Engineering Bulletin

EB601/D
7/2002

Initializing a BDLC-Based SAE J1850 Node on a GM Class 2 or Chrysler PCI Network

By Matt Ruff
8/16-Bit Systems Engineering
Austin, Texas

Introduction

This engineering bulletin describes the basic process for initializing any node which uses a Motorola microcontroller (MCU) with a byte data link controller (BDLC) module onto a Society of Automotive Engineers (SAE) J1850 network. This bulletin is intended to be a supplement to existing documentation, such as product data sheets, reference manuals, and block user guides. Please refer to those documents for exact register names and bit definition information for the specific device being used.

The J1850 node is more than simply a microcontroller, it is the entire application-specific design, including the MCU, circuit board, J1850 physical interface, and any other external circuitry required to perform the customer application. Of particular interest to this discussion, the MCU (with BDLC) and the J1850 physical interface establish communication with the rest of the vehicle's J1850 network and therefore the initialization procedure for the node must include all components of the communication chain.

Initialization Procedure

Why do we care about proper initialization? The primary reason is that the node can come on line at any point in the middle of bus traffic, so the node must be able to synchronize to the message traffic on the bus without disturbing it. J1850 is an arbitrated protocol, so all nodes must be synchronized in order for bitwise arbitration to function properly.

Initialization is the process of powering up a node and introducing it to the network traffic for the first time and must not be confused with waking the node up from a SLEEP or STOP condition. Refer to the *BDLC Reference Manual* (Motorola document order number BDLCRM/D) or BDLC block user guide for more information on recovery procedures from STOP or SLEEP modes. Initialization of the node involves setting up the BDLC for the first time, setting parameters such as clock selection (CLKS bit) and analog round trip delay time (BARD), which only need to be configured once after the initial power-on reset of the MCU. The initialization process presumes that all registers and external devices are in their initial hardware reset state and should only be performed one time following a reset of the MCU, such as power-on reset (POR) or other MCU hardware reset, such as a computer operating properly (COP) reset.

Some confusion has been introduced to the initialization process due to changes in device design. In early devices designed for J1850 networks, such as the HC05V12, the analog transceiver was incorporated into one package, so it was automatically placed into loopback mode when the analog loopback mode ALOOP bit was set. In subsequent designs, an external transceiver is required and the application software must use a general-purpose port pin to accomplish the same functionality. Most external J1850 transceiver devices have a loopback enable input pin to support this mode of operation. Figure 1 shows how the analog and digital loopback features operate on the BDLC, in conjunction with the transceiver device.

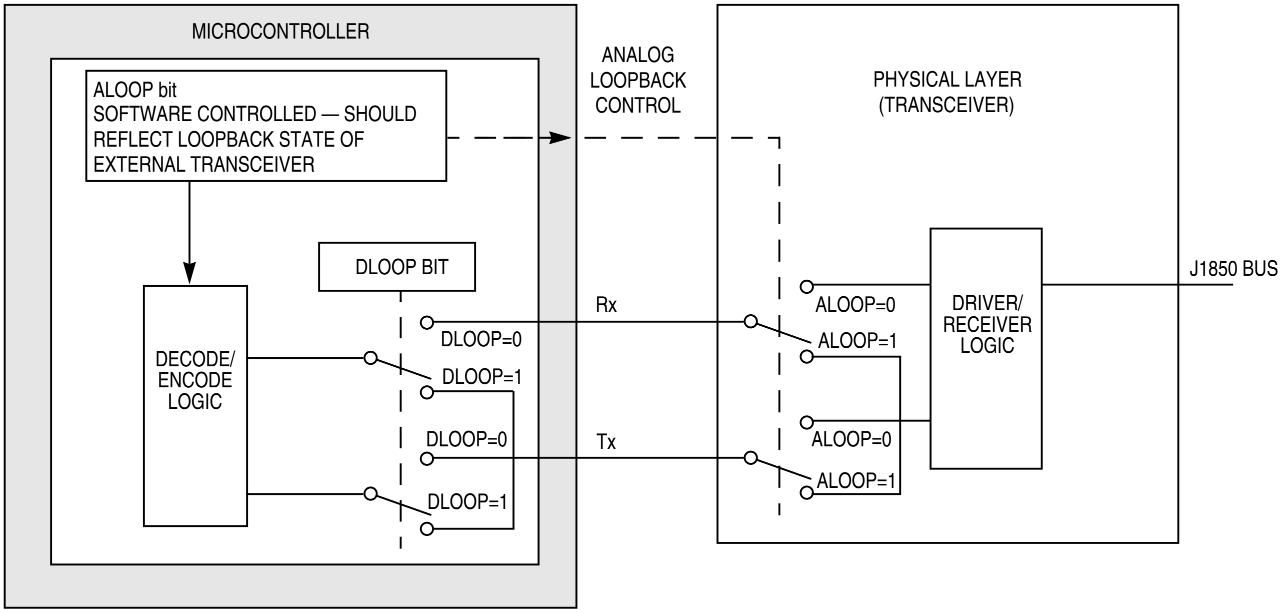

Figure 1. ALOOP and DLOOP Behavioral Model

It is the responsibility of the BDLC driver software to properly manage the synchronization of the ALOOP bit in the BDLC and the actual state of the transceiver device. The BDLC relies upon the state of the ALOOP bit to indicate the actual state of the analog stage of the node and if the two do not match, undesired behavior could result. It is critical, therefore, that the port pin assigned to control the transceiver loopback mode be initialized to match the reset state of the ALOOP bit out of device reset. This means that the port pin must be initialized to place the transceiver into loopback mode out of MCU reset. Most transceivers revert to loopback mode unless their loopback pin is driven to a particular state, so that a general-purpose port pin, which will initialize to a high-Z input, will result in the transceiver being placed into loopback mode. The pin could then be set to the appropriate level and switched to an output, keeping the transceiver in loopback and preventing any gaps in behavior. The transceiver can then be taken out of its loopback mode at any time.

With this in mind, let’s look at the initialization procedure for the BDLC module and incorporate the handling of the J1850 physical layer into that process. **Figure 2** shows the flowchart of the initialization procedure.

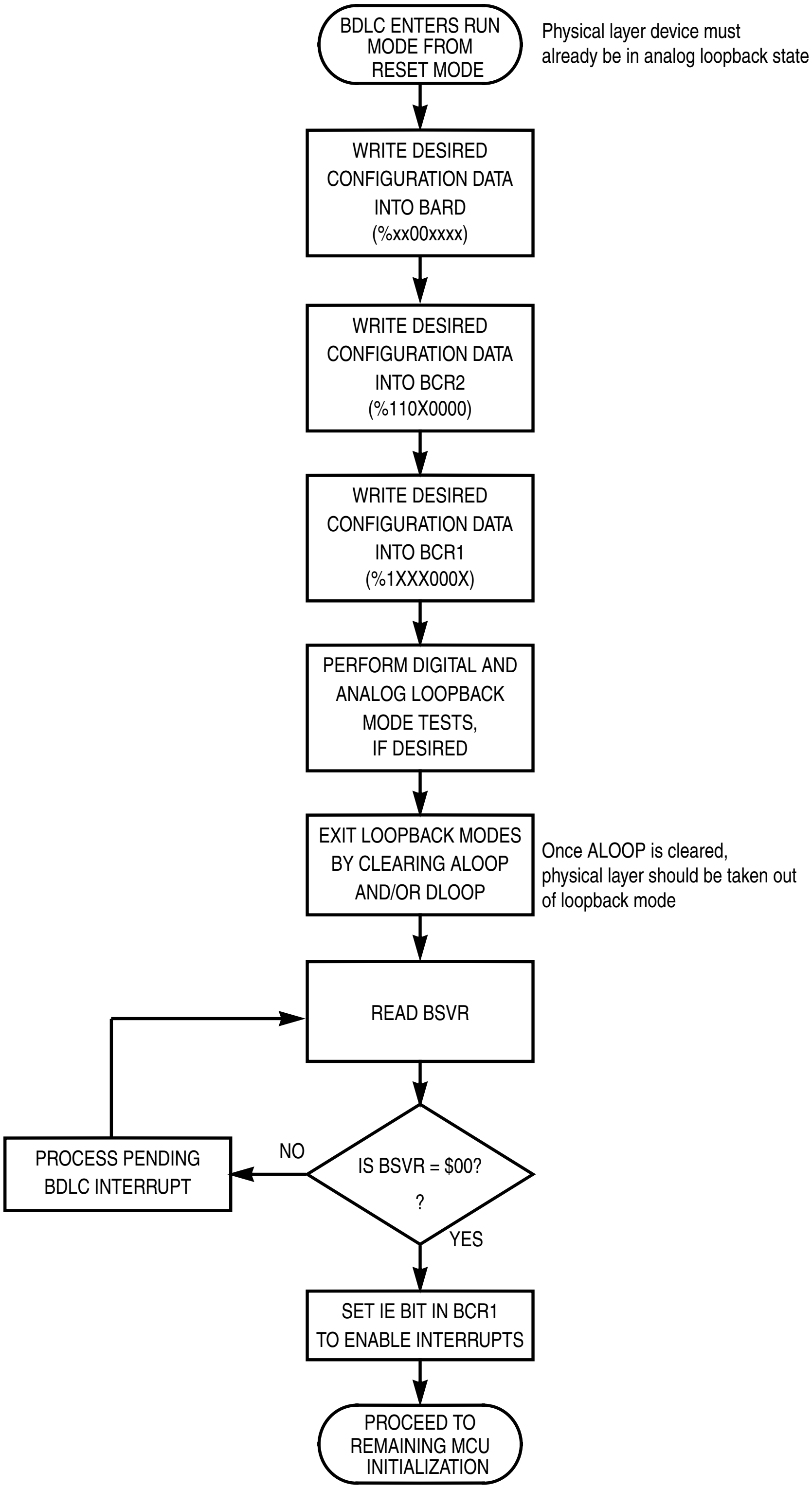
BDLC ENTERS RUN MODE FROM RESET MODE
Physical layer device must already be in analog loopback state
WRITE DESIRED CONFIGURATION DATA INTO BARD (%xx00xxxx)
WRITE DESIRED CONFIGURATION DATA INTO BCR2 (%110X0000)
WRITE DESIRED CONFIGURATION DATA INTO BCR1 (%1XXX000X)
PERFORM DIGITAL AND ANALOG LOOPBACK MODE TESTS, IF DESIRED
EXIT LOOPBACK MODES BY CLEARING ALOOP AND/OR DLOOP
Once ALOOP is cleared, physical layer should be taken out of loopback mode
READ BSVR
IS BSVR = $00? ?
NO
YES
PROCESS PENDING BDLC INTERRUPT
SET IE BIT IN BCR1 TO ENABLE INTERRUPTS
PROCEED TO REMAINING MCU INITIALIZATION

Figure 2. BDLC Initialization Flow Chart

As was stated before, one of the keys to the initialization process is proper handling of the physical layer device in the node. **Figure 2** shows the same basic initialization sequence shown in the *BDLC Reference Manual*, with some minor alterations to show the two critical points in dealing with the physical layer device to ensure that the BDLC knows the current state of the physical layer. From MCU reset state, the physical layer must be in its analog loopback state, which is reflected in the reset state of the ALOOP bit. After all analog and digital loopback tests are performed, loopback modes are exited and the physical layer can be taken out of its analog loopback state by the driver software properly setting or clearing the port pin to which the loopback enable/disable pin of the physical layer is connected. Details of this will depend on the physical layer being used and which MCU port pin is used to control this feature.

For detailed explanations of each step of this initialization process, please refer to the *BDLC Reference Manual* or the BDLC block user guide. For exact values to write into each register at each stage, please also refer to the BDLC section of the data sheet for the specific MCU being used, making certain that all parameters and values properly match the system into which the J1850 node is to be placed.

It is always very important to ensure that both ALOOP and DLOOP are cleared prior to exiting the BDLC initialization routine. In some systems, especially those where the transceiver device being used does not have an analog loopback mode available, it has been observed that in rare cases the clearing of the loopback bits is unsuccessful. This can result in the failure of the node to communicate on the network due to being in loopback mode when the application software believes the unit to be in normal operation.

One way to ensure that both loopback bits are cleared is to place a software loop into the initialization code which checks the state of the bits and attempts additional bit clears in the event that the first clearing attempt fails. For example:

```
loop:   MOV $00,BCR2
        BRSET 6,BCR2,loop
        BRSET 7,BCR2,loop
```

This method is extremely effective at ensuring clearing of the ALOOP and DLOOP bits, regardless of possible problems due to the absence of an analog loopback feature on the physical layer and takes very little extra execution time. This can provide extra assurance that the loopback modes are exited with very little performance impact.

What about the BDLC on HCS12?

The BDLC module on the HCS12 Family of MCUs does not have an ALOOP bit. For these devices, ensure that the BDLC is enabled before releasing the physical layer device from its analog loopback state. Further, the physical layer device must be kept in its loopback state while performing any analog loopback testing. Please refer to the BDLC block user guide for the HCS12 device being used for more information.

Conclusion

The initialization procedure for a J1850 node involves controlling the BDLC module, as well as the physical layer device in a system to ensure seamless merging of the node into network traffic. Due to the history of the evolution of the BDLC module and MCUs using this module, some confusion has arisen about how to handle the two now-distinct components of MCU and physical layer. It is essential to maintain synchronization of the MCU hardware, physical layer, and application software to ensure that proper initialization will occur every time the MCU resets.

References

- *BDLC Reference Manual* – HC08 and HC12 MCUs (Motorola document order number BDLCRM/D)
- HCS12 BDLC Block User Guide – HCS12DJxx MCUs

This page intentionally left blank.

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

http://www.motorola.com/semiconductors

Information in this document is provided solely to enable system and software implementers to use Motorola products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

MOTOROLA