CBAS Modbus Programming Guide

With the CBAS Modbus RTU interface, you communicate (monitor and command) with any manufacturers equipment.

A little Background

Modbus is an application layer messaging protocol that provides client/server communication between devices connected on different types of buses and networks. It is similar to RS485 and RS232, and has been a standard industrial protocol since 1979 when Modicon introduced the protocol in its PLCs (Programmable Logic Controller). Schneider Electric now owns Modicon.

Modbus is a request/reply protocol and offers services specified by function codes. These function codes are elements of the Modbus request/reply protocol data units (PDUs).

There are three categories of Modbus:

- Modbus Serial – either Modbus ASCII or Modbus RTU
    - Modbus ASCII – each eight-bit byte in a message is sent as two ASCII characters, main advantage is that it allows time intervals of up to one second to occur between characters without causing an error.
    - Modbus RTU – each eight-bit byte in a message contains two four-bit hexadecimal characters, main advantage is its greater character density allows better data throughput than ASCII for same baud rate.
    - Other Serial Modbus – several manufacturers have made modifications to the Modbus protocol to meet their specific application needs, these would include a Modbus Daniels, Modbus Omniflow, Modbus Tek Air and others.

Modbus RTU

Modbus RTU is the standard used by the Modbus channel in CBAS. In order to use this channel, an add-on license must be purchased for each site. This will enable you to program as many Modbus RTU channels as you need at the site, so you don’t have to put all Modbus devices on the same channel.

Veris Meters

When first added to CBAS in January of 2003, the Modbus channel was only intended for use with Veris Hawkeye 8035 and 8036 meters. If you program a Modbus channel in CBAS, you will find these in the list of controllers to add to the channel. There is also a text file (covered later) available for the Veris Hawkeye 8136 meter.

485 vs. 422

Most Modbus devices on the market offer a choice between 4-wire and 2-wire communications. Just like a RS485 channel, a Modbus RTU over 485 channel uses 2-wire communications, which is a misnomer because it uses 3 wires: +, -, and shield. Some manufacturers use only 4-wire communication, also known as RS422. In this situation, you will have to use the converter that the manufacturer recommends. In some situations, you will have to use Computrols RP1 to convert 485 to 232, then a 232 to 422 (4-wire 485) converter.

Master/ Slave

Modbus RTU is a version of RS 485, which is a Master/Slave protocol. In CBAS, all Modbus devices in the system must act as Slaves. The CBAS Server is the Master.

Modbus Generic

The Modbus RTU channel in CBAS started with just 2 controllers: The Veris Hawkeye 8035 and 8036 Meters. Because all Modbus RTU devices use the same standards and we were getting requests to add more and more devices to the channel, it was only natural to add a new controller to the channel. With CBAS version 2.0.1, the Modbus Generic controller type was added to the channel to handle any and all of these requests. Since then, many different manufacturers' equipment has been programmed using this controller type. Until recently, the channel had a limit of 64 addresses that could be programmed. As of CBAS version 2.2.4, up to 255 addresses can be programmed.

Each Modbus controller on the market has a list of points that are available to be monitored or commanded. You don't have to use all of them, but for each one that you want to monitor, you will need 2 or 3 pieces of information: Modbus Register or Position, Modbus Function, and Modbus Data Type or Range (in the case of Functions 3 or 6, which are analog). This information can be obtained from the equipment manufacturer or dealer, and can often be found on their web site.

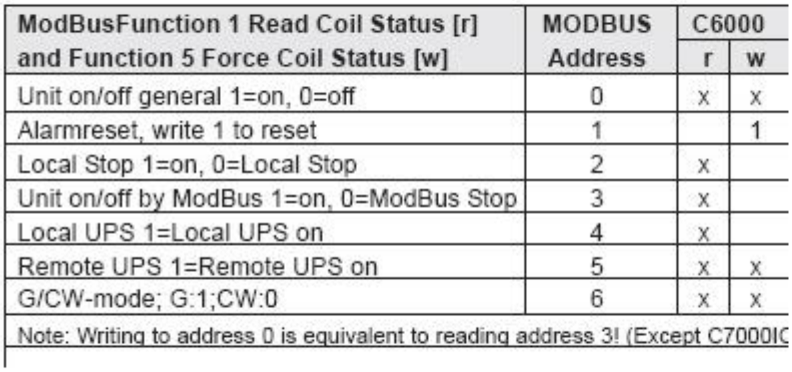
Modbus Registers and Functions

When Modbus was first added to CBAS, the range of Register addresses supported was limited to holding registers in one range. As of CBAS Version 3.1.9, the full range of registers is supported. When addressing points in CBAS on Modbus Generic controllers, it is very important to understand the basics of Modbus registers. There are several "functions" in the Modbus protocol, and they are related to address ranges. Many manufacturers of Modbus RTU protocol equipment do not give the full address, but a partial address and the Modbus Function. So it is up to you to know the address ranges of the Functions that CBAS supports:

Modbus Function 1 Read Coil Status. CBAS Modbus addresses in the range 1-9999. (Binary Inputs)
Modbus Function 2 Read Input Status. CBAS Modbus addresses in the range 10001-19999. (Binary Inputs)
Modbus Function 3 Read Holding Registers. CBAS Modbus addresses in the range 40001-49999. (Analog Inputs)
Modbus Function 4 Read Input Registers. CBAS Modbus addresses in the range 30001-39999. (Analog Inputs)
Modbus Function 5 Force Coil Status. CBAS Modbus addresses in the range 1-9999. (Same as Function 1, but Outputs instead of Inputs)
Modbus Function 6 Preset Single Register. CBAS Modbus addresses in the range 40001-49999. (Same as Function 3, but Outputs instead of Inputs)

Generally speaking, manufacturers' documentation will give a table of addresses in one function, like Function 2, Read Input Status. Sometimes, they will give you the full address, like 10001. Other times, they will give you 1 to 3 digit addresses. In this case, we know that Function 2 involves addresses in the range of 10001 to 19999. Some manufacturers specs force you to add a 1 and others don't. If the addresses given in the tables start with 0, then you would add 10001 to the address. If the addresses in the table start with 1, then you would just add 10000. In all cases, CBAS will actually subtract 1 from the address before sending out the request. It's a little confusing, but if it is not working, try subtracting or adding 1 to the address.

Here's an example:



ModBusFunction 1 Read Coil Status [r] and Function 5 Force Coil Status [w]	MODBUS Address	C6000	
		r	w
Unit on/off general 1=on, 0=off	0	x	x
Alarmreset, write 1 to reset	1		1
Local Stop 1=on, 0=Local Stop	2	x	
Unit on/off by ModBus 1=on, 0=ModBus Stop	3	x	
Local UPS 1=Local UPS on	4	x	
Remote UPS 1=Remote UPS on	5	x	x
G/CW-mode; G:1;CW:0	6	x	x

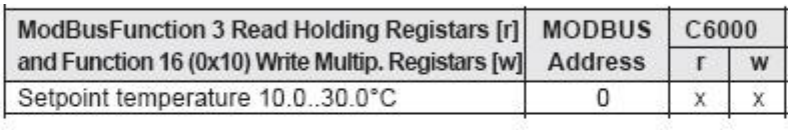
Note: Writing to address 0 is equivalent to reading address 3! (Except C7000IC

In the above table, Functions 1 and 5 are shown. If there is an x in the r (Read) column, it is possible to read the register, which means it is Function 1. Since the addresses start at 0 in this case, the address of the Unit On/Off General point would be programmed as 10001 in CBAS. CBAS will subtract 1 from the request before it goes out. When programming the point, make it a Binary Input, and CBAS will send it out as a Function 1 request.

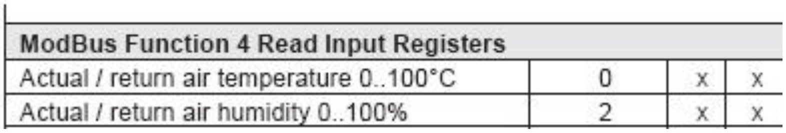
There is an x in the w (Write) column, so it is possible to command the Unit On/Off. That would be function 5 Force Coil Status. The address would be the same, 10001, and you would program it as a Binary Output. CBAS will subtract a 1 and send the request out as a Function 5.

Analog Example:

The tables below show examples of Functions 3, 16, and 4, which are analog functions.



ModBusFunction 3 Read Holding Registars [r] and Function 16 (0x10) Write Multip. Registars [w]	MODBUS Address	C6000	
		r	w
Setpoint temperature 10.0..30.0°C	0	x	x



ModBus Function 4 Read Input Registers			
Actual / return air temperature 0..100°C	0	x	x
Actual / return air humidity 0..100%	2	x	x

The first point, Setpoint Temperature, can be programmed in CBAS as address 40001 and configured as one of the analog point types. (More on those later) For Function 3, program it as an Input. CBAS does not, at this time, support Function 16. If the device supported Function 6, Preset Single Register, you would program it as an Output. Return Air Temperature and Return Air Humidity would be programmed as addresses 30001 and 30003 respectively. Because the addresses fall into the Function 4 range, the request will be sent out as a Function 4. Choose one of the analog point types to match the manufacturer specifications. There is no write function for Function 4 address points.

Programming Modbus in CBAS

Licensing

There are 2 types of Modbus RTU Channels in CBAS: Modbus RTU over TCP/IP and Modbus RTU on Controller. Both types require an add-on entry in the CBAS license file. Purchase the Modbus RTU Protocol add-on, as well as any other add-on, when you purchase CBAS. An add-on feature can also be purchased at a later time, and an updated license file can be emailed.
Note: The license file cannot be altered by the end user.

2 Types of Channels

When the Modbus RTU over TCP/IP channel is programmed, the Host controller merely passes data back and forth between the Modbus device and the CBAS Server. This is fine for monitoring and non-critical commands. There is a “traffic” screen (see Troubleshooting section) available on this channel that is not available on the “Modbus RTU on Controller” channel. Also, the On Scan/ Off Scan buttons work only on this channel type.

When the Modbus RTU on Controller channel is programmed, the database and its programming reside in the database on the Host controller. Program software points on the Host controller and place any logic on those points. This way, if communication between the server and Modbus device is lost or CBAS is in Editor Mode, the sequence of operations continues with the Host controller acting as the CBAS Server. Statuses are still updated on the CBAS server, and the “child” points can be monitored using a Handheld terminal at the Host controller. A controller on this channel can only be taken Off Scan by going to the Program screen of the controller and checking the Off Scan box. Restart CBAS for the change to take effect.

Note: Because the Modbus RTU on Controller is contained in the Host controllers’ database, it can only be accessed in CBAS through the Host controller/Channels. That is, the channel will not be listed in the Channels screen of Hardware View.

2 Types of Controllers

Originally, there was only one type of Modbus Generic which adhered to the standards expressed above, meaning that Holding Registers must have register addresses in the 40,000 range. It was found that manufacturers were now making points that were Holding Registers in any range, possibly starting at 1. In order to accommodate this, New Modbus Generic controller type was added. Now you can program a point at any register and choose what Function it is: Holding Register, Coil etc.

Programming Modbus

To add a Modbus RTU channel, in Hardware View:

- Locate the controller that will be acting as the Modbus host.
- Click the controller and click Channels.
- Click “Add a Channel” next to RS485 Host or RS485 Secondary.
- Give the channel a descriptive name and choose either Modbus RTU over TCP/IP or Modbus RTU on Controller for the configuration.
- Click Add Channel Now.

To add the controllers to the channel:

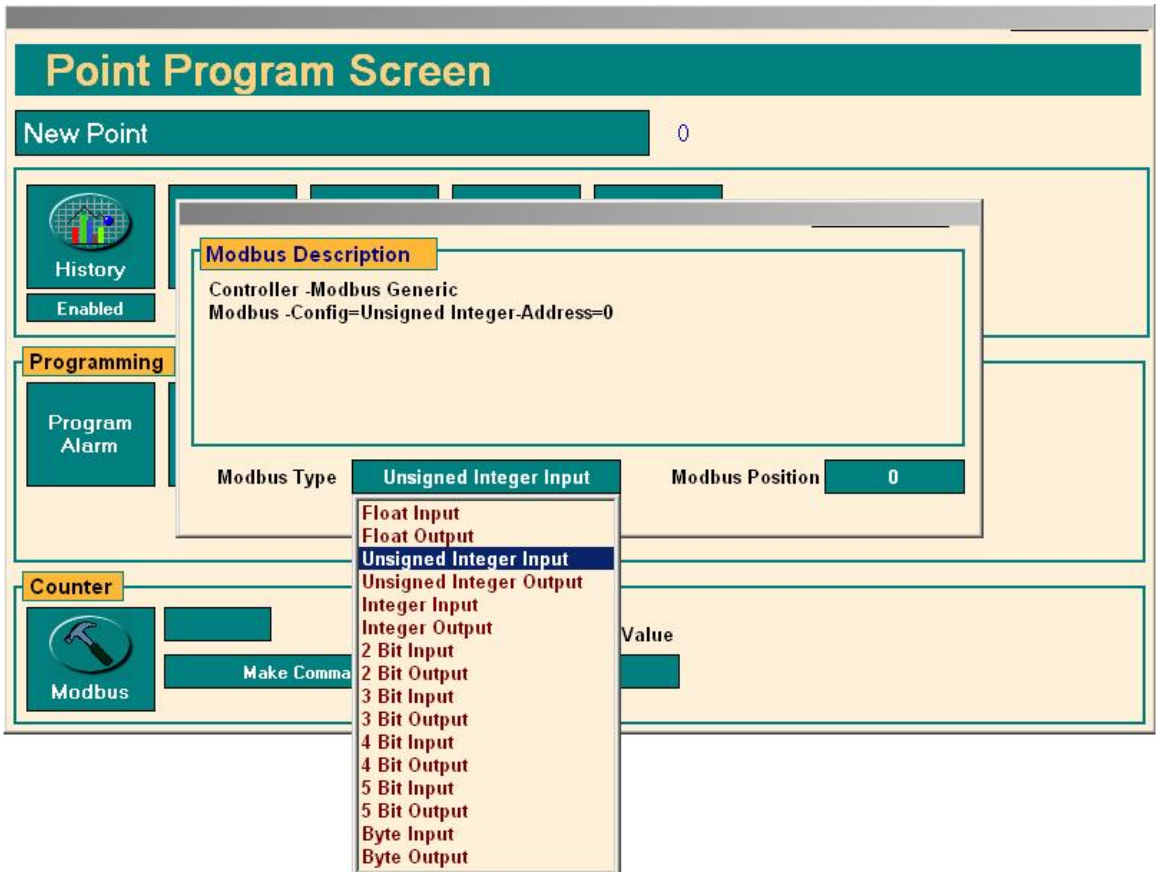
- Right-click twice and locate the Modbus channel you just created. If the channel is Modbus RTU on Controller type, then you must access it by clicking the Host controller, then Channels.
- Click the channel, click Controllers, and locate the line that has the address of the Modbus device.
- Click Add a Modbus Controller on that line and give the controller a descriptive name. Choose the Old Modbus Generic or New ModbusGeneric controller from the list under “Select Configuration”.
- Click Add Controller Now and you are finished.

To add points to the controller:

- Click the controller and click Points. You will see Add a Modbus Point.
- Add the points you want to use by clicking Add a Modbus Point and giving each point a descriptive name. If you have more than one Modbus controller, you will want to include the name of the controller in the point’s name.
- Based on the manufacturer’s point mapping specifications, choose a configuration type from the list. More configuration will be done in the next section.
- Click Add Point Now!

To configure the points:

- Click the point then the Modbus button in the bottom left of the Point Program screen. (See figure below)
- Click the Modbus Position field and enter the address, which may be a number within one of the standard register ranges listed above.
- If you need to change the Modbus Type, click on the field and select from the list.
- If New Modbus Generic, you must select the type of register.
- Right click or ESC to save.
- After changing any Modbus parameters in CBAS you must restart CBAS before the changes will take effect.



Controller -Modbus Generic
Modbus -Config=Unsigned Integer-Address=0

Modbus Type: Unsigned Integer Input

Modbus Position: 0

Modbus "text" files.

What is the easiest way of adding multiple identical Modbus controllers? Computrols controllers, VAV, 8X, 16X, etc, can be saved as templates. So, you program the first controller as completely as possible, then save it as a Template. For the rest of the controllers, you add then from the Template you saved, or you can copy the original controller directly.

With Modbus Generic controllers, you save the controller to a text file, and then add the rest of them using the text file.

To Save a Text file:

- In Editor Mode, click the completed Modbus Generic controller, then click Export Database.
- It will ask you to type in a description. Click OK, then type your description. When finished, right-click or ESC.
- A pop-up box will tell you that the file was saved to C:\CBAS\Bin\Modbus\ModbusGeneric.txt.
- If you will be making more than one text file because you have different configurations, find the above file and change its name to reflect the configuration.

To add the controllers using a text file:

- Click on the channel, click Controllers, and locate the line that has the address of the Modbus device.
- Click Add a Modbus Controller on that line and give the controller a descriptive name. Choose the Generic Modbus controller from the list under "Select Configuration".
- A pop-up box will ask you "Would you like to import the points into the newly added panel?" Click Yes.
- In the next window, select the correct text file and click OPEN.
- Next, you will be asked to "Please enter a prefix for all of the points to be added (max 12 characters)." Click OK, then enter a word that will differentiate the points from others in the database.
- After the progress window shows that the points have been added, another window will state how many points were added.
- Click OK, and you are finished adding the Generic Modbus controller.
- To view the points, click the controller, then Points.

More on the Modbus Generic controller

The Modbus Generic controller is limited to 512 points.

As you can see by the previous figure, there are several different data types supported by the various Modbus RTU Functions. Originally, the Modbus Generic controller was limited to Reading/Writing of Holding registers (Modbus Function 3 and 6). As stated earlier, Functions 1 through 6 are now supported. Holding Registers have a range of roughly 65,000 (65,536, or 256 x 256 to be exact). Some points will take up 2 Holding Registers. A Modbus register is 16 bits (or 1 WORD).

Below is an explanation of each type of Modbus data that is supported by CBAS.

Modbus Functions 3, 4, and 6.

- FLOAT (REAL) - This point type takes up 2 register addresses, which makes it 32 bits long. The first register is the upper 16 bits. Can be an Input or Output depending on the Function/Address.
- UNSIGNED LONG (UINT32) - Uses 2 registers. The first register is the Lower 16 bits. Range 0 - 999,999. Very similar to a Float, and the 2 types are interchangeable in CBAS.
- LONG (SINT32) – To some manufacturers, a signed 32-bit or signed integer 32 (SINT32) is a LONG, that can go negative or positive
- INTEGER (SINT16) - Takes up 1 Modbus register and is 16 bits long. Range -32,000 to 32,000. So it is really a Signed Integer - A signed 16-bit is a Integer in CBAS.
- UNSIGNED INTEGER (UINT16) - Uses 1 register. Range of 0 to 65,000. An “unsigned 16-bit” is an Unsigned Integer in CBAS.
- When it is “Signed”, that means the value can go negative, thus a minus sign.

Newly Added Nov 6 2009

- Scaled Integer (1 register) (SINT16) -32K to 32K
- Scaled Unsigned Integer (1 register) (UINT16) 0 to 65K
- Float (2 registers) 6 significant digits
- Signed Long (2 registers) (SINT32) new -2 billion to 2 billion
- Unsigned Long (2 registers) (UINT32) 0 to 4 billion

Example: Onicon 4200 Series Ultrasonic Flow Meter Page 43
Flow Velocity 1001 Input Register 4 Real 29.165
When it is 4 Bytes (32 bits) and Real, then that is a Float in CBAS.
When it is 4 Bytes (32 bits) and Integer, that is a Long in CBAS.

Scaling

We’ve added a scalable point type to each of the points above. We have seen several Modbus implementations where the point received had to be scaled in order to get the real value of the point. So previously you would have to create a software point and a logic statement to get the real value.

The new scalable point type is stored internally as a floating-point number. That means it stores 6 or 7 significant digits. So the number 123,456,789 would be truncated and stored as 1.23456 x 10 (8). The (8) means to the 8th power. Likewise .00000123456789 would be stored internally as 1.23456 x 10 (-6). So any point with more than 6 digits will lose some precision.

The scalable point type has a value that allows you to scale the received Modbus number by another floating point number. Leave this value at 1.0 to not scale. There is a Decimal Points box that allows you to select how many decimal points to display in the All Points view. There is a Change Value box that allows you to select the smallest amount of change that is required for the point to change status in CBAS.

Until further notice only Modbus channels running in the DPU will support the new point types (Modbus TCPIP and Modbus Interface). The new features are not in the firmware.

Modbus Functions 1, 2, and 5.

- Binary Input, Binary Output - Uses 1 register.
- Bits - Let you control single/multiple bits out of a register.
- A Byte is 8 Bits

Bit Points

Bit points are points that have the same register, but do different things, depending on the Bit you set.
You can monitor 1 bit out of the 16 bits in a single register. You need to add the "1 Bit" point type. Or, up to 16 of them for each address (40001 for example). Then in the configuration for the Modbus point, enter the
1-16 number in the "Start Bit Position".

So, for each Modbus bit address, you would have 1 point each of type "1 Bit"
all with the same Modbus address. The only difference between them all will
be their "Start Bit Position".

Multiple Bit Inputs and Outputs

Some manufacturers use multiple bits in the same Modbus register to represent things. In CBAS there are Modbus bit point types that represent 1-5 bits out of a Modbus register. 2 Bit, 3 Bit Inputs and Outputs, etc, are the same thing as a 1 bit, except that they return or send more than 1 Bit from the register.

If you have a 1 Bit with starting position 3, it will return the value of the
0000000000000x00 bit (value from 0 to 1).
If you have a 2 Bit with starting pos 3, it will return the value of the
000000000000xx00 bits (value from 0 to 3).
If you have a 3 bit with starting pos 3, it will return the value of the
00000000000xxx00 bits (value from 0 to 7).
Byte is the same as an 8 bit start at 3, it will return the value of the
000000xxxxxxxx00 bits (values from 0 to 255).

All of these items, 1 Bit thru Byte return multiple values out of a single register.

Take the following Modbus register for example:
40001 aaaaabbbbcccddef

aaaaa is Status 1
bbbb is Status 2
ccc is Status 3
dd is Status 4
e is Status 5
f is Status 6

So there are 6 status points in the 1 Modbus register. So, you would program the following 6 points in CBAS to see the status of all 6 points:

Modbus Type	Start Bit Position	Modbus Position	
5 Bit Input	11	40001	Status 1
4 Bit Input	7	40001	Status 2
3 Bit Input	4	40001	Status 3
2 Bit Input	2	40001	Status 4
1 Bit Input	1	40001	Status 5
1 Bit Input	0	40001	Status 6

The Modbus Type is how many bits you want to pick out of the register (1 to 5). Usually, if there are more than 5 bits, the manufacturer will use an entire Byte (8 bits).

The Start Bit is how many bits into the 16 bit Modbus Register to start. 0 = Start from beginning, 15 = Only look at the very last bit.

Other Considerations

Once a Modbus Generic controller is added to the database, it is possible to change the address or remove the controller completely.

Changing the address

- Click the controller, then Program.
- In the bottom right of the Program screen, click the address and change it.
- ESC to close, then ESC again to close the channel.
- Click the Channel, then Controllers to reopen the channel. You will see the controller at the new address.

Removing a controller

- In Editor Mode, click the Database Menu, then Remove a Controller.
- Choose the Channel where the controller is located.
- Select the controller to be removed.
- Click Yes if you are sure you want to remove it.
- You will be returned to the controller list for the channel, after the progress window closes.
- Choose another controller to remove, or ESC to finish removing controllers.

Baud Rate

Manufacturers of Modbus RTU devices use different baud rates, or communication speeds. It is possible to change the baud rate of a channel. All controllers on the channel must be capable of communicating at the same speed. Each host controller has 2 channels available. So, if you have devices that use different baud rates that can't be changed, just add another channel. The default for CBAS is 9600 baud. Other rates available are 19.2K baud and 38.4K baud.

Changing the baud rate:

- Click the Modbus Generic controller, then click Program.
- In the Channel Parameters section, click the Baud rate and select one of the 2 other choices.
- ESC to close the Channel program screen.
- If you are in Real Mode, close CBAS and reopen for the change to take effect.

MODBUS TCP/IP

Modbus TCP/IP is now available in CBAS. Of course there is no Host Controller needed, because the channel is actually the LAN and emanates from the network card on the CBAS DPU. You must add a MODBUS TCP/IP Channel to the Channels screen in Hardware View.
In Editor Mode, go to Hardware View and click “Add a TCP/IP Channel”.
Give the channel a descriptive name, then under Select Configuration, choose MODBUS TCP/IP.
Click “Add Point Now”
From there, everything is the same as adding a Generic Modbus Controller and Points to a Modbus RTU over TCP/IP Channel.

Troubleshooting

Channel Not Started

When you receive a yellow trouble banner, this means that a channel is programmed, but not added to the license file. You must have an add-on feature added to the licenseX.txt file, or the add-on channel will not work. You only get this alarm when CBAS first starts in Real Mode, and it only applies to add-on protocol channels, not Basnet or OPTO-22 on Controller.

On Scan/Off Scan

The On Scan/Off Scan buttons work the same on a Modbus RTU over TCP/IP controller as on a BASnet controller. However, in the case of a Modbus RTU on Controller channel, you must go to the Program screen of the controller and check the Off Scan box to take it Off Scan. You must restart CBAS for the change to take effect. To put it back On Scan, uncheck the box, then restart CBAS.

Viewing Traffic on the Channel

In CBAS version 7.1.9, after 9/24/07, a "show traffic" button was added to the program screen of the Modbus RTU over TCP/IP Channel (Modbus Interface). It is NOT available on the Modbus on Controller channel. It is similar to the traffic screen on a Basnet Channel, and will tell you if the Modbus device is responding back to CBAS. (see figure below)

It will also show you the NACK code if the Modbus controller is returning a negative acknowledgement. A NACK will be returned for several reasons, like invalid address specified, unsupported command, etc.

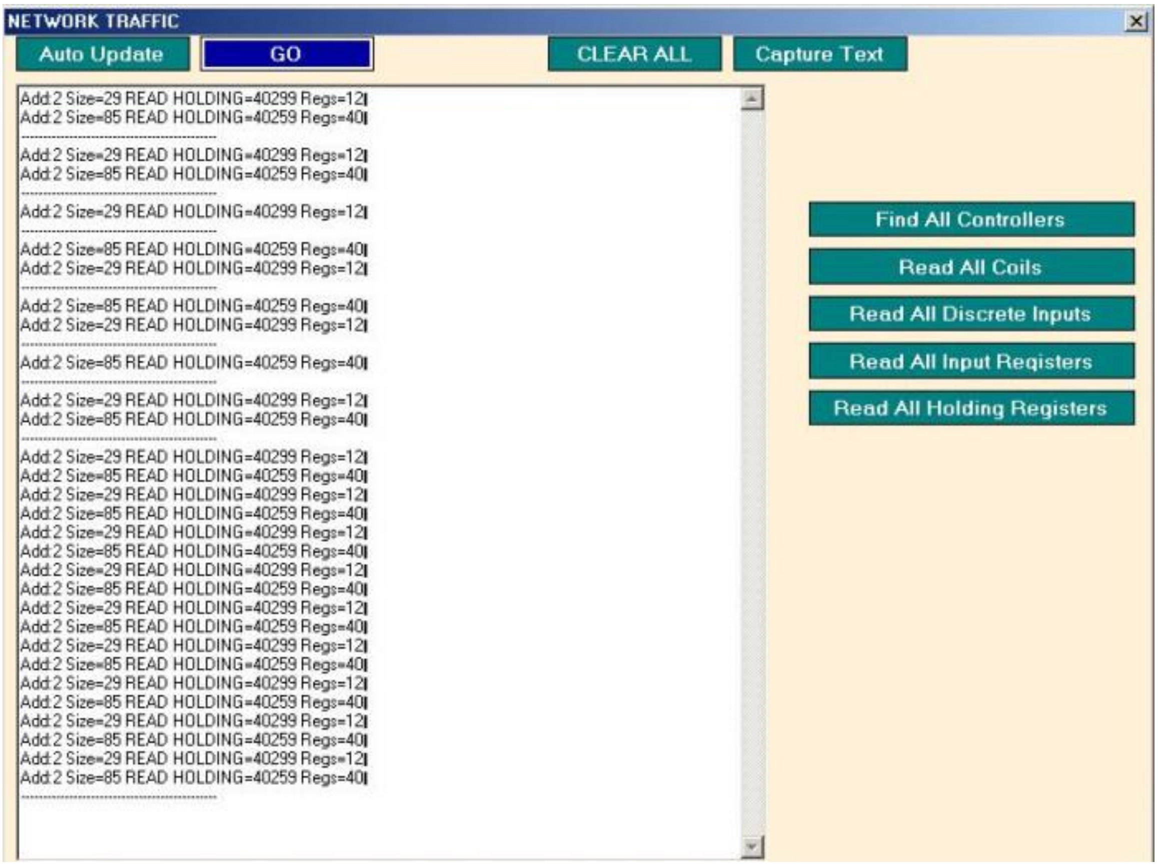
It will also show you the poll messages to the Modbus devices.

Below implemented 9/26/07.
Now, there is a 'Find All Controllers' button that will go out and send a 'REPORT_SLAVE_ID' message to all serial addresses from 1 to 250. The response will show up in the Traffic window.

Also there are buttons to 'Read All Coils', 'Read All Discrete Inputs', 'Read All Input Registers', 'Read All Holding Registers'. When you press these buttons first it will ask you for the Modbus controller address that you want to read from. Then it will read the first 1000 points for the section that you picked. If you press the button again it will read the next 1000 points. And so on. The response will be displayed in the Traffic window.

There is also a STOP button that you can press at any time to cancel one of the aforementioned scans.

There is also a 'Capture Text' button. When you press this button it will save everything that is displayed in the Traffic window to a text file. The text file will be "C:\CBAS\Data\Capturexxx.txt. Where xxx is the next unused number from 0 to 255.



```
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
--------------------------------------------
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
--------------------------------------------
Add:2 Size=29 READ HOLDING=40299 Regs=12
--------------------------------------------
Add:2 Size=85 READ HOLDING=40259 Regs=40
Add:2 Size=29 READ HOLDING=40299 Regs=12
--------------------------------------------
Add:2 Size=85 READ HOLDING=40259 Regs=40
Add:2 Size=29 READ HOLDING=40299 Regs=12
--------------------------------------------
Add:2 Size=85 READ HOLDING=40259 Regs=40
--------------------------------------------
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
--------------------------------------------
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
Add:2 Size=29 READ HOLDING=40299 Regs=12
Add:2 Size=85 READ HOLDING=40259 Regs=40
--------------------------------------------
```

```
READ HOLDING=40259 Regs=40
```

This line in the capture tells you that CBAS is reading registers 40259 thru 40299, all in one message. That means that you have points programmed on all of the addresses in between. When CBAS is polling, it checks for continuous addresses and will read all those registers in 1 message. This improves the scan rate.

Explanation Button
In the Modbus address program screen there is also a new "Explanation" button. This will open a box that gives a brief description of the Modbus protocol as implemented in CBAS.