

# TSI Modbus Register Map v2.0

**Revision A**

**September 3, 2010**

# Data Types and Algorithms

## Supported Modbus Functions Codes

TSI devices that conform to this specification respond to Modbus Function Code 3, Read Multiple Registers and Function Code 6, Write Single Register.

### Byte Ordering

Modbus uses big endian byte ordering (most significant byte first).

For computer systems that use "little endian" byte ordering (such as Windows and Windows CE), firmware and remote client software will have to swap the order of the bytes written to or read from the 16-bit Modbus registers.

**Every 16-bit value written to or read from the Modbus registers of this map should have the bytes swapped regardless of the ultimate data type represented (for Windows, Windows CE and other little-endian processors).**

## Data Types

### UINT16

16 bit value, unsigned, in the range 0 to 65535. Unless otherwise noted, single register data values use this type.

### INT16

16 bit signed integer value in the range of -32768 to +32767. See notes for UINT16 regarding byte ordering. This type is not currently used in the Register Map.

### UINT32

32 bit value, unsigned, in the range 0 to 4,294,967,295. These values will be split between 2, 16-bit registers that will be labeled, "High" and "Low" to designate the word ordering. For little-endian systems, the Most Significant word (16-bits) come first in memory (has the lowest memory address).

### INT32

32 bit value, signed in the range from -2,147,483,648 to 2,147,483,647. These values will be split between 2 16 bit registers that will be labeled, "High" and "Low" to designate the word ordering. For little-endian systems, the Most Significant word (16-bits) come first in memory (has the lowest memory address). This type is not currently used in the Register Map.

### FLOAT32

32 bit single precision floating point value in the IEEE 754 format (this is the standard format used for the "float" type in C, C++ and C# on the PC).

### ASCII Characters

Text strings read from or written to a compliant device may use 8-bit ASCII characters. Currently, the read-only ModelNumber and SerialNumber register sets return ASCII characters. For Recipe and Location labels, the device can use either ASCII or Unicode (UTF-16) characters, and must indicate in the DeviceFeatures\_1 Register which is supported by the device. Since each character is only 1 byte, each register read/write will contain 2 characters (byte-swapped). a NULL (0) character indicates the end of a text string, but is not required for strings that use all register bytes (see specific register descriptions). Applications should be ready to process strings that are not NULL terminated appropriately.

## UTF-16 Unicode Characters

Unicode is supported through UTF-16 encoding (as supported under Windows and Windows CE). This encoding allows for complete international character support.

## Algorithms

### Changing Device Configuration

The device configuration is affected by the registers in the Configuration Group. Some of the settings affect the instruments sampling parameters, others affect communication settings.

When changing values that are marked "In Recipe", the command code `CMD_SET_SAMPLING_CONFIG` should be written to finalize the configuration change. Any number of registers can be written before issuing the finalizing command. Note that not all devices will require this command code to finalize configuration changes, but issuing the command will ensure compatibility with all compliant instruments.

When changing TCP/IP configuration parameters, finalize the changes by writing the command code `CMD_SET_TCP_CONFIG` to the Command Register (for RS-485 devices, this would only apply to the Modbus Unit setting).

Configuration registers that are not marked "In Recipe" and that are not part of the TCP/IP configuration block do not need to be finalized with a command code. Changes to those register values will take effect immediately.

### Reading Sample Records from the Device

#### Sample Data Registers

The sample data register group starts at address 42001 and the actual sample record data is read from register addresses 42005 - 42037. Through this set of registers you can read any sample record recorded in the device buffer or the live readings from the device when it is sampling. The `DataRecordSelect` registers (410078/79) are used to address which data record to retrieve. The information below explains how to retrieve the sample records from the device.

All devices conforming to this specification should return the entire sample record data register set even if all locations are not returning meaningful data, such as unsupported extra channels or measurements. The various registers that report the number of channels support, measurements supported, etc. will inform the software of the validity of any specific reading, while still allowing the retrieval of the entire sample data record with a single read operation.

#### SampleRecordIndexHigh/Low Register Set (42003, 42004)

This read-only register set returns a 1's based, 32-bit index of the last recorded sample (first sample is 1, not 0). That index will increment each time a sample record is committed to the circular data buffer. When it reaches its maximum value (the last slot in the circular buffer), it circulates back to one. If the buffer is empty, a 0 is returned. Remote Modbus clients should monitor this register for change to determine if new data is available from the device.

#### SampleRecordSelectHigh/Low Register Set (41078, 41079)

Writing a 32-bit index value into the `SampleRecordSelectHigh/Low` register set will select which sample record from the device buffer is available through the Sample Data registers. You can address a specific data buffer location (1 – max. buffer address) or use one of 2 special modes that allow you to address the last saved record or the current live data.

**To read the last saved record without explicitly sending its index number**, write 0xFFFFFFFF to `SampleRecordIndex` registers, then read the data from the data registers.

**To read the "live" data as the device is sampling**, write 0 into the `SampleRecordIndex` registers and read the data from the data registers. Note that devices that do not support live data streaming will simply return the last saved sample when this mode is used.

**To read an arbitrary sample record from the buffer**, write the 1's based index of the record into `SampleRecordIndex` registers and read the data from the data registers.

### **SampleRecordCount (High/Low) Registers (42001, 42002)**

The 32 bit value from these registers reports how many samples are stored in the buffer. Since the buffer is circular, it will count up to the maximum number of records the buffer can hold and then remain at that count as the index recycles. A remote client can query this value to determine the maximum valid record index.

Remote clients must account for the fact that the sample record buffer is circular, so if the buffer index has recirculated, then the sample records in the buffer will not be in the correct time order. To read the sample records back in correct order, the first read SampleRecordIndex + 1 through SampleRecordCount, then read sample records 1 through SampleRecordIndex (or read all data in buffer order and reorder the data within the client software).

### **Recipe Creation and Modification**

"Recipes" allow a user to store and recall a device configuration set. This makes it easy to switch between configuration settings for different use cases. Not all instruments support the Recipe feature and some will offer text labels for Recipes, while others may support only numbered Recipes. Further, some instruments that support text labels may use ASCII text (8-bit) or Unicode (16-bit UTF-16) for international language support.

To determine if an instrument supports the Recipe feature, check the NumberOfRecipes register (40065). If it is 0, the Recipes feature is not supported, otherwise the value read there indicates how many recipes the instrument supports.

To determine if the instrument supports text labels, check the RecipeLabelLength register (40066). If it is 0, text Recipe labels are not supported, otherwise this value indicates the number of registers used to store each Recipe label (the Recipe labels are accessed through the "Location and Recipe Group" registers 43018 - 43034).

To determine if the instrument supports Unicode UTF-16 labels, check the value read from the DeviceFeatures\_1 register (40035). If bit 13 is set, UTF-16 labels are supported and values read from the Recipe label registers should be processed as 16-bit characters, otherwise the values in the label registers should be processed as 8-bit ASCII characters.

**Note:** If you are writing a software application that saves or deletes a number of Recipes to the instrument in a tight loop, it is advisable that you begin the session by first writing the command code CMD\_DISABLE\_LOCAL\_CONTROL (12) to the CommandRegister. When all Recipes have been written, close the session by writing command code CMD\_ENABLE\_LOCAL\_CONTROL (13). If you do not do this, the Recipes might not be stored correctly in the instrument.

### **Selecting a Recipe**

1. Write a value between 1 and the maximum supported Recipe for the instrument (found in NumberOfRecipes register 40065) into the RecipeSelect register (41081). 0 can also be selected to indicate that no recipe is currently selected.
2. Check the value from the SelectedRecipeValid register (41109). If it is 1, the selected Recipe is valid, if it is 0, the Recipe has not yet been defined.

### **Saving a Recipe**

1. Select the desired Recipe index by writing a value to the RecipeSelect register (41081).
2. Write configuration values to the configuration registers (sample timing, etc.).
3. **If** device supports text labels, select the Recipe label by writing the Recipe index into the RecipeLabelSelect register (43019) and then write the characters of the Recipe label into registers RecipeLabel\_1 - RecipeLabel\_16.  
Note: Do not exceed the length specified in the RecipeLabelLength register (40066).
4. Write the command code CMD\_SET\_SAMPLING\_CONFIG (9) to CommandRegister (41001)
5. Finally, write the command code CMD\_RECIPE\_SAVE (18) to the CommandRegister.

### **Deleting a Recipe**

1. Select the desired Recipe index by writing a value to the RecipeSelect register (41081).
2. Write the command code CMD\_RECIPE\_DELETE (19) to the CommandRegister.

## Location Creation and Modification

"Location" designations allow a user to mark sample data and associate it with a specific location such as a room in a facility. To date, all TSI instrument designed for contamination monitoring have supported Locations, however some use only numbered location while more advanced models support text labeled Locations. Further, some instruments that support text labels may use ASCII text (8-bit) or Unicode (16-bit UTF-16) for international language support.

To determine how many Locations an instrument supports check the NumberOfLocations register (40063). If it is 0, the Location feature is not supported by the device, otherwise the value read there indicates how many locations the instrument supports.

To determine if the instrument supports text labels for Locations check the LocationLabelLength register (40064). If it is 0, text Location labels are not supported and Locations are referred to by an index number only, otherwise this value indicates the number of registers used to store each Location label (the Location labels are accessed through the "Location and Recipe Group" registers 43001 - 43017).

To determine if the instrument supports Unicode UTF-16 labels, check the value read from the DeviceFeatures\_1 register (40035). If bit 13 is set, UTF-16 labels are supported and values read from the Location label registers should be processed as 16-bit characters, otherwise the values in the label registers should be processed as 8-bit ASCII characters.

**Note:** If you are writing a software application that saves or deletes a number of Locations to the instrument in a tight loop, it is advisable that you begin the session by first writing the command code CMD\_DISABLE\_LOCAL\_CONTROL (12) to the CommandRegister. When all Locations have been written, close the session by writing command code CMD\_ENABLE\_LOCAL\_CONTROL (13). If you do not do this, the Locations might not be stored correctly in the instrument.

## Selecting a Location

1. Write a value between 1 and the maximum number of supported Locations for the instrument (found in NumberOfLocations register 40063) into the LocationSelect register (41080). 0 can also be selected to indicate that no Location is currently selected.
2. For text labeled Locations only: Check the value from the SelectedLocationValid register (41110). If it is 1, the selected Location is valid, if it is 0, the Location has not yet been defined.

## Saving a Location

*Note: This only applies to devices that support text labeled Locations. Numbered locations do not have to be saved and cannot be deleted.*

1. Select the desired Location index by writing a value to the LocationSelect register (41080).
2. Select the Location label by writing the Location index into the LocationLabelSelect register (43001) and then write the characters of the Location label into registers LocationLabel\_1 - LocationLabel\_16. Note: Do not exceed the length specified in the LocationLabelLength register (40064).
3. To finalize the operation, write the command code CMD\_LOCATION\_SAVE (20) to the CommandRegister.

## Deleting a Location

*Note: This only applies to devices that support text labeled Locations. Numbered locations do not have to be saved and cannot be deleted.*

1. Select the desired Location index by writing a value to the LocationSelect register (41080).
2. Write the command code CMD\_LOCATION\_DELETE (21) to the CommandRegister.

## Device Information Group

Address	Mnemonic	Direction	Description
40001	MapRevRegister	R	Register map version: Major in 100ths, minor in tens
40002	FirmwareVersion	R	Device firmware version: Major in 100ths, minor in tens
40003	ModelNumber_1	R	Counter Model name registers: 1-16 <b>ASCII</b> characters, two characters per register, NULL terminated if less than 16.
40004	ModelNumber_2	R	
40005	ModelNumber_3	R	
40006	ModelNumber_4	R	
40007	ModelNumber_5	R	
40008	ModelNumber_6	R	
40009	ModelNumber_7	R	
40010	ModelNumber_8	R	
40011	SerialNumber_1	R	Counter Serial Number Registers: 1-16 <b>ASCII</b> characters, two characters per register, NULL terminated if less than 16.
40012	SerialNumber_2	R	
40013	SerialNumber_3	R	
40014	SerialNumber_4	R	
40015	SerialNumber_5	R	
40016	SerialNumber_6	R	
40017	SerialNumber_7	R	
40018	SerialNumber_8	R	
40019	Reserved_1	R	<b>Not used in this map revision</b>
40020	Reserved_2	R	
40021	Reserved_3	R	
40022	Reserved_4	R	
40023	Reserved_5	R	
40024	MfgDateYear	R	Date of Device Manufacture
40025	MfgDateMonth	R	
40026	MfgDateDay	R	
40027	LastCalDateYear	R	Date of last Calibration.
40028	LastCalDateMonth	R	
40029	LastCalDateDay	R	
40030	CalDueDateYear	R	Date next calibration is due
40031	CalDueDateMonth	R	
40032	CalDueDateDay	R	
40033	NominalFlowRateX100	R	Counter nominal flow rate times 100. Divide this integer by 100 to get nominal device flow rate. <b>IF</b> bit 15 is 1, then flow rate is in LPM (mask out bit 15 from the value), if it is 0, then flow is given in CFM. This gives a usable value range of 0.01 through 327.67 (32767 / 100).
40034	CounterChannels	R	Number of Counter channels
40035	DeviceFeatures_1	R	Device Features Word. Flag bits indicate device form factor and features (see Notes below).

Factory cutpoints			
40036	Channel_1_Size_nm	R	
40037	Channel_2_Size_nm	R	Channel 2 Factory Particle size cutpoint in nm
40038	Channel_3_Size_nm	R	Channel 3 Factory Particle size cutpoint in nm
40039	Channel_4_Size_nm	R	Channel 4 Factory Particle size cutpoint in nm
40040	Channel_5_Size_nm	R	Channel 5 Factory Particle size cutpoint in nm
40041	Channel_6_Size_nm	R	Channel 6 Factory Particle size cutpoint in nm
40042	Channel_7_Size_nm	R	Channel 7 Factory Particle size cutpoint in nm
40043	Channel_8_Size_nm	R	Channel 8 Factory Particle size cutpoint in nm
40044	Channel_9_Size_nm	R	Channel 9 Factory Particle size cutpoint in nm
40045	Channel_10_Size_nm	R	Channel 10 Factory Particle size cutpoint in nm
40046	Channel_11_Size_nm	R	Channel 11 Factory Particle size cutpoint in nm
40047	Channel_12_Size_nm	R	Channel 12 Factory Particle size cutpoint in nm
40048	Channel_13_Size_nm	R	Channel 13 Factory Particle size cutpoint in nm
40049	Channel_14_Size_nm	R	Channel 14 Factory Particle size cutpoint in nm
40050	Channel_15_Size_nm	R	Channel 15 Factory Particle size cutpoint in nm
40051	Channel_16_Size_nm	R	Channel 16 Factory Particle size cutpoint in nm
<hr/>			
40052	DeviceFeatures_2	R	Device Feature flags 2 (reserved for future)
40053	DeviceFeatures_3	R	Device Feature flags 3
<hr/>			
40054	Ethernet_MAC_addr_1	R	Device Ethernet MAC address 1
40055	Ethernet_MAC_addr_2	R	Device Ethernet MAC address 2
40056	Ethernet_MAC_addr_3	R	Device Ethernet MAC address 3
40057	MaxRecordsHigh	R	Maximum number of data records supported, High word
40058	MaxRecordsLow	R	Maximum number of data records supported, Low word
40059	LaserOnTimeHigh	R	Device Laser-on time, High word
40060	LaserOnTimeLow	R	Device Laser-on time, Low word
40061	SupportedMeasurements	R	Bit flags indicating auxilliary measurements supported by the device (some may require external probe attachment). Register Bit Definitions 0 Temperature 1 HumidityPercent 2 Velocity 3 Flow 4 CO2 5 CO 6 Pressure 7-15 Reserved for expansion
40062	Reserved_6	R	Reserved for future
40063	NumberOfLocations	R	The number of locations that are supported. May be 0 if location selection is not supported.
40064	LocationLabelLength	R	Max. Number of characters in the label. May be 0 if text labels not supported.
40065	NumberOfRecipes	R	The number of recipes that are supported. May be 0 if recipe selection is not supported.
40066	RecipeLabelLength	R	Max. Number of characters in the label. May be 0 if text labels not supported.

---

**Notes:****DeviceFeatures\_1 bit definitions:****Bit**

0	Handheld
1	Portable
2	Reserved_1
3	Reserved_2
4	Ethernet
5	USB
6	Printer
7	DisplayBrightness
8	PrintReverse
9	VariableBins
10	Modbus
11	Reserved_3
12	Reserved_4
13	UTF16
14	RelaySupport

**Some of these bits are used internally by the device firmware but are not useful for remote host control.**

**When bit is 1, it means:**

Is a Handheld Device

Is a Portable Device

*Reserved for future*

*Reserved for future*

Ethernet TCP/IP supported (used internally)

USB Communication (used internally)

Device supports a printer (used internally)

Adjustable display brightness (used internally)

Supports reverse printing (used internally)

Device supports user variable bin cutpoint sizes

Supports Modbus (used internally)

*Reserved for future*

*Reserved for future*

If 1, Device Location and Recipe labels are Unicode UTF-16.

If 0, 8bit ASCII

If 1, Device supports configurable alarm relay closure.

**DeviceFeatures\_2 bit definitions:**

Bits 0-15: Reserved for future expansion

**DeviceFeatures\_3 Register Bit Definitions:**

Bits 0-15: Reserved for future expansion



## Configuration Group: 41001-41999

Address	Mnemonic	Direction	Description	In Recipe
41001	CommandRegister	R/W	<b>Device Command Register</b> See Command Codes at the end of this section.  Device may optionally return a command status code when this register is read: 0 Command succeeded. 1 Command not recognized. 2 Command Failed. 3 A configuration value is out of range.	N
41002	StatusRegister	R	<b>Status Register</b> - Device running state: 0 Stopped: Pump is off, device is not sampling. 1 Start Delay: Device is started, but not yet sampling. 2 Holding: Device is in hold time between samples. 3 Sampling: Device is currently sampling. 32768 (0x8000) Device not ready to send valid data. <b>Note:</b> Not all instruments use all of these status codes	N
41003	BatteryStatus	R	Battery Status	N
41004	BatteryCharge	R	Battery Charge percentage	N
41005	LiveFlowStatus	R	Real-time flow status: 0 = OK, 1 = Error, 2 = stopped	N
41006	RTC_TimeYear	R/W	Real-Time-Clock year	N
41007	RTC_TimeMonth	R/W	Real-Time-Clock month	N
41008	RTC_TimeDay	R/W	Real-Time-Clock day	N
41009	RTC_TimeHour	R/W	Real-Time-Clock hour	N
41010	RTC_TimeMinute	R/W	Real-Time-Clock minute	N
41011	RTC_TimeSecond	R/W	Real-Time-Clock second	N
41012	StartDelayHigh	R/W	Auto-mode initial start delay seconds, high word	Y
41013	StartDelayLow	R/W	Auto-mode initial start delay seconds, low word	Y
41014	HoldTimeHigh	R/W	Auto-mode hold delay seconds, high word	Y
41015	HoldTimeLow	R/W	Auto-mode hold delay seconds, low word	Y
41016	SampleTimeHigh	R/W	Auto-mode sample time seconds, high word	Y
41017	SampleTimeLow	R/W	Auto-mode sample time seconds, low word	Y
41018	CyclesHigh	R/W	Auto-mode cycles, high word	Y
41019	CyclesLow	R/W	Auto-mode cycles, low word	Y
41020	ChannelDataEnable	R/W	Channel data enable flag bits. 16 bit word. LSB = channel 1, 1 = enabled, 0 = disabled	Y
41021	ChannelAlarmEnable	R/W	Channel alarm enable flag bits. 16 bit word. LSB = channel 1, 1 = enabled, 0 = disabled	Y
41022	AlarmThresholdMode	R/W	High Byte: 0 = differential, 1 = cumulative Low byte: 0 = count, 1 = FT3, 2 = m3  Informs the device how to interpret the Alarm Threshold values. Not supported by all devices.	Y
41023	AlarmThresholdHigh_1	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41024	AlarmThresholdLow_1	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41025	AlarmThresholdHigh_2	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41026	AlarmThresholdLow_2	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41027	AlarmThresholdHigh_3	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41028	AlarmThresholdLow_3	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41029	AlarmThresholdHigh_4	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41030	AlarmThresholdLow_4	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41031	AlarmThresholdHigh_5	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41032	AlarmThresholdLow_5	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41033	AlarmThresholdHigh_6	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41034	AlarmThresholdLow_6	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y

41035	AlarmThresholdHigh_7	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41036	AlarmThresholdLow_7	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41037	AlarmThresholdHigh_8	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41038	AlarmThresholdLow_8	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41039	AlarmThresholdHigh_9	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41040	AlarmThresholdLow_9	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41041	AlarmThresholdHigh_10	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41042	AlarmThresholdLow_10	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41043	AlarmThresholdHigh_11	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41044	AlarmThresholdLow_11	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41045	AlarmThresholdHigh_12	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41046	AlarmThresholdLow_12	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41047	AlarmThresholdHigh_13	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41048	AlarmThresholdLow_13	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41049	AlarmThresholdHigh_14	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41050	AlarmThresholdLow_14	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41051	AlarmThresholdHigh_15	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41052	AlarmThresholdLow_15	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y
41053	AlarmThresholdHigh_16	R/W	Particle alarm threshold, 32 bit unsigned int, high word	Y
41054	AlarmThresholdLow_16	R/W	Particle alarm threshold, 32 bit unsigned int, low word	Y

41055	SamplingMode	R/W	Sets the device sampling mode. 0 = Auto, 1 = Manual, 2 = Beep	Y
41056	Reserved_1	R/W		N
41057	Reserved_2	R/W		N
41058	Reserved_3	R/W		N
41059	Reserved_4	R/W		N
41060	Reserved_5	R/W		N

#### Unit settings for supported measurements.

If a device does not support changing these values, it will simply ignore a write request.

41061	TemperatureUnit	R/W	Unit code	N
41062	HumidityUnit	R/W	Unit code	N
41063	VelocityUnit	R/W	Unit code	N
41064	FlowUnit	R/W	Unit code	N
41065	CO2Unit	R/W	Unit code	N
41066	COUnit	R/W	Unit code	N
41067	Pressure	R/W	Unit code	N
41068	Measurement 8 units	R/W	Unit code (Reserved for future expansion)	N
41069	Measurement 9 units	R/W	Unit code (Reserved for future expansion)	N
41070	Measurement 10 units	R/W	Unit code (Reserved for future expansion)	N
41071	Measurement 11 units	R/W	Unit code (Reserved for future expansion)	N
41072	Measurement 12 units	R/W	Unit code (Reserved for future expansion)	N
41073	Measurement 13 units	R/W	Unit code (Reserved for future expansion)	N
41074	Measurement 14 units	R/W	Unit code (Reserved for future expansion)	N
41075	Measurement 15 units	R/W	Unit code (Reserved for future expansion)	N
41076	Measurement 16 units	R/W	Unit code (Reserved for future expansion)	N

#### Unit codes for measurements

0 = n/a, 1 = °F, 2 = °C, 3 = %, 4 = feet/min

5 = meters/second, 6 = cubic feet/minute

7 = cubic meters/second, 8 = cubic meters/hour

9 = liters/second, 10 = parts per million

11 = inH2O, 12 = PA, 13 = hPA

14 = kPA, 15 = mmHg, 16 = cmHG

17 = inHG, 18 = mmH2O, 19 = cmH2O

N

41077	OperationModeFlags	R/W	Bit 0: Unit will begin AutoMode counting on power-up. Bits 1-15: Reserved.	N
41078	SampleRecordSelectHigh	R/W	32bit value to select the sample record data that is to be made available through the Sample Data registers (42xxx group).	N

41079	SampleRecordSelectLow	R/W		N
41080	LocationSelect	R/W	Read to discover current location, write to select new	N
41081	RecipeSelect	R/W	Read to discover current recipe, write to select new	N

#### **TCP/IP Configuration Registers**

To set TCP/IP configuration, write registers, then issue command CMD\_SET\_TCP\_CONFIG.

**Note** that for most devices, TCP/IP can not be reconfigured via Modbus registers.

41082	DHCP_Enabled	R/W	Enable/Disable DHCP (1 = Enabled, 0 = Disabled)	N
41083	IP_Address_1	R/W	TCP/IP Configuration Registers	N
41084	IP_Address_2	R/W	TCP/IP Configuration Registers	N
41085	Subnet_Mask_1	R/W	TCP/IP Configuration Registers	N
41086	Subnet_Mask_2	R/W	TCP/IP Configuration Registers	N
41087	Gateway_1	R/W	TCP/IP Configuration Registers	N
41088	Gateway_2	R/W	TCP/IP Configuration Registers	N
41089	ModbusServerPort	R/W	TCP/IP Configuration Registers	N
41090	Modbus_Unit	R/W	Modbus Unit Address of device	N

#### **Programmable Cutpoint Registers**

For devices that support user selectable channel cutpoints.

41091	Channel_1_Size_nm	R/W	Channel 1 Currently active Particle size cutpoint in nm	Y
41092	Channel_2_Size_nm	R/W	Channel 2 Currently active Particle size cutpoint in nm	Y
41093	Channel_3_Size_nm	R/W	Channel 3 Currently active Particle size cutpoint in nm	Y
41094	Channel_4_Size_nm	R/W	Channel 4 Currently active Particle size cutpoint in nm	Y
41095	Channel_5_Size_nm	R/W	Channel 5 Currently active Particle size cutpoint in nm	Y
41096	Channel_6_Size_nm	R/W	Channel 6 Currently active Particle size cutpoint in nm	Y
41097	Channel_7_Size_nm	R/W	Channel 7 Currently active Particle size cutpoint in nm	Y
41098	Channel_8_Size_nm	R/W	Channel 8 Currently active Particle size cutpoint in nm	Y
41099	Channel_9_Size_nm	R/W	Channel 9 Currently active Particle size cutpoint in nm	Y
41100	Channel_10_Size_nm	R/W	Channel 10 Currently active Particle size cutpoint in nm	Y
41101	Channel_11_Size_nm	R/W	Channel 11 Currently active Particle size cutpoint in nm	Y
41102	Channel_12_Size_nm	R/W	Channel 12 Currently active Particle size cutpoint in nm	Y
41103	Channel_13_Size_nm	R/W	Channel 13 Currently active Particle size cutpoint in nm	Y
41104	Channel_14_Size_nm	R/W	Channel 14 Currently active Particle size cutpoint in nm	Y
41105	Channel_15_Size_nm	R/W	Channel 15 Currently active Particle size cutpoint in nm	Y
41106	Channel_16_Size_nm	R/W	Channel 16 Currently active Particle size cutpoint in nm	Y

#### **"Relay" contact closure configuration**

Check DeviceFeatures\_1, RelaySupport bit to verify that these configuration registers are supported.

"Relay" Contact Closure select bits.

Bit Function

0 - Optics Dirty Alert Flag

1 - Flow Alert Flag

2 - High Scatter Alert Flag

3 - 15 Open

41107	RelayEventSelect	R/W		N
41108	RelayChannelSelect	R/W	Bits indicate which of the 16 channels will cause contact closure when alarm is triggered.	N
41109	RelayChannelDelay	R/W	Number of samples to delay before closing relay.	N
41110	SelectedRecipeValid	R	When text Recipe labels are supported, a 1 in this register indicates that the selected location is valid.	N
41111	SelectedLocationValid	R	When text Location labels are supported, a 1 in this register indicates that the selected location is valid.	N

**Command Register Codes**

<b>Code</b>	<b>Mnemonic</b>	<b>Description</b>
1	CMD_CLEAR_DATA	Clear all data records
2	CMD_RESERVED_2	Reserved for future expansion
3	CMD_RESERVED_3	Reserved for future expansion
4	CMD_RESERVED_4	Reserved for future expansion
5	CMD_RESERVED_5	Reserved for future expansion
6	CMD_AUTO_START	Start automatic-mode sampling, obeying configured start delay, sample time, hold time and cycle count.
7	CMD_AUTO_STOP	Stop automatic-mode sampling
8	CMD_SET_RTC	Set the real time clock with the values written into the RTC set registers 41006-41011
9	CMD_SET_SAMPLING_CONFIG	Set automode sample timing parameters
10	CMD_RESERVED_10	Reserved for future expansion
11	CMD_RESERVED_11	Reserved for future expansion
12	CMD_DISABLE_LOCAL_CONTROL	Lock out the instrument user interface so instrument can only be configured via Modbus
13	CMD_ENABLE_LOCAL_CONTROL	Enable instrument user interface so it can be configured locally or via Modbus
14	CMD_SILENCE_DEVICE	Temporarily silences the audible alarm on the device. Setting is never stored, device power cycle turns audible alarm back on
15	CMD_UNSILENCE_DEVICE	Turn the audible alarm capability back on
16	CMD_RESERVED_16	Reserved for future expansion
17	CMD_SET_TCP_CONFIG	Set the TCP/IP configuration based on the values written to the TCP/IP Configuration Registers. Save the configuration to recipe number selected in RecipeSelect register. Also finalizes changes to Recipe text label if that feature is supported.
18	CMD_RECIPE_SAVE	Save the configuration to recipe number selected in RecipeSelect register. Also finalizes changes to Recipe text label if that feature is supported.
19	CMD_RECIPE_DELETE	Delete the recipe selected in RecipeSelect register.
20	CMD_LOCATION_SAVE	Save the Location label for the selected Location
21	CMD_LOCATION_DELETE	Delete the Location selected in LocationSelect register.
500-599	CMD_DIAGNOSTIC_1 - CMD_DIAGNOSTIC_100	<u>A Block of proprietary diagnostic/testing commands.</u>

## Sample Data Block: 42001-42999

Address	Mnemonic	Direction	Description
42001	SampleRecordCountHigh	R	Number of records in database, high word
42002	SampleRecordCountLow	R	Number of records in database, low word
42003	SampleRecordIndexHigh	R	Index of last recorded record, high word
42004	SampleRecordIndexLow	R	Index of last recorded record, high word
42005	SampleIDHigh	R	Record ID of record being read, high word
42006	SampleIDLow	R	Record ID of record being read, low word
42007	TimeStampYear	R	Record time stamp, Year
42008	TimeStampMonth	R	Record time stamp, month
42009	TimeStampDay	R	Record time stamp, day
42010	TimeStampHour	R	Record time stamp, hour
42011	TimeStampMinute	R	Record time stamp, minute
42012	TimeStampSecond	R	Record time stamp, second
42013	Reserved_1	R	
42014	Reserved_2	R	
42015	Reserved_3	R	
42016	Reserved_4	R	
42017	Reserved_5	R	
42018	DeviceStatus	R	Device Error Status flags, see below
42019	AlarmStatus	R	Channel alarm flags, bit 0 = ch. 1, bit 1 = ch. 2, etc.
42020	FlowRateX100	R	Counter flow rate multiplied by 100. Divide this integer by 100 to get device flow rate. IF bit 15 is 1, then flow rate is in LPM (mask bit 15 out of the value), if it is 0, then flow is given in CFM. This gives a usable value range of 0.01 - 327.67 (32767 / 100).
42021	SampleTimeHigh	R	Elapsed sample time, high word, units specified in SampleTimeUnits register
42022	SampleTimeLow	R	Elapsed sample time, low word, units specified in SampleTimeUnits register
42023	SampleTimeUnits	R	Units that the Elapsed Sample Time is reported in. 0 = mSec, 1 = seconds, 2 = 1/10 seconds
42024	CountMode	R	Indicates the format of the returned particle count data. High Byte: 0 = differential, 1 = cumulative Low byte: 0 = raw count, 1 = FT3, 2 = m3
42025	LocationNumber	R	The location number

---

			Particle count Registers.
			Each is a 32-bit unsigned int, High 16-bits, then low 16-bits.
42026	ParticleCountHigh_1	R	
42027	ParticleCountLow_1	R	
42028	ParticleCountHigh_2	R	
42029	ParticleCountLow_2	R	
42030	ParticleCountHigh_3	R	
42031	ParticleCountLow_3	R	
42032	ParticleCountHigh_4	R	
42033	ParticleCountLow_4	R	
42034	ParticleCountHigh_5	R	
42035	ParticleCountLow_5	R	
42036	ParticleCountHigh_6	R	
42037	ParticleCountLow_6	R	
42038	ParticleCountHigh_7	R	
42039	ParticleCountLow_7	R	
42040	ParticleCountHigh_8	R	
42041	ParticleCountLow_8	R	
42042	ParticleCountHigh_9	R	
42043	ParticleCountLow_9	R	
42044	ParticleCountHigh_10	R	
42045	ParticleCountLow_10	R	
42046	ParticleCountHigh_11	R	
42047	ParticleCountLow_11	R	
42048	ParticleCountHigh_12	R	
42049	ParticleCountLow_12	R	
42050	ParticleCountHigh_13	R	
42051	ParticleCountLow_13	R	
42052	ParticleCountHigh_14	R	
42053	ParticleCountLow_14	R	
42054	ParticleCountHigh_15	R	
42055	ParticleCountLow_15	R	
42056	ParticleCountHigh_16	R	
42057	ParticleCountLow_16	R	

---

			Channel cut point sizes for the sample. Some models have variable cutpoints and could have different sizes per sample.
42058	Channel_1_Size_nm	R	
42059	Channel_2_Size_nm	R	
42060	Channel_3_Size_nm	R	
42061	Channel_4_Size_nm	R	
42062	Channel_5_Size_nm	R	
42063	Channel_6_Size_nm	R	
42064	Channel_7_Size_nm	R	
42065	Channel_8_Size_nm	R	
42066	Channel_9_Size_nm	R	
42067	Channel_10_Size_nm	R	
42068	Channel_11_Size_nm	R	
42069	Channel_12_Size_nm	R	
42070	Channel_13_Size_nm	R	
42071	Channel_14_Size_nm	R	
42072	Channel_15_Size_nm	R	
42073	Channel_16_Size_nm	R	

---

#### Measurement block

For auxilliary measurements like temperature, humidity, etc.

42074	TemperatureUnit	R	Temperature measurement unit (0 = not present or disabled)
42075	TemperatureValueHigh	R	Value High 16 bits (float)
42076	TemperatureValueLow	R	Value low 16 bits (float)
42077	HumidityUnit	R	Humidity measurement unit (0 = not present or disabled)
42078	HumidityValueHigh	R	Value High 16 bits (float)
42079	HumidityValueLow	R	Value low 16 bits (float)
42080	VelocityUnit	R	Velocity measurement unit (0 = not present or disabled)
42081	VelocityValueHigh	R	Value High 16 bits (float)
42082	VelocityValueLow	R	Value low 16 bits (float)
42083	FlowUnit	R	Flow measurement unit (0 = not present or disabled)
42084	FlowValueHigh	R	Value High 16 bits (float)
42085	FlowValueLow	R	Value low 16 bits (float)
42086	CO2Unit	R	CO2 measurement unit (0 = not present or disabled)
42087	CO2ValueHigh	R	Value High 16 bits (float)
42088	CO2ValueLow	R	Value low 16 bits (float)
42089	COUnit	R	CO measurement unit (0 = not present or disabled)
42090	COValueHigh	R	Value High 16 bits (float)
42091	COValueLow	R	Value low 16 bits (float)
42092	PressureUnit	R	Pressure measurement unit (0 = not present or disabled)
42093	PressureValueHigh	R	Value High 16 bits (float)
42094	PressureValueLow	R	Value low 16 bits (float)
42095-42121	Reserved for future measurements	R	Future measurement types. 3 registers per measurement. 16 measurements total. Unit codes for measurements 0 = n/a, 1 = °F, 2 = °C, 3 = %, 4 = feet/min 5 = meters/second, 6 = cubic feet/minute 7 = cubic meters/second, 8 = cubic meters/hour 9 = liters/second, 10 = parts per million 11 = inH2O, 12 = PA, 13 = hPA 14 = kPA, 15 = mmHg, 16 = cmHG 17 = inHG, 18 = mmH2O, 19 = cmH2O
42122	MeasurementEnabled	R	Flags bits indicate whether measurements were enabled during sampling.

#### Device Status register bit map (RM\_DeviceStatus, 42019)

Bits	0-1	Sample Flow Status, 0 = flow OK, 1 = flow error, 2 = flow stopped
Bit	2	Laser OK = 0, Laser Error = 1
Bit	3	Laser Scatter Alert (or detector error)
Bit	4	Optics Dirty Alert
Bit	5	Calibration Corruption error (needs recalibration)
Bit	6-13	Reserved
Bit	14	Service Alert. Single bit indicator of any error.
Bits	15	Data valid = 0, Data invalid = 1. Data invalid indicates device-not-ready

## Location and Recipe Group: 43001-43999

Address	Mnemonic		Description
43001	LocationLabelSelect	R/W	Write to LocationLabelSelect to address location labels. First index is 1.
<b>Location Labels</b>			
43002	LocationLabel_1	R/W	Location label registers, up to 16 UTF-16 characters or 32 ASCII. Null terminated IF less than max.
43003	LocationLabel_2	R/W	
43004	LocationLabel_3	R/W	
43005	LocationLabel_4	R/W	
43006	LocationLabel_5	R/W	
43007	LocationLabel_6	R/W	
43008	LocationLabel_7	R/W	
43009	LocationLabel_8	R/W	
43010	LocationLabel_9	R/W	
43011	LocationLabel_10	R/W	
43012	LocationLabel_11	R/W	
43013	LocationLabel_12	R/W	
43014	LocationLabel_13	R/W	
43015	LocationLabel_14	R/W	
43016	LocationLabel_15	R/W	
43017	LocationLabel_16	R/W	
43018	RecipeLabelSelect	R/W	Write to RecipeLabelSelect to address recipe labels. First index is 1.
<b>Recipe Labels</b>			
43019	RecipeLabel_1	R/W	Recipe label registers, up to 16 UTF-16 characters or 32 ASCII. Null terminated IF less than max.
43020	RecipeLabel_2	R/W	
43021	RecipeLabel_3	R/W	
43022	RecipeLabel_4	R/W	
43023	RecipeLabel_5	R/W	
43024	RecipeLabel_6	R/W	
43025	RecipeLabel_7	R/W	
43026	RecipeLabel_8	R/W	
43027	RecipeLabel_9	R/W	
43028	RecipeLabel_10	R/W	
43029	RecipeLabel_11	R/W	
43030	RecipeLabel_12	R/W	
43031	RecipeLabel_13	R/W	
43032	RecipeLabel_14	R/W	
43033	RecipeLabel_15	R/W	
43034	RecipeLabel_16	R/W	