

Modbus CAN Adapter

ASY-529-X01

Rev 1.05

Contents

Getting Started.....	3
Overview	3
System Requirements	3
Initial Setup	3
Number Conventions	3
Limits.....	3
Specifications	4
Connections	5
Main Screen	6
STORE:.....	6
READ:.....	6
Version:.....	6
Updating the adapter with your settings.....	6
Communication Settings.....	7
CAN Bus.....	7
Serial.....	7
Clear register after reading	7
CAN Frame Windows	8
CAN Message Object:.....	8
DLC:	8
RTR:	8
Reference Name:	8
Modbus Register List:.....	8
Quick Set:	8
Entry Errors	9
Deleting an Entry.....	9
Using the Adapter	10
Best Practices	10

Getting Started

Overview

The Can Modbus adapter provides a means for CAN traffic to be read by a Modbus master device. CAN messages are received by the adapter and compared to the messages in the loaded receive message table. The data is then stored in the adapter in the defined Modbus register for retrieval by the Modbus Master.

Transmissions are made from the Modbus master by writing to the register or registers associated with the CAN identifier to be transmitted. The adapter will then store the information in the predefined register and transmit the CAN message.

CAN message objects and register associations can be defined in receive and transmit tables referenced to the CAN message action direction.

System Requirements

Operating system: Windows 10

Communications: USB 2.0

Initial Setup

The FTDI driver set should be loaded on your PC prior to running the program. This can be obtained by searching for FTDI drivers or from the following link:

<https://www.ftdichip.com/Drivers/VCP.htm>

Choose the windows operating system VCP driver for the x64 (64 bit) architecture and follow the prompts to install it.

Number Conventions

All numbers in the program are represented as decimal numbers except CAN objects. These are represented only in Hexadecimal which is the usual format for representing the COB or PID values.

Limits

The adapter can handle up to 250 CAN ID values on receive and 250 CAN ID values on transmit. Up to 8 registers can be assigned to each ID representing each byte of the message. A total of 4000 registers can be assigned.

Specifications

Hardware:

Input voltage*:	10 to 30 VDC or 12-18 VAC
Input current:	< 50mA
Input power:	< 1W
Operating Temperature range:	-20 to 60°C
Modbus inputs:	One A/B input
CAN bus inputs:	Two with H, L, and GND connections.
USB:	USB 2.0B

**Note: Device may also be USB powered without any connection to the input power connector.*

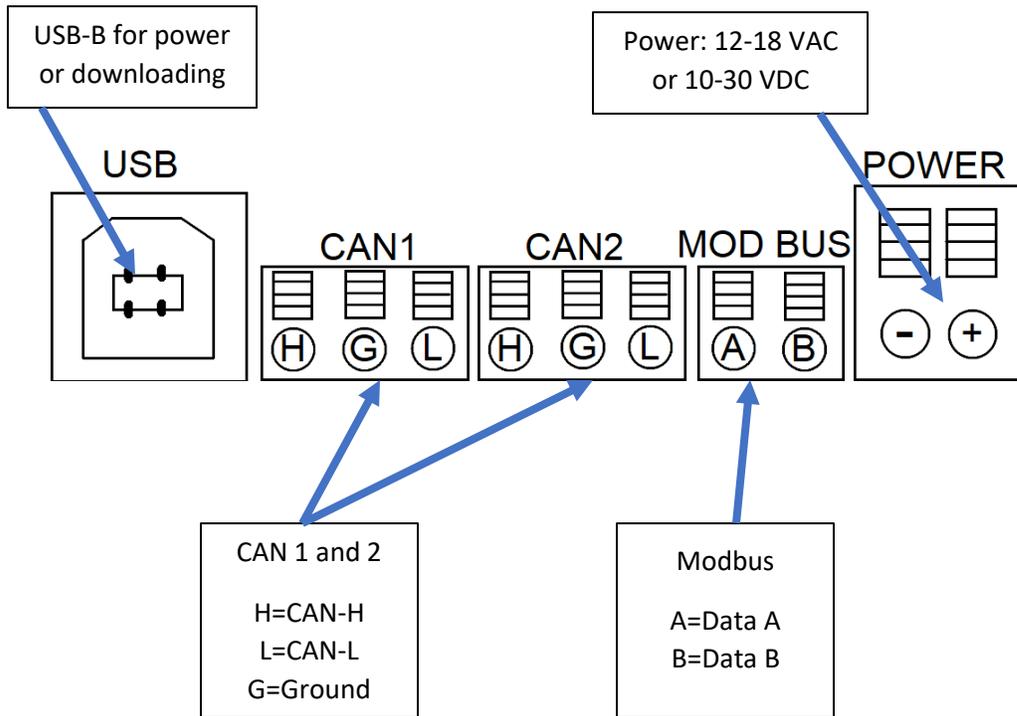
Modbus

Interface:	RS485
Protocol:	MODBUS RTU
RTU protocol default:	29 bits
Coding system:	8-bit binary
Start bits:	1
Data bits:	8, least significant bit sent first.
Parity:	Even
Stop bits:	1
Device address range:	1 to 247
Default Address:	1
Baud:	2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200, 250000
Parity:	Even, Odd, None
Stop bits:	Automatically selected based on parity.
Default baud:	115200
Minimum silent interval time:	30mS

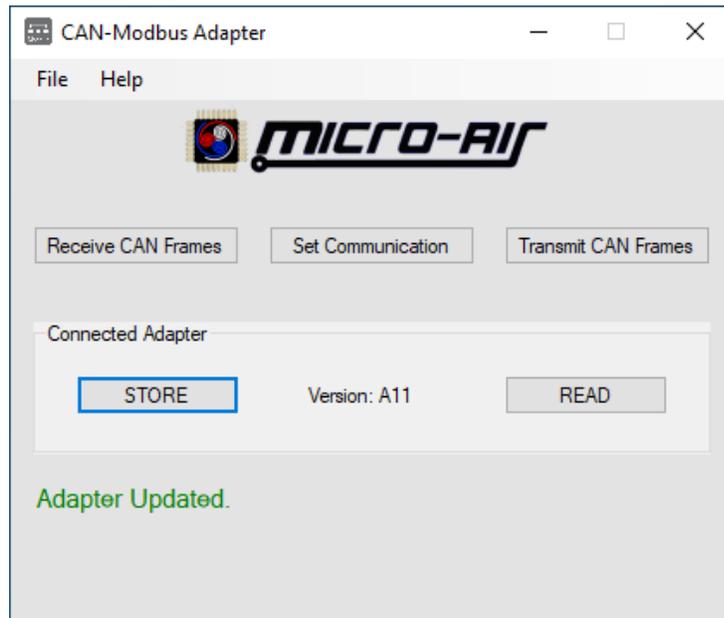
CAN bus

Interface Specification:	CAN rev 2.0A and 2.0B compliant, ISO11898.
Baud:	100K, 125K, 250K, 500K
Identifier:	11 or 29 bits
Termination:	Selectable 124-ohm terminator resistor
Isolation:	Fully isolated including ground connection.

Connections



Main Screen



The GUI main screen is the starting point for the program. An initial project is created the first time the program is started. Finish setting up the adapter by setting your communication settings, receive Can frames and Transmit CAN frames in the windows. Save the information to your project from the file menu or at the prompt when you close the program. Projects are stored in the documents directory on the PC in a folder called "CAN-Modbus Adapter".

STORE: This sends your project settings to the adapter.

READ: This over-writes the existing project settings in the PC with the project settings from the adapter. It is recommended that you create a new project before reading the device so the current project settings are not cleared.

Version: The firmware version of the adapter which is updated with a successful connection to the adapter or a firmware update.

Updating the adapter with your settings

Connect the USB cable to the adapter to download the project to the adapter. The USB cable will provide power to the adapter so a separate power connection is not necessary. Press the STORE button to download the project to the adapter. This takes the information from the received CAN frames, transmit CAN frames, and communication parameter windows and stores it in the adapter.

Communication Settings

CAN Bus

Baud rate: set the CAN baud rate to the system baud rate.

CAN 2.0A or 2.0B: Select the protocol used.

Data time out (sec): The maximum time to wait after an RTR transmission is sent.

Modbus register: The reserved register used to return a 1 for an RTR request reception that occurred within the timeout period or a 0 if the reception did not occur within the timeout period.

Serial

Baud rate: Set the Modbus baud rate required.

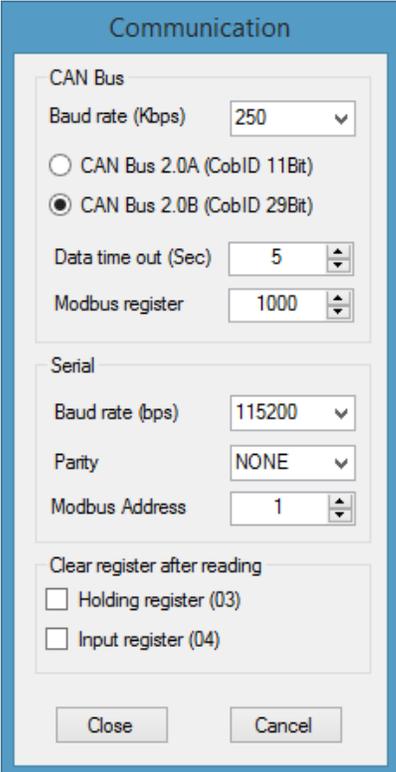
Parity: Set the parity required by the Modbus master.

Clear register after reading

Holding register: If the register was read using a read holding register command (Modbus function 03), the interface will clear the data in the register following the read. This allows the Modbus master to determine when new data is received from the CAN bus.

Input register: If the register was read using a read input register command (Modbus function 04), the interface will clear the data in the register following the read. This allows the Modbus master to determine when new data is received from the CAN bus.

Pressing close will save the selections to the selected project. Pressing cancel will ignore changes and return to the main screen.



The screenshot shows a dialog box titled "Communication" with a blue header. It is divided into two main sections: "CAN Bus" and "Serial".

- CAN Bus Section:**
 - Baud rate (Kbps): 250 (dropdown menu)
 - Protocol selection: CAN Bus 2.0A (CobID 11Bit) and CAN Bus 2.0B (CobID 29Bit)
 - Data time out (Sec): 5 (spin box)
 - Modbus register: 1000 (spin box)
- Serial Section:**
 - Baud rate (bps): 115200 (dropdown menu)
 - Parity: NONE (dropdown menu)
 - Modbus Address: 1 (spin box)
- Clear register after reading Section:**
 - Holding register (03)
 - Input register (04)
- Buttons:** "Close" and "Cancel" buttons at the bottom.

CAN Frame Windows

CAN Message Object (Hex)	DLC	RTR	Reference Name
0x17F00100	4	<input type="checkbox"/>	
0x17F00200	4	<input type="checkbox"/>	
0x17F00300	4	<input type="checkbox"/>	
0x17F00400	4	<input type="checkbox"/>	
0x17F00500	4	<input type="checkbox"/>	
0x17F00600	4	<input type="checkbox"/>	
0x17F00700	4	<input type="checkbox"/>	
0x17F00800	4	<input type="checkbox"/>	
0x17F00900	4	<input type="checkbox"/>	
0x17F00A00	4	<input type="checkbox"/>	
0x17F11500	1	<input type="checkbox"/>	
0x17F11600	1	<input type="checkbox"/>	
0x17F11700	1	<input type="checkbox"/>	
*		<input type="checkbox"/>	

Modbus Register List

CAN Data 0: 200, Lower Byte selected

CAN Data 1: 201, Lower Byte selected

CAN Data 2: 202, Lower Byte selected

CAN Data 3: 203, Lower Byte selected

CAN Data 4: 0, Not Used selected

CAN Data 5: 0, Not Used selected

CAN Data 6: 0, Not Used selected

CAN Data 7: 0, Not Used selected

Quick Set: Single L, L / U, Single U, U / L

CAN Message Object: This is also known as PGN, COB and other terms depending on the system you are using. Use the 0x convention to specify hexadecimal values and enter the value you expect to receive.

DLC: This is the length of the CAN data packet in bytes. From 1 to 8 bytes can be received or 0 to 8 bytes transmitted.

RTR: RTR is available on the Transmit CAN frames window only. Setting the checkbox forces the adapter to set the RTR bit on transmit.

Reference Name: This name is a locally stored name you can use to reference your entries. Projects are stored in the Documents folder on your PC in CSV format and can be edited to enter a list of messages and DLC values.

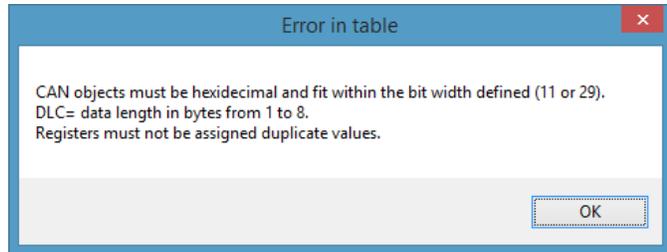
Modbus Register List: Set the Modbus register number in this list and the upper or lower byte to store the data. Entries should be sequentially numbered to allow multiple registers to be read or written in the same transmission. These numbers are in decimal and not hex.

Quick Set: Allows quickly setting the registers in sequence in Upper, Lower, or both bytes of the register referenced to the register set in CAN Data 0. It sets all registers above the DLC to 0 and sets them as unused.

Pressing close will save the entries to the selected project. Pressing cancel will ignore changes and return to the main screen.

Entry Errors

Lines with an error are highlighted in red and the table cursor is returned to the line with the error. The problem must be corrected before continuing to a different line. Line errors that exist when exiting the window will not be saved.



All CAN objects must conform to the 11 or 29 bit standard selected in the communication section. Entries that contain characters other than A to F or values exceeding those that can be put in an 11 or 29 bit format will be rejected. Positive values only are allowed.

CAN transmission lengths (DLC) must be between 0 and 8 for transmissions and 1 to 8 for receptions.

Register values cannot be duplicated in another CAN frame. This includes transmit registers on receive frames and receive registers on transmit frames. This check prevents creating inconsistent results where a frame could be set or received and then incorrectly read.

CAN ID's can only be duplicated on transmit frames with different DLC values. Duplicate receive frames will also result in an entry error.

Deleting an Entry

Press the delete key or press the right mouse button and select delete row to delete the currently selected row.

Using the Adapter

Connect the adapter to the CAN and Modbus lines as appropriate for your devices. Valid Modbus commands are listed in Table 1.

Table 1		
Access	Object type	Function Code
Read	Holding Registers	3
Read	Input Registers	4
Write	Single Holding Register	6
Write	Multiple Holding Registers	16

This adapter provides Modbus feedback for invalid registers, checksums, and transmissions as provided by the Modbus RTU specification. Your connected communication equipment should support identifying these error responses to make identifying configuration errors easier.

Best Practices

Always group transmit register values within a CAN message sequentially. This makes it easier to update all values in a single message in one transmission. The Modbus RTU specification limits the number of registers that can be read or written in a single exchange so keeping them sequential allows transmitting multiple messages in a single exchange while updating all values.

Receiving registers may also be grouped by message just like transmit registers. They also may be grouped as needed to simplify data acquisition. If for example, a CAN message has a temperature reading in byte 3 and 4 of the message and your PLC is updating this on the display, you can group these registers sequentially to provide an ordered feedback of just these values.