,X       ; GET MOD 8 CNTR
      LDAA   #32         ; SHIFT 3 LSB
      MUL   ; BITS OF B REG 5 PL'S LEFT TO
                        ; POS 7,6,5 AS REQ FOR THIS CYCLE

      LDAA   $15,X       ; SC1 BUFFER
      ANDA   #$1F        ; CLEAR ALL OPTION SEL BITS
      ABA     ; RESET OPTION SEL BITS
      STAA   $15,X       ; RESTORE IN SC1 BUFFER

      ;-----
      ;
      ;
      ;
      ;   BITS 3 & 4 = 1   OPRT1 DISCRETE ON
      ;                   FUEL PUMP 50% D.C.
      ;   BIT 3 = 0
      ;
      ;   BIT 4 = 1       OPRT1 DISCRETE OFF
      ;                   FUEL PUMP 50% D.C.
      ;

```

```

; BITS 3 & 4 = 0 OPRT1 50 % D.C.
; FUEL PUMP DISCRETE ON
;
;-----
LDD    #$D200          ; FOR 50% DUTH CYCLE
STD    L3FEA          ; ECM PWMI OF FMD 1

LDAA   $13,X          ; GET FROM FMD 1 BUFFER
BITA   #$10           ; TINE TO RE-START CYCLING
BEQ    LEF28          ; BR IF TIME
; ... else
SUBA   #8             ; ADJ FOR THIS CYCLE
BRA    LEF2A          ;

LEF28  ORAA   #$18     ; RESTART CYCLING
LEF2A  STAA   $13,X    ; SAVE NEW DATA IN FMD 1 BUFFER

;-----
; 100 Msec STEP IAC MOTOR
;-----
LDAA   $12,X          ; GET CURRENT IAC STATUS
ASLA                   ; ROTATE LEFT
ADCA   #0             ; WRAP W/CY BIT
STAA   $12,X          ; SAVE NEW IAC STATUS WD
ANDA   #3             ; MASK FOR CIOL BITS, 0000 0011b

LDAB   $4002          ; CPU Data latch, (PRP STATUS)
ANDB   #$00FC         ; CLR OLD COIL BITS
ABA    ; SET NEW COIL BITS
STAA   $4002          ; CPU Data latch, (PRP STATUS)
;-----

;-----
; EXERSIZE FUEL & SPK OUTPUTS
; ACCORDING TO RPM
;-----
LEF3E  LDAA   L0114    ; GET FMD 2 BUFFER
ORAA   #$18           ; SET EFIB, (SIMULTANIOUS FIRE)
STAA   L0114          ; WR TO FMD 2 BUFFER

LDD    L3FFC          ; CPU Cntl Reg STATUS
ANDB   #$DF           ; CLEAR BIT 5 FOR SIMUL FIRE
ORAB   #$10           ; BIT 4, ENABLE EST
PSHB                   ;
PSHA                   ;

LDX    L3FC0          ; GET REF PERIOD FROM ECM
PSHX                   ; XFER TO DBL ACCUM
PULA                   ;
PULB                   ;
LSRD                   ; DIVIDE BY 4
LSRD                   ;
CPX    #328           ; REF PER 5 MSEC
BCC    LEF6F          ; BR IF G.T. or E.Q. 5 Msec

PULX
STX    L3FFC          ; ECU CR
LDX    #66            ; SET BPW = 1 Msec
PSHX                   ; SET INJ DELAY 1 Msec

;
; CALC 2'S COMP TO GET REF PERIOD/4 FOR 45 DEG SPK ADV
;
COMA                    ;

```

```

COMB
ADDD      #1                ;

LDY      #197              ; SET DWELL TIME = 3 Msec
BRA      LEF9E            ; GO UPDATE SCI TABLE

LEF6F    CPX      #655      ; REF PER 10 Msec
        BHI      LEF8F      ; BR IF G.T. 10 Msec

        PULA                ; GET CPU DATA FM STACK
        PULB
        ORAB      #$20      ; BIT 5, FOR ATERNATING FUEL
        STD      L3FFC      ; ECU CR,

        LDAB      L0114      ; GET FM FMD 2 BUFFER
        ANDB      #$F7      ; SEL EFIA, ALTERNATING FUEL
        STAB      L0114      ; WR TO FMD 2 BUFFER

        LDX      #328        ; SET BPW = 5 Msec
        PSHX

        CLRB                ; CLR D FOR TDC SPK
        CLRA

        LDY      #262        ; SET DWELL 4 Msec
        BRA      LEF9E      ; GO UPDATE SCI TABLE

LEF8F    PULX                ; GET ECM DATA
        STX      L3FFC      ; ECU CR, SET FOR SIMUL FUEL

        LDX      #$0000      ; SET INJ DELAY = 0 BY PUSH TO STX
        PSHX

        LDX      #655        ; SET BPW = 10 Msec
        LDY      #328        ; SET DWELL = 5 Msec
        ;-----
        ; LEAVE D = REF/4 FOR 45 DEG RETARD
        ;-----

*****
* UPDATE SCI TABLE
*
*
*****

LEF9E:   STD      L3FF6      ; EST FALL CNT'R, WR FOR S.A. or S.R.
        BSR      LEFF9      ; DELAY

        STX      L3FD0      ; WR FOR FUEL PW
        BSR      LEFF9      ; DELAY

        STY      L3FDC      ; WR FOR DWELL TIME
        BSR      LEFF9      ; DELAY

        PULX                ; GET DELAY FM STX
        STX      L3FCE      ; SAVE FOR INJ DELAY
        SEI                ; TURN OFF INTERRUPTS

        LDX      #L0144      ; POINT TO CPU DATA TABLE
        LDD      L3FC0      ; GET REF PERIOD
        BSR      LEFED      ; WT TO TABLE

        LDY      #L011D      ; OLD PP2 TMR
        LDD      L3FC2      ; B Ctr, OLD PPI TMR
        BSR      LEFFA      ; WR TO TABLE

```



```

LDY    #L011F                ;
LDD    L3FC4                  ;
BSR    LEFFA                  ; WR TO TABLE

LDD    L3FC6                  ;
BSR    LEFED                  ;

LDD    L3FC8                  ; PA2 CNT'R
BSR    LEFED

LDD    L3FCA                  ;
BSR    LEFED                  ;

LDD    L3FE0                  ; PA4 CTR, Rd spd pulse cntr
BSR    LEFED                  ; WR DIFF TO TABLE

LDY    #L0121                ;
LDD    L3FF8                  ; PA1 TMR
BSR    LEFFA                  ; WR TO TABLE

LEFEA: CLI                    ; CLEAR & ENABLE INTERUPTS
      BRA    LEFEA            ; WAIT FOR NEXT INTERUPT
      *****

*****
* CK if LOW BYTE OF D REG pointed by X REG is next
* for TX BY SCI, If not, D REG Data is written to
* pointed addr. Up dating SCI table. else nothing is
* written
*****

LEFED: PSHX                    ; SAVE POINTER
      INX                    ; INC TO POINT LO BYTE
      CPX    L011A            ; LOW BYTE NEXT FOR SCI ?
      PULX                    ;
      BEQ    LEFF7            ; IF SO DON'T UPDATE TABLE
      ; ... else
      STD    0,X              ; IF NO. WR FRESH DATA TO TABLE

LEFF7: INX                    ; INC POINTER
      INX

LEFF9: RTS
      *****

*****
* CK IF DATA IN D reg IS DIFFERENT THAN Y POINTED
* DATA, IF SO THE DIFFERENCE IS WRITTEN TO SCI
* DATA TABLE USING PRIOR SUB ROUTINE (AT LEFED)
*
*****

LEFFA: SUBD    0,Y              ; IS THER A DIFF ?
      BNE    LF002            ; BR IF SO
      ; ... else
      INX                    ; NEXT ADDR
      INX

      RTS                    ; RET W/ NO UP DATA

LF002: BSR    LEFED            ; WD DIFF VALUES TO SCI TABLE

```

```

ADDD      0,Y                ; RETAIN ORIG D REG VALUE
STD       0,Y                ; SAVE AS REF FOR NEXT DIFF

RTS
*****

*****
* TOGGLE COP2 BIT OF FMD 1 & 2
* IN SERIAL DATA REG'S
*
*****

LF00B: PSHB                    ;
      ;
LDD      L0113                ; GET FM FMD BUFF'S
EORA     #$02                 ; TOGGLE FMD 1 COP 2 BIT
EORB     #$02                 ; TOGGLE FMD 2 COP 2 BIT
STD      L0113                ; SAVE NEW FMD BUFFER DATA
      ;
PULB
      ;

RTS
*****

*****
* INITIATE XMISSION OF XMIT DATA TABLE VIA
* THE SERIAL COMM I/F, (SCI).
*
*
*****

LF018: LDX      #L0119         ; ADDR OF STATUS TBL
      ;
LDAA     #$80                 ; CLR ALL BUT BIT 7
STAA    0,X                  ; WR TO STATUS TBL
      ;
LDY      #L0123               ; POINT TO DATA TABLE
STY     1,X                  ; SAVE IN STATUS TABLE
CLR     3,X                  ; CLR RUNNING CK SUM BYTE

LDAA     L0115                ; GET SC1 BUFFER
ORAA     #$01                 ; SET b0, ENABLE SXR
STAA    L0115                ; SAVE NEW SC1 DATA

BSR      LF077                ; XMIT TO SC1 CHIP
      ; (X NOW HAS PRP)

LDAA     6,X                  ; RD TO CLR INT'S
CLRA
STAA     8,X                  ; WR TO TDR SO ID CODE TX FIRST

BSET     5,X,#$88            ; SET b7 & b3

RTS
*****

*****
* HANDLE INTERRUPTS CAUSED BY TX DATA REG EMPTY
* DURING AN SCI XMISSION
*
*
*****

LF03B: LDX      #$0119         ; INDEX STATUS TABLE
BRCLR   0,X,#1,LF052        ; BR IF NOT b0, LAST BYTE FLAG
      ; BR IF NOT LAST BYTE

LDAA     3,X                  ; GET CK SUM BYTE

```

```

NEGA                                ; MAKE 2's COMPL

LDX    #$4007                        ; CPU TX/RX CNT'L REG, (8192 BAUD SCI)
BCLR   0,X,#$80                      ; CLR b7, DIASABLE TDRE INT'S

STAA   3,X                            ; XMIT 2'S COMP CK SUM
BSET   0,X,#$40                      ; ENABLE TC INTERUPTS

BRA    LF06C                          ; EXIT VIA RTI

LF052: LDY    1,X                      ; GET ADDR OF CURRENT BYTE
LDAA   0,Y                            ; GET CURRENT BYTE
STAA   L400A                          ; CPU TX DR, (SCI)
;
ADDA   3,X                            ; ADD CURRENT CK SUM BYTE
STAA   3,X                            ; SAVE AS NEW CK SUM
;
INY    ; FOR NEXT BYTE
STY    1,X                            ; SAVE ADDR IN STATUS TABLE
CPY    #$015B                          ; CK IF ALL BYTES SENT ?
BNE    LF06C                          ; BR IF NOT DONE
; ... else
INC    0,X                            ; SET LAST BYTE FLAG IN MODE WD

LF06C: RTI                            ; RETURN
*****

*****
* HANDLE INTERUPTS RESULTING FROM COMPLETION OF
* SCI XMISSIONS. A TC INTERUPT WHEN SCI HAS SENT
* LAST BYTE AND RDRE IS EMPTY.
*
* DISABLE XMITER, TC INT'S, CLR XMISSION IN
* WORK FLAG
*
*****

LF06D: CLR    L0119                    ; CLEAR SCI MODE WD
;
LDX    #$4007                        ; CPU TX/RX CNT'L REG, (8192 BAUD SCI)
BCLR   0,X,#$48                      ; DISABLE XMIT & TC INT'S

RTI
*****

*****
* USE FOR COMM WITH SC1 CHIP. ENTER W/ DATA IN A
* GOING TO SC1 CHIP
*
* DATA RX'ed FROM SC1 CHIP WILL BE IN A reg ON RET
*****

LF077: LDX    #$4002                    ; CPU Data latch, (PRP REG)
BCLR   0,X,#$80                      ; SELECT SC1 CHIP
;
BSR    LF0C0                          ; XMIT/RX SERIAL LINE
;
BSET   0,X,#$80                      ; UN-SELECT SCI CHIP

RTS
*****

*****
* COMM W/ FMD 1 ON ENTRY, ENTER WITH DATA TO GOING
* TO FMD 1 IN A REG,
* DATA FM FMD WILL BE IN A REG ON EXIT

```

```

*****
LF083:  LDX      #$4002                ; CPU Data latch, (RPR REG)
        BSET    0,X,#04              ; SELECT FMD 1
                                           ;
        BSR     LF0C0                ; XMIT/RX SERIAL LINE
                                           ;
        BCLR    0,X,#4               ; UN-SELECT FMD 1

```

RTS

```

*****
* COMM W/ FMD 2 ON ENTRY, ENTER WITH DATA TO GOING
* TO FMD 2 IN A REG,
* DATA FM FMD WILL BE IN A REG ON EXIT
*****

```

```

LF08F:  LDX      #$4002                ; CPU Data latch, (RPR REG)
        BSET    0,X,#$40            ; SELECT FMD 2
                                           ;
        BSR     LF0C0                ; XMIT/RX SERIAL LINE
                                           ;
        BCLR    0,X,#$40            ; UN-SELECT FMD 2

```

RTS

```

*****
* CK IF ALDL HAS A 3.9K RESISTOR TO GROUND
* IF SO THE CY FLAG IS SET TO INDICATE A FACTORY
* TEST REQUEST
*****

```

```

LF09B:  LDAA     #$70                 ; SEL ALDL A/D CH, (7)
        BSR     LF0D2                ; GO READ A/D
                                           ;
        CMPA    #100                 ; 2.00 VDC
        BCC     LF0A7                ; BR IF NOT FACT TEST
                                           ; ... ELSE
        SUBA    #40                   ;
        COMA    ; CMPL BIT 7 ...
        ASLA    ; SHIFT TO CY

```

LF0A7: RTS

```

*****
* 16 BIT CK SUM OF 8 BIT NUMBERS
* BEGINE W/ ADDR POINTED BY X TILL FFFF
* EXIT WITH CK SUM IN Y reg
*

```

```

LF0A8:  LDY      #$0000

```

```

LF0AD:  CLRA     ; ZERO COP TMR
        INCA    ; DO 256 LOOPS
        BNE     LF0B8                ; TILL DONW
                                           ; ... else
        PSHX    ; SAVE CURRENT CK SUM ADDR
                                           ;
        LDX     #$FF00                ; Set to arm COP
        STX     L400B                ; COP ARM
                                           ;
        PULX    ; GET CURRENT CK SUM ADDR

```

```

LF0B8: LDAB      0,X                ; GET EPROM DATA
      ABY                ; ADD B to Y
      INX                ; NEXT EPROM ADDR
      BNE      LF0AD      ; TILL DONE

```

```
RTS
```

```
*****
```

```
*****
```

```

* LF0C0.SRC
* XMIT ON SERIAL SHIFT REG, (SSR)

```

```
*****
```

```

LF0C0: PSHX
      LDX      #$4000                ;
      STAA     0,X                  ; WR DATA TO SERIAL SHFT REG

      BCLR     1,X,$80              ; SER I/O CSR, START XMISSION

      PSHB                ; WAIT 8 - 39 SYS CLK'S
      MUL                ; FOR TRANSMISION TO COMPLETE
      MUL
      MUL
      PULB

      LDAA     0,X                  ; GET RETUREND DATA FM SSR
      PULX                ; RESTORE X reg

```

```
RTS
```

```
*****
```

```
*****
```

```

* ADREAD.SRC
*
* Read serial A/D
*
* Call with:
*   Ch num in A reg
*
* Return with:
*   A/D result in A reg
*   or ) if A/D or serial fails
*
* TYPE $8D, ECM P/N 1227730

```

```
*****
```

```

LF0D2: PSHX
      LDX      #$4000                ; GM Register
      SEI                ; HOLD INTERRUPTS
      BCLR     2,X,$08              ; SELECT SERIAL A/D CHIP
      STAA     0,X                  ; CH TO DON IN SSR
      BCLR     1,X,$80              ; CLR b7, SEND CH NUM

      ;
      ; 8 SCLK's delay
      ;

      PSHB
      MUL
      MUL
      MUL
      PULB

      LDAA     0,X                  ; LD TEST CH RESULT FM SSR

```

```

        STAA    $01AE                ; SAVE RESULT
;
; delay 64 cycles
;
        PSHB
        MUL
        MUL
        MUL
        MUL
        MUL
        NOP
        PULB

;-----
; READ A/D TEST CH
;
;-----
        LDAA   #$B0                ; SET A/D CH 11 FOR TEST
        STAA   0,X                 ; WR TO SSR
        BCLR   1,X,$80            ; CLR b7, SEND CH NUM

;
; delay
;
        PSHB
        MUL
        MUL
        MUL
        PULB

        LDAA   0,X                 ; GET RESULT
        BSET   2,X,$08            ; DISABLE SAD
        CLI
        PULX                       ; INT'S BACK ON

        RTS
*****

```

```

*****
* Coolant var table #1, 4K PULL UP
*
* COOL = ((DEG C)+40) * (256/192)
*****

```

ORG \$F105

		TEMP DEG C	A/D		
		-----	---		
LF105:	FCB 225	; VERY HOT	0	151	1
	FCB 215	; 121	6		
	FCB 155	; 76	22		
	FCB 132	; 59	38		
	FCB 117	; 48	54		
	FCB 106	; 39	70		
	FCB 97	; 32	86		
	FCB 88	; 26	102		
	FCB 81	; 21	118		
	FCB 74	; 15	134		
	FCB 67	; 10	150		
	FCB 60	; 5	166		
	FCB 52	; -1	182		

```

FCB 44 ; -7 198
FCB 34 ; -14 214
FCB 22 ; -24 230
FCB 00 ; -40 246
*****

```

```

*****
* Coolant var table #2, 384 OHM TBL
*
*****
ORG $F116

```

```

; TEMP DEG C A/D
; -----
LF116: FCB 255 ; VERY HOT 0
FCB 255 ; VERY HOT 16
FCB 249 ; 147 32
FCB 223 ; 127 48
FCB 205 ; 114 64
FCB 191 ; 103 80
FCB 178 ; 94 96
FCB 167 ; 85 112
FCB 157 ; 78 128
FCB 147 ; 70 144
FCB 137 ; 63 160
FCB 127 ; 55 176
FCB 117 ; 47 192
FCB 104 ; 38 208
FCB 89 ; 27 224
FCB 67 ; 10 240
FCB 00 ; VERY COLD 256
*****

```

```

*****
* IAC STEPPER MOTOR STATE TABLE
*****
ORG $F127

```

```

FCB 1
FCB 2
FCB 3
FCB 4
*****

```

```

*****
* MAT TABLE FOR 1K PU
* ANHT
* 17 LINE TBL
*****
ORG $F12B

```

```

; TEMP DEG C A/D, (INV)
; -----
LF12B: FCB 0 ; -40 0
FCB 42 ; -8 16
FCB 61 ; 6 32
FCB 75 ; 16 48
FCB 85 ; 24 64
FCB 95 ; 31 80
FCB 103 ; 37 96
FCB 111 ; 43 112
FCB 120 ; 50 128
FCB 128 ; 56 144
FCB 137 ; 63 160
FCB 148 ; 71 176
FCB 161 ; 81 192

```

```

FCB 176      ; 92          208
FCB 199      ; 109        244
FCB 227      ; 130        240
FCB 255      ; 152        256
*****

```

```

*****
* MALF NUMBER TABLE FOR BLINK OUT
*
* TLB OUT IS 2 DIGIT BCD NUM
*****
ORG $F13C

```

```

LF13C:  FCB $12      ;
        FCB $12      ; DUMMY PLACE HOLDER
        FCB $13      ; o2 SENSOR
        FCB $14      ; HI COOLANT
        FCB $15      ; LOW COOLANT
        FCB $16      ; SHORTED INJECTOR
        FCB $21      ; HIGH TPS
        FCB $22      ; LOW TPS
        FCB $23      ; LOW MAT
        FCB $24      ; Vss
        FCB $25      ; HIGH MAT
        FCB $26      ; NOT USED
        FCB $31      ; NOT USED
        FCB $32      ; EGR
        FCB $33      ; HIGH MAP
        FCB $34      ; LOW MAP
        FCB $35      ; NOT USED
        FCB $36      ; NOT USED
        FCB $41      ; CYL SELECT ERR
        FCB $42      ; EST
        FCB $43      ; KNOCK FAIL
        FCB $44      ; LEAN o2
        FCB $45      ; RICH o2
        FCB $46      ; VATS
        FCB $51      ; EPROM ERROR
        FCB $52      ; LO OIL TEMP
        FCB $53      ; OVER VOLTAGE
        FCB $54      ; FUEL PUMP VLOTAGE
        FCB $55      ; NOT USED
        FCB $56      ; NOT USED
        FCB $61      ; NOT USED
        FCB $62      ; HIGH OIL TEMP
        FCB $63      ; NOT USED
        FCB $64      ; NOT USED
        FCB $65      ; NOT USED
        FCB $66      ; NOT USED
*****

```

```

ORG      $FFEE

*****
* VECTORS
* TYPE $8D, ECM P/N 122730
*****

```

```

L7FEE  FCB  $88CD      ; FOR ALDL TESTING

L7FF0  FCB  $6000      ; SWI
L7FF2  FCB  $B204      ; IRQ1
L7FF4  FCB  $E3ED      ; IRQ, NOT USED
L7FF6  FCB  $6000      ; ILLEGAL OP CODE

```



```
L7FF8 FCB $B000 ; ILLEGAL ADDRESS
L7FFA FCB $B000 ; COP TIMED OUT
L7FFC FCB $B000 ; CLOCK FAILED
L7FFE FCB $B000 ; HOT RESTART
*****
```

END