ПРЕОБРАЗОВАТЕЛИ СИГНАЛОВ

MFC 010 C

1. Introduction

The MFC010 is a stand alone signal converter designed to directly interface the OPTIMASS and OPTIGAS families of Coriolis mass flowmeters into control systems using the Modbus RTU protocol where there is no requirement for the extensive output control features provided by more expensive converter solutions.

The MFC010 performs three primary direct measurements, Mass flow, Density and Temperature. Using these primary measurements the MFC010 is able to calculate an array of secondary values such as Volume Flow, Velocity and Concentration.

Mass Flow – Mass flow measurement doesn't come any simpler, once installed just perform a "Zero Calibration", "Reset" the "Totalisers" and away you go. Where Process noise is a nuisance use the "Measurement Time Constant", "Low Flow Threshold" and "Pressure Suppression" features to provide reliable and repeatable results.

Density – Using the inverse relationship between the Density of the process product and the oscillation frequency of the measuring tube, the MFC010 can provide a very accurate and reliable Density reading. In order to maximise the excellent performance of the MFC010 the user should perform a density calibration after installation. The MFC010 provides two forms of Density Calibration, the simple "Single Point Calibration" and the more accurate "Two Point Calibration". Using the "Density Averaging" feature the user can reduce noisy readings caused by process installation and noise. **NB.** Density measurement is not available with the OPTIGAS 5000 meters.

Concentration – Using the "Density" and "Temperature" measurements the MFC010 is capable of calculating the concentration of a product in the process medium, from one of a number of pre-defined industry standards, such as "°Brix" and "°Baumé", as well as user defined mixtures using the programmable coefficients. Concentration measurement is a function that comes with a comprehensive manual and a Coefficient calculation software package which will take the users own process data and convert it into compatible coefficients to permit the MFC010 to automatically calculate the concentration of the target process.

Velocity – Using the measured mass flow and density, the velocity of the product is calculated using the "Pipe Diameter" setting. By default this is set to the measuring tube internal diameter to calculate the velocity of the product passing through the sensor, but it can be set to calculate the velocity in a section of the connecting pipe work.

Process Control – Where precise process conditions are required, the "Process Control" function can be used to detect adverse variations in the "Density" or "Temperature" measurements and, as well as indicating the condition, it can take one of a number of predefined actions according to the users requirements.

2. Mechanical Installation

Refer to the installation guidelines and instructions for mounting the sensor in the process pipe work provided in the handbook on the CD supplied with the sensor.

3. Electrical Installation

The MFC010 is provided with four electrical terminal connections.

- V+ The power supply input terminal.
- V- The power supply return path and “Common” for the Modbus interface.
- A The inverting (RS485-) terminal for the Modbus interface.
- B The non-inverting (RS485+) terminal for the Modbus interface.

These terminals can be accessed in the terminal compartment of the sensor.

3.1 Electrical Input Specifications for the MFC010

NB all voltages, unless otherwise stated, are with reference to the “V-” terminal.

V+ Terminal		
	Min. Input Voltage	11.4V DC
	Max. Input Voltage	12.6V DC
	Max. Input Current	200mA DC
A & B *		
	Min. Input Voltage	-7V DC
	Max. Input Voltage	+11.8V DC
	Min. Output Voltage	-6V DC
	Max. Output Voltage	+6V DC

*The Modbus protocol requires that the communications interface to the MFC010 complies with the limitations of the EIA/TIA-485 (RS485) specification.

For a standard, non hazardous area, sensor the input impedance of the MFC010 is equivalent to 1/8 of a standard RS485 load, i.e. an input impedance of >96kΩ, permitting it to be connected to the Modbus bus in accordance with the Modbus requirements. However, when installed in a Hazardous area the MFC010 requires that suitable barrier devices must be fitted between the MFC010 and the Modbus main bus, see sections 4.1 & 4.2 for details of suggested barrier devices and connection. If the main Modbus Bus is configured for multidrop operation, a Modbus compatible RS485 repeater is required to connect the barrier devices to the bus, see section 4.3 for further details.

3.2 Recommended Cable Specification

The cable used to connect the MFC010 to the Modbus master control system should be an overall screened twisted pair cable, with two twisted pairs of a minimum 20 AWG conductor. The total cable capacitance should not exceed 50nF and the conductor inductance should not exceed 200µH. The external cable insulation should be specified appropriately for the environment into which the device is to be installed. The outside diameter of the cable should be between 6.5 mm and 9.5 mm to ensure proper sealing is achieved when passed through the cable gland entry.

an supply suitable cable that can be ordered to the required length, the part numbers are as follows

External Insulation Colour Grey - Part No. X5871059989

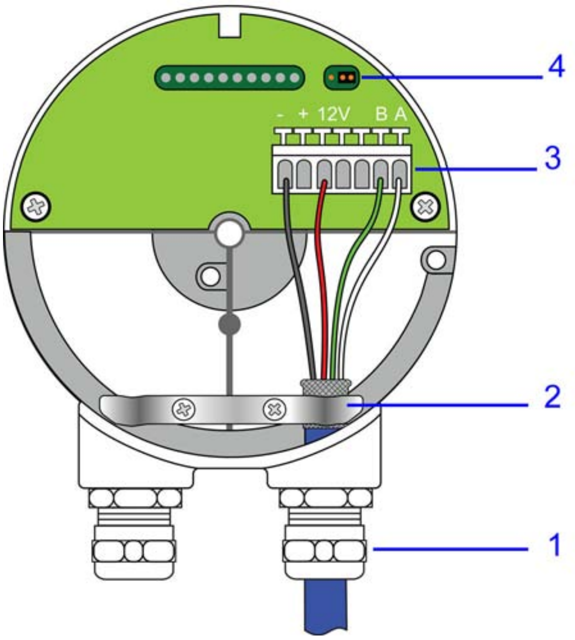
External Insulation Colour Blue - Part No. X5871069989
(For hazardous area installations)

The maximum cable length from the MFC010 to the bus "Master" device is 300m when using the default Modbus transmission speed of 19200 Baud. There are further limitations on the cable length when installing the system into a Hazardous area, refer to section 4.3 on page 17 for details.

3.3 Connection to the MFC010

1. Unscrew the fixing screw on the junction box cover.
2. Release the two fixing screws holding the cable grip in place and remove the grip.
3. Strip approx. 50mm/2” of the outer casing of the signal cable.
4. Split the screen away from the cores and fold it back on the outer cable.
5. Fit the cable grip and secure, making sure that the screen is gripped under the grip.
6. Connect the four cores to the terminals marked A,B, 12V, - as shown

NOTE: The spring loaded connections are released by depressing the white lever above each connection

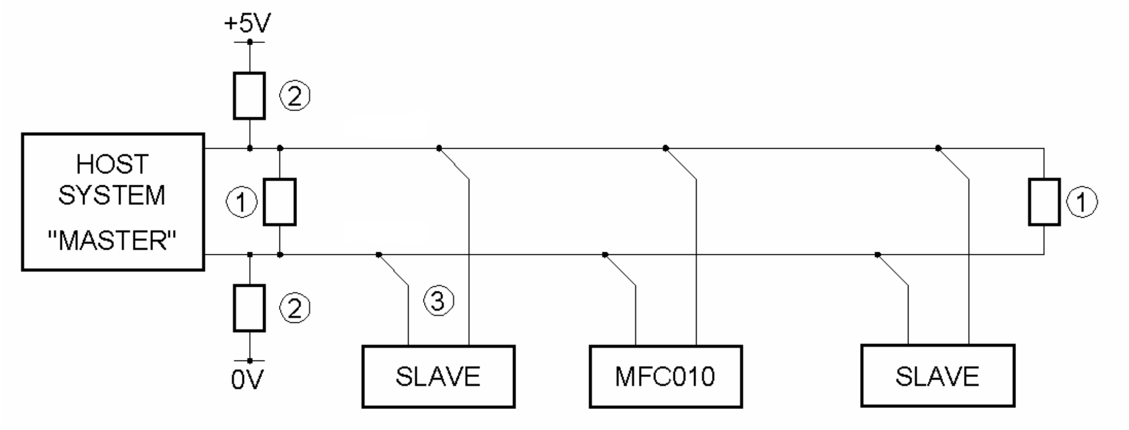


1 Cable Gland
2 Cable Grip/Earth
3 Terminal connections
4 Jumper for EOL resistor (not supplied) – off in position as shown, on in other position

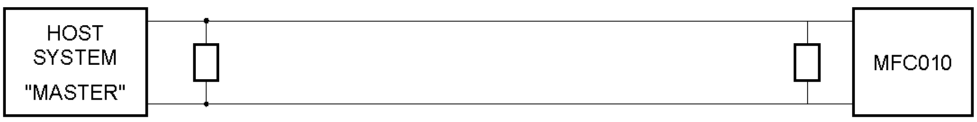
Terminal	Input Connection
12V	V+
-	V-
A	A (RS485-)
B	B (RS485+)

3.4 Connection to the Modbus Bus

The MFC010 is designed to be connected as a Slave device onto the 2-wire bus implementation of the Modbus physical layer definition. In this configuration the receiver and transmitter lines for each device are connected together, Transmitter A to Receiver A and Transmitter B to Receiver B, and operated in Half Duplex mode, where the master transmits a request and only after receiving it does the nominated slave device transmit a reply. When not responding to a direct request from the Master device, the Slave devices remain passive, monitoring the bus and awaiting a suitable request from the Master device. In addition to the A and B signal lines the bus MUST include a “Common” signal line to act as a ground reference point for the A and B signals.

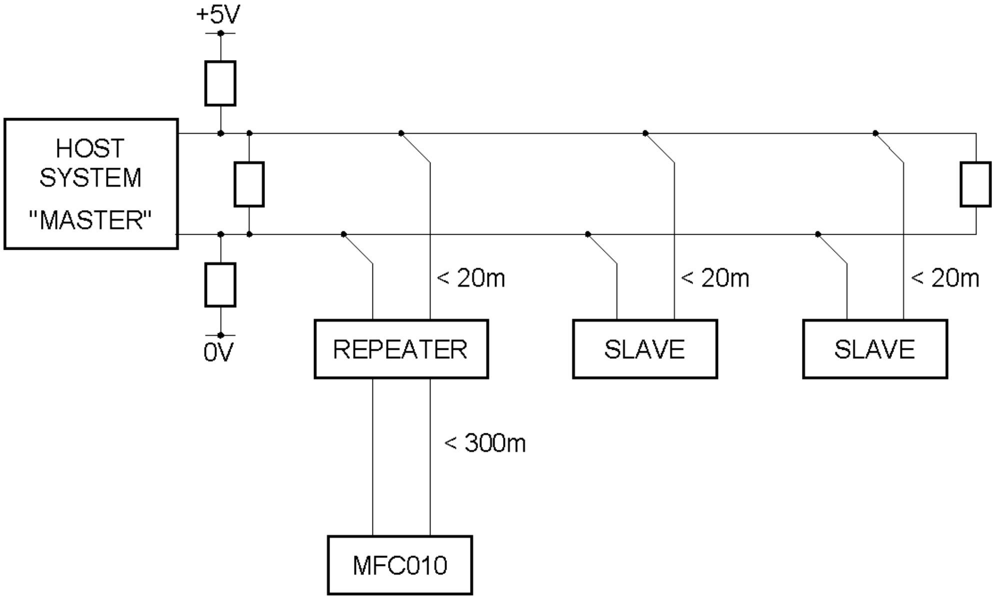


① The master bus must be terminated at its physical end points by suitable termination networks connected between the A (D0) and B (D1) signal lines. When not using bus-biasing resistors, see next paragraph, each termination network may consist of a single 150 Ohm, 0.5W resistor. However, when bus-biasing resistors are required, a more suitable termination network would consist of a 1nF capacitor in series with a 120 Ohm, 0.25W resistor. NB It is common that the Host system “Master” is physically at one end of the bus, so one of the termination resistors is fitted at its terminals, but it should be realised that this is not always the case and care should be taken to ensure that the termination network is at the physical end of the bus. In a point-to-point configuration, when only one Slave device is fitted to the bus, then the terminating networks can simply be situated at the connecting terminals of the master and slave devices.

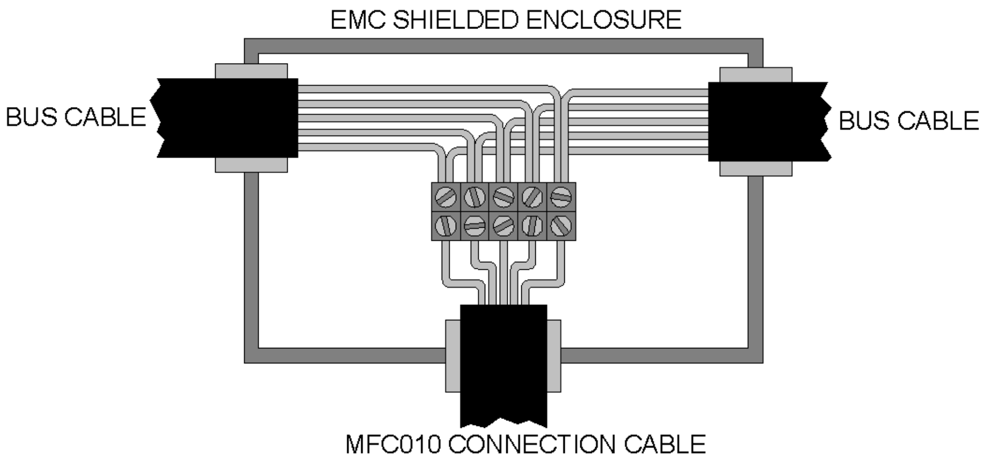


② Some slave devices require that Bus-Biasing resistors are fitted to ensure that the bus is in a defined state when none of the transmitting devices are active. The MFC010 does NOT require Bus-Biasing resistors to be fitted but is compatible with their presence on the bus if one or more of the other slave devices on the bus require them to be fitted, as long as they comply with the Line Polarization requirements of the Modbus specification.

③ In a multidrop bus configuration the slave devices are connected to the main bus cable by branch connections at intervals along the length of the main bus. The branch connections, Derivations as they are termed in the Modbus specification, must be less than 20m in length from the main bus cable to the slave device. Some slave devices permit direct connection to the main bus, known as “Daisy Chaining”, in some cases by providing extra terminals and cable access points. However, as indicated in the previous sections, access to the terminal compartment of the MFC010 is limited; therefore it is not practical to directly connect the MFC010 to the main bus. Instead, the installation should utilise a short branch connection. If a greater length of cable is required between the MFC010 device and the main Modbus bus, the user should install a suitable RS485 repeater between the MFC010 and the bus (refer to the diagram below).

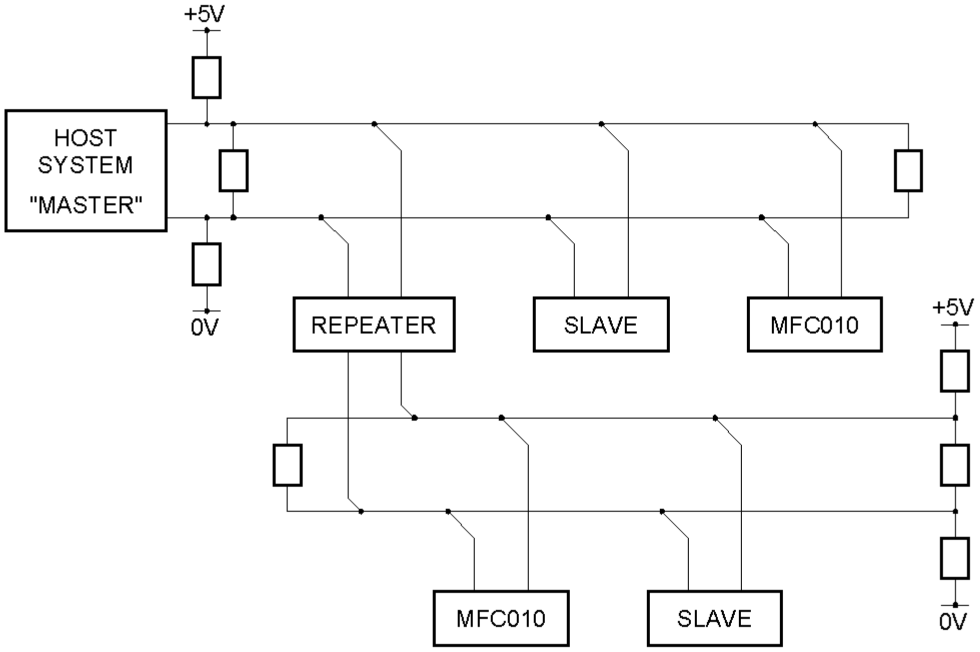


Because the connection to the bus requires exposing the signal wires, the connection to the main bus should be made within a suitable EMI shielded enclosure. This connection should include the “Common” signal connection, the power supply connection (if a suitable one is provided by the bus), and the drain wire when available. Each of the cable screens must be properly terminated to the enclosure by means of appropriate EMC cable glands. For example:



If the Bus does not provide a suitable power supply for the MFC010, a separate suitable power supply connection should be made at this point.

An RS485 Repeater can be used to extend the length of the Bus and the number of slave devices that are attached to the bus (refer to the figure below). However, if the bus is extended in such a fashion, termination and polarization networks should be fitted according to the same rules as used for the main bus (see descriptions above).



NB For Hazardous Area applications the user should refer to section 4.3, on page 17, for connection details.

NB For multidrop systems, ensure cycle times are properly calculated to ensure bus speeds are adequate for the application.

3.5 Installation Guidelines for Electromagnetic Compatibility

Whilst the MFC010 and its associated sensor has been designed, tested and certified to meet the requirements on international standards of Electromagnetic Compatibility (EMC), it is the users responsibility to ensure that the connection guidelines described in this document are followed. In addition the user should use recognised good practise in the location and cable routing of the MFC010 in relation to its surrounding environment. The user should consider the following suggestions when installing an MFC010 into a system.

1. Every effort should be made to avoid significant lengths (>50mm) of unshielded signal wire when connecting to the system, any terminal connections should be housed in a suitably shielded enclosure.
2. Avoid routing the cable in groups with or alongside other power carrying cables.
3. Avoid locating the MFC010 or routing the connection cable in close proximity to large electrically powered equipment, such as pumps, inverters etc.
4. If necessary, route the connection cable through a suitably earthed metal conduit.

4. Installation in Hazardous Area Applications

Before installation the user MUST ENSURE that the equipment to be installed is the Hazardous area approved equipment.

Copies of the appropriate certificates can be found on the website at www.om.

Before installation the user MUST refer to the hazardous area installation document, supplied with this equipment, and strictly adhere to the relevant installation instructions indicated therein.

When the MFC010 is used in Hazardous area installations, suitable barrier devices MUST be fitted. The Safety Parameters for the MFC010 are as follows. All interface and barrier devices must be appropriately certified to meet these parameters.

ATEX

V+ & V-	Input Voltage, U_i	16.5V
	Input Current, I_i	340mA
	Input Power, P_i	1.3W
	Input Capacitance, C_i	35nF
	Input Inductance, L_i	10µH
A & B	Input Voltage, U_i	11.8V
	Input Current, I_i	40mA
	Input Power, P_i	120mW
	Input Capacitance, C_i	35nF
	Input Inductance, L_i	10µH

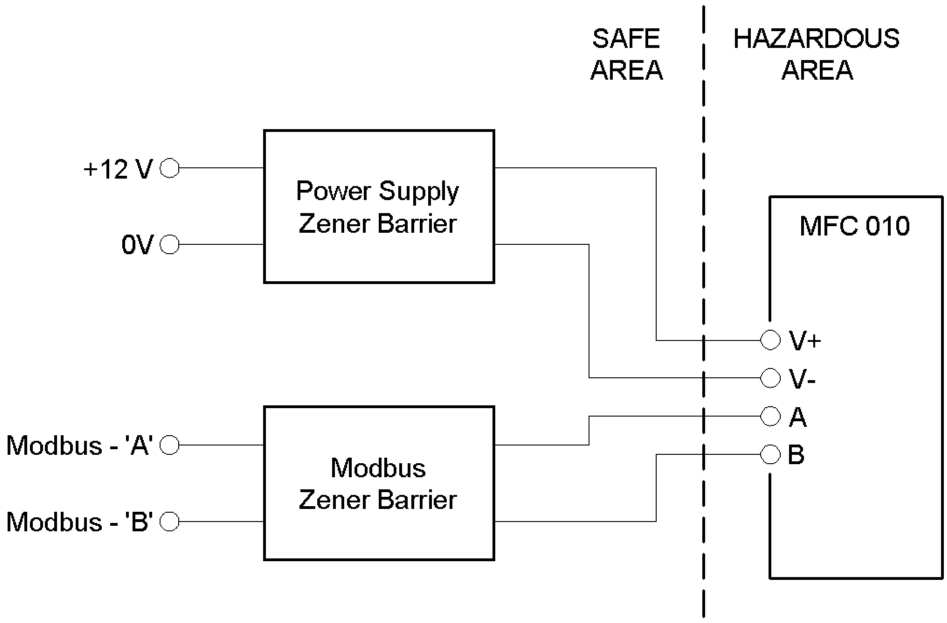
FM

V+ & V-	Input Voltage, U_i	16.2V
	Input Current, I_i	317mA
	Input Power, P_i	1.28W
	Input Capacitance, C_i	35nF
	Input Inductance, L_i	10µH
A & B	Input Voltage, U_i	11.8V
	Input Current, I_i	34mA
	Input Power, P_i	90mW
	Input Capacitance, C_i	35nF
	Input Inductance, L_i	10µH

The output safety parameters of the barrier devices must not exceed the Voltage, Power and Current limits set out above. The output Capacitance parameter for the barrier devices must exceed the sum of the MFC010 input Capacitance, specified above, and the maximum cable Capacitance. The output Inductance parameter for the barrier devices must exceed the sum of the MFC010 input Inductance, specified above, and maximum cable Inductance. To summarise:

$\mathbf{U_o}$ Barrier	$<$	$\mathbf{U_i}$ MFC010
$\mathbf{I_o}$ Barrier	$<$	$\mathbf{I_i}$ MFC010
$\mathbf{P_o}$ Barrier	$<$	$\mathbf{P_i}$ MFC010
$\mathbf{C_o}$ Barrier	$>$	$\mathbf{C_{cable}} + \mathbf{C_i}$ MFC010
$\mathbf{L_o}$ Barrier	$>$	$\mathbf{L_{cable}} + \mathbf{L_i}$ MFC010

NB The In-line resistance of the Modbus barrier **MUST NOT EXCEED** 1000 Ohms for each of the A and B Input terminals.



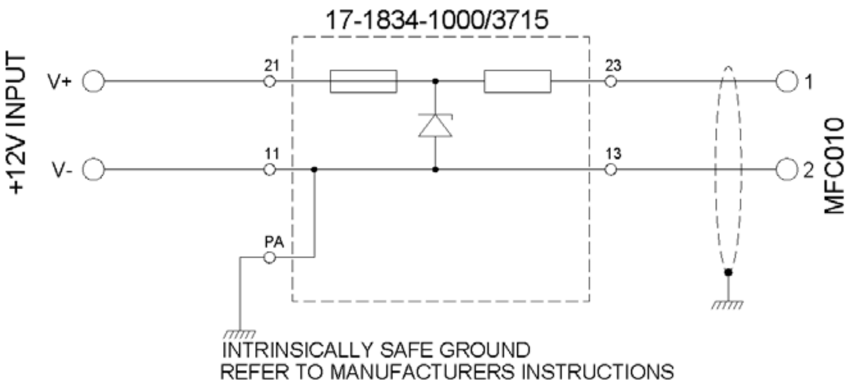
The Zener barrier devices must be installed in an EMI shielded enclosure and the cable screen(s) kept intact right up to the barrier terminals as far as is practical. The cable screen(s) should be terminated to the enclosure, chassis Earth connection, and kept SEPARATE from the intrinsically safe Earth connections of the Barrier devices. The user MUST adhere to the barrier manufacturer's instructions for connecting the intrinsically safe Earth connection to the barrier devices.

4.1 Power Supply Barrier Devices

The following Zener Barrier devices are those that are recommended for use on the V+ & V- power supply input connections to the MFC010.

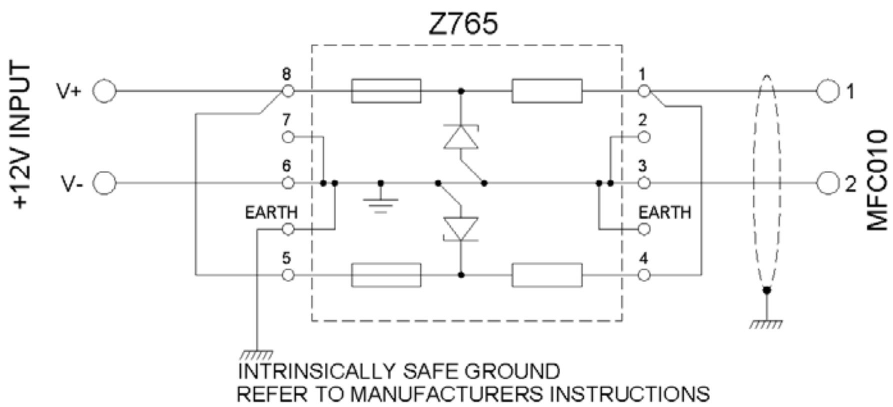
Manufacturer : Bartec
Part Number : 17-1834-1000/3715
Ex Approvals : EEx ia/ib IIC

Connection :



Manufacturer : Pepperl & Fuchs
Part Number : Z765
Ex Approvals : EEx ia IIC
FM and CSA Approved

Connection :



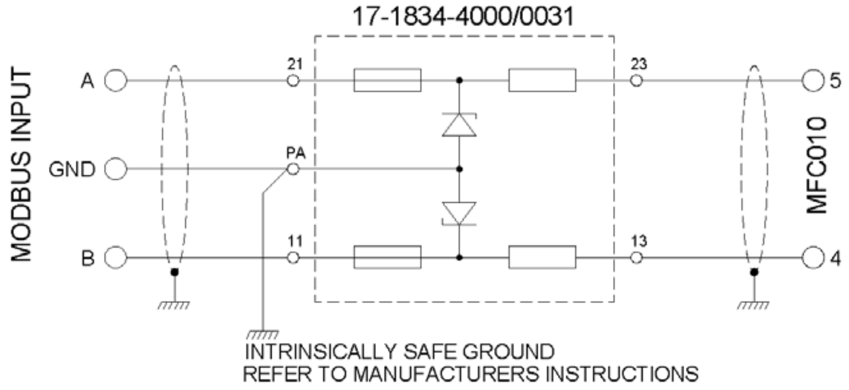
Note: For Optimass 2000 the supply voltage to the barrier should be +14V to ensure maximum voltage is supplied to the meter.

4.2 Modbus Barrier Devices

The following Zener Barrier devices are those that are recommended for use on the A & B Modbus input connections to the MFC010. When specifying alternate devices the user must ensure that the in-line resistance of the Modbus barrier **DOES NOT EXCEED** 1000 Ohms for each of the A and B Input terminals.

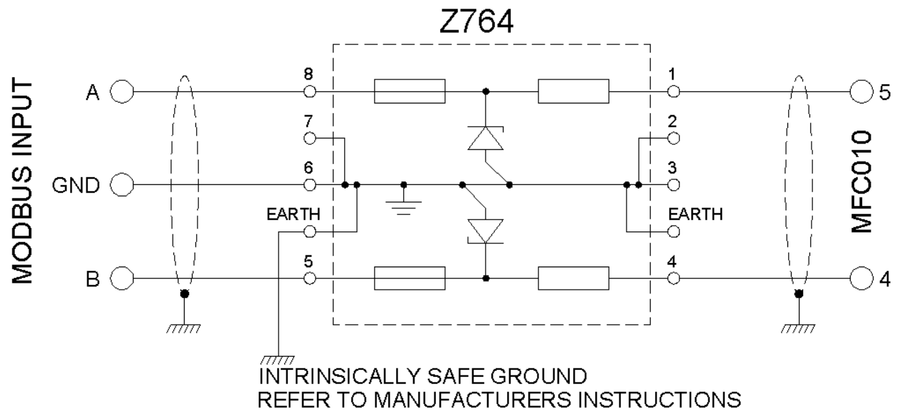
Manufacturer : Bartec
Part Number : 17-1834-4000/0031
Ex Approvals : EEx ia/ib IIC

Connection :



Manufacturer : Pepperl & Fuchs
Part Number : Z764
Ex Approvals : EEx ia IIC
FM and CSA Approved

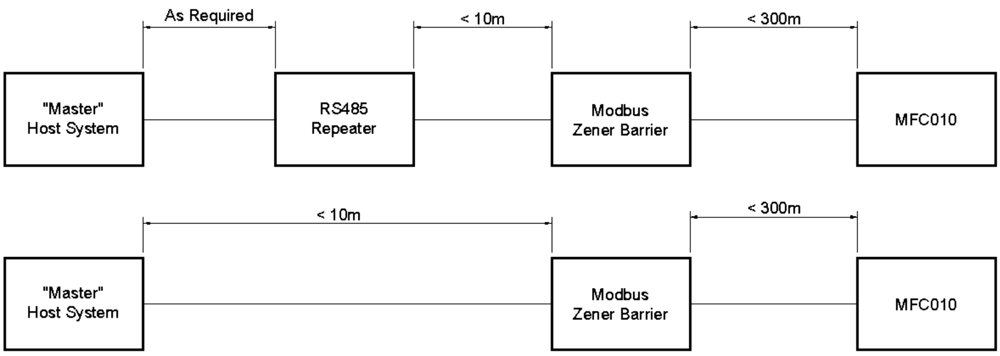
Connection :



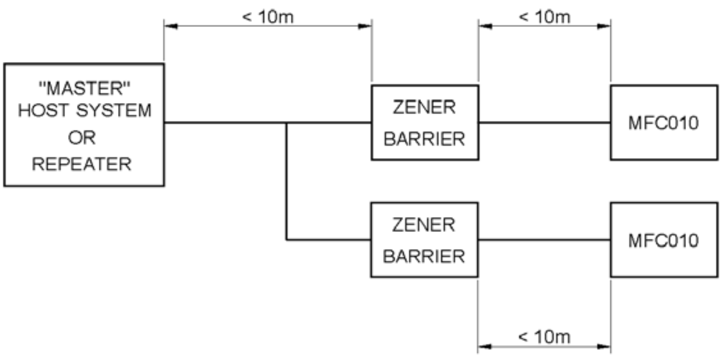
4.3 Connection To The Modbus Bus

When installed in hazardous area application the MFC010 interface is not directly compatible with the Modbus interface standard due to the presence of the required Zener Barrier devices.

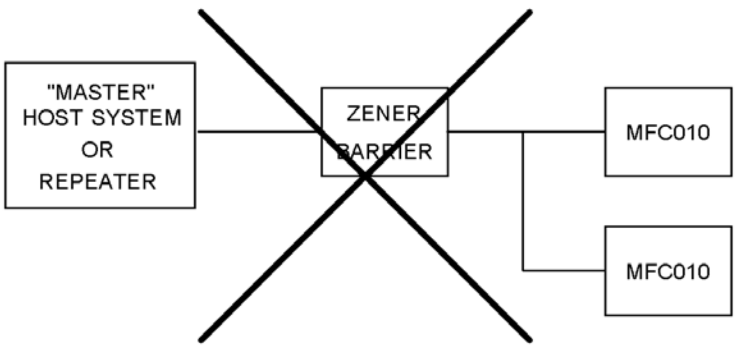
In a point-to-point configuration, when the MFC010 is the only device on the bus, the cable length from the barriers to the "Master" host system must not exceed 10m in length. If a greater distance is required, the use of a suitable RS485 repeater is recommended, in which case the repeater connection to the Zener Barrier devices should not exceed 10m in length. The maximum cable length between the Repeater and the "Master" host system is determined by the operating limits of those two devices. The cable length from the Modbus barrier device to the MFC010 must be less than 300m i.e.



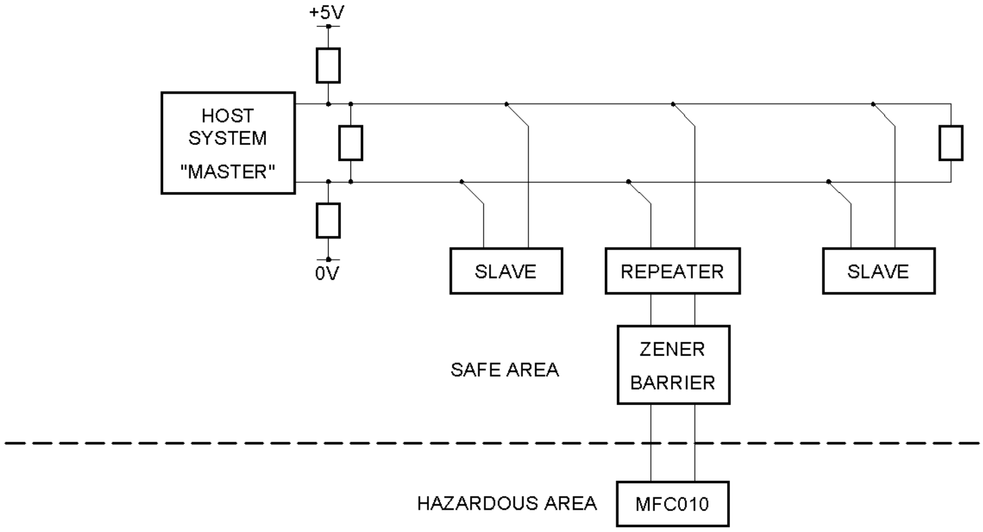
Where the distance from the barrier device to the MFC010 is less than 10m, two barrier devices may be connected in parallel to the "Master" host system, or the repeater if one is being used, refer to the diagram below. The overall cable length from the Host/repeater to the barrier devices must still be less than 10m as described previously. If more than two devices are required to be connected then a dedicated repeater should be used for each.



Each MFC010 MUST have its own dedicated barrier interface; they MUST NOT be connected in parallel on the hazardous area side of the system.



In a multidrop installation, see figure below, the Zener Barrier devices must be connected to the bus using a suitable RS485 repeater, with the connecting cable between the Barrier devices and the repeater not exceeding 10m. The connection of the repeater to the Modbus bus must then follow the rules and restrictions of the Modbus protocol as indicated previously in section 3.4.



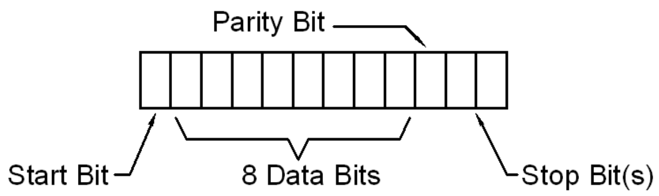
5. Modbus Protocol Interface

The interface to the MFC010 is implemented in the Modbus RTU communications protocol, and is done so in accordance with the specification and requirements of the "Modbus Protocol Reference Guide" (PI-MBUS-300 Rev J). The physical electrical parameters of the Modbus specification are defined by the EIA/TIA-485 (RS485) standard and the "Modbus over Serial Line - Specification and Implementation Guide V1.0" interface definition.

In a serial communications system such as the Modbus protocol, data is transmitted as a series of voltage levels along the connecting data wires. A "bit", or binary digit, value is determined by the logical level (high or low) of the connecting interface over a set time period. The time period for each bit is determined by the transmission speed, known as the baud rate. For a baud rate of 9600, the bit period is 1/9600 = 104.2 microseconds. The MFC010 supports Baud rates of 1200, 2400, 4800, 9600, 19200, 38400 and 57600 baud (see the Baud rate setting in Holding Register No. 1005). The higher transmission speeds require careful attention to the cable installation in order to function reliably and error free (see Section 3.2 on page 7 for installation details).

5.1 Character Transmission Format

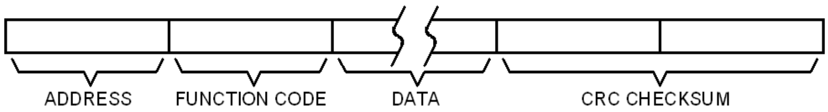
Data is transmitted in sets of 8 bit data blocks, known as "Bytes" or "Characters". Each character is preceded and followed by framing bits that permit the correct detection of the transmitted character. The first "Bit" transmitted will be the "Start" bit, this permits the receiving device to detect that a character is being transmitted. The "Start" bit is then followed by the 8-bit data byte. A "Parity" bit may then follow the 8-bit character. This "Parity" bit is optional (see the Transmission format setting in Holding Register No. 1004), it allows the system to validate the contents of the 8-bit data byte to ensure that no errors have occurred during transmission. Following the "Parity" bit is the "Stop" bit that indicates the end of the transmitted character to the receiving device. If no parity bit is used, two stop bits must be implemented, this ensures a consistent character length of 11 bits is maintained.



The 8 data bits are annotated from bit 0 (the least significant bit, LSB) to bit 7 (the most significant bit, MSB). The character is transmitted "MSB first", i.e. the first bit after the start bit, is bit 7 of the data byte. See appendix B for more details on binary coding.

5.2 Modbus Telegram Format

The messages between the Modbus master and slave devices are transmitted as groups of characters, as described above, collectively known as telegrams. Each telegram is preceded and followed by a "Quiet" period on the Modbus bus of 3½ character periods. The "Quiet" period following the telegram is used to indicate the end of the telegram.



The first character received in the telegram identifies the slave device to which or from which the telegram is being transmitted. This first character is known as the Slave ID or Slave Address. In multidrop configurations (see Section 5.4 below) this address character is used by the master device to individually communicate with one of the instruments on the bus connection. The Slave device returns this value to indicate to the master the source of the response telegram. The Slave Address value for the MFC010 can be set using Holding register No. 1006 (see Section 7.5).

The second character received in the telegram is the function command code requested by the master device. A list of the function codes supported by the MFC010 can be found in Section 6 (See Page 23) along with a description of each.

The last two characters received form a 16-bit checksum value. This checksum value is used to ensure that the data received in the telegram has not been corrupted. The checksum is calculated and appended to the telegram by the transmitting device (Slave or Master) and the receiving device compares the received checksum value against the value it calculates from the received data. If the data has been corrupted in some way during transmission, then the checksum calculated by the receiving device will be different than that which it received with the telegram. The receiving device will then ignore the telegram knowing that the data within is unreliable. See Appendix A for information on the Modbus Checksum calculation.

Between the function code character and the CRC checksum at the end of the telegram is the telegram data. The contents and format of these data characters is dependant upon the function code requested.

5.3 Data Types in Modbus

There are two data types used to transmit information on a Modbus data bus, the "Bit" and the "Register". The "Bit" represents a single binary state, whether as an output or an input condition. The "Register" is a 16-bit integer transmitted as two 8-bit characters. Using multiple "Registers" the Modbus interface can transmit higher accuracy values such as "Floating Point" and "Double Precision Floating Point" numbers.

"Bit" variables are packed into 8 bit bytes, so each character sent or received can contain up to 8 "Bit" variables. The Master and Slave devices use only as many 8 bit data characters as are required to transmit the information. Any unused bits in the data characters are ignored. The bit that is first indexed by the Master request address is transmitted in the LSB, Bit 0, of the first data character. The next "Bit" value is transmitted in the next bit, Bit 1, of the first data character. This continues until the last bit location, Bit 7, of the first data character is used. The next "Bit" value is then transmitted in the LSB, Bit 0, of the following data character, this continues until all of the requested values have been transmitted. Any unused bit locations in the last data character are filled out with "0"s

For simple single register variables the Most Significant Character of the register is transmitted first, with the Least Significant character following immediately after. However, for variables that require multiple registers, i.e. the "Floating Point" and "Double Precision Floating Point" variables, the transmission order is a little more complicated. i.e.

Single 16 Bit Register Variables, Data Transmission Order

Byte 1, MS Byte	Byte 0, LS Byte

Long Integer & Floating Point Variables, Data Transmission Order

Byte 1	Byte 0, LS Byte	Byte 3, MS Byte	Byte 2
Requested Register		Requested Register + 1	

Double Precision Floating Point Variables, Data Transmission Order

1	0	3	2	5	4	7	6
Requested Register		Register + 1		Register + 2		Register + 3	

5.4 Multidrop Operation

A "Master" device, such as a PC or PLC, can be used to control and interrogate a number of "Slave" devices, such as an MFC010, connected to the Modbus bus in a "Multidrop" configuration. The "Master" device always initiates the communication interchange with the "Slave" devices, each of which waits for instructions or requests from the "Master" before transmitting data on the bus in response to the instruction. Although the Modbus specification allows for up to 247 "Slave" devices to be physically connected to the bus at any time, the Master device can only request information from one "Slave" device at a time. A unique ID number or "Address" is allocated to each of the "Slave" devices to allow the "Master" to differentiate between them. Although it does not matter in which order the "Slave" devices are interrogated, the "Master" must wait for the response, or for a suitable period after the request, before making a request to any other of the slave devices on the bus.

Under some limited conditions, i.e. when the instruction to the "Slave" device does not require a detailed response, the "Master" device can send a "Broadcast" command, indicated by a "Slave" ID Address of "0", to all of the slave devices simultaneously.

5.5 Calculating Data Transmission Rates

Careful attention should be made to ensuring that the bus installation can support the amount and rate of data transmission required. Consideration of the limitations of the physical installation, as previously described, should not be ignored. The maximum usable transmission speed, baud rate, will depend entirely upon the installation.

The transmission format also needs to be carefully considered. In the Modbus standard, each transmitted character is 11 bits long, depending upon the setting of the transmission format. At the Modbus default transmission speed of 19200 baud, each character will have a transmission period of 573 microseconds.

For a simple data transfer of one Input value (see section 6.4 on page 26 for details) between the master and slave will require an 8 character (+ 2 x 3½ character "Quiet" periods) telegram in the request from the master, and a 9 character (+ 2 x 3½ character "Quiet" periods) telegram in the response from the slave. If the Slave responds immediately the cycle from the Master sending the request to receiving the response will be at least 31 characters long, or 17.8 milliseconds at 19200 baud. Therefore the maximum rate of data requests that could be made is 56 every second.

In most cases far more data will be required, and in multidrop systems the master device may be requesting data from up to 64 units. In these circumstances the user must ensure that there is sufficient time interval between requests for the measured values to be received without overlapping the Master request telegram with the previous slave reply telegram.

To achieve the required update rates the user may have to consider whether, in a multidrop configuration, the number of devices on a bus must be limited or whether the cable installation will support one of the higher data transmission speeds which are available.

This is especially important where fast response is required (such as batch filling operations).

5.6 Error Messages in Modbus

When the MFC010 detects an error in the request received in a properly formatted telegram, it will respond with an error message. The error message response telegram is formatted as follows.

Address	Function	Error Code	CRC	CRC

The most significant bit of the requested function code is set (add 128, 80_{16}) in the response telegram to indicate that an error has been detected. For example, if an error were detected in a Function 1 request, then the returned function code would be 81_{16} (129).

The single data character in the response telegram will indicate the type of error detected. These are as follows.

1 Illegal Function — The requested function code is not supported by the MFC010 or is not valid due to the current settings of the device.

2 Illegal Data Address — The Register requested is not valid.

3 Illegal Data Value — The requested data (in Write operations only) is invalid for the register being written.

6 Slave Device Busy — The MFC010 is unable to process the requested command because an EEPROM save is in progress.

Errors due to communications faults (CRC errors, Parity errors etc) are logged but no response is returned because the data in the received telegram is deemed unreliable. The Master system can read the error logs by using the diagnostic command (Function 08, see Section 6.8).

6. Modbus Functions Supported by the MFC010

6.1 01 (01_{16}): Read Coil Status

This function permits the user to read the state of a number of consecutive Discrete Outputs, or “Coil”, registers. Within the MFC010 the majority of the Discrete Outputs are used to initiate command functions; when read, response will be “1” whilst command is being processed and “0 when the command is completed (See Section 7.2 on page 39 for details of the individual Status Output registers). The format of the Master request telegram for this function should be as follows.

Request Character	Field	For Example	
1	Slave Address	01_{16}	Request to Slave ID 1
2	Function	01_{16}	“Read Coil Status”
3	Start Address Hi	03_{16}	Start Address = 1002
4	Start Address Lo	$E9_{16}$	
5	No of Points Hi	00_{16}	No. of Points = 5
6	No of Point Lo	05_{16}	(“Coils” 1002 – 1006)
7	CRC Lo	$2D_{16}$	CRC Checksum
8	CRC Hi	$B9_{16}$	

The MFC010 will respond to such a request with a telegram formatted as follows.

Response Character	Field	For Example	
1	Slave Address	01_{16}	Response from Slave ID 1
2	Function	01_{16}	“Read Coil Status”
3	Data Bytes in Response	01_{16}	1 byte (5 States requested < 8 Bits)
4	Data Byte 1	15_{16}	Data = 00010101_2
5	CRC Lo	90_{16}	CRC Checksum
6	CRC Hi	47_{16}	

The number of data bytes in the response will depend upon the number of Discrete Outputs requested. The appropriate bit in each of the data bytes received will indicate each Discrete Output state requested. Therefore, each data byte in the response will contain a maximum of 8 Discrete Output “Coil” states. For example, if 19 Discrete Outputs are requested, then three data characters will be returned, with the first group of 8 output states encoded in the first data byte, the second group of 8 output states coded in the second data byte, and the last 3 output states coded in the first three bit locations of the last data byte. Bit 0 of the first response data byte will correspond to the “Start Address” Discrete Output register specified by the request telegram. Bit 0 of the second response data byte will correspond to the “Start Address” + 8 Discrete Output register and so on.

In the example above, 5 Discrete Outputs are requested, so only one data byte is required in the response. The response data value shown above indicates that registers 1002, 1004 and 1006 are active, and that registers 1003 and 1005 are inactive.

6.2 02 (02_{16}): Read Discrete Input

This function permits the user to read the state of a number of consecutive Discrete Input registers. (See Section 7.3, on page 41, for details of the individual registers). The format of the Master request telegram for this function should be as follows.

Request Character	Field	For Example	
1	Slave Address	01_{16}	Request to Slave ID 1
2	Function	02_{16}	"Read Discrete Input"
3	Start Address Hi	03_{16}	Start Address = 1001
4	Start Address Lo	$E8_{16}$	
5	No of Points Hi	00_{16}	No. of Points = 12
6	No of Point Lo	$0C_{16}$	("Inputs" 1001 – 1011)
7	CRC Lo	$F8_{16}$	CRC Checksum
8	CRC Hi	$7F_{16}$	

The MFC010 will respond to such a request with a telegram formatted as follows.

Response Character	Field	For Example	
1	Slave Address	01_{16}	Response from Slave ID 1
2	Function	02_{16}	"Read Discrete Input"
3	Data Bytes in Response	02_{16}	2 bytes
4	Data Byte 1	CD_{16}	Data = 11001101_2
5	Data Byte 2	09_{16}	Data = 00001001_2
6	CRC Lo	$2D_{16}$	CRC Checksum
7	CRC Hi	$2E_{16}$	

The number of data bytes in the response will depend upon the number of Discrete Inputs requested. The appropriate bit in each of the data bytes received will indicate each Discrete Input state requested. Therefore, each data byte in the response will contain a maximum of 8 discrete input states. For example, if 19 Discrete Inputs are requested, then three data characters will be returned, with the first group of 8 input states encoded in the first data byte, the second group of 8 input states coded in the second data byte, and the last 3 input states coded in the first three bit locations of the last data byte. Bit 0 of the first response data byte will correspond to the "Start Address" Discrete Input register specified by the request telegram. Bit 0 of the second response data byte will correspond to the "Start Address" + 8 Discrete Input register and so on.

In the example above, 12 Discrete Inputs are requested, so two data bytes are required in the response.

6.3 03 (03_{16}): Read Holding Registers

This function permits the user to read the value of a number of consecutive Holding registers. (See Section 7.5 on page 49 for details of the individual registers). The format of the Master request telegram for this function should be as follows.

Request Character	Field	For Example	
1	Slave Address	01_{16}	Request to Slave ID 1
2	Function	03_{16}	"Read Holding Registers"
3	Start Address Hi	03_{16}	Start Address = 1023
4	Start Address Lo	FE_{16}	
5	No of Points Hi	00_{16}	No. of Points = 3
6	No of Point Lo	03_{16}	(Input Registers 1023 – 1025)
7	CRC Lo	64_{16}	CRC Checksum
8	CRC Hi	$7F_{16}$	

The MFC010 will respond to such a request with a telegram formatted as follows.

Response Character	Field	For Example	
1	Slave Address	01_{16}	Response from Slave ID 1
2	Function	03_{16}	"Read Holding Registers"
3	Data Bytes in Response	06_{16}	6 bytes (3 x 2 Byte Registers)
4	Data Byte 1	$3F_{16}$	Register 1023 = 16201
5	Data Byte 2	49_{16}	
6	Data Byte 3	02_{16}	Register 1024 = 724
7	Data Byte 4	$D4_{16}$	
8	Data Byte 5	$F1_{16}$	Register 1025 = 61730
9	Data Byte 6	22_{16}	
10	CRC Lo	$7D_{16}$	CRC Checksum
11	CRC Hi	BD_{16}	

6.4 04 (04_{16}): Read Input Registers

This function permits the user to read the value of a number of consecutive Input registers. (See Section 7.5 on page 493 for details of the individual registers). The format of the Master request telegram for this function should be as follows.

Request Character	Field	For Example	
1	Slave Address	01_{16}	Request to Slave ID 1
2	Function	04_{16}	"Read Input Registers"
3	Start Address Hi	$0B_{16}$	Start Address = 3001
4	Start Address Lo	$B8_{16}$	
5	No of Points Hi	00_{16}	No. of Points = 2
6	No of Point Lo	02_{16}	(Input Registers 3001 – 3002)
7	CRC Lo	$F3_{16}$	CRC Checksum
8	CRC Hi	CA_{16}	

The MFC010 will respond to such a request with a telegram formatted as follows.

Response Character	Field	For Example	
1	Slave Address	01_{16}	Response from Slave ID 1
2	Function	04_{16}	"Read Input Registers"
3	Data Bytes in Response	04_{16}	4 bytes (2 x 2 Byte Registers)
4	Data Byte 1	94_{16}	Register 3001 / 3002 = 75.29
5	Data Byte 2	$7B_{16}$	
6	Data Byte 3	42_{16}	
7	Data Byte 4	96_{16}	
8	CRC Lo	17_{16}	CRC Checksum
9	CRC Hi	63_{16}	

In the example above the Input register requested contains a floating point number and needs to be accessed as a pair of registers (3001/3002). The resulting 4 bytes in the data response can then be decoded into a floating-point number (See Section 5.3 on page 20 and Appendix C on page 93 for further details on encoding and decoding floating point numbers).

6.5 05 (05_{16}): Force Single Coil

This function permits the user to set the state of a single Discrete Output “Coil” register. (See Section 7.2 on page 39 for details of the individual registers). In the MFC010 implementation, these registers are used to initiate commands and functions. Setting the Output state initiates the function, attempting to clear the Output state will result in a data error. The format of the Master request telegram for this function should be as follows.

Request Character	Field	For Example	
1	Slave Address	01_{16}	Request to Slave ID 1
2	Function	05_{16}	“Force Single Coil”
3	Coil Address Hi	03_{16}	Coil Address = 1001
4	Coil Address Lo	$E8_{16}$	
5	Force Data Hi	FF_{16}	Set Coil “Active”
6	Force Data Lo	00_{16}	
7	CRC Lo	$0C_{16}$	CRC Checksum
8	CRC Hi	$4A_{16}$	

The MFC010 will respond to such a request with a telegram formatted as follows.

Response Character	Field	For Example	
1	Slave Address	01_{16}	Response from Slave ID 1
2	Function	05_{16}	“Force Single Coil”
3	Coil Address Hi	03_{16}	Coil Address = 1001
4	Coil Address Lo	$E8_{16}$	
5	Force Data Hi	FF_{16}	Set Coil “Active”
6	Force Data Lo	00_{16}	
7	CRC Lo	$0C_{16}$	CRC Checksum
8	CRC Hi	$4A_{16}$	

The MFC010 (slave) response telegram should be an exact duplicate of the master request telegram.

6.6 06 (06_{16}): Preset Single Register

This function permits the user to set the value of a single Holding register. For this reason this command cannot be used to write to variables that occupy multiple consecutive registers such as floating point and long integer variables (See Section 7.5 on page 49 for details of the individual registers). The format of the Master request telegram for this function should be as follows.

Request Character	Field	For Example	
1	Slave Address	01_{16}	Request to Slave ID 1
2	Function	06_{16}	"Preset Single Register"
3	Register Address Hi	03_{16}	Register Address = 1020
4	Register Address Lo	FB_{16}	
5	Preset Data Hi	00_{16}	Set Register 1020 = 35
6	Preset Data Lo	23_{16}	
7	CRC Lo	$B9_{16}$	CRC Checksum
8	CRC Hi	$A6_{16}$	

The MFC010 will respond to such a request with a telegram formatted as follows.

Response Character	Field	For Example	
1	Slave Address	01_{16}	Response from Slave ID 1
2	Function	06_{16}	"Preset Single Register"
3	Register Address Hi	03_{16}	Register Address = 1020
4	Register Address Lo	FB_{16}	
5	Preset Data Hi	00_{16}	Set Register 1020 = 35
6	Preset Data Lo	23_{16}	
7	CRC Lo	$B9_{16}$	CRC Checksum
8	CRC Hi	$A6_{16}$	

The MFC010 (slave) response telegram should be an exact duplicate of the master request

6.7 07 (07_{16}): Read Exception Status

When the Master device requests this command function, the MFC010 will respond with a single 8 bit data character summarizing the status of the instrument. The Master query telegram format is.

Request Character	Field	For Example	
1	Slave Address	01_{16}	Request to Slave ID 1
2	Function	07_{16}	"Read Exception Status"
3	CRC Lo	41_{16}	CRC Checksum
4	CRC Hi	$E2_{16}$	

The MFC010 will respond to such a request with a telegram formatted as follows.

Response Character	Field	For Example	
1	Slave Address	01_{16}	Response from Slave ID 1
2	Function	07_{16}	"Read Exception Status"
3	Status	$3D_{16}$	Data = 00111101_2
4	CRC Lo	$E3_{16}$	CRC Checksum
5	CRC Hi	$E1_{16}$	

The Status character received in the response will be formatted as follows :

Bit 0 (LSB) Bit 1	System State :	00_2 = Measuring, 01_2 = Standby, 10_2 = Stop, 11_2 = Start-up.
Bit 2	EEPROM Save Status :	0 = All Data Saved to EEPROM 1= Data Write to EEPOM Pending
Bit 3	Process Control Status : 0 = Process Control Inactive (Control Condition Invalid) 1 = Process Control Active (Control Condition Valid)	
Bit 4	Zero Calibration Status :	0 = Zero Calibration OK 1 = Zero Calibration Error
Bit 5	Density Calibration Status :	0 = Density Calibration OK 1 = Density Calibration Error
Bit 6	Process Warning Status :	0 = No Process Warning Flag(s) Detected 1 = Process Warning Flag(s) Detected
Bit 7 (MSB)	System Error Status :	0 = No System Error Flag(s) Detected 1 = System Error Condition Flag(s) Detected

6.8 08 (08_{16}): Diagnostics

This command function permits the user to perform one of several diagnostic operations, such as retrieving the error and event logs. For further details on this command function, refer to the Modbus specification.

6.9 11 ($0B_{16}$): Fetch Comm. Event Counter

This function allows the master device to determine if request telegrams are being properly processed. The Event count returned is a count of the number of request telegrams which have been received and processed without errors occurring. By fetching the Event count before and after a series of messages the master can determine whether the messages were handled normally. When the Master device requests this command function the MFC010 will respond with a two character (16 bit) status value and a two character event count. The Master request telegram should be formatted as follows.

Request Character	Field	For Example	
1	Slave Address	01_{16}	Request to Slave ID 1
2	Function	$0B_{16}$	"Fetch Comm. Event Counter"
3	CRC Lo	41_{16}	CRC Checksum
4	CRC Hi	$E7_{16}$	

The MFC010 will respond to such a request with a telegram formatted as follows.

Response Character	Field	For Example	
1	Slave Address	01_{16}	Response from Slave ID 1
2	Function	$0B_{16}$	"Fetch Comm. Event Counter"
3	Status Hi	FF_{16}	Instrument Status
4	Status Lo	FF_{16}	
5	Count Hi	$1E_{16}$	Event Count = 7891
6	Count Lo	$D3_{16}$	
7	CRC Lo	EC_{16}	CRC Checksum
8	CRC Hi	12_{16}	

The status value is either $FFFF_{16}$, in which case the slave is still processing a command, or 0000_{16}, in which case the slave is ready to receive the next command request.

6.10 16 (10_{16}): Preset Multiple Registers

This function permits the user to set the value of a number of consecutive Holding registers. This command function must be used to write to variables which occupy multiple consecutive registers such as floating point and long integer variables (See Section 7.5 on page 49 for details of the individual registers). Some of the Variables which occupy single (16 bit) registers cannot be set using this command, the Password registers (1001-1003) in particular. Command Function 06 (06_{16}) must be used to set these registers. The format of the Master request telegram for this function should be as follows.

Request Character	Field	For Example	
1	Slave Address	01_{16}	Request to Slave ID 1
2	Function	10_{16}	“Preset Multiple Registers”
3	Starting Address Hi	03_{16}	Starting Register Address = 1020
4	Starting Address Lo	FB_{16}	
5	No. of Registers Hi	00_{16}	Number of Registers = 2
6	No. of Registers Lo	02_{16}	
7	Byte Count	04_{16}	No of Bytes = 4 (2 x 2)
8	Data Hi	00_{16}	Set Register 1020 = 17
9	Data Lo	11_{16}	
10	Data Hi	00_{16}	Set Register 1021 = 18
11	Data Lo	12_{16}	
12	CRC Lo	79_{16}	CRC Checksum
13	CRC Hi	$A0_{16}$	

The MFC010 will respond to such a request with a telegram formatted as follows.

Response Character	Field	For Example	
1	Slave Address	01_{16}	Response from Slave ID 1
2	Function	10_{16}	“Preset Multiple Registers”
3	Starting Address Hi	03_{16}	Starting Register Address = 1020
4	Starting Address Lo	FB_{16}	
5	No. of Registers Hi	00_{16}	Number of Registers = 2
6	No. of Registers Lo	02_{16}	
7	CRC Lo	30_{16}	CRC Checksum
8	CRC Hi	$7D_{16}$	

6.11 17 (11_{16}): Report Slave ID

The Report Slave ID command is useful to retrieve all of the identification information from the system with one simple short request. The master request telegram should be 4 bytes long and formatted as follows.

Request Character	Field	For Example	
1	Slave Address	01_{16}	Request to Slave ID 1
2	Function	11_{16}	"Report Slave ID"
3	CRC Lo	$C0_{16}$	CRC Checksum
4	CRC Hi	$2C_{16}$	

The MFC010 response telegram will be 57 characters long (including the two CRC checksum bytes appended to the telegram) and is structured as follows.

Response Character	Field	For Example	
1	Slave Address	01_{16}	Response from Slave ID 1
2	Function	11_{16}	"Report Slave ID"
3	Byte Count	34_{16}	"No of bytes in Reply " = 52
4	Device ID	00_{16}	00 = MFC010
5	Run Indicator	FF_{16}	0 = Off, FF_{16} = On
6	Sensor Type (See Holding Register No. 1012)		
7	Sensor Size (See Holding Register No. 1013)		
8	Sensor Material (See Holding Register No. 1014)		
9 – 20	Software Version - 12 Character ASCII String		
21 – 34	Software Number - 14 Character ASCII String		
35 – 46	Software Compilation Date - 12 Character ASCII String		
47 – 49	MFC010 Serial Number – 24 bit Integer (Most Significant Byte First)		
50 – 52	Sensor Serial Number – 24 bit Integer (Most Significant Byte First)		
53 - 55	System Serial Number – 24 bit Integer (Most Significant Byte First)		
56	CRC Lo		CRC Checksum
57	CRC Hi		

Алматы (7273)495-231
Ангарск (3955)60-70-56
Архангельск (8182)63-90-72
Астрахань (8512)99-46-04
Барнаул (3852)73-04-60
Белгород (4722)40-23-64
Благовещенск (4162)22-76-07
Брянск (4832)59-03-52
Владивосток (423)249-28-31
Владикавказ (8672)28-90-48
Владимир (4922)49-43-18
Волгоград (844)278-03-48
Вологда (8172)26-41-59
Воронеж (473)204-51-73
Екатеринбург (343)384-55-89
Иваново (4932)77-34-06
Ижевск (3412)26-03-58
Иркутск (395)279-98-46
Казань (843)206-01-48
Калининград (4012)72-03-81
Калуга (4842)92-23-67
Кемерово (3842)65-04-62
Киров (8332)68-02-04
Коломна (4966)23-41-49
Кострома (4942)77-07-48
Краснодар (861)203-40-90
Красноярск (391)204-63-61
Курган (3522)50-90-47
Курск (4712)77-13-04
Липецк (4742)52-20-81
Магнитогорск (3519)55-03-13
Москва (495)268-04-70
Мурманск (8152)59-64-93
Набережные Челны (8552)20-53-41
Нижний Новгород (831)429-08-12
Новокузнецк (3843)20-46-81
Новосибирск (383)227-86-73
Ноябрьск(3496)41-32-12
Омск (3812)21-46-40
Орел (4862)44-53-42
Оренбург (3532)37-68-04
Пенза (8412)22-31-16
Пермь (342)205-81-47
Петрозаводск (8142)55-98-37
Псков (8112)59-10-37
Ростов-на-Дону (863)308-18-15
Рязань (4912)46-61-64
Самара (846)206-03-16
Санкт-Петербург (812)309-46-40
Саранск (8342)22-96-24
Саратов (845)249-38-78
Севастополь (8692)22-31-93
Симферополь (3652)67-13-56
Смоленск (4812)29-41-54
Сочи (862)225-72-31
Ставрополь (8652)20-65-13
Сургут (3462)77-98-35
Сыктывкар (8212)25-95-17
Тамбов (4752)50-40-97
Тверь (4822)63-31-35
Тольятти (8482)63-91-07
Томск (3822)98-41-53
Тула (4872)33-79-87
Тюмень (3452)66-21-18
Улан-Удэ (3012)59-97-51
Ульяновск (8422)24-23-59
Уфа (347)229-48-12
Хабаровск (4212)92-98-04
Чебоксары (8352)28-53-07
Челябинск (351)202-03-61
Череповец (8202)49-02-64
Чита (3022)38-34-83
Якутск (4112)23-90-97
Ярославль (4852)69-52-93

Россия +7(495)268-04-70 **К**азахстан +7(7172)727-132 **К**иргизия +996(312)96-26-47

https://opti.nt-rt.ru || opti@nt-rt.ru