MFT B-Series Serial Communications

The Kurz Instruments MFT B-Series (Mass Flow Transmitter) Models have three modes of data communications using the USB or RS-485 interface. They are available on the MFTB firmware version 1.x or later.

The three communication modes are as follows:

1. Remote Terminal
2. Data Logging
3. Modbus Protocol, ASCII and RTU

The Remote Terminal and Data Logging modes of data communication use a common serial port of the microprocessor therefore only one mode can be used at time.

The Remote Terminal mode is used with the USB interface and requires the Kurz USB Driver to communicate with a computer terminal (PC or Laptop). This mode can be used to setup the configuration of the MFT B-Series and to Upload or Download the configuration files.

The Data Logging mode is used with the USB interface and requires the Kurz USB Driver to log data from the MFT B-Series. Log output is in a comma separated value (CSV) format for ease of importing into a spreadsheet program. The data logging can be setup to log with a periodic time interval. It is advisable to turn off the terminal echo to prevent undesirable data from being transmitted along with the logged data (when terminal echo is active, data displayed on the meter's local LCD display is echoed to the remote terminal through the USB port and will be logged along with the periodic log data).

The Modbus mode is used with RS-485 (point to point or multi-drop). The RS-485 can only be configured as Half Duplex. Modbus is a network communication protocol that was originally developed then released in 1979 by Modicon (*owned by Schneider Electric)* for their controllers. The controllers can communicate using master-slave technique, in which only one device (*the master)* can initiate transactions (*called the queries).* The other devices (*the slaves)* respond by supplying the requested data to the master, or by taking the action requested in the query. The master can address individual slaves, and the slave will respond by returning a message (*called response)* to the query that is individually addressed to it. The Modbus protocol establishes the format for the master's query by placing the following into the query: device address, a function code defining the requested action, any data to be sent and an error-checking field. The slave's response message is also constructed using Modbus protocol. It contains a field confirming the action taken, any data to be returned and an error-checking field. If an error occurred in receipt of the message or the slave is unable to perform the requested action, the slave will construct an error message

and send it as its response. The ASCII or RTU transmission modes are available in the MFT B-Series models.

Communication Setup

Portions of this section are also located in the Installation section under USB Serial Communications.

Setup for Remote Terminal Mode

Remote Terminal communication mode is a default. This is provided through RS232 serial communication via a standard USB connector on the SC board. Remote Terminal communication allows the user to interface with the MFT B-Series via a computer running a terminal emulation program. It also allows the user to use the computer keyboard to navigate the meter's onboard menu system.

To use the serial interface, connect the flow meter to a PC via a standard USB cable. The cable is a USB Type A Male to USB Type Mini-b male, similar to the one shown in Figure J-1.

Figure J-1, USB 2.0 A to Mini-b 1
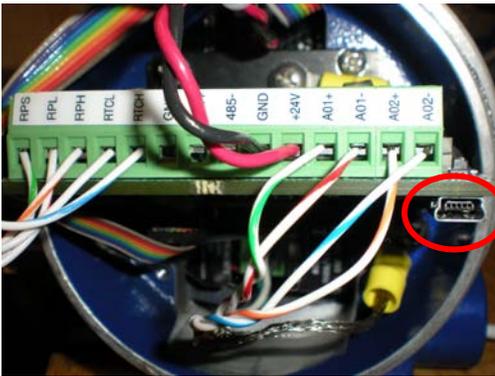
The USB port on the flow meter can be accessed by removing the enclosure lid on the electronics side of the meter. The USB port is located on the SC board as shown in Figure J-2.

The meter must be turned on to establish communication with the remote terminal software and setup the communications parameters.

Figure J-2, MFTB USB Connector

Any terminal emulation program may be used to act as a remote terminal to the MFT B-Series. HyperTerminal, which comes with all Windows versions prior to Windows Vista, can be used. However, HyperTerminal has known issues with correctly opening the port. We recommend using Tera Term 4.62 or newer. Both programs support Xmodem for transferring the MFT B-Series configuration files. This takes about five seconds at 9600 baud, the only baud rate supported by the Kurz USB driver. The program KzComm may also be used to interface with the meter through the USB port, but it is not a terminal emulator.

If HyperTerminal is not installed on your computer it may be added by going to the Add/Remove Programs within the Control Panel and clicking the Windows Setup tab. Then choose the Communication group and within it you will find the button to select HyperTerminal. After this you may be asked for your Windows disk to continue with the installation. Refer to your Windows documentation if you need more information on HyperTerminal. PCs using Windows Vista as the operating system do not include HyperTerminal so you will need to download and install it or another terminal emulator program. HyperTerminal is available for purchase and download from its designer at the following web site: http://www.hilgraeve.com/hyperterminal.html.

Tera Term, our recommended terminal emulator, is freely available for download at the following websites. Open source version 4.65, at the time of this writing, can be found at http://ttssh2.sourceforge.jp/ and version 3.13, which is limited to COM ports 1 through 4, can be found at http://www.ayera.com/teraterm. The open source version does not have the COM port limitation. Tera Term works with Windows 95 through 7 and has English, Japanese, and German versions. You may want to put a shortcut to this program in your Start menu and on your desktop.

The following screen shots show the communication and setup of the TeraTerm terminal emulator program with the MFT B-Series flow meter. The example assumes that the Kurz USB driver has previously been installed on the computer that will be running TeraTerm. The flow meter must be connected to the PC using a USB cable described above and the meter must be powered on.

At the PC, start TeraTerm. The following window will appear when TeraTerm is first started.
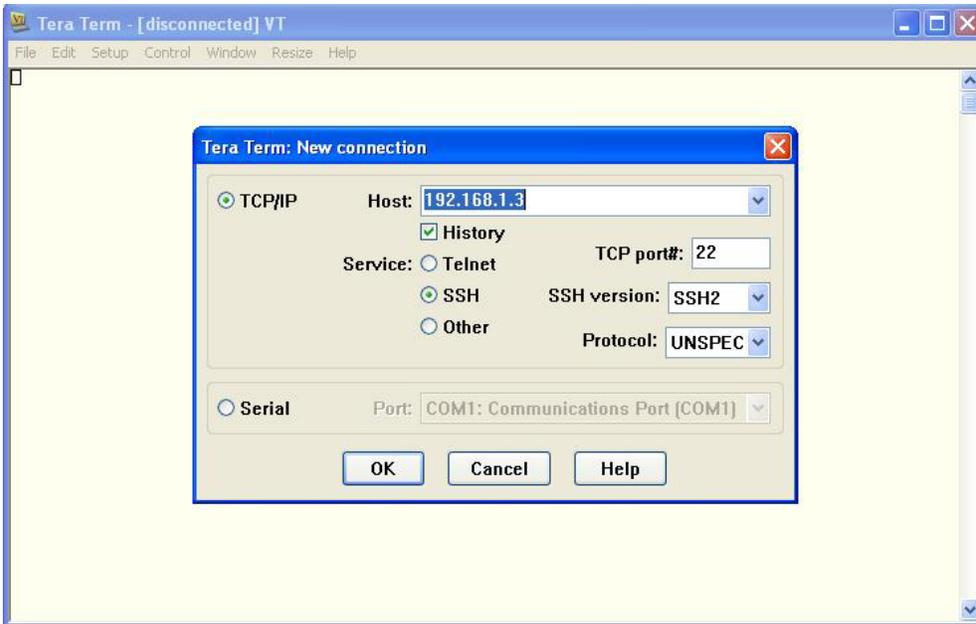
Select the "Serial" radio button and select the COM port assigned to the Kurz Instruments USB Driver. It will be identified as 'COMx: Kurz USB-HID -> COM Device (COMx). In the example, the Kurz USB Driver has instantiated as COM3. The actual COM port designation may be different on your PC.
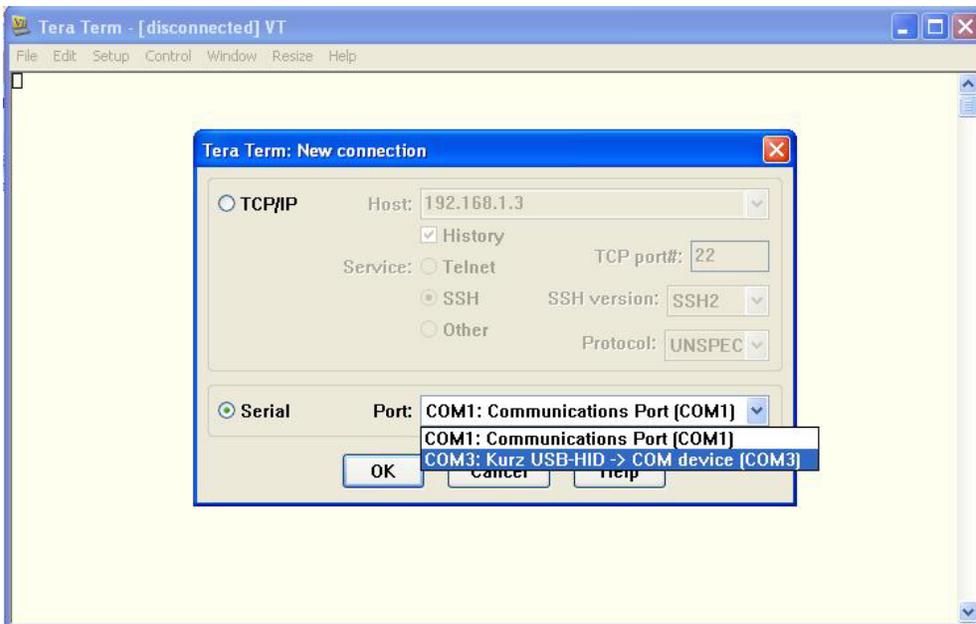
TeraTerm auto configures the COM Port based on the Windows Device Manager setting for the COM Port so the initial communication parameters may need to be changed in order to communicate properly with the meter. The TeraTerm terminal screen may display unrecognizable characters as a result.
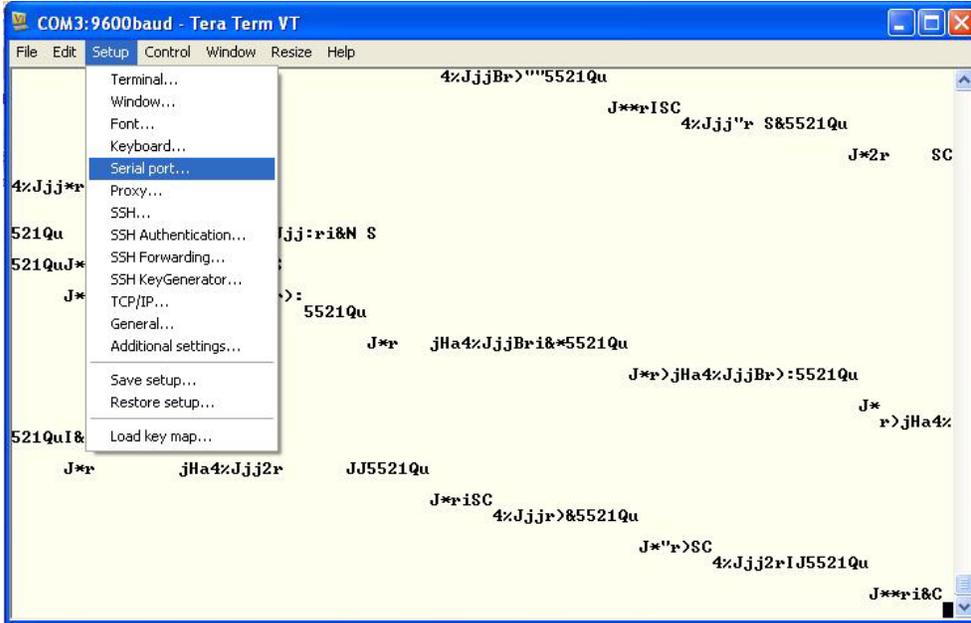
Click on the Setup menu and select 'Serial port…' from the pulldown list. The Serial Port Setup window will appear.
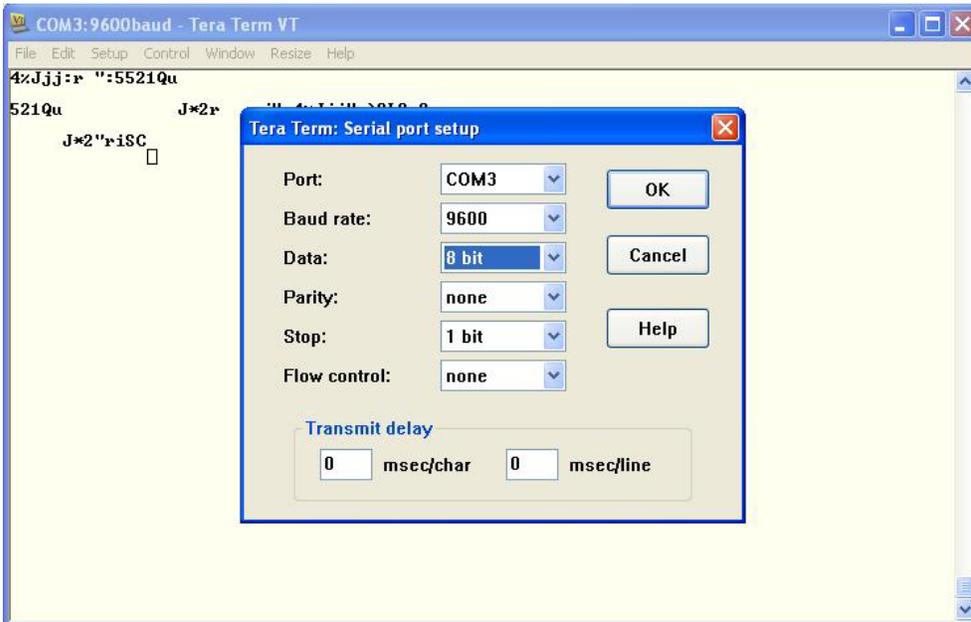
Set the communications port parameters as follows:
1. 9600 baud
2. 8 data bits
3. 1 stop bit
4. No parity
5. No flow control

Press 'OK' after adjusting the serial port parameters. If unrecognizable characters are still being displayed after entering the proper communication port parameters, reset the port by clicking on the 'Control' menu and selecting 'Reset port' from the pulldown list.
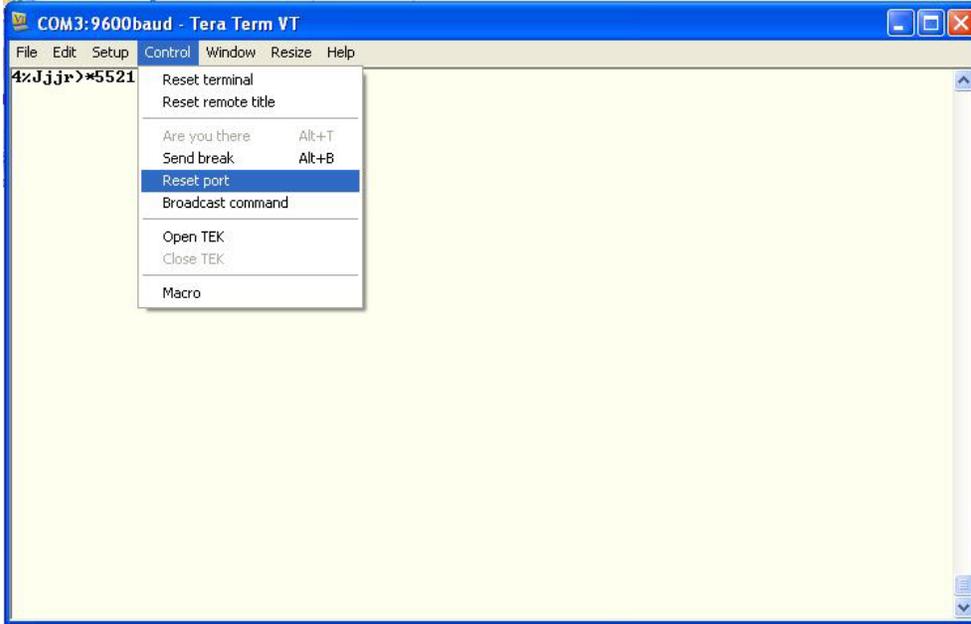
When synchronized communication is established, the MFT B-Series will echo all the information that is being displayed on the local LCD display to the terminal emulator program on the computer (ie Terminal Echo is ON).
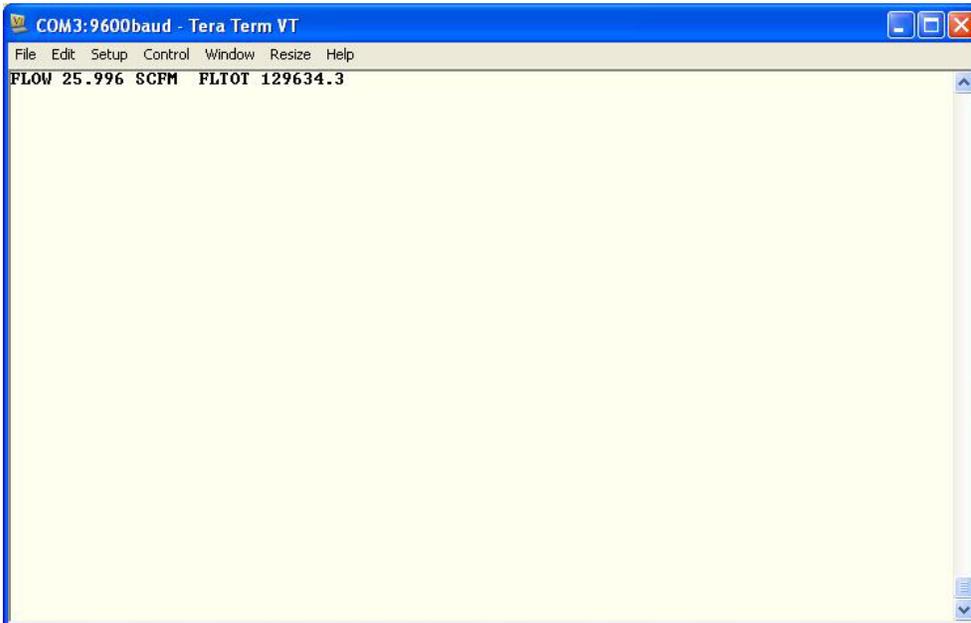
For the example above, the local LCD screen on the meter would have the following displayed:

```
FLOW 25.996  SCFM
FLTOT 129634.3
```

Termina Echo can be turned OFF by pressing the <SHIFT>+ key (ie press the plus key '+', while simultaneously pressing the <SHIFT> key). However, the Terminal Echo must be ON in order to use the computer keyboard to navigate the meter's onboard menu system using the terminal emulator program.

The keys used on the computer keyboard to navigate the meter's menu system are the lowercase equivalent of the keys used on the meter keypad. The Table below defines the navigation keys on the PC keyboard and the local keypad.

Keypad	Remote Keyboard	Function
P	p (lowercase P)	Invoke Program Mode (requires an Access Code); Or Skip over data with no entry, during data entry
D	D <BACKSPACE>	Invoke Display Mode (no Access Code required); Or delete 1 character to the left, during data entry
L	l (lowercase L)	Invoke Log Mode (no Access Code required)
E	<ENTER>	Invoke Extended Utilities Mode (requires an Access Code); Or accept a data entry, during data entry
C	c (lowercase C)	Clear the displayed/entered value, during data entry; Or acknowledge an active system fault
H	h (lowercase H)	HOME - Return to Run Mode; Or back out of a menu
	+	Toggles terminal echo OFF/ON
^ / YES	^ (Shift-6)	Scroll the selection list forward
v / NO	v (lowercase V)	Scroll the selection list backward
-	-	Minus (hyphen) used in numeric and text data
.	.	Period (decimal) used in floating point and text data
(0-9)	(0-9)	Number keys for numeric data including access codes and entering text data on the local keypad

MFT B-Series ASCII Commands

The MFT B-Series allows the user to access internal data remotely using a PC or Laptop computer with a USB Port and Kurz USB Driver (DCN 451035) via ASCII commands. Using a terminal emulation program, the user can manually type in the ASCII commands and the command response can be viewed on the PC terminal. Or a custom Host program can send the ASCII commands to collect the process data from the meter. The format of the ASCII command is:

> *<ESC>*[command]*<CR>*

> where:
> > <ESC> is the escape character 0x1B; or ESC key

[command] is one of the commands listed in the Table below
<CR> is the carriage return character 0x0D; or ENTER key

The response values are sent as printable ASCII characters terminated by
<CR><NL> (carriage return character followed by the newline character).

Some of the ASCII commands will initiate an XMODEM send or receive in order
to transfer blocks of data between the PC and the flow meter.

The table below is the list of supported ASCII commands. Note that the PC
commands are case sensitive and must be in lowercase as shown.

ASCII Command	Description
download	Download a configuration file from the PC to the MFT B-Series; this command initiates an XMODEM send
upload	Upload the configuration file from the MFT B-Series to the PC; this command initiates an XMODEM receive
qflow	Query the Flow Rate
qvel	Query the Velocity
qtemp	Query the Temperature
qmeterid[*]	Query the Flow Meter's Tag Name
qrppower	Query the Rp Power
qrpcurrent	Query the Rp Current
qrpres	Query the Rp Resistance
qrptemp	Query the Rp Temperature
qrtcres	Query the Rtc Resistance
qsnumber[*]	Query the Flow Meter's Serial Number
qruntime	Query the Run Time Counter
qfunit	Query the Flow Units
qmeter1[*]	Query the Flow Meter's flow data information
qai1[*]	Query the Flow Meter's External Input information
qao1	Query the Analog Output channel #1
qao2	Query the Analog Output channel #2
qalarm	Query the Alarm status

[*] These commands are only available in firmware versions 1.05 and newer.

The ASCII commands should be used while the flow meter is in *Run Mode*.

Refer to the topic "Setup for Remote Terminal Mode" for the setup.

Here is an example of how to query the velocity of the MFT B-Series from a PC
running with a terminal emulator program. In this example the HyperTerminal

emulator program was used. Once the communication between the MFT B-Series and the PC has been established, if the terminal echo is ON (HyperTerminal is receiving data that is displayed by the MFT B-Series) turn it OFF by typing the plus '**+**' key on the keyboard for a cleaner screen. The terminal echo will stop after the MFT B-Series receives the command. Carefully type in the command since it will not be displayed on the terminal screen. Start with the Escape key followed by **qvel** and terminate with Enter. Once the MFT B-Series receives the command it will respond with the velocity reading at that time. The command and response look like the line below;

> Command   Response
> *<ESC>*qvel*<CR>*   1000.00

The format for the qmeter1 and qai1 outputs are in a comma separated value format.

qmeter1:
>
> flow meter ID, runtime (hrs), flowrate, MUNIT, totalizer, TOTUNIT, elapsed time (min), velocity, VUNIT, ref. density, density unit, flow area, Area unit, correction factor, PRP or IRP, PRP/IRP value, PRP/IRP units, raw flow or velocity, raw flow or velocity unit

qai1:
>
> One of the following formats will be used depending on the External Input usage:
> > current (mA)   or
> > current, current units, engineering scaled value, engineering units

UPLOAD and DOWNLOAD Commands

These two commands will allow the user to save the configuration of the MFT B-Series to the PC disk drive and update the configuration of the MFT B-Series from the PC using the configuration file. These ASCII commands will initiate an XMODEM file transfer. The following are examples of how to save and update the configuration of the MFT B-Series using the UPLOAD and DOWNLOAD command, respectively. These examples assume that the MFT B-Series and the PC or Laptop have already established communications using the HyperTerminal terminal emulation program.

Saving the Configuration to the PC disk drive

If the terminal echo is ON, press the plus '**+**' key on the keyboard to toggle terminal echo OFF. Type in the command *<ESC>*upload*<CR>,* the MFT B-Series will respond with a prompt to initiate an XMODEM file transfer. The following will be displayed at the HyperTerminal window

>MFT-B Ready to Transmit File
>XMODEM Receive File from MFT-B

From this point, click Transfer on the HyperTerminal menu bar and click Receive File, Figure 2 will be displayed.

Figure 1: Receive File

Enter or browse to the folder where you want to save the configuration file. Select Xmodem under 'Use receiving protocol'.   Click the Receive button when you are done. A new window, as shown in Figure 2, will be displayed. Enter the filename of the configuration file with a .cf extension (e.g. FT20001A.cf). Press the OK button to continue.
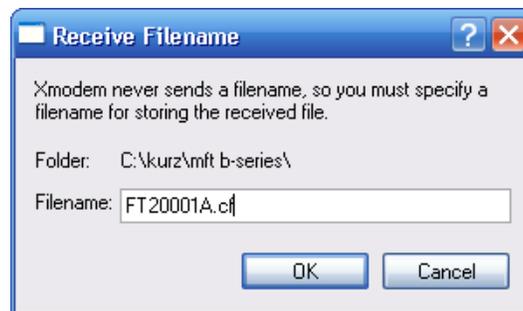
Figure 2: Receive Filename

At this point the Xmodem protocol is receiving the file as seen on the Xmodem activity window shown in Figure 3.
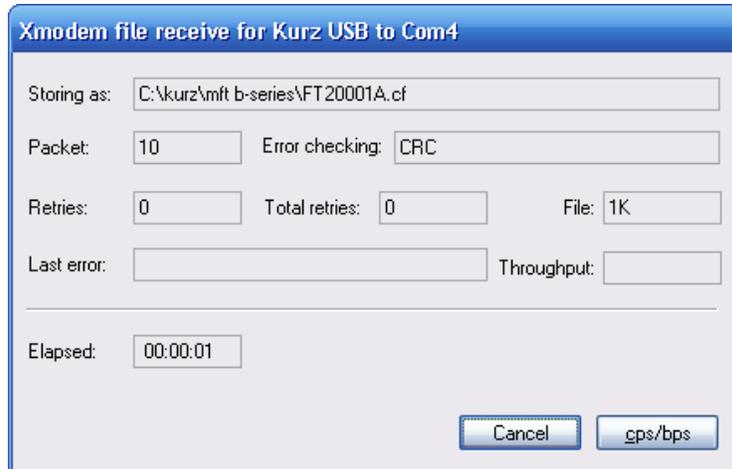
Figure 3: Xmodem Receive Activity Window

Updating the MFT B-Series Configuration from a Configuration File

If the terminal echo is ON, press the plus '**+**' key on the keyboard to toggle terminal echo OFF. Type the command <*ESC*>download<*CR*>, the MFT B-Series will respond with a prompt to initiate an XMODEM file transfer. The following will be displayed at the HyperTerminal window

>MFT-B Ready to Receive File
>XMODEM Transmit File to MFT-B

From this point, click Transfer on the HyperTerminal menu bar and click Send File, Figure 4 will be displayed.

Figure 4: Send File

Enter or browse the filename of the configuration file to be sent and select Xmodem under the 'Protocol' option pulldown. Click the Send button when finished.

Note:
During this process the command may time out and Xmodem can not successfully send the file. If this happens, try it again, this time it will be

faster since the folder for your file is already setup from the previous attempt.

After the Send button is clicked, the Xmodem activity window will popup, as seen in Figure 5.
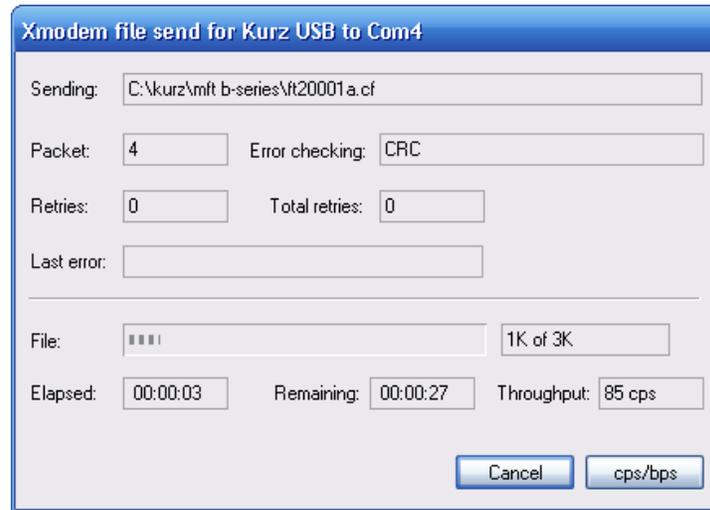
Figure 5: Xmodem Send Activity Window

An alternative method to perform the upload/download of the configuration file is through the Windows program KzComm that is supplied with the MFT B-Series unit. The User's Guide is Kurz Instruments DCN: 280128.

USB to PC Communication Problems

An issue between how Windows handles the virtual COM port created by the Kurz USB Device Driver for the PC can cause the terminal emulator program to display unrecognizable characters. This can sometimes be caused by not exiting or disconnecting from the terminal emulator program before disconnecting or powering down the MFT B-Series unit connected to the PC via the USB cable. The following solutions can be used to remedy this problem.
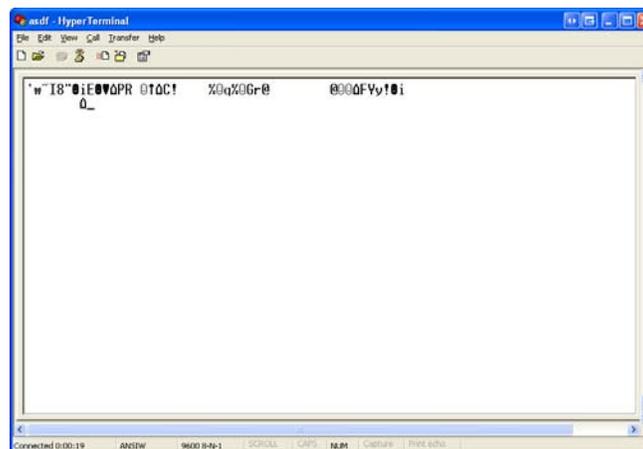
Figure 6: Terminal Emulator Displaying Gibberish

General Notes: Always exit/close the terminal emulator program before disconnecting or powering down the MFT B-Series unit.

KzComm: See Reset XMODEM COM Port in the KzComm Help or Users Guide.

Terminal Emulator:

    Solution 1: Use Tera Term 4.64 or latest (Supports COM1 through COM15).

        1. Open Tera Term.
        2. Connect
            a. Tera Term only allows connections to COM1 through 15. Use the Device Manager (type **devmgmt.msc** from the **Start->Run** window) to change the COM port to a lower number.
            b. Only the COM port needs to be chosen.
        3. If gibberish is seen then select **Control->Reset port** tabs in Tera Term.
        4. Save the setup by selecting **Setup->Save Setup...**
        5. Create a shortcut to Tera Term on your desktop.

    Solution 2: Use Tera Term Pro 3.1.3 (Only supports COM1 through 4).

    See Solution 1, but only supports COM1 through 4.

    Solution 3: Use HyperTerminal
        1. Setup HyperTerminal with the correct COM port, 9600 baud rate, 8 data bits, no parity, 1 stop bit, and no flow control.
        2. **File->Save As** and save it to your desktop.
        3. Open the connection. If gibberish is displayed, disconnect and exit HyperTerminal and repeat until the 3rd failure.
        4. If gibberish continues to be displayed after repeating the above steps, disconnect and close HyperTerminal. Open the Device Manager (type **devmgmt.msc** from the **Start->Run** window) to change the COM port baud rate to any other value not previously used and exit the COM properties window. Disconnect the USB from the Computer wait a few seconds and restart HyperTerminal.
        5. Go to Step 4 and repeat until working.

Setup for Modbus Mode

The Modbus communication mode is a default. It can be selected as either Modbus ASCII or Modbus RTU. RTU is set as the factory default. The MFT B-Series must be used only as a slave device and it will work only if used with a master device. The RS-485 Interface can be used to communicate with the MFT B-Series via the Modbus protocol. The RS-485 can only be used as Half Duplex and can be used in point to point or multi-drop (1 Master and 1 or more Slaves) operations. When used in a multi-drop network, each slave device must have a unique device address within the network. The individual slave device address can be assigned in the range of 1-247.

WARNING

Do not connect the MFT B-Series to the Modbus Network until it is properly configured.

The MFT B-Series may be setup for Modbus communications in *Program Mode* as follows:

Enter *Program Mode*, press **P**, the **654321** access code, and **E**. Press **2** to invoke the *Quick Jump* option entry method and select **Option #19** for the Modbus Communication Setup menu.

The meter will prompt for the device address of the flow meter.

```
DEV MODBUS ADDR
>1
```

The default address set at the factory is 1. Use the numeric keys to enter the device address and press **E** to accept the entry. The address can be in the range (1-247).

The meter will next prompt for the MODBUS MODE. The MODBUS MODE defines whether the Master/Slave device will communicate using the Modbus ASCII or Modbus RTU protocol.

```
MODBUS MODE
>MODBUS ASCII  ^v
```

Use the **^** or **v** key to select between MODBUS ASCII or MODBUS RTU and press **E** to accept the selection. The flow meter is set to MODBUS RTU at the factory.

If MODBUS ASCII is selected, the communication parameters are
> 7 Databits
> No Parity
> 2 Stop bits

If MODBUS RTU is selected, the communication parameters are
> 8 Databits
> No Parity
> 1 Stop bit

The meter will next prompt for the BAUD RATE or the rate of data transmission between the Master and Slave device (flow meter). The factory default is 38400 BPS.

```
MODBUS  BAUD RATE
>38400   BPS      ^v
```

Use the **^** or **v** to select the BAUD RATE used by the Modbus Master and press **E** to accept the selection. The selection list has the following Baud Rates to choose from 9600, 14400, 19200, 38400, 57600.

The next Modbus parameter that needs to be specified is the byte order of the modbus registers. This configuration is particularly important for the Modbus Master to correctly interpret the floating point data from the modbus registers. There are 2 options to select from - 'BYTE #1 2   3 4' and 'BYTE # 3 4   1 2'. These indicate the order of the modbus registers when 2 registers are used for a device parameter. The factory default is 'BYTE #1 2   3 4' which means that the low order byte is sent first, followed by the high order byte (little-Endian style). The selection 'BYTE # 3 4   1 2' means that the high order byte is sent first, followed by the low order byte (big-Endian style).

```
REGISTER ORDER
>BYTE #12   34   ^v
```

Use the **^** or **v** keys to scroll between the BYTE ORDER selections and press **E** to accept the selection or **P** to skip the data entry. Pressing **E** or **P** at the REGISTER ORDER screen will exit from the Modbus Communication Setup menu.

Note: If an issue occurs with reading floating point numbers try changing the REGISTER ORDER parameter.

MFT B-Series Modbus Command and Registers

The MFT B-Series firmware version 1.00 and later supports the reading and writing of coils and registers. Firmware version 1.03 and later support device identification.   Coils are represented by a single bit. A register is defined as a 16-bit storage of data. This means that a 16 bit integer number will occupy one register, 32 bit floating point numbers will occupy two registers, and two 8 bit ASCII characters are stored in one register. Note that registers with ASCII characters are represented as a char array and need to be terminated with a null character (0x00). The MFT (A-Series) supports reading data via modbus, but only using Function 03 -Read Hold Registers. That data can now be read with the MFT B-Series using the Function 04 – Read Input Registers.

Modbus functions operate on memory mapped to registers. Modbus registers are organized into the following reference types identified by the leading number of the reference address:

Modbus Register Reference Address	Description
0xxxx	Read/Write Discrete Outputs or Coils. A 0x reference address is used to drive output data to a digital output channel.
1xxxx	Read Discrete Inputs. The ON/OFF status of a 1x reference is controlled by the corresponding digital input.
3xxxx	Read Input Registers. A 3x reference register contains a 16-bit number received from an external source – e.g. an analog signal.
4xxxx	Read/Write Output or Holding Registers. A 4x register is used to store 16-bits of numerical data (binary or decimal), or to send data from the CPU to an output channel.

Register Map Notes:
1. The "x" following the leading character represents a four-digit address location in memory. The leading character is generally implied by the function code and omitted from the address specifier for a given function. The leading character also identifies the I/O data type.
2. The ON/OFF state of discrete inputs and outputs is represented by a 1 or 0 value assigned to an individual bit in the 16-bit data word. With respect to mapping, the LSB of the word maps to the lowest numbered coil of a group and coil numbers increase sequentially as you move towards the MSB. Unused bits are set to zero.

Function 05 (0x05) – Write Single Coil
Function 01 (0x01) – Read Coils
Modbus Register Reference Address 0xxxx

Modbus Coil	Coil Description
0	Start the Zero Drift Check
1	Start the Mid-Span Drift Check
2	Start the Span Drift Check
3	Start the Drift Check Cycle
4	Abort On-Going Drift Check
5-7	Reserved
8	Start a Purge (Requires Purge Option)

Function 02 (0x02) – Read Discrete Inputs
Modbus Register Reference Address 1xxxx

Modbus Coil	Coil Description
0	Zero Drift Check started
1	Mid-Span Drift Check started
2	Span Drift Check started
3	Drift Check Cycle started
4-7	Reserved
8	Purge Cycle started (Requires Purge Option)
9-15	Reserved
16	Error - Rp Resistance Above High Limit
17	Error - Rp Resistance Below Low Limit
18	Error - Rtc Resistance Above High Limit
19	Error - Rtc Resistance Below Low Limit
20	Error - Wire Loop Resistance Above High Limit
21	Error - Rps Sensor Lead Open Circuit
22	Error - High Sensor or Wire Leakage
23	Error – Flow Rate Above Design Limit
24	Error - Meter Kick Out High
25	Error - Meter Kick Out Low
26	Error - ADC Failed to Convert Measurement
27	Error - Sensor Control Drive Stopped Responding
28	Error - Sensor Over-Voltage Crowbar Engaged
29	Error - Sensor Type Does Not Match Configuration
30	Error - Abnormal Sensor Node Voltages
31	Error - Unable to Write the Configuration File to the EEPROM
32	Error – Sensor Type Does Not Match Board Build[2]
33	RESERVED[2]
34	RESERVED[2]

35	RESERVED^2
36	RESERVED^2
37	RESERVED^2
38	RESERVED^2
39	RESERVED^2
40	RESERVED^2
41	RESERVED^2
42	RESERVED^2
43	RESERVED^2
44	RESERVED^2
45	Sensor Leakage Warning: S-Gnd below 100 k^3
46	Logging Power On Event^2
47	Logging Change Made to the Configuration Event^2
48	Alarm 1 Triggered^1
49	Alarm 2 Triggered^1

[1] These features are only available in firmware versions 1.05 and newer.

[2] These features are only available in firmware versions 1.20 and newer.

[3] These features are only available in firmware versions 1.10 and newer and have a factory configuration option to allow operation up to 600 °C for the FD2 Sensor and the warning code may be followed by the error at coil 22.

Function 04 (0x04) – Read Input Registers **Modbus Reference Register Address 3xxxx**		
Modbus Register	**Register Description**	**Data Type C-Compilers**
0-1	Flow Rate	Float
2-3	Velocity	Float
4-5	Temperature	Float
6-7	Total Flow	Float
8-9	Elapsed Time	Float
10-11	Flow Rate Correction Factor	Float
12-13	Temperature Correction Factor	Float
14-15	Density	Float
16-20	Serial Number	ASCII (char)
21-23	Velocity Unit	ASCII (char)
24-26	Flow Rate Unit	ASCII (char)
27-29	Total Flow Unit	ASCII (char)
30-32	Temperature Unit	ASCII (char)
33-34	Sensor Rp Current (IRP)	Float
35-36	Sensor Rp Power (PRP)	Float
37-38	Electronic Temperature	Float
39-40	Voltage Input for Zero Drift Check	Float
41-42	Voltage Output for Zero Drift Check	Float
43-44	Percent Difference of the Zero Drift Check	Float
45-46	Voltage Input for Mid Drift Check	Float
47-48	Voltage Output for Mid Drift Check	Float
49-50	Percent Difference of the Mid Drift Check	Float
51-52	Voltage Input for Span Drift Check	Float
53-54	Voltage Output for Span Drift Check	Float
55-56	Percent Difference of the Span Drift Check	Float
57-58	Current Runtime Counter (sec)[*]	Unsigned 32 bit Integer
59-60	AO1 Current (mA)[*]	Float
61-62	AO2 Current (mA)[*]	Float

[*]These features are only available in firmware versions 1.05 and newer.

Function 03 (0x03) – Read Holding Registers **Function 06 (0x06) – Write Single Register** **Modbus Reference Register Address 4xxxx**		
Modbus Register	**Register Description**	**Data Type C-Compilers**
0-5	Reserved	
6-7	Flow Area	Float
8-14	Flow Meter Identification	ASCII (char)
15-21	Temperature Meter Identification	ASCII (char)

22-23	Analog Out 1 - 4 mA Scale	Float
24-25	Analog Out 1 - 20 mA Scale	Float
26-27	Analog Out 2 - 4 mA Scale	Float
28-29	Analog Out 2 - 20 mA Scale	Float
30	Purge Width (ms)	Unsigned Integer
31	Purge Hold Mask (ms)	Unsigned Integer
32-33	Purge Interval (min)	Unsigned 32 bit Integer
34-35	Drift Check Zero Scale Value (%)	Float
36-37	Drift Check Mid Scale Value (%)	Float
38-39	Drift Check Span Scale Value (%)	Float
40	Drift Check Zero Check Duration (sec)	Unsigned Integer
41	Drift Check Mid Check Duration (sec)	Unsigned Integer
42	Drift Check Span Check Duration (sec)	Unsigned Integer
43	Drift Check Time Interval (hrs)	Unsigned Integer
44-45	PID Reference Data	Float

Read Device Identification

Function 43 (0x2B) – Encapsulated Interface Transport
MEI Type 14 (0x0E) – MODBUS Encapsulated Interface (MEI)

Object Id	Object Name / Description	Type	Category
0x00	Vendor Name	ASCII String	Basic - Madatory
0x01	Product Code	ASCII String	Basic - Madatory
0x02	Major Minor Revision	ASCII String	Basic - Madatory

Data Formats for Kurz Floating Point, Straight (Daniels Extensions) and Classic Modicon Modbus

Kurz has implemented the straight binary mapping of the 32 bit floating point variables into the Modbus registers. The Kurz byte order is 1 2 3 4. This method may also be known as Reverse-32 on some systems as the classic Modbus order uses 3 4 1 2 as the byte order. When viewing the Kurz variables like

Temperature and Flow, if the numbers do not look normal, try the other formats for interpreting the data registers.

Modbus Biasing

The MFT B-Series slave device will auto detect the polarity of its wiring and correct a polarity error in wiring, but it requires a biased bus for this functionality to work properly.   See the MFT B-Series Field Wiring Diagram for a larger version of the following diagram.

Modbus ASCII Compatibility Issues

The MFT B-Series MODBUS setup for ASCII transmission framing is not supported byKzComm. If KzComm is to be used over MODBUS, then the RTU transmission framing must be used.

MODBUS MULTIPOINT SERIAL CONNECTIONS
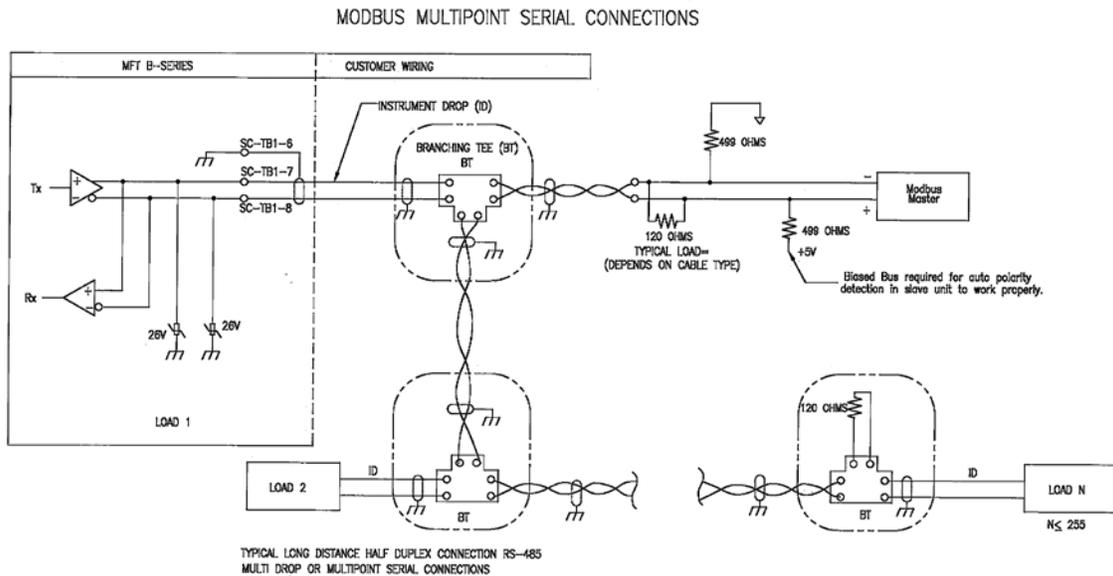
Half-Duplex Modbus Serial Communications Example on an RS-485 Bus
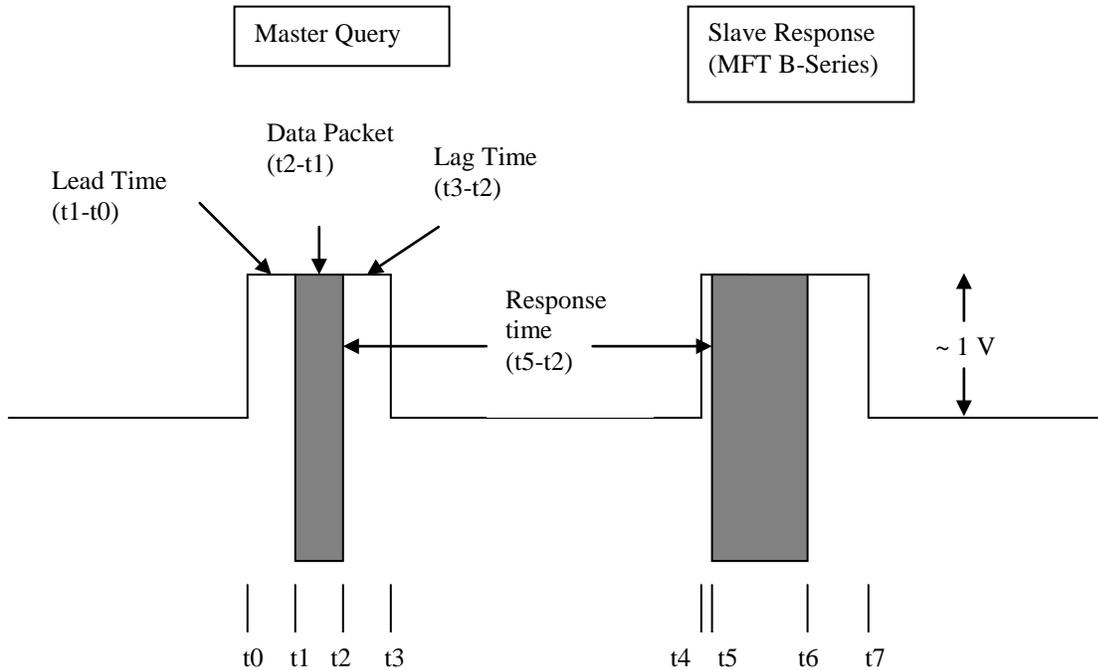
The Modbus master issues a command with the slave responding. Note that between the transmissions we have no signal. RS-485 is a tri-state (logic low, high and off). The delay between asserting the RS-485 bus and the serial data is known as the lead time (t1-t0) or (t5-t4). The data packet length (t2-t1) or (t6-t5) depends on the message and the baud rate. The delay between the packet and release of the RS-485 bus is the lag time (t3-t2) or (t7-t6).   The delay between the master and slave packets is the response time (t5-t2).

MFT B-Series Specifications (RTU protocol, 38.4 kbits/s)
           Lead time: 184 us
           Lag time: 14 ms max
           Response time: 18 ms max

Many of the master devices are programmable for their baud rate, silent interval, timeout times, number of retries, framing, and protocol. Note that if the master lag time is too long, this transmission will collide with the slave response and will increase the data transmission errors.

Kurz recommends the following values:
           Baud Rate: 38.4 kbaud
           Silent Interval: 35 ms
           Timeout Interval: 100 ms
           Number of Retries: 2
           Framing: Remote Terminal Unit (RTU)

Because of these lead/lag and response times, the MFT B-Series is limited to approximately 15 transactions/sec at 38.4 kbaud. For large Multipoint systems this still allows a complete 12 point duct measurement to be made in under a

second. Because the CPU in the MFT B-Series is used for many different functions, it may not always be ready to respond to a command from the Modbus master. This is a common limitation for Modbus devices. For this reason it is recommended that you pole it multiple times or just skip this address if it does not respond during an address scan of the devices on the Modbus network. Many of the Modbus drivers are designed to pole a device multiple times or just skip it on a polling cycle and not report an error unless it does not respond after X tries.