

MERID Communication Driver

Driver for Serial Communication
with MERIDIAN Universal Gas Detector

Contents

CONTENTS1

INTRODUCTION2

GENERAL INFORMATION.....3

 DEVICE SPECIFICATIONS.....3

 NETWORK SPECIFICATIONS.....3

 DRIVER CHARACTERISTICS3

 CONFORMANCE TESTING 15

SELECTING THE DRIVER 16

CONFIGURING THE DEVICE 17

CONFIGURING THE DRIVER 17

 CONFIGURING THE COMMUNICATION SETTINGS 17

 CONFIGURING THE DRIVER WORKSHEETS 18

EXECUTING THE DRIVER 25

TROUBLESHOOTING 26

SAMPLE APPLICATION 28

REVISION HISTORY..... 29

Introduction

The MERID driver enables serial communication between the Studio system and the MERIDIAN Gas Detector from Scott Safety through a proprietary variation of the Modbus protocol.

This document will help you to select, configure and execute the MERID driver, and it is organized as follows:

- **Introduction:** This section, which provides an overview of the document.
- **General Information:** Identifies all of the hardware and software components required to implement communication between the Studio system and the target device.
- **Selecting the Driver:** Explains how to select the MERID driver in the Studio system.
- **Configuring the Device:** Describes how the target device must be configured to receive communication from the MERID driver.
- **Configuring the Driver:** Explains how to configure the MERID driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver:** Explains how to execute the MERID driver during application runtime.
- **Troubleshooting:** Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **Sample Application:** Explains how to use a sample application to test the MERID driver configuration
- **Revision History:** Provides a log of all changes made to the driver and this documentation.

Notes:

- This document assumes that you have read the “Development Environment” chapter in Studio’s *Technical Reference Manual*.

General Information

This chapter identifies all of the hardware and software components required to implement serial communication between the MERID driver in Studio and remote devices using the MERIDs protocol.

Device Specifications

To establish serial communication, your target device must meet the following specifications:

- **Manufacturer:** Scott Safety (a Tyco business)
- **Compatible Equipment:** MERIDIAN Universal Gas Detector
- **Programmer Software:** None

Network Specifications

To establish communication, your device network must meet the following specifications:

- **Device Communication Port:** Device's built-in Serial Port
- **Physical Protocol:** RS232 / RS485
- **Logic Protocol:** Proprietary variation of the Modbus protocol
- **Device Runtime Software:** None
- **Specific PC Board:** None
- **Cable Wiring Scheme:** Standard RS485 connection (see device's manual for details)

Driver Characteristics

The MERID driver package consists of the following files, which are automatically installed in the `\DRV` subdirectory of Studio:

- **MERID.INI:** Internal driver file. *You must not modify this file.*
- **MERID.MSG:** Internal driver file containing error messages for each error code. *You must not modify this file.*
- **MERID.PDF:** This document, which provides detailed information about the MERID driver.
- **MERID.DLL:** Compiled driver.

You can use the MERID driver on the following operating systems:

- Windows XP/Vista/7/8/2003/2008/2012
- Windows XPe/Embedded Standard
- Windows Embedded and CE 5.x, 6.x, 7

The MERID driver supports the following registers:

STANDARD MODBUS REGISTERS (Read Only)

Register Type	Data Type	Length	Bit	String Size	Remarks
4x	B	1 Byte	•		Signed Byte
	UB	1 Byte	•		Unsigned Byte
	W	1 Word	•		Signed Word (Default)
	UW	1 Word	•		Unsigned Word
	BCDDW	2 Words	•		Binary-coded Decimal Double Word
	F	2 Words			Floating Point
	S	Varies		•	String

MERIDIAN PROPERTY NAME (TRANSMITTER)

Meridian Property Name	Data Type	Access	Address		Remarks
			MERIDIAN	MODBUS	
TXTemperature	Integer	R	MERIDIAN:0:TXTemperature	4X:B2	-128 tp +127°C
TXVoltage	Integer	R	MERIDIAN:0:TXVoltage	4x:UW117	Voltage / 10. Example: 214=24.1V
RelayStatus	Integer	R	MERIDIAN:0:RelayStatus	4x:UW129	1=Energized bit0: Relay 1 Status bit1: Relay 2 Status bit2: Relay 3 Status bit3: Relay 4 Status bit7: GUI Edit Mode
CurrentHumidity	Integer	R	MERIDIAN:0:CurrentHumidity	4x:UW130	
TXStatus	Integer	R	MERIDIAN:0:TXStatus	4x:UW132	bit0: TX Fault 9fault record holds the fault code) bit1: Loop 1 current halt, 1=halt bit2: Loop 2 current halt, 1=halt bit3: Loop 3 current halt, 1=halt bit5: System inhibit, 1=inhibit bit6: GUI Edit Mode bit11: remote configuration lockout
TXOptions	Integer	R	MERIDIAN:0:TXOptions	4x:UW133	bit0: 2 wire bit1: 3-4 wire bit2: battery option bit3: wired HART expansion bit4: wireless HART expansion bit5: wireless ISA 100.11A
CECStatus	Integer	R	MERIDIAN:0:CECStatus	4x:UW142	Lower byte: CEC1Status Upper byte: CEC2Status If 0=No CEC detected bit0 = CEC EEPROM detected bit1 = CEC uP comms up bit2 = CECE request DataBase bit3 = DB download complete bit4 = DB mismatch bit5 = DB CRC error bit6 = Protocol error bit7 = Undefined error
TXCurrentDate	Integer	RW	MERIDIAN:0:TXCurrentDate	4x:BCDDW177	transmitter's current date. BCD format:00,year,month,day

Meridian Property Name	Data Type	Access	Address		Remarks
			MERIDIAN	MODBUS	
TXCurrentTime	Integer	RW	MERIDIAN:0:TXCurrentTime	4x:BCDDW179	transmitter's current time. BCD format:00,year,month,day
LoopCurrent1	Real	R	MERIDIAN:0:LoopCurrent1	4x:F263	
LoopCurrent2	Real	R	MERIDIAN:0:LoopCurrent2	4x:F265	
LoopCurrent3	Real	R	MERIDIAN:0:LoopCurrent3	4x:F267	
MBSlaveAddress	Integer	R	MERIDIAN:0:MBSlaveAddress	4x:UB513	TX MODBUS address, 1-247
MBSlaveBaudIndex	Integer	R	MERIDIAN:0:MBSlaveBaudIndex	4x:UB514	TX MODBUS baudrate, 1 = 9600, 2 = 19200, 2 is default
MBSlaveParity	Integer	R	MERIDIAN:0:MBSlaveParity	4x:UB515	TX MODBUS parity type 0 = none, 1 = odd, 2 = even, even is default When parity even/odd. 1 stop bit hen parity none, 2 stop bits
MBSlaveStopBit	Integer	R	MERIDIAN:0:MBSlaveStopBit	4x:UB516	
Regional	Integer	RW	MERIDIAN:0:Regional	4x:UB518	Languages, Dates format, Daylight saving time bit3-0:Languages 0=English(en) 1=Spanish(es) 2=Portuguese(pt) 3=French(fr) 4=Russian(ru) 5=Chinese(zh) 6-15=Reserved bit5-4: Date Format 0=MDY 1=DMY 2=YMD 3=Reserved bit7-6:Daylight saving time 0=Off 1=On 2-3=Reserved
Alarm1Logic	Integer	RW	MERIDIAN:0:Alarm1Logic	4x:UB519	When latching, acknowledge is required bit0: 0=nonlatching, 1=latching bit1: 0=nonfailsafe, 1=failsafe
Alarm2Logic	Integer	RW	MERIDIAN:0:Alarm2Logic	4x:UB520	When latching, acknowledge is required bit0: 0=nonlatching, 1=latching bit1: 0=nonfailsafe, 1=failsafe
Alarm3Logic	Integer	RW	MERIDIAN:0:Alarm3Logic	4x:UB521	When latching, acknowledge is required bit0: 0=nonlatching, 1=latching bit1: 0=nonfailsafe, 1=failsafe
PrevGasType	Integer	R	MERIDIAN:0:PrevGasType	4x:B529	
Alarm1OffTimeDelay	Integer	RW	MERIDIAN:0:Alarm1OffTimeDelay	4x:UW594	This time, in seconds, is the delay on or off time for the alarm relay/LED pair.
Alarm2OffTimeDelay	Integer	RW	MERIDIAN:0:Alarm2OffTimeDelay	4x:UW596	This time, in seconds, is the delay on or off time for the alarm relay/LED pair.
Alarm3OffTimeDelay	Integer	RW	MERIDIAN:0:Alarm3OffTimeDelay	4x:UW598	This time, in seconds, is the delay on or off time for the alarm relay/LED pair.
UserPassword	Integer	RW	MERIDIAN:0>UserPassword	4x:UW599	Access for enabling operator level functions. Binary: 4 digit number. le '0000'
SysMgrPassword	Integer	RW	MERIDIAN:0:SysMgrPassword	4x:UW601	Access for enabling sys mgr functions. Binary: 4 digit number. le '0000'
TXDataBaseVersion	Integer	R	MERIDIAN:0:TXDataBaseVersion	4x:UW607	High byte major, low byte minor
InhibitCurrent_FP32	Real	RW	MERIDIAN:0:InhibitCurrent_FP32	4x:F659	The 4-20mA output that is sent when the device is in inhibit mode in .1mA steps.
TXName	String	RW	MERIDIAN:0:TXName	4x:S753:18	User defined name/location 2 bytes per register - 16 chars ASCII and NULL, NULL.
TXModelNumber	String	R	MERIDIAN:0:TXModelNumber	4x:S762:10	Mode Number 8 chars, ASCII

Meridian Property Name	Data Type	Access	Address		Remarks
			MERIDIAN	MODBUS	
TXSerialNumber	String	R	MERIDIAN:0:TXSerialNumber	4x:S767:16	sn example 115Ayywwnnnnnn 115 is the company (Monroe) A is assembly (s is subassy) 08 is the year of manufacture 16 is the week of manufacture xxxxxx is the sequential number produced within the week. 14 chars ASCII
TXCodeVersion	String	R	MERIDIAN:0:TXCodeVersion	4x:S775:6	Firmware version 4 chars, ASCII
TXLatitude	String	RW	MERIDIAN:0:TXLatitude	4x:S778:12	10 chars. Null terminated
TXLongitude	String	RW	MERIDIAN:0:TXLongitude	4x:S784:12	10 chars. Null terminated
ResetTransmitter	Trigger	W	MERIDIAN:0:ResetTransmitter	N.A.	Reset transmitter when the respective tag changes of value.
AlarmAcknowledge	Trigger	W	MERIDIAN:0:AlarmAcknowledge	N.A.	Alarm Acknowledge when the respective tag changes of value.
Enter	Trigger	W	MERIDIAN:0:Enter	N.A.	Enter (Enter is used for confirm)
Esc	Trigger	W	MERIDIAN:0:Esc	N.A.	Esc (Esc/Back is used for cancel)
SensorsEnabledAndOffline	Integer	RW	MERIDIAN:0:SensorsEnabledAndOffline	4x:UB522	Bit0 = Sensor 1 Enabled Bit1 = Sensor 2 Enabled Bit2 = Sensor 3 Enabled Bit3 = Sensor 1 Offline Bit4 = Sensor 2 Offline Bit5 = Sensor 3 Offline

MERIDIAN PROPERTY NAME (SENSORS)

Meridian Property Name	Data Type	Access	Sensor	Address		Remarks
				MERIDIAN	MODBUS	
Alarm1Status	Boolean	R	1	MERIDIAN:1:Alarm1Status	4x:UW118.0	1=Sensor in alarm
			2	MERIDIAN:2:Alarm1Status	4x:UW118.1	
			3	MERIDIAN:3:Alarm1Status	4x:UW118.2	
Alarm2Status	Boolean	R	1	MERIDIAN:1:Alarm2Status	4x:UW119.0	1=Sensor in alarm
			2	MERIDIAN:2:Alarm2Status	4x:UW119.1	
			3	MERIDIAN:3:Alarm2Status	4x:UW119.2	
Alarm3Status	Boolean	R	1	MERIDIAN:1:Alarm3Status	4x:UW120.0	1=Sensor in alarm
			2	MERIDIAN:2:Alarm3Status	4x:UW120.1	
			3	MERIDIAN:3:Alarm3Status	4x:UW120.2	
FaultStatus	Boolean	R	1	MERIDIAN:1:FaultStatus	4x:UW121.0	1=Sensor in fault
			2	MERIDIAN:2:FaultStatus	4x:UW121.1	
			3	MERIDIAN:3:FaultStatus	4x:UW121.2	
InhibitStatus	Boolean	R	1	MERIDIAN:1:InhibitStatus	4x:UW122.0	1=Sensor in inhibit
			2	MERIDIAN:2:InhibitStatus	4x:UW122.1	
			3	MERIDIAN:3:InhibitStatus	4x:UW122.2	
CalStatus	Boolean	R	1	MERIDIAN:1:CalStatus	4x:UW123.0	1=Sensor Cal Due
			2	MERIDIAN:2:CalStatus	4x:UW123.1	
			3	MERIDIAN:3:CalStatus	4x:UW123.2	

Meridian Property Name	Data Type	Access	Sensor	Address		Remarks
				MERIDIAN	MODBUS	
SensorConnected	Boolean	R	1	MERIDIAN:1:SensorConnected	4x:UW124.0	1=Sensor Connected
			2	MERIDIAN:2:SensorConnected	4x:UW124.1	
			3	MERIDIAN:3:SensorConnected	4x:UW124.2	
SensorStatusHL	Integer	R	1	MERIDIAN:1:SensorStatusHL	4x:UW134	High level status of sensor bit0: Sensor enabled bit1: Alarm 1 trip bit2: Alarm 2 trip bit3: Alarm 3 trip bit4: Sensor fault bit5: Sensor inhibit bit6: Sensor connected bit7: Sensor calibrating bit8: Sensor Cal due bit9: Loop signaling (0-live, 1=fixe) bit14: Reserved bit15: Reserved
			2	MERIDIAN:2:SensorStatusHL	4x:UW135	
			3	MERIDIAN:3:SensorStatusHL	4x:UW136	
SensorGasConc10Bit	Integer	R	1	MERIDIAN:1:SensorGasConc10Bit	4x:UW137	200 = 0%Full Scale, 1000 = 100%Full Scale, 0 = -25%Full Scale
			2	MERIDIAN:2:SensorGasConc10Bit	4x:UW138	
			3	MERIDIAN:3:SensorGasConc10Bit	4x:UW139	
SensorGasConc	Real	R	1	MERIDIAN:1:SensorGasConc	4x:F257	Linearized and temp compensated gas concentration. These values are floats that match the decimal place and displayed.
			2	MERIDIAN:2:SensorGasConc	4x:F259	
			3	MERIDIAN:3:SensorGasConc	4x:F261	
SensorGasConcASCII	String	R	1	MERIDIAN:1:SensorGasConcASCII	4x:S337:8	6 char string. This is what is displayed on the LCD, includes decimal point.
			2	MERIDIAN:2:SensorGasConcASCII	4x:S344:8	
			3	MERIDIAN:3:SensorGasConcASCII	4x:S351:8	
SensorGasUnitsASCII	String	R	1	MERIDIAN:1:SensorGasUnitsASCII	4x:S341:6	4 char string. This is what is displayed on the display, gas units.
			2	MERIDIAN:2:SensorGasUnitsASCII	4x:S348:6	
			3	MERIDIAN:3:SensorGasUnitsASCII	4x:S355:6	
CurrentLoopOffset	Integer	RW	1	MERIDIAN:1:CurrentLoopOffset	4x:B523	signed byte -128 to 127
			2	MERIDIAN:2:CurrentLoopOffset	4x:B524	
			3	MERIDIAN:3:CurrentLoopOffset	4x:B525	
CurrentLoop20mAOffset	Integer	RW	1	MERIDIAN:1:CurrentLoop20mAOffset	4x:B530	signed byte -128 to 127
			2	MERIDIAN:2:CurrentLoop20mAOffset	4x:B531	
			3	MERIDIAN:3:CurrentLoop20mAOffset	4x:B532	
ZeroOffset	Integer	R	1	MERIDIAN:1:ZeroOffset	4x:UW4155	
			2	MERIDIAN:2:ZeroOffset	4x:UW8251	
			3	MERIDIAN:3:ZeroOffset	4x:UW12347	
CurrentCalTemperature	Real	R	1	MERIDIAN:1:CurrentCalTemperature	4x:F4193	Updated by the sensor when span is complete. xx.x°C
			2	MERIDIAN:2:CurrentCalTemperature	4x:F8289	
			3	MERIDIAN:3:CurrentCalTemperature	4x:F12385	
CalGasConc	Real	RW	1	MERIDIAN:1:CalGasConc	4x:F4199	Default is 50% FS
			2	MERIDIAN:2:CalGasConc	4x:F8295	
			3	MERIDIAN:3:CalGasConc	4x:F12391	

Meridian Property Name	Data Type	Access	Sensor	Address		Remarks
				MERIDIAN	MODBUS	
CurrentCalFactor	Real	R	1	MERIDIAN:1:CurrentCalFactor	4x:F4201	Updated by the sensor after a successful span
			2	MERIDIAN:2:CurrentCalFactor	4x:F8297	
			3	MERIDIAN:3:CurrentCalFactor	4x:F12393	
SensorStatus	Integer	R	1	MERIDIAN:1:SensorStatus	4x:UW4262	bit0: Normal bit1: Set_Defaults (factory defaults) bit2: CRC_Fault bit3: Comb_Under_Volt_Fault (combust undervolt) bit4: Comb_Over_Fault (combust overvolt) bit5: Comb_Over_Rng (combustible overrange flag) bit8: Under_Volt_Fault (processor undervolts) bit9: Over_Volt_Fault (processor overvolts) bit10: Sensor_Fault (mainly ADC overrange) bit12: NV_MEM_Fault (read verify failed after write) bit13: WDT_Fault bit14: Startup_fault (EEPROM or proc DCO) bit15: IIC_Fault (I2C bus error)
			2	MERIDIAN:2:SensorStatus	4x:UW8358	
			3	MERIDIAN:3:SensorStatus	4x:UW12454	
SensorVoltage	Integer	R	1	MERIDIAN:1:SensorVoltage	4x:UW4263	mV
			2	MERIDIAN:2:SensorVoltage	4x:UW8359	
			3	MERIDIAN:3:SensorVoltage	4x:UW12455	
GasAtoD	Integer	R	1	MERIDIAN:1:GasAtoD	4x:UW4264	10bit, 0-1023, current AtoD output
			2	MERIDIAN:2:GasAtoD	4x:UW8360	
			3	MERIDIAN:3:GasAtoD	4x:UW12456	
GasConc	Real	R	1	MERIDIAN:1:GasConc	4x:F4305	Linearized and temp compensated gas concentration.
			2	MERIDIAN:2:GasConc	4x:F8401	
			3	MERIDIAN:3:GasConc	4x:F12497	
SensorTemp	Real	R	1	MERIDIAN:1:SensorTemp	4x:F4307	°C
			2	MERIDIAN:2:SensorTemp	4x:F8403	
			3	MERIDIAN:3:SensorTemp	4x:F12499	
GasType	Integer	RW	1	MERIDIAN:1:GasType	4x:UB4385	This is used by the sensor to determine which code to run. Each sensor will have a specific number. Toxics: 1-127, IR & CB: 128-255. Note: The values are used for internal purposes to identify sensors to the transmitter.
			2	MERIDIAN:2:GasType	4x:UB8481	
			3	MERIDIAN:3:GasType	4x:UB12577	
GasUnits	Integer	R	1	MERIDIAN:1:GasUnits	4x:UB4386	
			2	MERIDIAN:2:GasUnits	4x:UB8482	
			3	MERIDIAN:3:GasUnits	4x:UB12578	
GasRangeIndex	Integer	RW	1	MERIDIAN:1:GasRangeIndex	4x:UB4387	Indicates current Gas Range.

Meridian Property Name	Data Type	Access	Sensor	Address		Remarks
				MERIDIAN	MODBUS	
			2	MERIDIAN:2:GasRangeIndex	4x:UB8483	Upper nibble displays decimal points. Bits 0-3: Index to ranges/gain indexes. Bi4,5:1,2:display DP 0-0, 1=1, 2=2
			3	MERIDIAN:3:GasRangeIndex	4x:UB12579	
DisplayNegative	Integer	RW	1	MERIDIAN:1:DisplayNegative	4x:UB4389	Display negative or 0 when negative bit0: 0=no, 1 = yes
			2	MERIDIAN:2:DisplayNegative	4x:UB8485	
			3	MERIDIAN:3:DisplayNegative	4x:UB12581	
SensorLife	Integer	R	1	MERIDIAN:1:SensorLife	4x:UB4390	Percentage of sensor life left. 0-100
			2	MERIDIAN:2:SensorLife	4x:UB8486	
			3	MERIDIAN:3:SensorLife	4x:UB12582	
SensorWarmupTime	Integer	R	1	MERIDIAN:1:SensorWarmupTime	4x:UW4449	The amount of time, in 10 seconds, that the sensor needs to warmup. Inhibit during this time. Seconds x 10.
			2	MERIDIAN:2:SensorWarmupTime	4x:UW8545	
			3	MERIDIAN:3:SensorWarmupTime	4x:UW12641	
MfgCalTime	Integer	R	1	MERIDIAN:1:MfgCalTime	4x:UW4450	Calibration times. hh:mm (BCD format)
			2	MERIDIAN:2:MfgCalTime	4x:UW8546	
			3	MERIDIAN:3:MfgCalTime	4x:UW12642	
InstallCalTime	Integer	R	1	MERIDIAN:1:InstallCalTime	4x:UW4451	Calibration times. hh:mm (BCD format)
			2	MERIDIAN:2:InstallCalTime	4x:UW8547	
			3	MERIDIAN:3:InstallCalTime	4x:UW12643	
PriorCalTime	Integer	R	1	MERIDIAN:1:PriorCalTime	4x:UW4452	Calibration times. hh:mm (BCD format)
			2	MERIDIAN:2:PriorCalTime	4x:UW8548	
			3	MERIDIAN:3:PriorCalTime	4x:UW12644	
CurrentCalTime	Integer	R	1	MERIDIAN:1:CurrentCalTime	4x:UW4453	Calibration times. hh:mm (BCD format)
			2	MERIDIAN:2:CurrentCalTime	4x:UW8549	
			3	MERIDIAN:3:CurrentCalTime	4x:UW12645	
MfgZeroOffset	Integer	R	1	MERIDIAN:1:MfgZeroOffset	4x:UW4454	Zero offsets. ADC count
			2	MERIDIAN:2:MfgZeroOffset	4x:UW8550	
			3	MERIDIAN:3:MfgZeroOffset	4x:UW12646	
InstallZeroOffset	Integer	R	1	MERIDIAN:1:InstallZeroOffset	4x:UW4455	Zero offsets. ADC count
			2	MERIDIAN:2:InstallZeroOffset	4x:UW8551	
			3	MERIDIAN:3:InstallZeroOffset	4x:UW12647	
PriorZeroOffset	Integer	R	1	MERIDIAN:1:PriorZeroOffset	4x:UW4456	Zero offsets. ADC count
			2	MERIDIAN:2:PriorZeroOffset	4x:UW8552	
			3	MERIDIAN:3:PriorZeroOffset	4x:UW12648	
CalibrationPeriod	Integer	RW	1	MERIDIAN:1:CalibrationPeriod	4x:UW4457	Number of days between calibrations
			2	MERIDIAN:2:CalibrationPeriod	4x:UW8553	
			3	MERIDIAN:3:CalibrationPeriod	4x:UW12649	
InhibitTimer	Integer	RW	1	MERIDIAN:1:InhibitTimer	4x:UW4458	This particular sensor inhibit timer. Used during Span. In seconds (Min. of 5 Mins., Max. of 10 Mins).
			2	MERIDIAN:2:InhibitTimer	4x:UW8554	
			3	MERIDIAN:3:InhibitTimer	4x:UW12650	

Meridian Property Name	Data Type	Access	Sensor	Address		Remarks
				MERIDIAN	MODBUS	
PurgeTimer	Integer	RW	1	MERIDIAN:1:PurgeTimer	4x:UW4459	Time after span that output is inhibited for shit sensor. In Seconds (Min. of 5 Mins., Max. of 10 Mins)
			2	MERIDIAN:2:PurgeTimer	4x:UW8555	
			3	MERIDIAN:3:PurgeTimer	4x:UW12651	
GasRange0	Integer	R	1	MERIDIAN:1:GasRange0	4x:UW4472	Lowest range. Range; lowest to highest. Note: The number of ranges vary based on the sensor. The min ans max balues vary based on the sensor.
			2	MERIDIAN:2:GasRange0	4x:UW8568	
			3	MERIDIAN:3:GasRange0	4x:UW12664	
GainEunits0	Integer	R	1	MERIDIAN:1:GainEunits0	4x:UW4473	Gain 1,0 & Eunits Split into 2 bytes: Highest byte Reserved The low byte is the Eunits Eunits: 01 = ppm 02 = ppb 03 = %LEL -4 = %V/V bit 7 indicates default range. Example: 0x0A01 Eunits=ppm
			2	MERIDIAN:2:GainEunits0	4x:UW8569	
			3	MERIDIAN:3:GainEunits0	4x:UW12665	
MaxCalFactor0	Integer	R	1	MERIDIAN:1:MaxCalFactor0	4x:UW4474	Minimum acceptable cal factor
			2	MERIDIAN:2:MaxCalFactor0	4x:UW8570	
			3	MERIDIAN:3:MaxCalFactor0	4x:UW12666	
GasRange1	Integer	R	1	MERIDIAN:1:GasRange1	4x:UW4475	Lowest range. Range; lowest to highest. Note: The number of ranges vary based on the sensor. The min ans max balues vary based on the sensor.
			2	MERIDIAN:2:GasRange1	4x:UW8571	
			3	MERIDIAN:3:GasRange1	4x:UW12667	
GainEunits1	Integer	R	1	MERIDIAN:1:GainEunits1	4x:UW4476	Gain 1,0 & Eunits Split into 2 bytes: Highest byte Reserved The low byte is the Eunits Eunits: 01 = ppm 02 = ppb 03 = %LEL -4 = %V/V bit 7 indicates default range. Example: 0x0A01 Eunits=ppm
			2	MERIDIAN:2:GainEunits1	4x:UW8572	
			3	MERIDIAN:3:GainEunits1	4x:UW12668	
MaxCalFactor1	Integer	R	1	MERIDIAN:1:MaxCalFactor1	4x:UW4477	Minimum acceptable cal factor
			2	MERIDIAN:2:MaxCalFactor1	4x:UW8573	
			3	MERIDIAN:3:MaxCalFactor1	4x:UW12669	
GasRange2	Integer	R	1	MERIDIAN:1:GasRange2	4x:UW4478	Lowest range. Range; lowest to highest. Note: The number of ranges vary based on the sensor. The min ans max balues vary based on the sensor.
			2	MERIDIAN:2:GasRange2	4x:UW8574	
			3	MERIDIAN:3:GasRange2	4x:UW12670	
GainEunits2	Integer	R	1	MERIDIAN:1:GainEunits2	4x:UW4479	Gain 1,0 & Eunits Split into 2 bytes: Highest byte Reserved
			2	MERIDIAN:2:GainEunits2	4x:UW8575	

Meridian Property Name	Data Type	Access	Sensor	Address		Remarks
				MERIDIAN	MODBUS	
			3	MERIDIAN:3:GainEunits2	4x:UW12671	The low byte is the Eunits Eunits: 01 = ppm 02 = ppb 03 = %LEL -4 = %V/V bit 7 indicates default range. Example: 0x0A01 Eunits=ppm
MaxCalFactor2	Integer	R	1	MERIDIAN:1:MaxCalFactor2	4x:UW4480	Minimum acceptable cal factor
			2	MERIDIAN:2:MaxCalFactor2	4x:UW8576	
			3	MERIDIAN:3:MaxCalFactor2	4x:UW12672	
GasRange3	Integer	R	1	MERIDIAN:1:GasRange3	4x:UW4481	Lowest range. Range; lowest to highest. Note: The number of ranges vary based on the sensor. The min ans max balues vary based on the sensor.
			2	MERIDIAN:2:GasRange3	4x:UW8577	
			3	MERIDIAN:3:GasRange3	4x:UW12673	
GainEunits3	Integer	R	1	MERIDIAN:1:GainEunits3	4x:UW4482	Gain 1,0 & Eunits Split into 2 bytes: Highest byte Reserved The low byte is the Eunits Eunits: 01 = ppm 02 = ppb 03 = %LEL -4 = %V/V bit 7 indicates default range. Example: 0x0A01 Eunits=ppm
			2	MERIDIAN:2:GainEunits3	4x:UW8578	
			3	MERIDIAN:3:GainEunits3	4x:UW12674	
MaxCalFactor3	Integer	R	1	MERIDIAN:1:MaxCalFactor3	4x:UW4483	Minimum acceptable cal factor
			2	MERIDIAN:2:MaxCalFactor3	4x:UW8579	
			3	MERIDIAN:3:MaxCalFactor3	4x:UW12675	
GasRange4	Integer	R	1	MERIDIAN:1:GasRange4	4x:UW4484	Lowest range. Range; lowest to highest. Note: The number of ranges vary based on the sensor. The min ans max balues vary based on the sensor.
			2	MERIDIAN:2:GasRange4	4x:UW8580	
			3	MERIDIAN:3:GasRange4	4x:UW12676	
GainEunits4	Integer	R	1	MERIDIAN:1:GainEunits4	4x:UW4485	Gain 1,0 & Eunits Split into 2 bytes: Highest byte Reserved The low byte is the Eunits Eunits: 01 = ppm 02 = ppb 03 = %LEL -4 = %V/V bit 7 indicates default range. Example: 0x0A01 Eunits=ppm
			2	MERIDIAN:2:GainEunits4	4x:UW8581	
			3	MERIDIAN:3:GainEunits4	4x:UW12677	
MaxCalFactor4	Integer	R	1	MERIDIAN:1:MaxCalFactor4	4x:UW4486	Minimum acceptable cal factor
			2	MERIDIAN:2:MaxCalFactor4	4x:UW8582	
			3	MERIDIAN:3:MaxCalFactor4	4x:UW12678	
GasRange5	Integer	R	1	MERIDIAN:1:GasRange5	4x:UW4487	Lowest range. Range; lowest to highest. Note: The number of ranges
			2	MERIDIAN:2:GasRange5	4x:UW8583	

Meridian Property Name	Data Type	Access	Sensor	Address		Remarks
				MERIDIAN	MODBUS	
			3	MERIDIAN:3:GasRange5	4x:UW12679	vary based on the sensor. The min ans max balues vary based on the sensor.
GainEunits5	Integer	R	1	MERIDIAN:1:GainEunits5	4x:UW4488	Gain 1,0 & Eunits Split into 2 bytes: Highest byte Reserved The low byte is the Eunits Eunits: 01 = ppm 02 = ppb 03 = %LEL -4 = %V/V bit 7 indicates default range. Example: 0x0A01 Eunits=ppm
			2	MERIDIAN:2:GainEunits5	4x:UW8584	
			3	MERIDIAN:3:GainEunits5	4x:UW12680	
MaxCalFactor5	Integer	R	1	MERIDIAN:1:MaxCalFactor5	4x:UW4489	Minimum acceptable cal factor
			2	MERIDIAN:2:MaxCalFactor5	4x:UW8585	
			3	MERIDIAN:3:MaxCalFactor5	4x:UW12681	
GasRange6	Integer	R	1	MERIDIAN:1:GasRange6	4x:UW4490	Lowest range. Range; lowest to highest. Note: The number of ranges vary based on the sensor. The min ans max balues vary based on the sensor.
			2	MERIDIAN:2:GasRange6	4x:UW8586	
			3	MERIDIAN:3:GasRange6	4x:UW12682	
GainEunits6	Integer	R	1	MERIDIAN:1:GainEunits6	4x:UW4491	Gain 1,0 & Eunits Split into 2 bytes: Highest byte Reserved The low byte is the Eunits Eunits: 01 = ppm 02 = ppb 03 = %LEL -4 = %V/V bit 7 indicates default range. Example: 0x0A01 Eunits=ppm
			2	MERIDIAN:2:GainEunits6	4x:UW8587	
			3	MERIDIAN:3:GainEunits6	4x:UW12683	
MaxCalFactor6	Integer	R	1	MERIDIAN:1:MaxCalFactor6	4x:UW4492	Minimum acceptable cal factor
			2	MERIDIAN:2:MaxCalFactor6	4x:UW8588	
			3	MERIDIAN:3:MaxCalFactor6	4x:UW12684	
GasRange7	Integer	R	1	MERIDIAN:1:GasRange7	4x:UW4493	Lowest range. Range; lowest to highest. Note: The number of ranges vary based on the sensor. The min ans max balues vary based on the sensor.
			2	MERIDIAN:2:GasRange7	4x:UW8589	
			3	MERIDIAN:3:GasRange7	4x:UW12685	
GainEunits7	Integer	R	1	MERIDIAN:1:GainEunits7	4x:UW4494	Gain 1,0 & Eunits Split into 2 bytes: Highest byte Reserved The low byte is the Eunits Eunits: 01 = ppm 02 = ppb 03 = %LEL -4 = %V/V bit 7 indicates default range. Example: 0x0A01 Eunits=ppm
			2	MERIDIAN:2:GainEunits7	4x:UW8590	
			3	MERIDIAN:3:GainEunits7	4x:UW12686	
MaxCalFactor7	Integer	R	1	MERIDIAN:1:MaxCalFactor7	4x:UW4495	Minimum acceptable cal

Meridian Property Name	Data Type	Access	Sensor	Address		Remarks
				MERIDIAN	MODBUS	
			2	MERIDIAN:2:MaxCalFactor7	4x:UW8591	factor
			3	MERIDIAN:3:MaxCalFactor7	4x:UW12687	
MfgCalDate	Integer	R	1	MERIDIAN:1:MfgCalDate	4x:BCDWD4545	Calibration dates 00,yy,mm,dd (BCD format) (1st byte always 0x00)
			2	MERIDIAN:2:MfgCalDate	4x:BCDDW8641	
			3	MERIDIAN:3:MfgCalDate	4x:BCDDW12737	
InstallCalDate	Integer	R	1	MERIDIAN:1:InstallCalDate	4x:BCDDW4547	Calibration dates 00,yy,mm,dd (BCD format) (1st byte always 0x00)
			2	MERIDIAN:2:InstallCalDate	4x:BCDDW8643	
			3	MERIDIAN:3:InstallCalDate	4x:BCDDW12739	
PriorCalDate	Integer	R	1	MERIDIAN:1:PriorCalDate	4x:BCDDW4549	Calibration dates 00,yy,mm,dd (BCD format) (1st byte always 0x00)
			2	MERIDIAN:2:PriorCalDate	4x:BCDDW8645	
			3	MERIDIAN:3:PriorCalDate	4x:BCDDW12741	
CurrentCalDate	Integer	R	1	MERIDIAN:1:CurrentCalDate	4x:BCDDW4551	Calibration dates 00,yy,mm,dd (BCD format) (1st byte always 0x00)
			2	MERIDIAN:2:CurrentCalDate	4x:BCDDW8647	
			3	MERIDIAN:3:CurrentCalDate	4x:BCDDW12743	
KFactor	Real	R	1	MERIDIAN:1:KFactor	4X:F4625	This is the ratio between the calgastype and the gastype
			2	MERIDIAN:2:KFactor	4X:F8721	
			3	MERIDIAN:3:KFactor	4X:F12817	
MfgCalFactor	Real	R	1	MERIDIAN:1:MfgCalFactor	4X:F4627	Cal factors
			2	MERIDIAN:2:MfgCalFactor	4X:F8723	
			3	MERIDIAN:3:MfgCalFactor	4X:F12819	
InstallCalFactor	Real	R	1	MERIDIAN:1:InstallCalFactor	4X:F4629	Cal factors
			2	MERIDIAN:2:InstallCalFactor	4X:F8725	
			3	MERIDIAN:3:InstallCalFactor	4X:F12821	
PriorCalFactor	Real	R	1	MERIDIAN:1:PriorCalFactor	4X:F4631	Cal factors
			2	MERIDIAN:2:PriorCalFactor	4X:F8727	
			3	MERIDIAN:3:PriorCalFactor	4X:F12823	
MfgCalTemperature	Real	R	1	MERIDIAN:1:MfgCalTemperature	4X:F4633	xx.x°C
			2	MERIDIAN:2:MfgCalTemperature	4X:F8729	
			3	MERIDIAN:3:MfgCalTemperature	4X:F12825	
InstallCalTemperature	Real	R	1	MERIDIAN:1:InstallCalTemperature	4X:F4635	xx.x°C
			2	MERIDIAN:2:InstallCalTemperature	4X:F8731	
			3	MERIDIAN:3:InstallCalTemperature	4X:F12827	
PriorCalTemperature	Real	R	1	MERIDIAN:1:PriorCalTemperature	4X:F4637	xx.x°C
			2	MERIDIAN:2:PriorCalTemperature	4X:F8733	
			3	MERIDIAN:3:PriorCalTemperature	4X:F12829	
Alarm1SetPoint	Real	RW	1	MERIDIAN:1:Alarm1SetPoint	4X:F4639	xx.x (default to a 10% of FS)
			2	MERIDIAN:2:Alarm1SetPoint	4X:F8735	
			3	MERIDIAN:3:Alarm1SetPoint	4X:F12831	
Alarm1ResetPoint	Real	RW	1	MERIDIAN:1:Alarm1ResetPoint	4X:F4641	xx.x (default to a 9% of FS)

Meridian Property Name	Data Type	Access	Sensor	Address		Remarks
				MERIDIAN	MODBUS	
			2	MERIDIAN:2:Alarm1ResetPoint	4X:F8737	
			3	MERIDIAN:3:Alarm1ResetPoint	4X:F12833	
Alarm2SetPoint	Real	RW	1	MERIDIAN:1:Alarm2SetPoint	4X:F4643	xx.x (default to a 25% of FS)
			2	MERIDIAN:2:Alarm2SetPoint	4X:F8739	
			3	MERIDIAN:3:Alarm2SetPoint	4X:F12835	
Alarm2ResetPoint	Real	RW	1	MERIDIAN:1:Alarm2ResetPoint	4X:F4645	xx.x (default to a 22% of FS)
			2	MERIDIAN:2:Alarm2ResetPoint	4X:F8741	
			3	MERIDIAN:3:Alarm2ResetPoint	4X:F12837	
Alarm3SetPoint	Real	RW	1	MERIDIAN:1:Alarm3SetPoint	4X:F4647	xx.x (default to a 50% of FS)
			2	MERIDIAN:2:Alarm3SetPoint	4X:F8743	
			3	MERIDIAN:3:Alarm3SetPoint	4X:F12839	
Alarm3ResetPoint	Real	RW	1	MERIDIAN:1:Alarm3ResetPoint	4X:F4649	xx.x (default to a 45% of FS)
			2	MERIDIAN:2:Alarm3ResetPoint	4X:F8745	
			3	MERIDIAN:3:Alarm3ResetPoint	4X:F12841	
NegativeFaultSetpoint	Real	RW	1	MERIDIAN:1:NegativeFaultSetpoint	4X:F4651	x.x% (5% is max)
			2	MERIDIAN:2:NegativeFaultSetpoint	4X:F8747	
			3	MERIDIAN:3:NegativeFaultSetpoint	4X:F12843	
Deadband	Real	RW	1	MERIDIAN:1:Deadband	4X:F4653	x.x% (5% is max)
			2	MERIDIAN:2:Deadband	4X:F8749	
			3	MERIDIAN:3:Deadband	4X:F12845	
GasNameLine1	String	RW	1	MERIDIAN:1:GasNameLine1	4X:S4753:10	ASCII gas name
			2	MERIDIAN:2:GasNameLine1	4X:S8849:10	
			3	MERIDIAN:3:GasNameLine1	4X:S12945:10	
GasNameLine2	String	RW	1	MERIDIAN:1:GasNameLine2	4X:S4758:10	ASCII gas name
			2	MERIDIAN:2:GasNameLine2	4X:S8854:10	
			3	MERIDIAN:3:GasNameLine2	4X:S12950:10	
SensorName	String	RW	1	MERIDIAN:1:SensorName	4X:S4763:18	User defined name/location
			2	MERIDIAN:2:SensorName	4X:S8859:18	
			3	MERIDIAN:3:SensorName	4X:S12955:18	
SensorModelNumber	String	R	1	MERIDIAN:1:SensorModeNumber	4X:S4772:10	ASCII date, 2 bytes per register
			2	MERIDIAN:2:SensorModeNumber	4X:S8868:10	
			3	MERIDIAN:3:SensorModeNumber	4X:S12964:10	
SensorSerialNumber	String	R	1	MERIDIAN:1:SensorSerialNumber	4X:S4777:16	sn example 115Ayywwnnnnnn 115 is the company (Monroe) A is assembly (s is subassy) 08 is the year of manufacture 16 is the week of manufacture xxxxxx is the sequential number produced within the week.
			2	MERIDIAN:2:SensorSerialNumber	4X:S8873:16	
			3	MERIDIAN:3:SensorSerialNumber	4X:S12969:16	
SensorCodeVersion	String	R	1	MERIDIAN:1:SensorCodeVersion	4X:S4785:6	4 char string, ASCII

Meridian Property Name	Data Type	Access	Sensor	Address		Remarks
				MERIDIAN	MODBUS	
			2	MERIDIAN:2:SensorCodeVersion	4X:S8881:6	
			3	MERIDIAN:3:SensorCodeVersion	4X:S12977:6	
SensorProperties	String	R	1	MERIDIAN:1:SensorProperties	4X:S4788:10	8 char string, ASCII
			2	MERIDIAN:2:SensorProperties	4X:S8884:10	
			3	MERIDIAN:3:SensorProperties	4X:S12980:10	
Zero	Trigger	W	1	MERIDIAN:1:Zero	N.A.	Zero when the respective tag changes of value.
			2	MERIDIAN:2:Zero	N.A.	
			3	MERIDIAN:3:Zero	N.A.	
Span	Trigger	W	1	MERIDIAN:1:Span	N.A.	Span when the respective tag changes of value.
			2	MERIDIAN:2:Span	N.A.	
			3	MERIDIAN:3:Span	N.A.	
Inhibit	Boolean	W	1	MERIDIAN:1:Inhibit	N.A.	Inhibit (System and Sensor) 1=Inhibit, 0=Not inhibit Inhibit timer set to 5 Mins
			2	MERIDIAN:2:Inhibit	N.A.	
			3	MERIDIAN:3:Inhibit	N.A.	

Note:
 The Modbus address offset base is 40001. Consider subtracting one of the Modbus address when communicating directly with the Modbus addresses.

Conformance Testing

The following hardware/software was used for conformance testing:

- **Configuration:**
 - **MERIDIAN Gas Detector Port:** Serial RS485 (with external adapter to RS232)
 - **Protocol:** Proprietary variation of the Modbus protocol
 - **Computer COM Port:** COM1
 - **Baud Rate:** 19200
 - **Data Bits:** 8
 - **Stop Bits:** 1
 - **Parity:** None

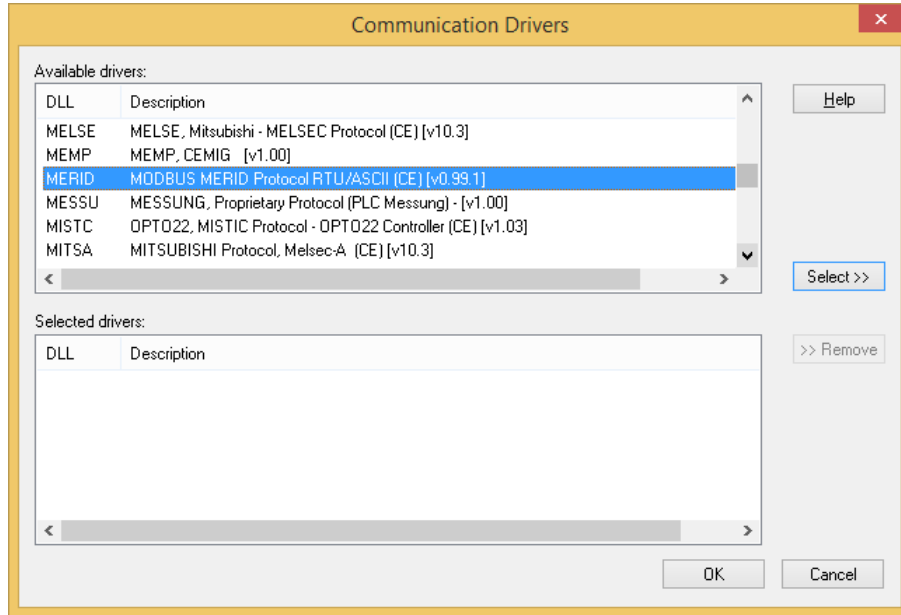
- **Cable:** RS232 with converter to RS485

Driver Version	Studio Version	Operating System (development)	Operating System (target)	Equipment
1.4	7.1 + SP3	Windows 8	Windows 7 Windows CE 6.0	MERIDIAN Universal Gas Detector

Selecting the Driver

When you install Studio, all of the communication drivers are automatically installed in the `\DRV` subdirectory but they remain dormant until manually selected for specific applications. To select the MERID driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.
2. Select the **MERID** driver from the *Available Drivers* list, and then click the **Select** button.



Communication Drivers Dialog

3. When the **MERID** driver is displayed in the **Selected Drivers** list, click the **OK** button to close the dialog. The driver is added to the *Drivers* folder, in the *Comm* tab of the *Workspace*.

Attention:

For safety reasons, you must take special precautions when installing any physical hardware. Please consult the manufacturer's documentation for specific instructions.

Configuring the Device

The MERIDIAN Universal Gas Detector supports different communication settings. Consult the device’s manual for detailed instructions on how to configure its serial communication settings through its built-in display interface.

Configuring the Driver

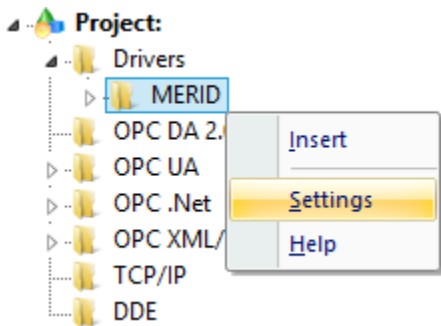
Once you have selected the MERID driver in Studio, you must properly configure it to communicate with your target device. First, you must set the driver’s communication settings to match the parameters set on the device. Then, you must build driver worksheets to associate database tags in your Studio application with the appropriate addresses (registers) on the device.

Configuring the Communication Settings

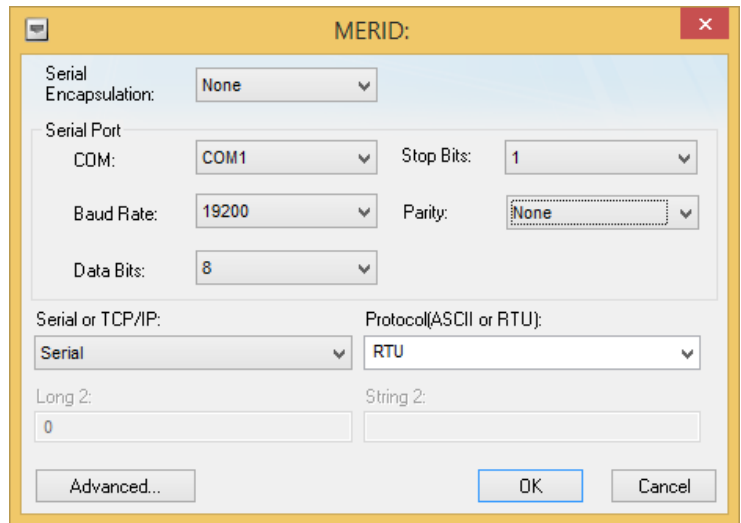
The communication settings are described in detail in the “Communication” chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only MERID driver-specific settings and procedures will be discussed here. To configure the communication settings for the MERID driver:

1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The MERID driver is listed here as a subfolder.
2. Right-click on the *MERID* subfolder and then select the **Settings** option from the pop-up menu. The *MERID: Communication Parameters* dialog is displayed:



Select Settings from the Pop-Up Menu



MERID: Communication Parameters Dialog

3. Verify the *Serial Port* settings, and change them if necessary.

4. Configure the additional driver-specific settings, as described in the following table:

Setting	Default Value	Valid Values	Description
Serial or TCP/IP	Serial	Serial	The driver implements a custom protocol based on the standard Modbus Serial protocol to communicate with the device.
		TCP/IP	The driver implements a custom protocol based on the standard Modbus TCP protocol to communicate with the device.
Protocol (ASCII or RTU)	RTU	RTU	Each eight-bit Word is sent as two four-bit hexadecimal, allowing for greater density and faster throughput. NOTE: In most cases, we recommend using this protocol.
		ASCII	Each eight-bit Word is sent as two four-bit ASCII characters, allowing for a time interval between characters without causing errors.

- Consult the Technical Reference Manual later for more information about configuring the *Advanced settings*, especially if you are using a Data Communication Equipment (DCE) converter (e.g., 232/485) between your PC and your target device
- Click **OK** to close the *Advanced Settings* dialog, and then click **OK** to close the *Communication Settings* dialog.

Configuring the Driver Worksheets

Each selected driver includes a Main Driver Sheet and one or more Standard Driver Worksheets. The Main Driver Sheet is used to define tag/register associations and driver parameters that are in effect at all times, regardless of application behavior. In contrast, Standard Driver Worksheets can be inserted to define additional tag/register associations that are triggered by specific application behaviors.

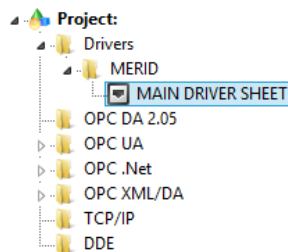
The configuration of these worksheets is described in detail in the “Communication” chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only MERID driver-specific parameters and procedures are discussed here.

MAIN DRIVER SHEET

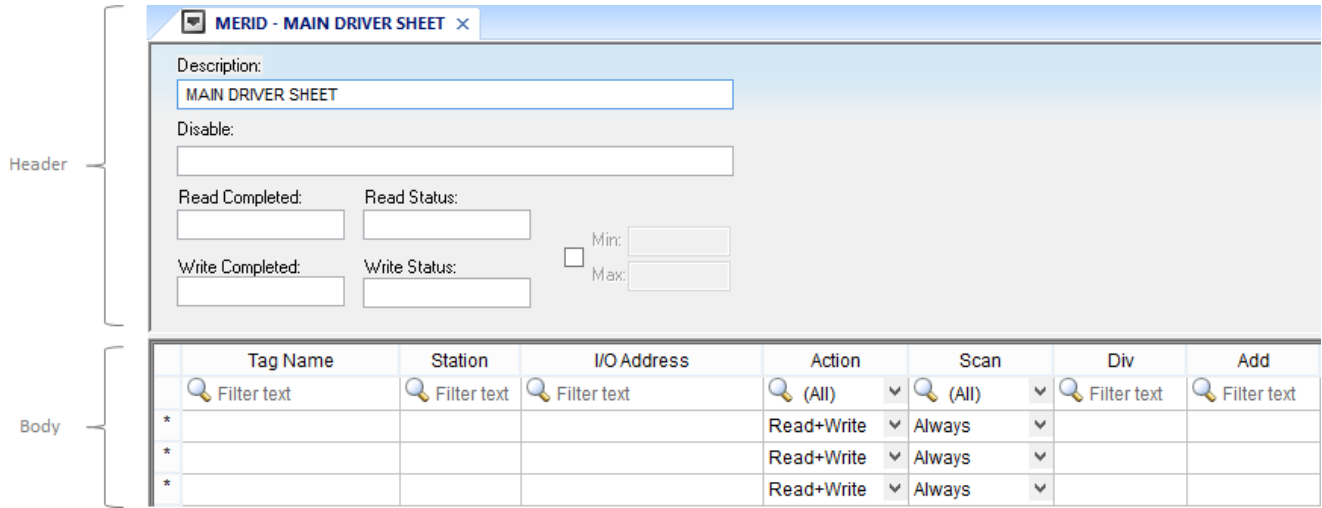
When you select the MERID driver and add it to your application, Studio automatically inserts the *Main Driver Sheet* in the *MERID* driver subfolder. To configure the Main Driver Sheet:

- Select the *Comm* tab in the *Workspace* pane.
- Open the *Drivers* folder, and then open the *MERID* subfolder:



Main Driver Sheet in the MERID Subfolder

- Double-click on the **MAIN DRIVER SHEET** icon to open the following worksheet:



Main Driver Sheet

Most of the fields on this sheet are standard for all drivers; see the “Communication” chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station** and **I/O Address** fields use syntax that is specific to the MERID driver.

- For each table row (i.e., each tag/register association), configure the **Station** and **I/O Address** fields as follows:

- STATION**

The value of the Station field depends on the Communication Settings “Serial or TCP/IP” as follows:

Communication Settings	Station		Remarks
	Syntax	Example	
Serial or TCP/IP			
Serial	<ModbusSlaveID>	1	Modbus Slave ID of the device (1..247)
TCP/IP	<IP>:<TCP Port>:<ModbusSlaveID>	192.168.1.1:502:1	<IP>: IP Address of the device <TCP Port>: TCP Port of the device (typically 502 for Modbus TCP) <ModbusSlaveID>: Modbus Slave ID of the device (1..247)

- I/O ADDRESS**

Specify the address of the associated device register.

You can either use the STANDARD MODBUS REGISTERS syntax or the MERIDIAN PROPERTY NAME syntax to read values from the device. You must use the MERIDIAN PROPERTY NAME syntax to write values to the device. In other words, the STANDARD MODBUS REGISTERS is not supported to write values to the device.

Notice that even using the MERIDIAN PROPERTY NAME syntax, some properties are Read Only.

Syntax Type	I/O Address		Remarks
	Syntax	Example	
STANDARD MODBUS REGISTERS	<RegisterType>:[<DataType>]<Address>[.<Bit>]	4X:B2	<p><RegisterType>: The device supports only the Data Type 4x, which means Holding Register.</p> <p><DataType>: Optional parameter to specify the Data Type. See list of all possible Data Types supported by the driver in the table STANDARD MODBUS REGISTERS (Read Only) from the General Information chapter.</p> <p><Address>: Modbus numerical address for the respective Register Type.</p> <p><Bit>: Optional parameter with the bit number (0..32)</p>
MERIDIAN PROPERTY NAME (TRANSMITTER)	MERIDIAN:0:<PropertyName>	MERIDIAN:0:TXTemperature	<p><PropertyName>: MERIDIAN Property Name. See table MERIDIAN PROPERTY NAME (TRANSMITTER) from the General Information chapter for a complete list of supported Property Names for the transmitter.</p>
MERIDIAN PROPERTY NAME (SENSORS)	MERIDIAN:<Sensor>:<PropertyName>	MERIDIAN:1:Alarm1Status	<p><Sensor>: Sensor Number (1, 2, or 3).</p> <p><PropertyName>: MERIDIAN Property Name. See table MERIDIAN PROPERTY NAME (SENSORS) from the General Information chapter for a complete list of supported Property Names for the sensors.</p>

Refer to the General Information chapter for a complete list of addresses supported by the MERIDIAN Universal Gas Detector.

Since the MERIDIAN Universal Gas Detector supports different block sizes for reading commands depending on the type of properties, the MAIN DRIVER SHEET implements the reading command of each Modbus address (or its respective Meridian Property Name) individually. Therefore, in order to optimize the reading performance, consider using the STANDARD DRIVER SHEET to read data from the device.

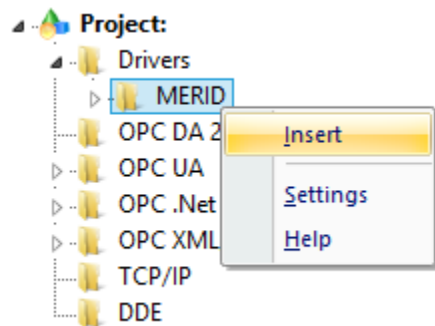
STANDARD DRIVER WORKSHEET

When you select the MERID driver and add it to your application, it has only a Main Driver Sheet by default (see previous section). However, you may insert additional Standard Driver Worksheets to define tag/register associations that are triggered by specific application behaviors. Doing this will optimize communication and improve system performance by ensuring that tags/registers are scanned only when necessary – that is, only when the application is performing an action that requires reading or writing to those specific tags/registers.

Note:
We recommend configuring device registers in sequential blocks in order to maximize performance.

To insert a new Standard Driver Worksheet:

1. In the *Comm* tab, open the *Drivers* folder and locate the *MERID* subfolder.
2. Right-click on the *MERID* subfolder, and then select **Insert** from the pop-up menu:



Inserting a New Worksheet

A new MERID driver worksheet is inserted into the *MERID* subfolder, and the worksheet is opened for configuration:

Standard Driver Worksheet

Most of the fields on this worksheet are standard for all drivers; see the “Communication” chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station**, **Header**, and **Address** fields use syntax that is specific to the MERID driver.

The **Write Trigger** command is NOT supported by the MERID driver.

3. Configure the **Station** and **Header** fields as follows:

- **STATION**

The value of the Station field depends on the Communication Settings “Serial or TCP/IP” as follows:

Communication Settings	Station		Remarks
	Syntax	Example	
Serial or TCP/IP			
Serial	<ModbusSlaveID>	1	Modbus Slave ID of the device (1..247)
TCP/IP	<IP>:<TCP Port>:<ModbusSlaveID>	192.168.1.1:502:1	<IP>: IP Address of the device <TCP Port>: TCP Port of the device (typically 502 for Modbus TCP) <ModbusSlaveID>: Modbus Slave ID of the device (1..247)

▪ **HEADER**

The Header supports the following syntaxes:

Syntax Type	Header		Remarks
	Syntax	Example	
STANDARD MODBUS REGISTERS	<RegisterType>:<AddressReference>	4X:10	<RegisterType>: The device supports only the Data Type 4x, which means Holding Register. <AddressReference>: Modbus numerical address for the respective Register Type, used as an offset for the addresses configured in the IO Address columns of the driver worksheet's body.
MERIDIAN PROPERTY NAME	MERIDIAN	MERIDIAN	You just need to write the text MERIDIAN in the Header of the worksheet to configure the I/O Address using the MERIDIAN PROPERTY NAME syntax.

▪ **I/O ADDRESS**

Specify the address of the associated device register.

You can either use the STANDARD MODBUS REGISTERS syntax or the MERIDIAN PROPERTY NAME syntax to read values from the device. You must use the MERIDIAN PROPERTY NAME syntax to write values to the device. In other words, the STANDARD MODBUS REGISTERS is not supported to write values to the device.

Notice that even using the MERIDIAN PROPERTY NAME syntax, some properties are Read Only.

Syntax Type	I/O Address		Remarks
	Syntax	Example	
STANDARD MODBUS REGISTERS	[<DataType>]<Address>[.<Bit>]	1	<DataType>: Optional parameter to specify the Data Type. See list of all possible Data Types supported by the driver in the table STANDARD MODBUS REGISTERS (Read Only) from the General Information chapter. <Address>: Modbus numerical address for the respective Register Type. <Bit>: Optional parameter with the bit number (0..32)
MERIDIAN PROPERTY NAME (TRANSMITTER)	0:<PropertyName>	0:TXTemperature	<PropertyName>: MERIDIAN Property Name. See table MERIDIAN PROPERTY NAME (TRANSMITTER) from the General Information chapter for a complete list of supported Property Names for the transmitter.
MERIDIAN PROPERTY NAME (SENSORS)	<Sensor>:<PropertyName>	1:Alarm1Status	<Sensor>: Sensor Number (1, 2, or 3). <PropertyName>: MERIDIAN Property Name. See table MERIDIAN PROPERTY NAME (SENSORS) from the General Information chapter for a complete list of supported Property Names for the sensors.

Refer to the General Information chapter for a complete list of addresses supported by the MERIDIAN Universal Gas Detector.

The STANDARD DRIVER SHEET executes reading commands in blocks. In other words, all addresses configured in each worksheet is executed as a single request (block of data). The MERIDIAN Universal Gas Detector requires specific blocks of Modbus addresses (and respective Meridian Property Names), according to the following table. It is important to make sure that each STANDARD DRIVER SHEET does not mix addresses (or their respective Meridian Property Names) from different blocks, represented as rows in the following table:

Type	Block Range		Block Size (Bytes)
	From	To	
Transmitter and Sensors	4X:2	4X:2	2
	4X:117	4X:142	27
	DW:177	DW:179	6
	FP:257	FP:267	14
	ST:337:8	ST:355:6	24
	4X:513	4X:532	21
	4X:594	4X:607	15
	FP:659	FP:659	4
Sensor 1 Only	ST:753:18	ST:784:12	43
	4X:4155	4X:4155	2
	FP:4193	FP:4201	12
	4X:4262	4X:4264	4
	FP:4305	FP:4307	6
	4X:4385	4X:4390	7
	4X:4449	4X:4495	47
	DW:4545	DW:4551	10
Sensor 2 Only	FP:4625	FP:4653	32
	ST:4753:10	ST:4788:10	45
	4X:8251:0	4X:8251:0	2
	FP:8289:0	FP:8297:0	12
	4X:8358:0	4X:8360:0	4
	FP:8401:0	FP:8403:0	6
	4X:8481:0	4X:8486:0	7
	4X:8545:0	4X:8591:0	47
Sensor 3 Only	DW:8641:0	DW:8647:0	10
	FP:8721:0	FP:8749:0	32
	ST:8849:10	ST:8884:10	45
	4X:12347:0	4X:12347:0	2
	FP:12385:0	FP:12393:0	12
	4X:12454:0	4X:12456:0	4
	FP:12497:0	FP:12499:0	6
	4X:12577:0	4X:12582:0	7
Sensor 3 Only	4X:12641:0	4X:12687:0	47
	DW:12737:0	DW:12743:0	10
	FP:12817:0	FP:12845:0	32
	ST:12945:10	ST:12980:10	45

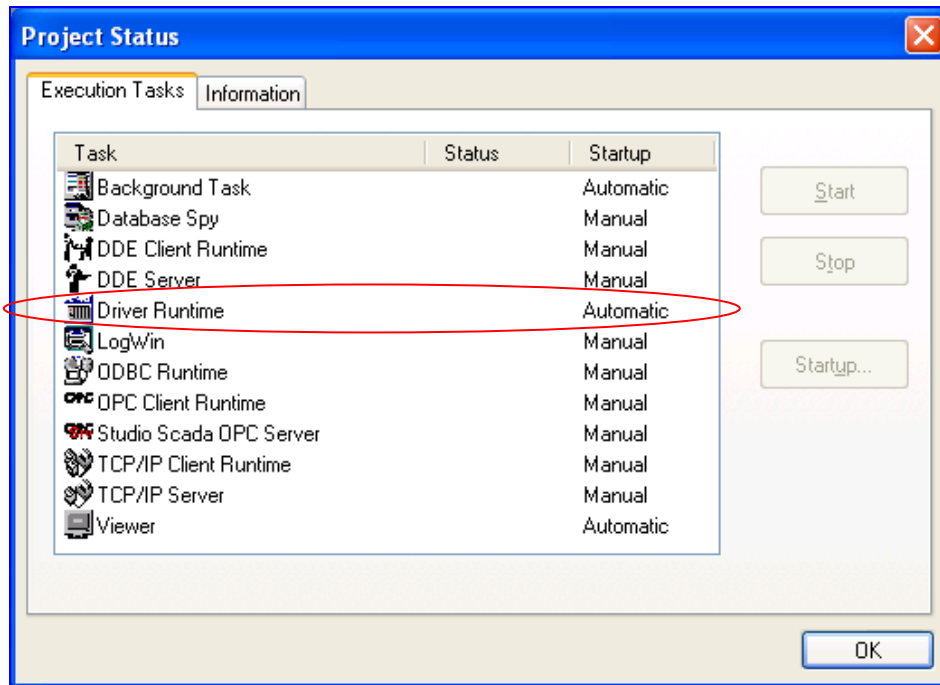
If you configure addresses (or their respective Meridian Property Names) from more than block (row from the previous table), the device will return an error indicating illegal data type or invalid block size.

Executing the Driver

By default, Studio will automatically execute your selected communication driver(s) during application runtime. However, you may verify your application's runtime execution settings by checking the *Project Status* dialog.

To verify that the communication driver(s) will execute correctly:

1. From the main menu bar, select **Project** → **Status**. The *Project Status* dialog displays:



Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.
 - If the setting is correct, then proceed to step 3 below.
 - If the **Driver Runtime** task is set to **Manual**, then select the task and click the **Startup** button to toggle the task's *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

If the MERID driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** or **Write Status** fields of the Main Driver Sheet and Standard Driver Sheet will receive an error code. Use this error code and the following table to identify what kind of failure occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None required
2	Illegal Data Value	Try to read an address that is not present in the PLC, such as 49999	Type a valid address.
3	Invalid data values	Specified address does not exist on the device so protocol received invalid data	Verify that specified address exists on the device.
4	Equipment failed	Equipment failed or out of order	Check equipment state.
5	Ack	The PLC may be busy and sent this Ack to acknowledge that it received the message but is unable to respond at the time	Wait until the PLC is available again and restart communicating with it
6	Equipment in use	Command invalid when equipment is in use	Studio commands cannot generate this error.
7	Negative Ack	Ack action error during communication	Check device and Studio Communication Parameters. Make sure that the property you are trying to update is not read only.
8	Memory parity error	Invalid Communication Parameter	Check driver Communication Parameters.
10	Invalid Header field	Specified invalid tag value in Header field	Specify a valid Header tag value.
11	Invalid Address field	Specified invalid Address	Specify a valid address.
12	Invalid block size	Offset greater than maximum allowed	Specify a valid offset or create a new worksheet. Typically, maximum offset is 64.
13	Invalid CRC	Invalid CRC in response message	<ul style="list-style-type: none"> ▪ Check the cable wiring. ▪ Check the station number. ▪ Check the RTS/CTS configuration (see <i>Studio Technical Reference Manual</i> for valid configurations).
18	Invalid BCD Value	Tried reading an invalid BCD value	Verify that PLC value is valid.
19	Invalid BCD Value	Tried writing a negative BCD value	Only positive BCD values are valid.
100	Illegal Operation	Tried to write to Read Only addresses (1x and 3x)	Writing operations are possible only in Coil Status and Holding Registers areas
102	Invalid Tag	Invalid tag	Check the possible tag names on the documentation
104	String too long	String length is greater than the maximum supported for the specific Meridian String Tag	Check the size of the string
105	Error writing TXCommand	Communication problems	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the RTS/CTS configuration (see <i>Studio Technical Reference Manual</i> for valid configurations). Check the maximum registers configuration (see <i>Studio Technical Reference Manual</i> for valid configurations).
106	Meridian Write Group not Supported	The driver does not support writing many Meridian tags using write trigger	<ul style="list-style-type: none"> ▪ Enable 'Write on Tag Change'
107	Write only Tags	Tag is write only	<ul style="list-style-type: none"> ▪ Disable the read operations for this Tag

-15	Timeout Start Message	<ul style="list-style-type: none"> ▪ Disconnected cables ▪ PLC is turned off, in stop mode, or in error mode ▪ Wrong station number ▪ Wrong RTS/CTS control settings 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the RTS/CTS configuration (see <i>Studio Technical Reference Manual</i> for valid configurations). ▪ Check the maximum registers configuration (see <i>Studio Technical Reference Manual</i> for valid configurations).
-17	Timeout between rx char	<ul style="list-style-type: none"> ▪ PLC in stop mode or in error mode ▪ Wrong station number ▪ Wrong parity ▪ Wrong RTS/CTS configuration settings 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the RTS/CTS configuration (see “Network Specifications” for valid RTS/CTS configurations). ▪ Check the maximum registers configuration (see <i>Studio Technical Reference Manual</i> for valid configurations).
20	Read out of sync	<ul style="list-style-type: none"> ▪ Timeout value 	<ul style="list-style-type: none"> ▪ Increase timeout value

⇒ **Tip:**
 You can monitor communication status by establishing an event log in Studio’s *Output* window (*LogWin* module). To establish a log for **Field Read Commands**, **Field Write Commands** and **Serial Communication**, right-click in the *Output* window and select the desired options from the pop-up menu.

Sample Application

A sample application that employs the MERID driver may be provided on the Studio installation CD. We strongly recommend that you use this sample application to test the driver *before* you develop your own applications, for the following reasons:

- To better understand the information and instructions provided in this document;
- To verify that your driver configuration is working satisfactorily with the target device; and
- To ensure that the all of hardware used in the test (i.e. the device, adapter, cable, and PC) is functioning safely and correctly.

 **Note:**

The following instructions assume that you are familiar with developing project applications in Studio. If you are not, then please review the relevant chapters of the Studio *Technical Reference Manual* before proceeding.

To use the sample application:

1. Configure the device's communication settings according to the manufacturer's documentation.
2. Run Studio.
3. From the main menu bar, select **File → Open Project**.
4. Insert the Studio installation CD and browse it to find the sample application. It should be located in the directory `\COMMUNICATION EXAMPLES\MERID`.
5. Select and open the sample application.
6. Configure and test the driver, as described in the rest of this document.

When you have thoroughly tested the driver with your target device, you may proceed with developing your own Studio application projects.

 **Tip:**

You can use the sample application screen as the maintenance screen for your own applications.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	1.00	Fabio Terezinho	Jun/5/2014	<ul style="list-style-type: none"> ▪ First driver version
B	1.1	Paulo Balbino	Jun/27/2014	<ul style="list-style-type: none"> ▪ Solved problem writing TXCommand registers
C	1.2	Paulo Balbino	Jul/3/2014	<ul style="list-style-type: none"> ▪ Solved issue writing the Meridian Property TXLongitude ▪ Included the driver files compiled for Windows Embedded Compact (Windows CE).
D	1.3	Paulo Balbino	Jul/7/2014	<ul style="list-style-type: none"> ▪ Implemented Byte Swap for the Meridian properties with String data type ▪ Solved issue writing the Meridian Property Span to sensor 3 ▪ Added the new Meridian property commands Enter and Esc ▪ Updated the access for the Meridian properties SensorOffline and CalGasConc to read only.
E	1.4	Felipe Andrade	Sep/09/2014	<ul style="list-style-type: none"> ▪ Removed SensorEnabled and SensorOffline ▪ Added property SensorsEnabledAndOffline ▪ Made the property CalGasConc writable.