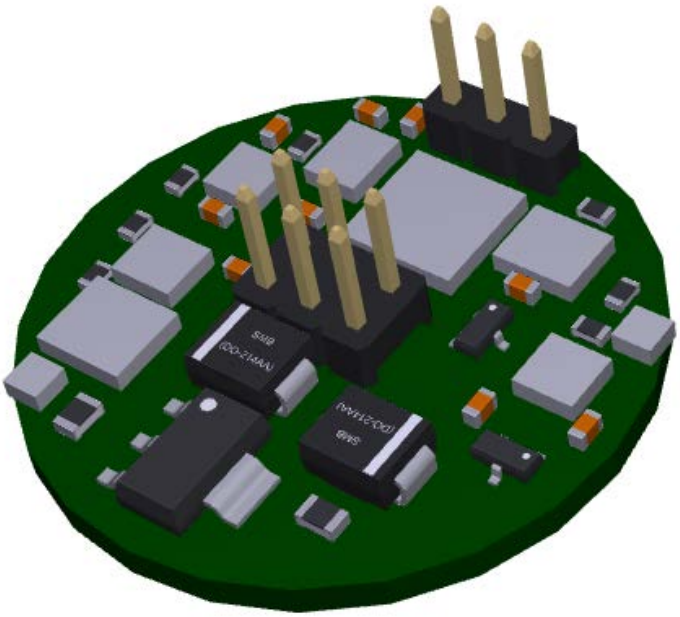
TX1 Sensor Controller Manual



CO2METER.COM
CO2 MEASUREMENT SPECIALISTS

4 January 2019 Rev. A
C:\Users\morgan.morris\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\FR247NXA\Atex_Board_Manual.docx

Document History

Date	Revision	Description
30 November 2018	Draft A	Initial Draft
1 December 2018	Draft A	Revised Draft
2 December 2018	Draft A	Added Appendix
15 December 2018	Rev. A	Firmware 1 build 0, clarification and detail; HR, IR and CMD

Table of Contents

Table of Figures

Introduction

About this Manual

This document is a technical reference for the CO2Meter.com TX1 Sensor Controller. It provides the information necessary to use the sensor controller as well as information about configuring the controller for various applications. The sensor current supports the SST Luminox O2 sensors.

Software Version

This manual applies to Software Version 1.

Chapter 1 : Overview

General

The TX1 Sensor Controller is a high-performance controller supporting various types of gas sensor modules. The controller supplies an RS485 Modbus RTU interface to the host system and a separate UART interface to the gas sensor module. The controller also provides a wide range (6 to 24Volt) power supply and a 4 20ma 3-wire current loop output.

Measurements

The TX1 receives continuous measurements streamed from the sensor, typically once per second. The controller performs various computations at each measurement, including adjusting the raw controller reading to the field calibrated values. The measurements are made available to a host system through the Modbus interface. The primary gas sensor reading is made available as a 4-20ma current loop output.

Chapter 2 : Communications Interface

General

The TX1 communicates with a host system through an RS485 Modbus RTU interface. This interface allows many slave devices to be connected to a host controller over a single twisted cable. A common ground reference is normally required. For long lines, the twisted-pair should be terminated with a 120-ohm resistor at each end.

Physical Connections

The TX1 uses a 6 pin (2x3) header with 0.1-inch centers. In some cases, the header is not populated and appropriate wire terminals are soldered directly to the header pads.

Connector Assignments

Pin	Function	Pin	Function
1	Loop Output	2	Reserved
3	GND	4	RS485 B (Data +)
5	DC Power (+6 – 24V)	6	RS485 A (Data -)

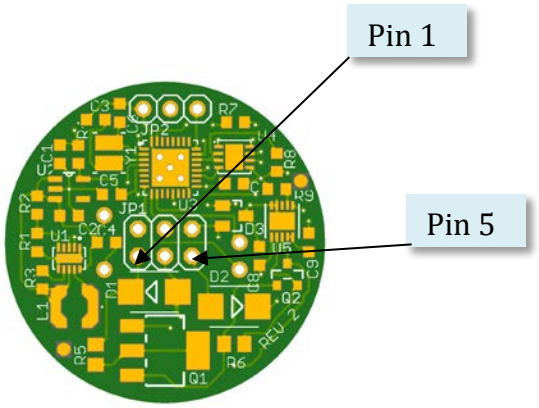


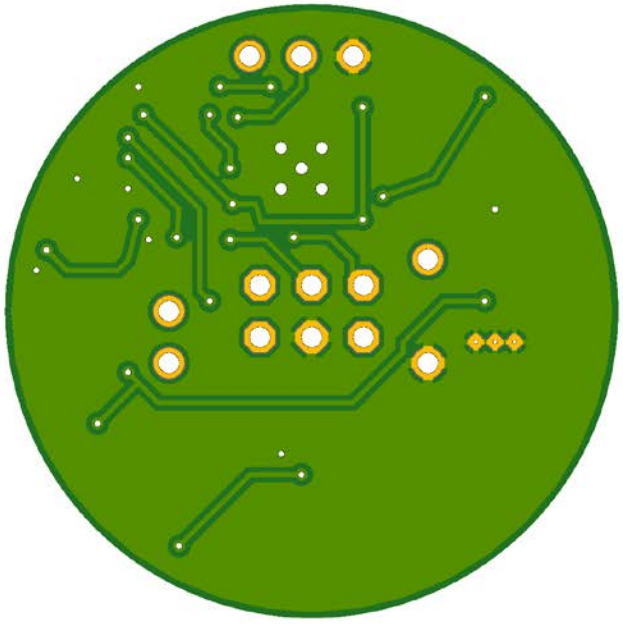
Figure 2-1 TX1 Sensor Controller

The DC Power and GND connections are protected from reverse polarity. The RS485 Data lines are protected against high voltage transients. The Loop Output pin is also protected against transient voltages. There is no protection on reserved Pin 2 which connects directly to the microcontroller.

> WARNING: Use care when connecting DC power. While the DC power and Ground connections are protected against reverse polarity, all other pins on the connector will be damaged if DC voltages > 5 V are continuously applied.

Chapter 3 Sensor Connections

The Luminox sensor is installed centered on the solder side of the TX1 PCB. The sensor is normally soldered to the board. 5V power is supplied to the sensor from a switching power supply.



Chapter 4 : Registers

The TX1 device provides a total of 32 Input Registers and 32 Holding Register that may be accessed through the Modbus RTU protocol. General information on the protocol as used on CO2METER Inc. devices can be found in the RS485 Modbus RTU Interface manual. The following descriptions are specific to the register assignments and commands supported by the TX1.

Note that the register numbers are as defined in the Modbus Specifications. Some PLC controllers will reference these numbers. In the Modbus Specifications the Input Registers are numbered from 30001 to 30999. These correspond to addresses of 0 in 998 used with "Read Input Register" function. Similarly, the holding registers are numbered from 40001 to 40999. These correspond to addresses of 0 to 998 used with the "Read Holding Register" and "Write Multiple Registers". The TX1 currently uses addresses in the range 0 to 31. Addresses outside this range result in a Modbus error (Illegal Data Address).

Note also that only the Holding Registers can be written from the host. Input Registers are read/only.

Input Registers

The TX1 supports 32 16-bit input registers. These registers are maintained by the controller during operation and can be read at any time. The "int" type is a 16-bit signed quantity. The "uint" type is a 16-bit unsigned. The float type is a 32-bit IEEE764 formatted number. For floating point values, the lower numbered register contains the 16-bit word with the exponent and upper bits of the significant. The higher numbered register contains the remaining bits of the significant.

The status bits are described separately in a following section.

Table 2-1 Input Registers (Read Only)

Reg	Addr	Type	Description
IR30001	0	int	Busy/Error Status (see below)
IR30002	1	int	Reserved
IR30003	2	int	O2 Concentration in 10's of ppm
IR30004	3	int	O2 Partial Pressure in tenths of millibars
IR30005	4	int	Temperature in 10ths of a degree C
IR30006	5	int	Pressure in millibars
IR30007	6	uint	Error Status received from Sensor
IR30008	7	uint	Manufacturer ID Code
IR30009	8	uint	Model Number
IR30010	9	uint	Firmware Version
IR30011	10	uint	Controller Serial Number
IR30012	11	uint	Received PDU Count
IR30013	12	uint	Received PDU Errors

Reg	Addr	Type	Description
IR30014	13	uint	Transmit PDU Count
IR30015	14	uint	Count of Received Measurements from Sensor
IR30016	15	int	Raw %O2 as received from Sensor
IR30017	16	float	Floating Point % O2
IR30018	17		
IR30019	18	float	Floating Point Partial Pressure O2
IR30020	19		
IR30021	20	float	Floating Point Temperature
IR30022	21		
IR30023	22	float	Floating Point Pressure
IR30024	23		Reserved
IR30025	24		Reserved
IR30026	25	-	Reserved
IR30027	26		Reserved
IR30028	27	-	Reserved
IR30029	28	-	Reserved
IR30030	29	-	Reserved
IR30031	30	-	Reserved
IR30032	31	-	Current DAC value to current loop (0 – 4095) as calculated from IR30003

Input Register Descriptions

IR30001 (Address 0) Busy/Error Status

This register provides the current Busy/Error status of the device. The bit assignments are as follows:

Bit 15 (MSB) Busy.
Bits 4-14 Reserved.
Bit 3 Command Failure.
Bit 2 Flash Error.
Bit 1 Invalid Command.
Bit 0 (LSB) Power Fail. (Set at Power Up)

All bits can be cleared by executing the “Error Reset” device command.

IR30003 (Address 2) O2 Concentration

This register provides the current concentration of O2 in 10’s of ppm.

IR30004 (Address 3) Partial Pressure O2

This register provides the current partial pressure of O2 in tenths of a millibar.

IR30005 (Address 4) Temperature

This register provides the current temperature in tenths of a Degree C.

IR30006 (Address 5) Pressure
Barometric pressure from sensor Barometer.

IR30007 (Address 6) Error Status
This register indicates an error status received from the sensor.

IR30008 (Address 7) Manufacturer Id
This register provides a manufacturer ID. All CO2Meter Inc. devices will have 35642

IR30009 (Address 8) Model Number
This register provides a model number. TX1 devices have a value of 1.

IR30010 (Address 9) Firmware Version
This register indicates the firmware level of the device.

IR30011 (Address 10) Serial Number
This register provides the serial number of the controller. A value of 65535 indicates that a serial number has not yet been assigned.

IR30012 (Address 11) PDU Count
This register contains the total number of packets received by the device since power up.

Holding Registers
There are 32 Holding Registers. The Holding Registers are read/write. They are initially loaded from non-volatile memory at power up. They may then be altered by the host. Note that holding registers are *not* automatically saved to non-volatile memory. Registers are saved to non-volatile memory only as a result of a specific write command.

Table 2-2 Holding Registers (Read/Write)

Reg	Address	Default	Description
HR40001	0	N/A	Checksum Value for FLASH Storage
HR40002	1	0	Reserved
HR40003	2	0	Reserved
HR40004	3	0	Reserved
HR40005	4	0	Reserved
HR40006	5	0	Reserved
HR40007	6	2	Gas Type
HR40008	7	0	Zero Value, reported in zero calibration to N2
HR40009	8	0	K Value of calibration slope of calibration K/32768 applied after subtracting 0 value.
HR40010	9	0	Calibration Concentration, value used in most recent calibration.
HR40011	10	0	Span- not used in this application place holder for software compatibility.

Reg	Address	Default	Description
HR40012	11	25000	Full Scale, full scale value for current loop 20mA=25%.
HR40013	12	10	Multiplier to get to PPM
HR40014	13	0	Reserved
HR40015	14	0	Reserved
HR40016	15	21	Modbus Address
HR40017	16	0	Comm Configuration (Stop bits and Parity)
HR40018	17	8	Baud Rate (Multiple of 1200)
HR40019	18	0	Reserved
HR40020	19	0	Loop Test (when non-zero)
HR40021	20	0	Reserved
HR40022	21	550	Current Loop 4ma Level, ADC Value corresponding to 4ma loop current.
HR40023	22	2740	Current Loop 20ma Level, ADC Value corresponding to 20ma loop current.
HR40024	23	0	Reserved
HR40025	24	0	Reserved
HR40026	25	0	Reserved
HR40027	26	0	Reserved
HR40028	27	0	Reserved
HR40029	28	0	Reserved
HR40030	29	0	Reserved
HR40031	30	0	Command Parameter
HR40032	31	0	Command

Holding Register Descriptions

HR40001 (Address 0) Checksum

This register contains the checksum of the initial holding register values as read from non-volatile memory. This register may be freely written from the host. When holding register content is saved to non-volatile storage, a new checksum value will be computed and written to non-volatile memory.

HR40007 (Address 6) Gas Type

This register contains a code that indicates the type of Gas sensed by the sensor. This value is always 2 which indicates O2.

HR40016 ADR 15 Modbus Address.

Default 21, Sensors will respond to the configured address or to the 0xFE. Note that the address returned in the message is the address that is used by the host. This is necessary to

comply with the protocol rules. Note also address 0xFE is outside the range of legal addresses.

Note that theoretically the sensor reacts also to the broadcast address 0. However, they are presently no implemented commands that use the broadcast address. Protocol rules require that no response be made to broadcast messages. In the future we might implement functions at this address e.g. reset. This allows the PLC to put all devices in a known reset state.

HR40017 ADR 16 Comm configuration (stop and parity)

The low order three bits define the Comm Configuration.
Bit 0 (lsb) is 0 for 1 stop bit (default) and 1 for two stop bits.
Bit 1 is 0 for no parity, 1 for parity.
Bit 2 is 0 for even parity, 1 for odd parity.

NOTE: These are not implemented in Ver 1 Build 0.

HR00020 ADR 19 Current loop test.

When >0<4096 will cause 4-20 loop to operate at that level for trim test.

Chapter 5 Device Commands

General

Device commands initiate operations at the device in response to messages from the host. Note that Device commands are not defined in the Modbus Specifications but are specific to the device (TX1).

Commands are used for operations e.g. calibration, error handling, and saving holding register content to non-volatile storage. Commands are also used for certain factory function associated with manufacturing and test.

Command Register

Holding Register 40032 (Address 0x1f) is used to initiate a command. In some cases, a command may require a parameter. The parameter to a command, if used, must be written to Holding Register 40031 (Address 0x1e) in the same Modbus message that writes the command or a prior message.

Modbus function code 16 (0x10), Write Multiple Holding Registers, allows the parameter and command to be sent in a single PDU.

Command Busy

A command can only be sent to the device when it is not Busy. The Busy indication is provided as the most significant bit in the Status Register (Input Register 30001 at address 0).

Command Execution

Commands begin executed as soon as they are received. If a command is not valid for the device, one of the Command Error bits will be set in the status register. Error bits are valid when the status indicates not busy.

Holding Register 40032 (Address 0x31) is automatically cleared when the command complete. Holding Register 40031 (Address 0x30) is also cleared if the command was successful. If an error occurred, Holding Register 40031 may contain an additional error code.

Note that some commands will execute immediately. In this case, by the time the host requests status the status will return non-busy the current error status. The error status is always valid for the last command when the device status indicates not busy.

Command Descriptions

Device Command 1 (0x0001) Error Reset

This command resets all error bits in the status word. Note that certain hardware errors may cause bits to be set again.

Device Command 2 (0x0002) Factory Use
This command is reserved for factory use.

Device Command 16 (0x0010) Save Holding Registers to Flash
This command saves the holding registers to Flash.

Device Command 17 (0x0011) Restore Factory Presets
This command will restore the factory presets to the holding registers. There are 7 groups of presets numbered 0 to 6. The command parameter indicates which group should be restored. Note that the retrieved values are not automatically saved to flash. To save the registers to flash, a separate Device Command 16 is required.

Device Command 32 (0x0020) Zero Calibration
This command sets the zero value for the sensor. The sensor should be shown 100% nitrogen for approximately 1 minute prior to issuing this command. Holding registers are automatically saved upon execution of this command.

Device Command 33 (0x0021) Calibrate to Span
This command calibrates the sensor based on the known concentration provided as the command parameter. This should be expressed in 10's of ppm. For example, for 20.9% the value should be 20900. Holding registers are automatically saved upon execution of this command.

Device Command 9985 (0x2701) Factory Use
This command is reserved for factory use.

Chapter 6 : Testing

WARNING: These instructions are intended for engineering personnel familiar with the RS485 Modbus protocol.

Modbus Tool

The RS485 Modbus RTU protocol is a packet-oriented protocol. All of the characters of the packet must be sent in a single burst with no gaps and a valid two-byte CRC. The absence of a character for 2.5-character times indicates the end of a packet.

The slave device will not react to a packet unless the CRC is correct and the address field is that of the device. Due to these characteristics, it is not practical to test device operation from a terminal emulator program.

Fortunately, there are several tools available for testing Modbus RTU devices. The following sections describe the use of one of these tools, Modbus Tool.

Modbus Tool is a free and open source Modbus Protocol testing program available as a free download. The tool can be downloaded from:

https://github.com/graham22/ModbusTool/blob/master/Setup/ModbusTool.msi

This windows installer package includes two programs. One program, “Modbus Master” is used to simulate a host and a second program “Modbus Slave” is used to simulate a device. The following instructions will use only the Modbus Master program.

Serial Port

Modbus tool, as well most other PC based tools for Modbus RTU devices, requires a serial RS485 port that appears as a com port on the local system. FTDI manufactures a smart cable well suited for this purpose. The manufacturers part number is USB-RS485-WE-1800-BT. This cable is available Digi-key, Mouser, and other electronic distributers. A version with a longer cable is also available.

Drivers for FTDI cables are pre-installed on most Windows systems. The latest drivers are also available from FTDI at the following link:

https://www.ftdichip.com/FTDrivers.htm

Cable Connections

The FTDI cables are color coded. The labeling on the TX1 uses “B Data +” and “A Data -“conforming to the definitions of A and B in the RS485 standard. The definitions on the FTDI cable are correct for the Data Polarity but opposite for A and B.
The yellow FTDI wire should be connected to Pin 6 of the TX1. The orange wire should be connected to Pin 4. Note that, in general, the Data+ and Data- labeling of devices is

consistent across most RS485 devices. The labelling for A and B is often opposite of that of the standard (Due to a different labeling convention used by manufacturers of the semiconductor interface devices).

Power Connections

The TX1 must be powered for proper operation. The DC power supply input is specified from 6V to 24V. Many devices will operate with power supply as low as 5V. If the current loop is not used, the TX1 device may be powered from the 5V USB available on the FTDI connector. Note, however, that this is not recommended.

> WARNING: Use care when connecting DC power. While the DC power and Ground connections are protected against reverse polarity, all other pins on the connector will be damaged if DC power is applied.

Configuring Modbus Tool

Start the Modbus Master tool. A graphical interface will appear similar to Figure 5-1 Holding Registers.

The connection should be set to RTU. Select the COM port of your FTDI RS485 cable (or other RS485 device). The other communication settings should match the figure.

Before connecting, set the slave ID to 21. This is the default Modbus address of co2meter.com devices. Alternatively, you can set the ID to 254. All co2meter.com devices will respond to address 254, regardless of their configuration.

Using the Modbus Tool

Press the connect button. This will connect the program to the FTDI smart cable. If the TX1 device is powered and the RS485 connections are correctly made, the Modbus Master tool will be able to issue commands to the device. The red and green led lights on the FTDI cable indicate transmit and receive activity.

The Modbus Master tool allows you to specify a starting address and a number of registers. As the TX1 has 32 registers in both the Holding Register and Input Register ranges, set the Start Address to 0 and the size to 32. May sure you hit the “Apply” button after changing any of these values.

The TX1 supports Reading Holding Register, Read Input Register, Write Single Register and Write Multiple Registers. The other functions are not supported.

Communication Log

Modbus Master provides a log of the TX and RX communication activity in the lower portion of the screen. This provides the hexadecimal values of the byte sent and received.

Reading Holding Registers

Begin by pressing the “Read holding register” button. If everything is operating correctly, the display should be populated in addresses 0 to 31 similar to the figure.

Reading Input Registers

The Input Registers can be read by pressing “Read input register”. If everything is operating correctly, the display should be populated in addresses 0 to 31. The display should similar to the figure but the actual values will be different.

Writing Holding Registers

The TX1 supports both the Write Single Register and Write Multiple Register functions. When using the Modbus Master tool is usually easiest to use the Write Multiple Register function. This allows you to leave the starting address and size at 32.

Always to perform a “Read Holding Registers” to get the current value of registers. You can then modify one or more registers and press the “Write Multiple Registers” button. This will update all 32 registers.

Figure 6-1 Holding Registers

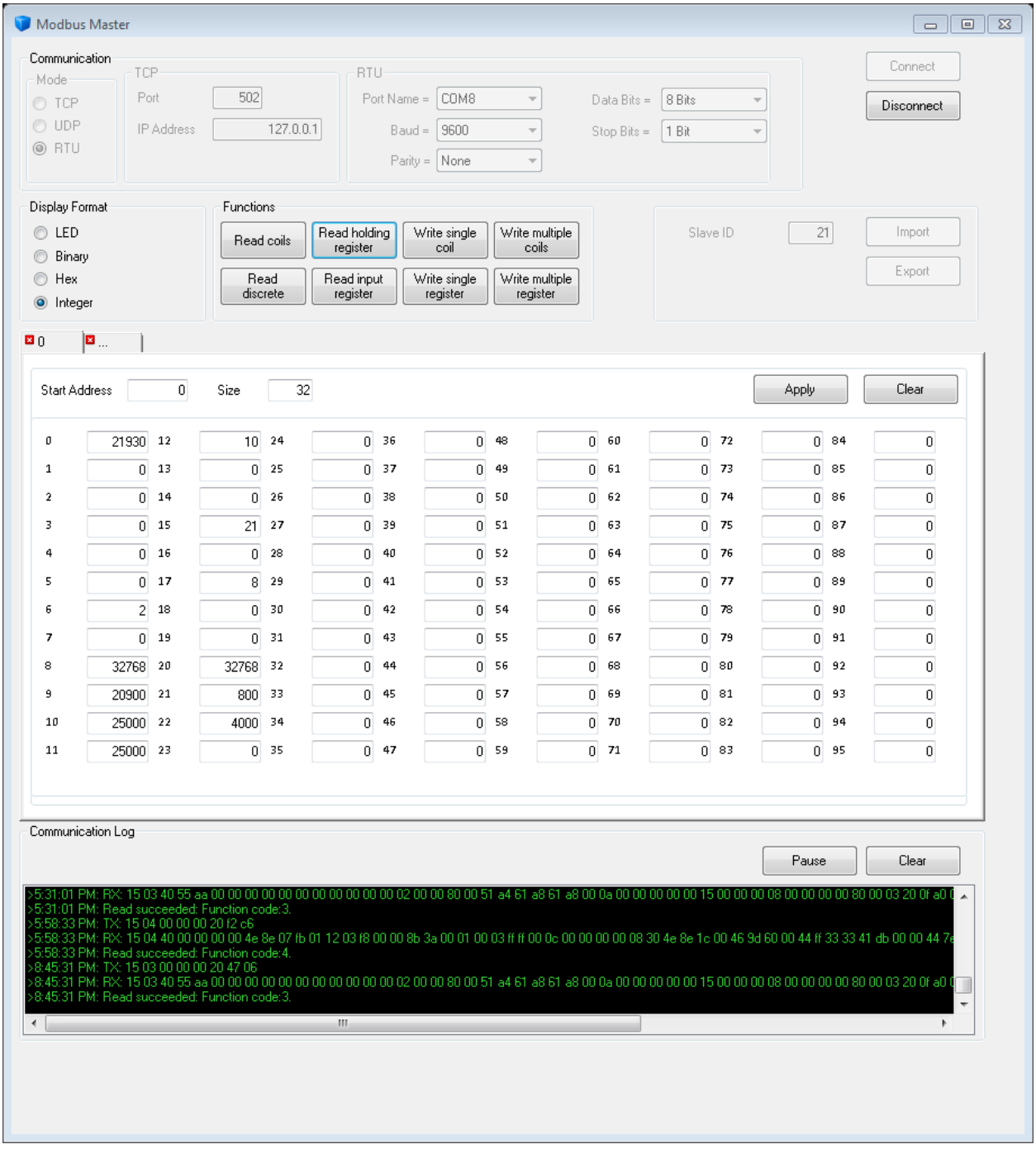
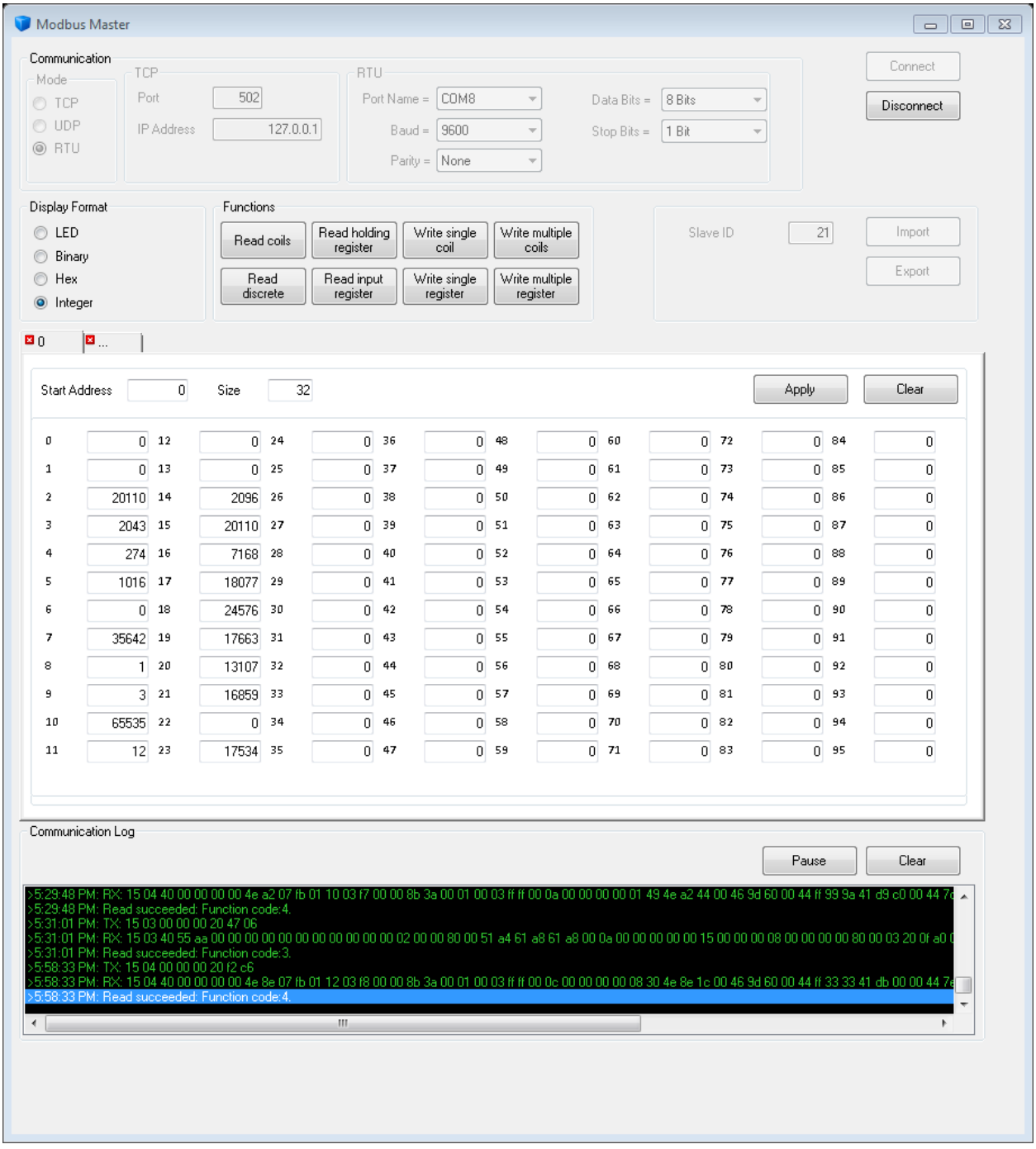


Figure 6-2 Input Registers



Chapter 7 Firmware Updates

General

This chapter provides instructions for update the firmware of the device.

> WARNING: Programming the device creates a risk of damage as the result of ESD (Electrostatic Discharge). These instructions are intended for engineering and manufacturing personnel trained in the proper control of ESD.

Programming Configuration

Processor

The AX1 uses an ST Microelectronics STM32F051K8U6 microcontroller. This device incorporates a 32-bit ARM Cortex M0 core and 64KB of flash memory.

Programming Interface

The device is programmed through a 3-wire SWD (Single Wire Debug) port. The interface consists of two signals (CLK and DIO) and GND.

Programming Voltage

The microcontroller is internally powered by a regulated 3.3V supply derived from the input voltage. The programming device must be set for 3.3V as target VCC is not available on the 3-wire interface. The means of establishing the target voltage levels is programmer specific.

Device Power

During programming, the TX1 device must be powered by an external 6V to 24VDC supply connected to the power pins. Input voltage must be applied prior to programming.

Connections

The connections during program are illustrated in Figure 7-1 Programming Connections. As the SWD programming pins carry high speed signals, the wiring should be short, less than 4 inches if open wires are used and less than 12 inches if flat cable with alternating grounds is used. For open wires, it is recommended that the wires be twisted.

Programming Device

The recommended programmer is the ST-LINK/V2 from ST Microelectronics. This is a low cost (< US$25) device compatible with the recommended software. Other devices e.g. the ST-LINK V3SET may be used if configured for 3.3V targets. This may require a separate 3.3V power supply, depending on the device.

To configure the ST-LINK.V2 for 3.3 programming over a 3-wire interface, it is necessary to jumper Pins 1 and 19. Pin 19 contains 3.3V source from the programmer. Pin 1 is the target voltage input to the programmer.

Figure 7-1 Programming Connections

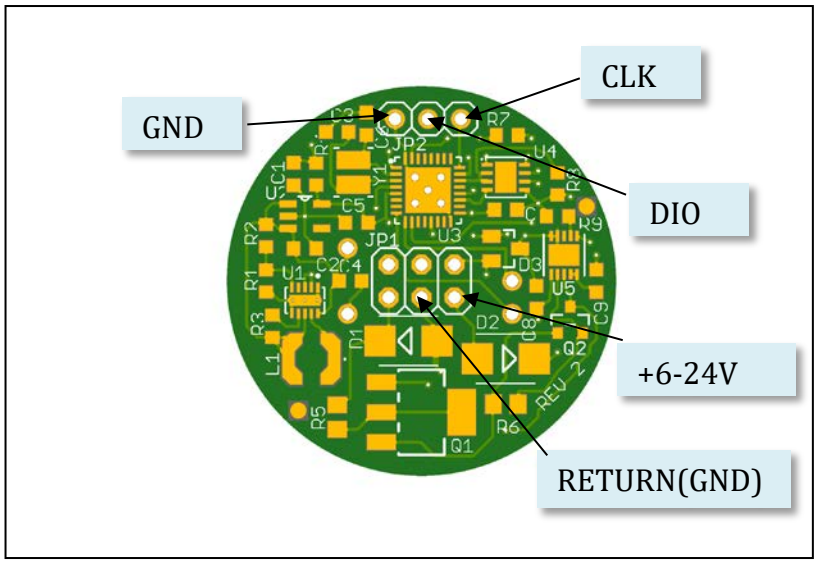
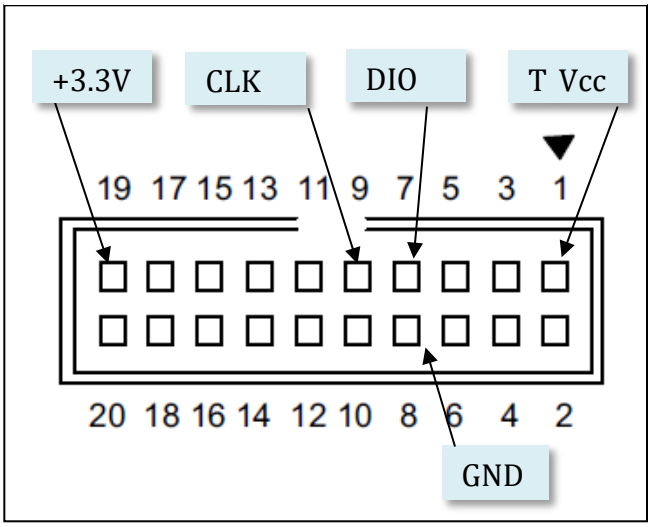


Figure 7-2 ST-LINK/V2 Connections



Programming Software

The recommended programming software is the STM32 CubeProgrammer from ST Microelectronics (Version 1.3 or later). This software is available for free from ST. Note that STM32 CubeProgrammer requires a Java Runtime Environment (8 or later) as a prerequisite.

It is also possible to program the device using the ST-LINK Utility version 4.2 or later.

The ST-LINK/V2 device should normally be updated to the latest version of its device firmware. This update is normally built-in to the utility.

Firmware Files

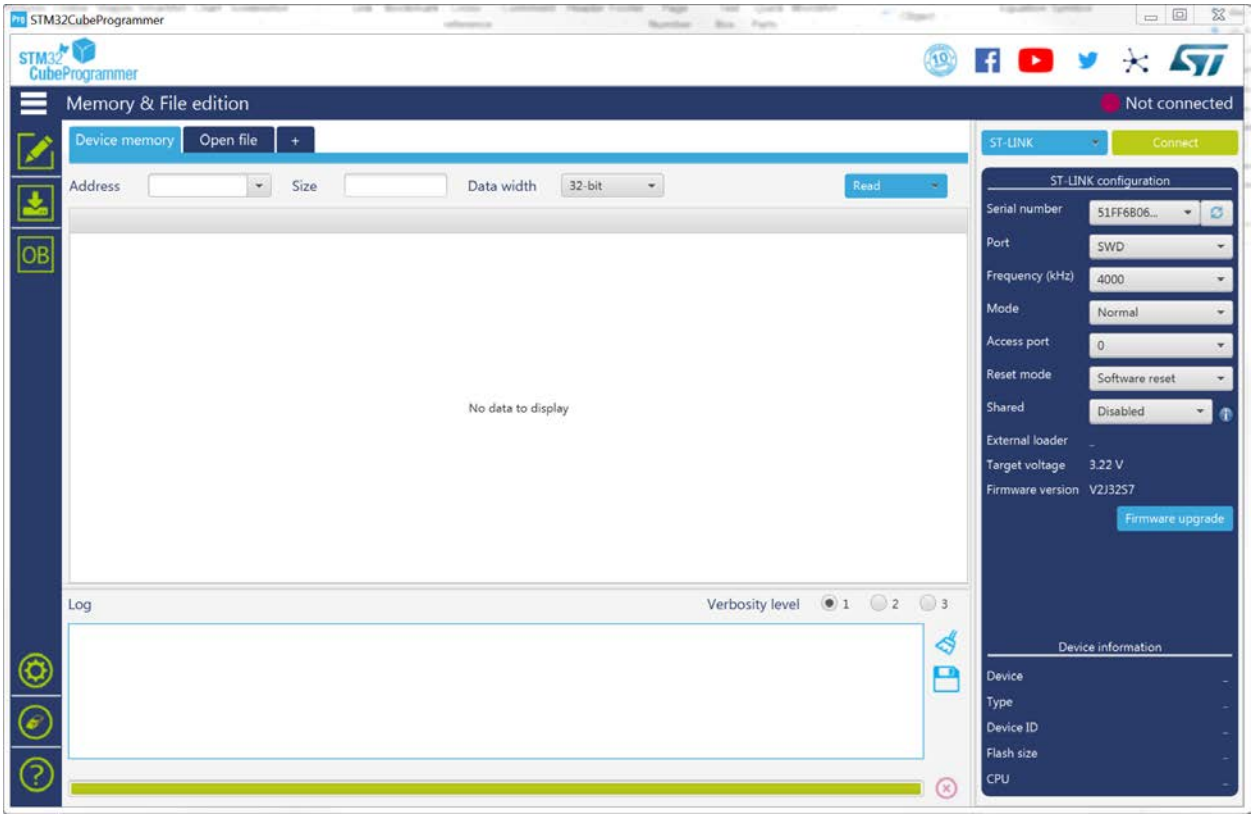
Firmware updates are supplied in the SREC file format which is compatible with both the STM32 CubeProgrammer and ST-LINK Utility tools.

Programming Procedure

Before starting the programmer insure that the proper device connections are made and that the device under test is powered up. Insure that the programming device is connected to the system (normally USB).

Start the STM Cube Programmer. The display should similar to Figure 7-3 Initial Screen.

Figure 7-3 Initial Screen



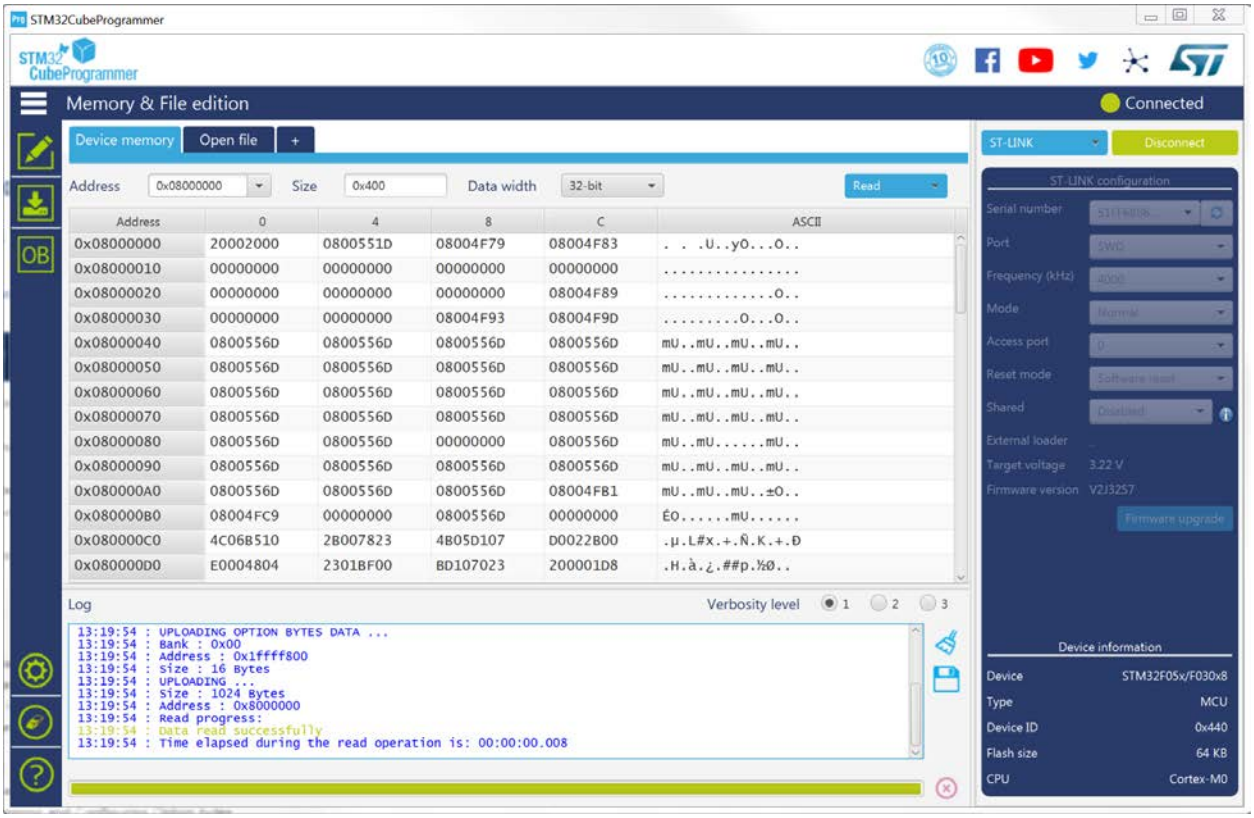
Insure that the ST LINK shows in the upper right corner of the screen and that there is a serial number present for the ST-LINK device.

Check the ST-LINK configuration. It should match the settings shows on the figure.

Connect to Device

Press the connect button. The screen should change to similar to Figure 7-4 Screen After Connection.

Figure 7-4 Screen After Connection



Device Erase

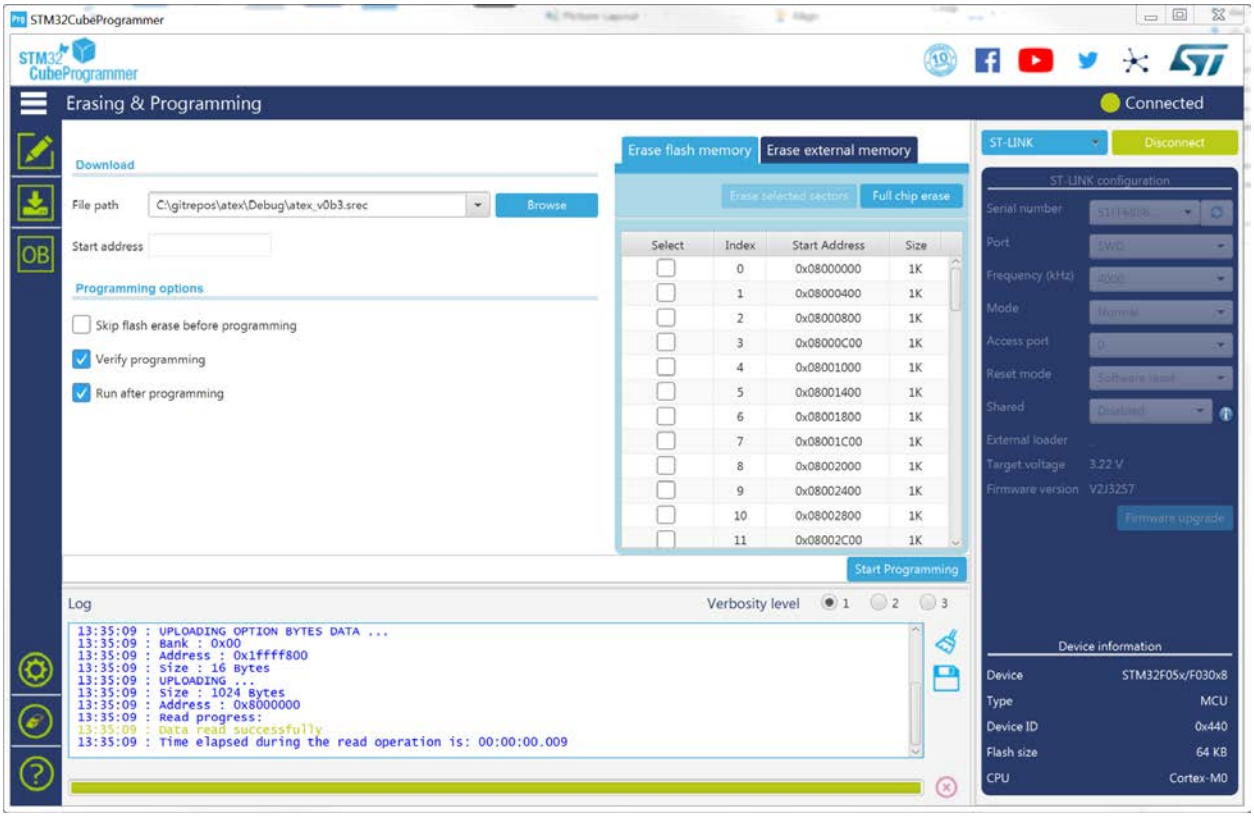
A mass chip erase function is available on a button on the lower left corner of the window. Should not normally perform a chip erase. Doing so will overwrite data stored in flash memory outside of the program space. This includes the configuration and calibration information and device serial number. When performing a normal programming operation, the tool will erase those sectors associated with the programming.

Programming Firmware

To program new firmware, click on the download ICON on the left side of the screen. This is second green ICON from the top. This will bring up the “Erasing and Programming” windows as illustrated in Figure 7-6 Erase and Programming Screen.

Press the browse button next to the “File path” to select the file to be programmed. This file should end with a “.srec” extension. Make sure that the “Verify programming” button is checked and that the “Skip Flash erase” button is *not* checked.

Figure 7-5 Erase and Programming Screen



Chapter 8 : Support

The quickest way to obtain technical support is via email. Please send all support inquires to **support@co2meter.com**.

Please include a clear, concise definition of the problem and any relevant troubleshooting information or steps taken so far, so we can duplicate the problem and quickly respond to your inquiry.

Warranty

This device comes with a 90 day (warranty period) limited manufacturer's warranty, starting from the date the meter was shipped to the buyer.

During this period of time, CO2Meter.com warrants our products to be free from defects in materials and workmanship when used for their intended purpose and agrees to fix or replace (at our discretion) any part or product that fails under normal use. To take advantage of this warranty, the product must be returned to CO2Meter.com at your expense. If, after examination, we determine the product is defective, we will repair or replace it at no additional cost to you.

This warranty does not cover any products that have been subjected to misuse, neglect, accident, modifications or repairs by you or by a third party. No employee or reseller of CO2Meter.com's products may alter this warranty verbally or in writing.

Liability

All liabilities under this agreement shall be limited to the actual cost of the product paid to CO2Meter.com. In no event shall CO2Meter.com be liable for any incidental or consequential damages, lost profits, loss of time, lost sales or loss or damage to data, injury to person or personal property or any other indirect damages as the result of use of our products.

Returns

If the product fails under normal use during the warranty period, a RMA (Return Material Authorization) number must be obtained from CO2Meter.com. After the item is received CO2Meter.com will repair or replace the item at our discretion.

To obtain a RMA number, call us at or email us at (386) 256-4910 support@co2meter.com.

When requesting a RMA please provide reason for return and original order number.
If we determine that the product failed because of improper use (water damage, dropping, tampering, electrical damage etc.), or if it is beyond the warranty date, we will inform you of the cost to fix or replace the product.

For more information visit our website: www.CO2Meter.com/pages/faq

Contact Us

We are here to help!

For information or technical support, please contact us.

support@co2meter.com
(386) 256-4910 (Technical Support)
(386) 872-7665 (Sales)

www.co2meter.com
Address:
CO2Meter, Inc.
131 Business Center Drive
Ormond Beach, FL 32174
USA