AN122 – EC100 Digital Communication Overview

The EC100 sensor supports both I2C and MODBUS communication standards. These protocols are designed to be robust and easy to implement, both in terms of physical hardware, requiring only two I/O lines each, and in terms of protocol implementation, using well-documented, and widely supported standards.

The advantages of using a digital protocol over analog voltage readings include bidirectional communication support, allowing modification of sensor operating parameters, an expanded set of sensor readings available (including proportional temperature data), and increased resolution in the measured value.

The purpose of this document is to demonstrate the basic communication procedures behind both of these protocols to enable rapid development of integrated applications.

I2C Communication

Hardware

The EC100 sensor's I2C interface is internally pulled up to 3.3V with weak pull-up resistors. These will be adequate for typical board-level communication bus distances. If the I2C bus will stretch over a distance greater than 12 inches additional pull-up resistors should be attached, to provide suitable transient response on clock and data edges. The I2C bus operates at the standard rate of 100Khz.

I2C is a pull-down protocol, so without additional pull-up resistors the EC100 is compliant with any system that detects a high level as being at 3.3V or higher, however if external pull-up resistors are attached the line voltage should not exceed 5V, doing so can cause permanent damage to the sensor.

Protocol

Timing is compliant with the standard I2C implementation. For a reference implementation guide please visit i2c-bus.org. The following byte streams demonstrate a standard I2C transaction with the sensor:

Communication Sequence:

(start)	0x30	0x40	(restart)	0x31	0x41	(val byte 1)	(val byte 2)	(filter val)	(stop)

Where (start), (restart), and (stop) are the conditions as specified in the implementation guide. This command reads the current sensor value. The value is represented as a 2-byte signed integer using fixed decimal-point storage. Scaling for sensors is shown below:

Oxygen	0.001 (Units: %)
Carbon Monoxide	0.1 (Units: ppm)
Ethylene	0.01 (Units: ppm)
Ethylene Oxide	0.1 (Units: ppm)

Scaling Coefficients and Units for Electrochemical Sensors

Addressing

The EC100 sensor's default I2C address is 0x30.

Function List

The following I2C functions are available for use.

Function	Direction	Format	Description

(0x10) – Write Command Byte	Write-Only	TO: 0x10 {VAL}
		FROM: 0x11
			Writes {VAL} to command byte for calibrations

(0x20) – Read EEPROM	Read-Only	TO: 0x20 {OFFSET} {NUM}
		FROM: 0x21 {DATA_N}
			Reads {NUM} bytes from {OFFSET} in EEPROM

(0x30) – Write EEPROM	Write-Only	TO: 0x30 {OFFSET} {NUM} {DATA_N}
		FROM: 0x31
			Writes {NUM} bytes to {OFFSET} in EEPROM.

(0x40) – Read Sensor Value	Read-Only	TO: 0x40 FROM: 0x41 {SENSOR VAL} {FILTER VAL}	Reads the current sensor value.

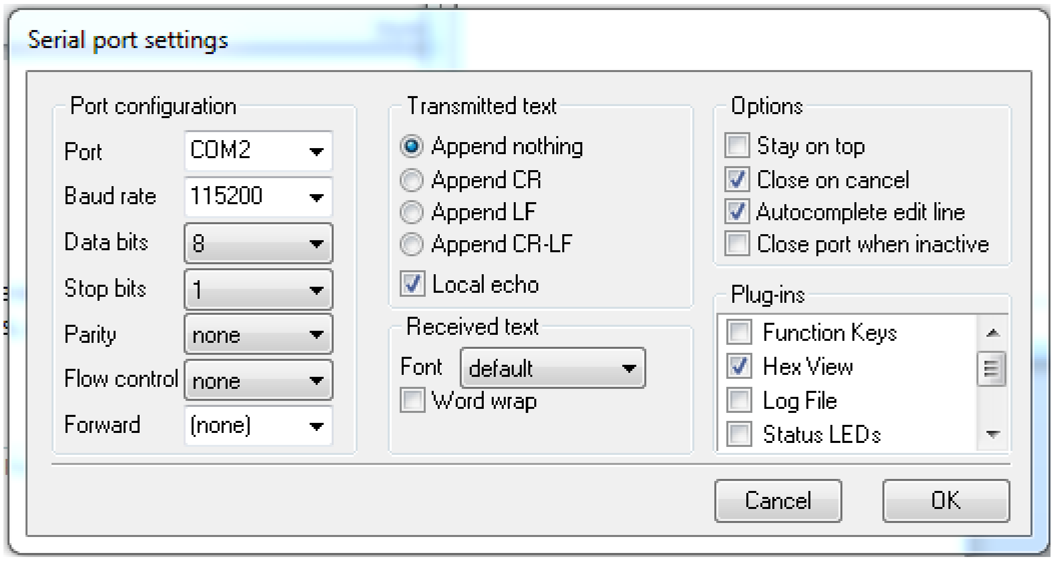
MODBUS Communication

Hardware

Application note AN117 documents in detail the hardware setup required for a MODBUS session. For testing purposes we recommend purchasing one of our development kit cables, and downloading the free Termite serial program, available here:

http://www.compuphase.com/software_termite.htm

Configure the program as shown below for direct sending of hexadecimal values, selecting the correct COM port as found in device manager.



Serial port settings

Port configuration
Port: COM2
Baud rate: 115200
Data bits: 8
Stop bits: 1
Parity: none
Flow control: none
Forward: (none)

Transmitted text
● Append nothing
○ Append CR
○ Append LF
○ Append CR-LF
☑ Local echo

Received text
Font: default
□ Word wrap

Options
□ Stay on top
☑ Close on cancel
☑ Autocomplete edit line
□ Close port when inactive

Plug-ins
□ Function Keys
☑ Hex View
□ Log File
□ Status LEDs

After configuring the program hexadecimal values can be sent in standard C hex notation, separating each byte with a space.

Protocol

The supported MODBUS command syntax is detailed in AN117, the supported commands are summarized below for convenience. For full implementation details please see the application note, including checksum calculating, byte fields, and return codes.

Command	Byte Code
Read Holding Register	0x03
Write Holding Register	0x06

Read Input Register	0x04

The EC100 has the following registers available for reading and writing:

Input Registers

Address: **0x01**	Sensor Value (UInt16_t)
Address: **0x02**	Temperature Value (UInt16_t)

Holding Registers

Address: **0x01**	Command Register (UInt16_t)

Command Status Register Bit Functions
The command register can be used to perform calibrations. The two least significant bits are used to cause calibration.

Bit 0:	Span calibration (span value dependent on sensor type)
Bit 1:	Zero calibration (recommended nitrogen or other nonreactive gas)

An example command sequence is presented below:

Example Command Sequences

These examples assume you have a single slave device attached to the MODBUS network. We will be using the "Any Sensor" MODBUS address for communication. A RS-485 communication cable is available on the CO2 Meter website for purchase.

Read Sensor:

Master Transmit:
0xFE 0x04 0x00 0x01 0x00 0x01 0x74 0x05

Slave Reply:
0xFE 0x04 0x02 0x01 0x90 0xAC 0xD8

In this reply we notice that the slave returned 2 bytes, 0x01 and 0x90. These bytes combine to form a single signed integer of 0x190, equivalent in decimal notation to 400. This value, if the sensor is a O2 sensor, corresponds to 0.400% O2, in accordance with the scaling table above.