Getting Started with your D1700M Series Module

This document contains step-by-step instructions to quickly connect and communicate with your D1700M modules. The modules require a one-time configuration to properly operate within a data acquisition system. Parameters such as the device address, communications speed (baud rate) and communications parity type must be determined before the configuration can begin. Each module is shipped from the factory with the same default communications values: ASCII character address = 1 (0x31), Baud rate = 300 and Parity type = None.

The first step is to connect the module to a host computer. Use Figure 1.0 to connect an RS-232 module to a host computer or Figure 2.0 to connect an RS-485 module to a host computer. The communications connections are similar or the same for all models of the D1700M series.

Configuration of each module can be performed using either the DGH Utility Software (shipped on CD-ROM with each order) or using Windows Hyper-terminal to manually configure a module. The DGH Utility Software CD-ROM can be downloaded from our website (www.dghcorp.com). See the Installation Instructions below to install the DGH Utility software on your computer.

Use the same connections and cable to pre-configure each module before installing it into the final application system. The 10-pin screw terminal plug on each module is removable. Therefore, make one cable and plug each individual module onto the computer to perform the setup and configuration.

Helpful Hints:

1. Typical Modbus slave addressing begins with address ‘0x01’. Each following module address increments sequentially from there. If you are installing modules into an existing Modbus application then you must determine a free address value before setting up the module. This will prevent communications errors due to address overlaps.

2. Before configuring the module, users must also determine the host serial port baud rate, number of stop bits, and the parity type. The modules must be programmed to the same communications values as the host serial port. Otherwise, the host will not communicate with the modules.

Quick Start Steps:

1. Connect the module to host computer via appropriate drawing (Figure 1.0 or Figure 2.0).

2. Install the Utility Software from CD-ROM provided, using instructions below.
3. Start the Utility Software 2000 from Window, Start, Programs menu selection.
4. Setup the module using ‘Software/Module Setup Instructions’ below.

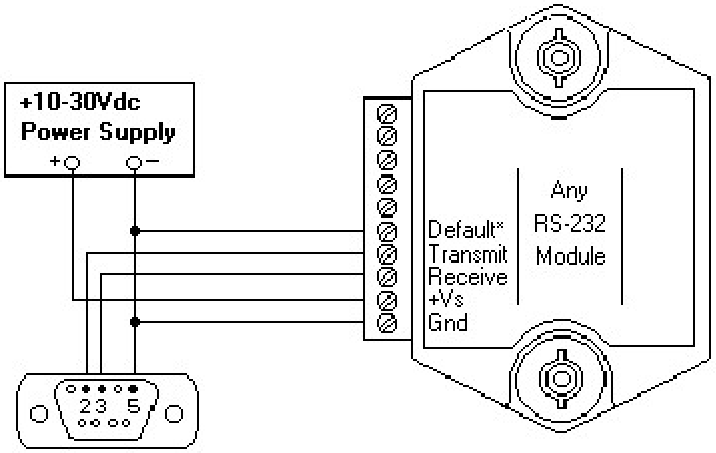
CD-ROM Software Installation Instructions:

1. Insert the DGH Utility Software CD-ROM into the computer CD-ROM drive.
2. Click on the “My Computer” icon on your Windows Desktop, a window will open that lists all of the computer drives.
3. Click on the drive icon that says “Dgh”. A new window will open.
4. Click on the file “setup.exe” and run this program.
5. Follow the installation prompts and complete the entire process. You may close the “My Computer” and “Dgh” windows now. A new menu selection has now been added to your “Start”/”Programs” menu. The selection is “DGH Data Acquisition” and it contains all the DGH Datasheets, Users Manuals, and Utility Software. The software is now installed and ready to run.

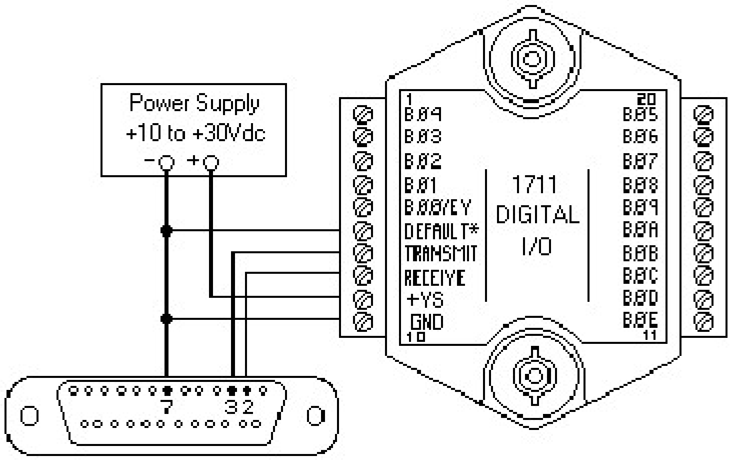
Figure 1.0, RS-232 Module Connection to Host PC:

The following connection diagrams illustrate connections between a D1700M module and a host computer serial port. The diagrams indicate connection to either a DB-9 or DB-25 pin serial port connector. Determine your connector type and use the appropriate connection drawing below.

Please note that these diagrams will work for any DGH RS-232 module regardless of model number. The connections are always the same during the configuration and setup process.



Connection to DB-9 host computer serial port



Remove connection from Default* to Gnd when configuration is completed.

Connection to DB-25 host computer serial port

Notes:

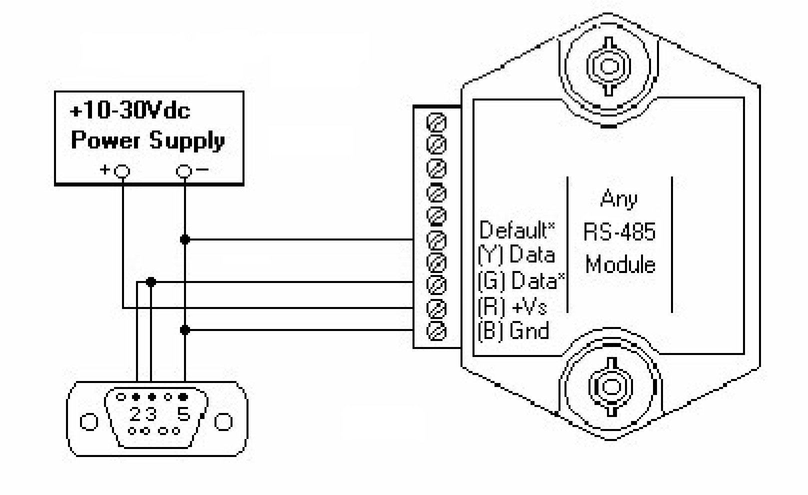
1. The module DEFAULT* pin is connected to GND in each of these RS-232 illustrations. Whenever, DEFAULT* is grounded, the module will only communicate at 300 baud. The parity setting should be No parity, eight data bits and one stop bit. Some computers may require Mark parity and seven data bits.

Figure 2.0, RS-485 Module Connection to Host PC:

The following connection diagrams illustrate how to connect a RS-485 module to a host computer. The first illustration shows how to connect an RS-485 module to an RS-232 DB-9 host computer serial port. This illustration should only be used for configuration of modules. It is not recommended, or suitable for installed applications.

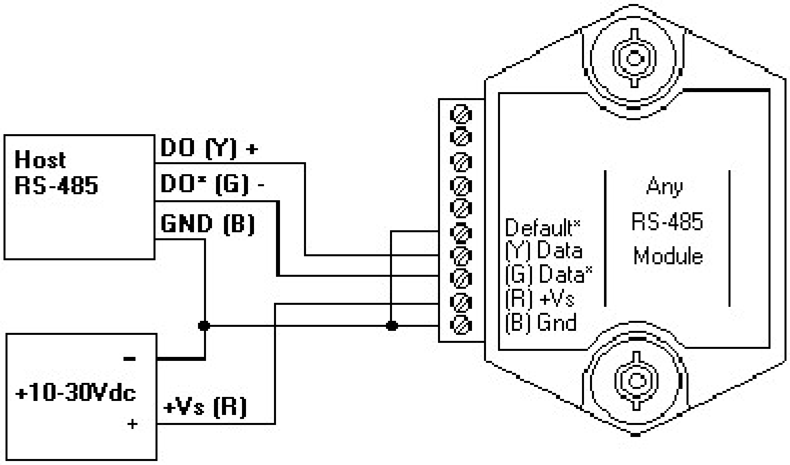
The second drawing illustrates connections to a RS-485 host serial port. The RS-485 data line polarity is indicated and should be noted when making the connections.

RS-485 Module to RS-232 Serial Port:



Connection to DB-9 host computer serial port

RS-485 Module to Host RS-485 Serial Port:



The polarity of the DGH RS-485 lines is indicated above. It should be noted that ‘(Y) Data’ is the Positive (+) line. The ‘(G) Data*’ line is the Minus (-) line.

Note: The module DEFAULT* pin is connected to GND in each of these RS-485 illustrations. Whenever, DEFAULT* is grounded, the module will only communicate at 300 baud. The parity setting should be No parity, eight data bits and one stop bit. Some computers may require Mark parity and seven data bits.

Setup Overview:

The D1700M contains two different protocol sets: the Modbus RTU protocol and the D1700 ASCII protocol. The D1700M ships from the factory in the D1700 ASCII protocol mode. Users must configure the module in the ASCII protocol before installation in a Modbus application.

The D1700 ASCII protocol makes configuring the module easier with programs such as Hyper-Terminal or the Utility Software. Further, configuration should always be performed with the DEFAULT* line connected to ground as shown in Figures 1 and 2.

While the DEFAULT* line is connected to ground the module will only communicate in the D1700 ASCII protocol mode at 300 baud, mark parity, seven data bits, and one stop bit. This mode provides a known set of communications parameters and the module will answer to any ASCII character address. Thus, allowing easy access to setup information. Setup values such as the Modbus Slave Address, application Baud Rate and Parity Type must be determined and written into the module. In addition to writing the new setup values, the Modbus RTU protocol must be ‘enabled’ in the module before installing the module into the application.

Utility Software Setup Instructions:

1. To run the software, click on the Windows “Start” button in lower left corner of the screen.
2. Select “Programs” and select “Dgh Data Acquisition”
3. Select “Utility Software” and select “Utility Software 2000” to run this program.
4. From the software main menu select “Edit” and “Serial Ports”
5. Select the correct COM Port, COM1:..COM32: and press “Open” button. All other port parameters are configured for a module in DEFAULT mode.
6. Plug the cable into the host computer serial port.
7. Connect the module to host computer using cable illustrations in this document.
8. Turn the power on and at the software main menu select “Setup” and “Modules”.
9. Ensure that the proper COM Port is displayed.

10. The program default address of ‘1’ is usable when the Default* pin is connected to Ground.
11. Select the four-digit model number from the “Model” drop-down list box.
12. Press the ‘Read Setup’ button.
13. A progress bar will appear while the program is reading the module parameters. If no communications errors are displayed then the module is communicating properly with the host computer.
14. Make the required setup adjustments to the ‘on-screen’ values and ‘APPLY’ the changes.

After the new settings have been ‘APPLY’d then you can disconnect the module from the 10-pin connector, insert the next module, and repeat this same process for the remaining modules. Once all the modules have been configured then they can be installed into the system application.

Manual Setup Instructions

The D1700M can be configured manually using a terminal emulation program such as Windows Hyper-Terminal. This configuration method will require an understanding of the binary and hexadecimal numbering systems to interpret the setup configuration tables in the D1700M Users Manual. The setup configuration tables illustrate the user-selectable bits that control features such as the module Baud Rate, Parity Type, and number of stop bits. Once the new bit values are determined then use the SetUp (SU) command to send the new value.

The Modbus Protocol mode can be manually enabled using the MBR command. All of this information can be found in the D1700M Users Manual. The MBR and MBD commands are described further down in this document.

Overview

Once the configuration process is completed then the D1700M modules can be installed into the system application. The D1700M modules will use the Modbus RTU protocol mode to communicate with the host Programmable Logic Controller (PLC) or PC based computer. The Modbus RTU protocol is widely used in data acquisition and control applications that utilize either a PLC or a PC based computer and data acquisition software program.

Data acquisition software programs and Modbus OPC servers are commonly used in process control applications today to read and write values to control modules. All commercial data acquisition software programs in the market today include a Modbus RTU protocol driver, providing instant connectivity to your D1700M module. The D1700M analog output module can be used to control devices such as valves, actuators, annunciators and more. Consult the factory if you need further assistance with compatibility questions about a specific program.

The Modbus RTU protocol handles data values in registers and discrete I/O bits as numbers. A complete register map of locations, functions and data values can be found at the end of this document.

The D1700M Modbus RTU protocol mode supports five functions that are part of the Modbus RTU protocol specification. The function numbers and their descriptions have been detailed below. If you need more information about the Modbus RTU protocol then you can find it on the internet at www.modbus.org.

D1700M Modbus RTU Protocol

The D1700M series modules support several of the Modbus RTU protocol functions. These functions read and write values to register locations within the D1700M. The values and register locations control features such as the Digital Outputs and Reading the Digital Inputs. Table 1.0 below contains a list of the supported Modbus functions and their descriptions.

Function	Description
01	Read Coil Status
03	Read Holding Registers
05	Force Single Coil
06	Preset Single Register
15	Force Multiple Coils

Table 1.0. D1700M Modbus Functions.

Modbus RTU Command Structure

The Modbus RTU serial protocol operates using a command and response protocol. The command structure begins an address, commonly referred to as the Modbus Slave Address. The address is followed by the command value or the Modbus function number. The function number indicates what action the device should perform. The required data values for the command function will follow and the message is terminated by a cyclical redundancy check (CRC) value.

The response message contains the same address and function values along with the requested data. Each response is terminated by a CRC checksum of the response bytes.

The Modbus protocol contains provisions for error checking. The Modbus standard contains a standardized set of values that can be returned from a Modbus Slave device when a problem or error is detected. These error values are referred to as “exception codes” and the DGH implementation of the Modbus RTU protocol utilizes these exception codes when applicable. See Table 3.0 for a complete list of supported exception codes.

A typical Modbus RTU command to read the digital input values from a D1700M module whose slave address is 0x03 would be:

```
Command:    03  03  00  00  00  01  85  E8
Response:   03  03  02  00  7D  01  A5
```

In the event that an error is detected, an error response would be generated and return one of the standardized error codes. The most significant bit (MSB) of the function number will always be set in any exception response. Therefore, an error response message that was generated from a function ‘03’ command would return the function value of 0x83. The value is followed by the standardized error code (02) and the two-byte CRC checksum. A typical exception response would be:

```
Command:     03  03  00  0A  00  01  A5  EA
Response:    03  83  02  61  31
```

Modbus RTU Functions

Function (01) Read Coil Status

Modbus function (01) Read Coil Status will read the status of discrete digital inputs. Modbus relay input coils are considered the digital input bits on the D1700M series modules.

Exception error messages will be generated when attempting to read more digital inputs than contained in the D1700M device. The following example can be used to read the status of the first three digital input bits:

```
Command:     03 01 00 00 00 03 7D E9
Response:    03 01 01 07 11 F2
```

In the command string:

03	is the slave address
01	is the Read Coil Status command
00 00	is the starting coil number
00 03	is the number of bits to read
7D E9	is the CRC to this message

In the response string:

03	is the slave address
01	is the Read Coil Status command
01	is the number of data bytes returned
07	is the status of B00, B01, and B02 bits, set High, Logic “1”
11 F2	is the CRC for this message

Function (03) - Read Holding Registers

Read Holding Register function (03) can be used to read data values from registers within the D1700M module. This function supports reading of up to three registers starting from Modbus slave register 40001 through 40003.

Some data acquisition and monitoring programs and some programmable logic controllers (PLC) may use zero based register addressing. In that case, the registers are addressed starting from zero meaning registers 40001-40003 are addressed as 0-2. Consult your software or PLC manual for the proper addressing mode.

The data for each register is returned as two bytes that represent a 16-bit binary value. The binary values can range from 0000-FFFF (hexadecimal). Consult the register map in Table 4.0 to determine the exact meaning of the response value in each register.

A typical command and response to read the Events Counter LSB value from a D1700M whose slave address is ‘03’ would be:

Command: 03 03 00 02 00 01 24 28
Response: 03 03 02 00 00 C1 84

In the command string:

03   is the slave address
03   is the Read Event Counter LSB Register command
00 02   is the starting register to be read (Register 40003)
00 01   the number of registers to be read
24 28   is the CRC for this message

In the response string:

03   is the slave address
03   is the command
02   indicated the number of data bytes in the message
00 00   is the Event Counter LSB data
C1 84   is the CRC for this message

Function (05) - Force Single Coil

The Force Single Coil function is used to turn “On” and “Off” digital output bits on a D1700M. The Modbus specification refers to digital outputs as “Output Coils”. Function 05 can turn on and off any bit that is configured as a digital output in the D1700M.

The number of digital outputs available in a D1700M is determined by both the model number and the “Setup” message within the module. The D1700M modules each contain up to 15 physical digital output bits. The D1700M board level products contain up to 64 bits of digital outputs.

The Setup Message contains a user-selectable variable called Word Length that determines how many bytes of physical I/O bits are available. Word Length values from 1..8 are possible, indicating 8..64 individual bits in the product being used. Since this value is user-selectable it is possible to set the D1700M such that the number of I/O bits is less than the physical quantity.

Modbus Address Exception error codes will be returned when attempting to turn “On” or “Off” a bit whose address is higher than the quantity of bits enabled in the Setup message.

A typical command and response to turn “On” digital output B00
would be:

Command:	03 05 00 00 FF 00 8D D8
Response:	03 05 00 00 FF 00 8D D8

In the command string:

03	is the slave address
05	is the Force Single Coil command
00 00	is the bit number to turn on (Bit # B00)
FF 00	the Turn On data value
8D D8	is the CRC for this message

In the response string:

03	is the slave address
05	is the Force Single Coil command
00 00	is the bit number to turn on (Bit # B00)
FF 00	is the Turn On data data
8D D8	is the CRC for this message

Function (06) - Preset Single Register

The Preset Single Register function (06) can be used to write values to a single register. Primary uses for this command are writing to the control register in the D1700M module.

This command can be used to suspend the Modbus RTU protocol mode and return the D1700M back to the DGH ASCII protocol mode. Write a value of 0000 to Modbus register 40001 to temporarily suspend the Modbus RTU protocol mode. The module will then communicate using the D1700M ASCII protocol command set.

The D1700M ASCII protocol command set can be used to verify or alter setup data or for troubleshooting purposes. The module will continue to communicate using the ASCII protocol until either a Remote Reset (RR) command (RR) is received or the power is cycled. At which time, the module will return to the Modbus RTU protocol mode. Refer to the D1700M ASCII Modbus Disable command (MBD) for more information on permanently disabling the Modbus RTU protocol mode.

The following command will write a value of one to the Control Register register, causing the D1700M to reset the Events Counter.

Command:	03 06 00 00 00 01 49 E8
Response:	03 06 00 00 00 01 49 E8

In the command string:

03	is the slave address
06	is the Preset Single Register command
00 00	is the starting register to be read (Register 40001)
00 01	specifies the data value written.
49 E8	is the CRC for this message

In the response string:

03	is the slave address
06	is the command
00 00	is the starting register to be read (Register 40001)
00 01	specifies the data value written.
49 E8	is the CRC for this message

Function (15) - Force Multiple Coils

The Force Multiple Coils function (15) can be used to turn on and of the D1700M digital output bits. The digital output bits begin at address location 00001 and there are up to 00064 bits that can be controlled using this function.

The following command will turn on bits 00001 (B00) through 00007 (B07).

Command: 03 0F 00 00 00 08 01 FF BF 04
Response: 03 0F 00 04 00 01 41 EA

In the command string:

03 is the slave address
0F is the Force Multiple Coils command
00 00 is the starting register to be read (Coil 0001)
00 08 specifies the quantity of coils written.
01 is the data byte count
FF is the data value written
3F 0C is the CRC for this message

In the response string:

03 is the slave address
0F is the command
00 00 is the starting register to be read (Register 40005)
00 08 specifies the quantity of registers written.
55 EF is the CRC for this message

Modbus Exception Codes

The Modbus protocol contains provisions for error checking. The Modbus standard contains a standardized set of values that can be returned from a Modbus Slave device when a problem or error is detected. These error values are referred to as “exception codes” and the DGH implementation of the Modbus RTU protocol utilizes these exception codes when applicable. See Table 3.0 below for a complete list of supported exception codes.

In the event that an error is detected, an error response would be generated and return one of the standardized error codes. The most significant bit (MSB) of the function number will always be set in any exception response. Therefore, an error response message that was generated from a function ‘03’ command would return the function value of 0x83. A typical exception response would be:

```
Command:     03  03  00  0A  00  01  A5  EA
Response:    03  83  02  61  31
```

The following error codes may be returned from a DGH D1700M Modbus RTU compatible module.

DGH Modbus Exception Codes		
Exception	**Name**	**Description**
01	Illegal Function	This exception code is generated when the module does not recognize the function code.
02	Illegal Data Address	This exception code is generated when the module does not support the specified data address in the command.
03	Illegal Data Value	This exception code is generated if the command data is out of range for the function.
06	Slave Busy	This exception code is generated during the 3 seconds after the module is reset or powered up.

Table 3.0. DGH Modbus Exception Codes.

D1700M Modbus Register Map

Function	Register	Direction	Value	Description
03	40002	Read	0.00FF	Events Counter HI
03	40003	Read	0.FFFF	Events Counter LO
01	00001	Read	0,1	Digital Input/Output B00
01	00002	Read	0,1	Digital Input/Output B01
01	00003	Read	0,1	Digital Input/Output B02
01	00004	Read	0,1	Digital Input/Output B03
01	00005	Read	0,1	Digital Input/Output B04
01	00006	Read	0,1	Digital Input/Output B05
01	00007	Read	0,1	Digital Input/Output B06
01	00008	Read	0,1	Digital Input/Output B07
01	00009	Read	0,1	Digital Input/Output B08
01	00010	Read	0,1	Digital Input/Output B09
01	00011	Read	0,1	Digital Input/Output B0A
01	00012	Read	0,1	Digital Input/Output B0B
01	00013	Read	0,1	Digital Input/Output B0C
01	00014	Read	0,1	Digital Input/Output B0D
01	00015	Read	0,1	Digital Input/Output B0E
.	.	.	.	.
01	00064	Read	0,1	Bit #B3F on H1770
05	00001	Write	B00	Digital Input/Output B00
05	00002	Write	B01	Digital Input/Output B01
05	00003	Write	B02	Digital Input/Output B02
05	00004	Write	B03	Digital Input/Output B03
05	00005	Write	B04	Digital Input/Output B04
05	00006	Write	B05	Digital Input/Output B05
05	00007	Write	B06	Digital Input/Output B06
05	00008	Write	B07	Digital Input/Output B07
05	00009	Write	B08	Digital Input/Output B08
05	00010	Write	B09	Digital Input/Output B09
05	00011	Write	B0A	Digital Input/Output B0A
05	00012	Write	B0B	Digital Input/Output B0B
05	00013	Write	B0C	Digital Input/Output B0C
05	00014	Write	B0D	Digital Input/Output B0D
05	00015	Write	B0E	Digital Input/Output B0E
.	.	.	.	.
05	00064	Write	B3F	Bit #B3F on H1770
06	40001	Write	0	Temporarily suspend Modbus RTU protocol and return back to DGH ASCII protocol.
06	40001	Write	1	Clear Events Counter D1711, D1712 Only.
15	1..64	Write	-	Turn On/Off BO0..B3F

Table 4.0. DGH Modbus Register Map.

Modbus RTU Enable (MBR)
To place any D1700M module in Modbus protocol mode use the Modbus RTU (MBR) command. The MBR command must be used to specify the Modbus device address and enable the Modbus protocol mode. The device address consists of a two character hexadecimal value and is stored in EEPROM. The two-byte address specified is translated to a one byte, 8 bit address required by the Modbus protocol. The example below can be used to specify a Modbus slave address of “01”.

```
Command:    $1MBR01
Response:   *

Command:    #1MBR01
Response:   *1MBR019D
```

Note: This address value is different and separate from the address value specified in the Setup (SU) message of the module. After the Modbus address is specified, a reset is necessary to activate the Modbus protocol mode. The reset may be accomplished in one of three ways:

1) Removing power for about 10 seconds to perform a power-up reset.
2) Momentarily grounding the Default* pin.
3) Issue a Write Enable (WE) command followed by a Remote Reset (RR) command.

After a reset is performed, the module is in Modbus protocol mode.

Modbus Disable (MBD)
The Modbus Disable (MBD) command is used to disable the Modbus RTU protocol. Any D1700M series module in Modbus mode can be returned to D1700 ASCII protocol mode by connecting a jumper wire between module pins GND and Default* pin. This places the module in Default Mode, where the module will only communicate at 300 baud, no parity, D1700 ASCII protocol, and answer to any address. While in Default mode, transmit an MBD command to internally disable the Modbus protocol.

Following the MBD command a device reset must occur. The reset is necessary to activate the D1700M ASCII protocol. A reset can occur by removing the Default* jumper, performing a power-up reset or by transmitting a Write Enable (WE) and Remote Reset (RR) command sequence. After a reset is performed, the module is in D1700M ASCII protocol mode.

Command: $1MBD
Response: *

Command: #1MBD
Response: *1MBD2E

Testing and Troubleshooting:

1. The RS-485 type D1700M modules “**cannot**” be tested in the Modbus RTU protocol mode using the DB-9 connection in Figure 2.0. The RS-485 Quick Hook-up Diagram to an RS-232 serial port causes the command to be read back into the computer before the response from the module. This causes parsing problems with Modbus drivers. The diagram in Figure 2.0 is only for setup and configuration of the module.

2. Communications with USB converters can sometimes lead to Modbus Timeout Errors. Meaning, that it appears as though the D1700M does not respond in Modbus RTU protocol mode. This is sometimes caused by converters that place gaps in between the data characters in the transmitted command. If a converter places gaps in between the characters, and the gap is more than 3.5 character times at the specified baud rate then the module will consider the message ended. Therefore, it will not receive a complete message and will never respond. This problem usually appears at the higher baud rates where the character times are much quicker.

3.