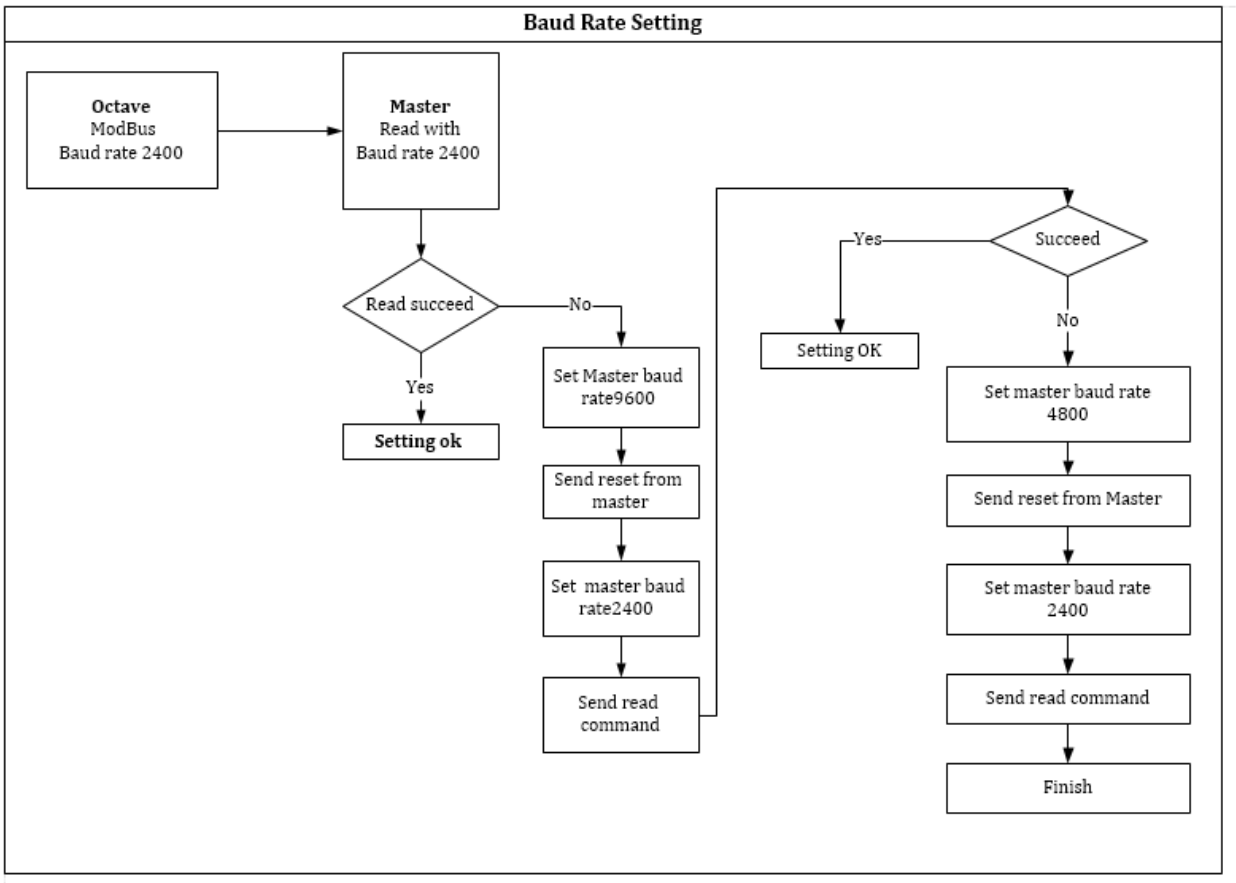
PIPERSBERG

octave

Modbus Module Format and Description

Contents



1.1 Modbus supported commands

The Module should support the following function codes for:

1.1.1 "Access data": 03, 04, 06, and 16.

				Function Codes			
				code	sub code	(hex)	Section
Data Access	16 bits access	Physical Input Registers Internal Registers Or Physical Output Registers	Read Holding Registers	03		03	6.3
			Read Input Registers	04		04	6.4
			Write Single Register	06		06	6.6
			Write Multiple Registers	16		10	6.12
Diagnostics			Diagnostic	08	0-4,10-18,19	08	6.8
			Identification	43			

03 (0x03) Read Holding Registers

This function code is used to read the contents of a contiguous block of holding registers in Modbus module.

04 (0x04) Read Input Registers

This function code is used to read from 1 to 125 contiguous input registers in a remote device.

06 (0x06) Write Single Register

This function code is used to write a single **holding** register in a remote device. The normal response is an echo of the request, returned after the register contents have been written.

16 (0x10) Write Multiple registers

This function code is used to write a block of contiguous registers (1 to 123 registers) in a remote device. The requested written values are specified in the request data field. Data is packed as two bytes per register. The normal response returns the function code, starting address, and quantity of registers written.

The Request PDU specifies the starting register address and the number of registers. In the PDU Registers are addressed starting at zero. Therefore, input registers numbered 1-16 are addressed as 0-15.
The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

1.2 Read Input Registers

1.2.1 The next registers are used by Modbus Master-Slave protocol to read values from the Modbus Module:

Address			Register Name	Type	Bit	Value
Hi	**Low**	**Dec.**				
0	0x00	0	Alarms	Bitwise Int16	- 0 1 2 3 4 5 6 7 8 9 10 11 12	0x0000 - OK 0x0001 - Leakage 0x0002 - Pipe Burst 0x0004 - Reverse Flow 0x0008 - Dry 0x0010 - Critical Configuration 0x0020 - Measurement Fail 0x0040 - Tamper 0x0080 – Octave Battery 0x0100 - Units Change 0x0200 - Watch Dog 0x0400 - Service Required 0x0800 – Flow Rate Cut Off 0x1000 – Module battery
0	0x1-0x10	1-16	AMR Serial Number	Int16	16 registers array [MSB][LSB] [0][x][0][x][0][x]...[0][x]	
0	0x11	17	RTC- week day	Int16	Meter Date and Time	1...7 - [0][x]
0	0x12	18	RTC - day	Int16		1...31 - [0][x]
0	0x13	19	RTC - month	Int16		1...12 - [0][x]
0	0x14	20	RTC - year	Int16		14...99 - [0][x]
0	0x15	21	RTC - hours	Int16		0...23 - [0][x]
0	0x16	22	RTC - minutes	Int16		0...59 - [0][x]
0	0x17	23	Volume unit	Int16	See System Units/Resolutions table	
0	0x18	24	Forward volume	double	Forward volume	
0	0x20	32	Reverse volume	double	Reverse volume	
0	0x28	40	Volume Resolution index	Int16	Value to multiply Forward/Reverse volume	
0	0x29	41	Current Flow	double	Current flow	
0	0x31	49	Flow Resolution index	Int16	Value to multiply Current flow	
0	0x32	50	Flow Units	Int16	See System Units/Resolutions table	
0	0x33	51	Flow Direction	Int16	See System Units/Resolutions table	
0	0x34	52	Temperature Value	Int16	See System Units/Resolutions table	
0	0x35	53	Temperature Unit	Int16	See System Units/Resolutions table	
0	0x36	54	Forward volume_ui32	Uint32	Forward volume (int representation)	
0	0x3E	58	Reverse volume_ui32	Uint32	Reverse volume (int representation)	
0	0x42	62	Current Flow_i32	Int32	Current flow (int representation)	
0	0x46	66	Signed volume	double	Signed volume	
0	0x4E	74	Unsigned volume	double	Unsigned volume	
0	0x56	82	Signed volume_i32	Int32	Signed volume (int representation)	
0	0x5A	86	Unsigned volume_ui32	Uint32	Unsigned volume (int representation)	

1.3 Write Input Registers

1.6.1. Current values can be sending by request from Modbus Master (remote) through RS485 communication to Modbus Module (Slave).

1.6.2. The next registers can update/set values into the Modbus Module:

Address		Register Name	Type	Value		Usage
Hi	Low					
0	0x00	Alarms	Int 16 Bitwise	0-Clear 1-System Reset		
0	0x01	Set Date and Time	6 registers of Int 16 array: Week day, day, month, year, hours, minutes	RTU	ASCII	Module will Convert
				[0][wd], [0][d], [0][m], [0][y], [0][h], [0][m]	wd -[Msb][Lsb], d-[Msb][Lsb], m-[Msb][Lsb], y-[Msb][Lsb], h-[Msb][Lsb], m-[Msb][Lsb]	
0	0x07	Set Volume Resolution index	Int16	See System Units/Resolutions table		
0	0x08	Set Volume Flow index	Int16	See System Units/Resolutions table		

1.6.3. In case of received 'System Reset' value into Alarm register (with Broadcast address), the Modbus module will reset and perform request from Octave full configuration. The reason for this requirement that we don't know current module configuration.

1. "Get Octave version" for detect Octave system version: only from version 15 support communication to Modbus module through MSI protocol.
2. "Get Module Configuration" – get full configuration of Modbus module.
3. "Get Octave measurement values".

1.6.4. All registers are presented in RTU format. For ASCII representation every byte of RTU value need to be converted to 2 bytes before response time as described in "ASCII representation" example.

1.4 System Units/Resolution

1.7.1. The next table are collects and summarizes all parameters units are used into the Modbus Module system.

Name	Value type
Serial communication Data mode	0 – RTU (**default**) 1 – ASCII
Serial communication Parity Configuration	0 – Even (**default**) 1 – Odd 2 – None
Flow Units	0 – Cubic Meters/Hour (**default**) 1 – Gallons/Minute 2 – Litres/Second 3 – Imperial Gallons/ Minute 4 – Litres/Minute 5 – Barrel/Minute
Flow Resolution	[0] = 0.0001x, [1] = 0.001x , [2] = 0.01x, [3] = 0.1x, [4] = 1x (**default**), [5] = 10x, [6] = 100x, [7] = 1000x [8] = 10000x For example: if index= 3-> Flow resolution = 0.1
Volume Units	0 – Cubic Meters (**default**) 1 – Cubic Feet 2 – Cubic Inch 3 – Cubic Yards 4 – US Gallons 5 – Imperial Gallons 6 – Acre Feet 7 – Kilolitres 8 – Litres 9 – Acre-Inch 10 – Barrel
Volume Resolution	[0] = 0.0001x, [1] = 0.001x , [2] = 0.01x, [3] = 0.1x, [4] = 1x (**default**), [5] = 10x, [6] = 100x, [7] = 1000x [8] = 10000x For example: if index= 3-> Volume resolution = 0.1
Flow direction	0 - No flow 1 - Forward flow 2 - Backward flow
Temperature Units	0 - Not Active 1 - Celsius 2 - Fahrenheit