



AN-086
Protégé System
Modbus Remote Configuration

PUBLICATION INFORMATION

This application note covers the use of the protégé system in a particular configuration and may be used in conjunction with other publications. Specifications may change without notice, for updates visit the Integrated Control Technology website at www.integratedcontroltechnology.com.

CONTENTS

Contents	1
Protégé System	2
Introduction	2
Application Note Information	2
Software Application Version	2
Third Party Software Applications.....	2
General.....	3
Introduction	3
Project Files	3
Getting Started.....	3
Controlling Areas	8
Introduction	8
Area Follows PGM Status	8
Testing Your Programming	10
Testing From Modbus	10
Area Arm On Pulse PGM.....	12
Area Disarm On Pulse PGM	13
Controlling Automation.....	14
Introduction	14
Controlling Automation From Modbus	14
Bridging Modbus and Other Protocols	15
Information	16
Introduction	16
Contact.....	16

PROTÉGÉ SYSTEM

Introduction

The Protégé System is a powerful integrated alarm and access control management system designed to provide integration with building automation, apartment complex control and HVAC in one flexible package. Communicating through a proprietary high speed protocol across an encrypted local area network and AES Encrypted Proprietary RS-485 module network using modular-based hardware design, system installers have the flexibility to accommodate any installation from small or large, residential or commercial.

Application Note Information

This application note explains the operation of the Protege System Controller when communicating with a ModBUS Master Device. This application note details the connectivity and should be read in conjunction with Application Note AN023 Protege Controller ModBUS Slave configuration. This details the registers that should be used.

The Protege system can be configured to operate using a number of Protocols and Port Ranges providing many variations to the information provided here. Where possible the author has attempted to keep the IP and Port configurations as default however in most cases these are changed to suit the networking environment the devices are installed on.

Software Application Version

This application note has been tested with version 2.08 Build 3020 or higher of the controller firmware although the configuration parameters are relevant to most previous versions. The Protégé System Management Suite User Interface and Server should be Release 430 or higher to take advantage of the configuration parameters.

Third Party Software Applications

This application note uses the Topserver Modbus to OPC application from Software Toolbox which can be obtained from the vendors website www.softwaretoolbox.com you can download the demo application from <http://www.toolboxopc.com/Features/Demo/demo.shtml> and it is recommended to only install the Modbus and BacNET components.

GENERAL

Introduction

The following shows how to connect to the Protege System Controller to the Topserver application and assumes that you have already read and configured your controller service from the information in application note 23. Please make sure that once you have programmed the service that you start it.

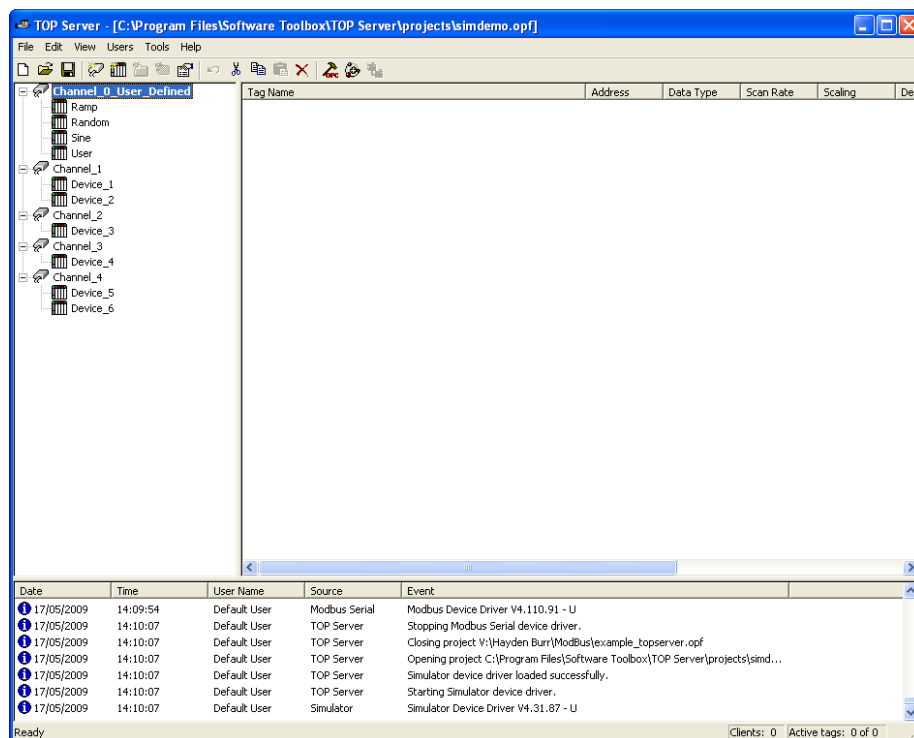
Project Files

In this application note we make extensive use of the Top Server Project Files and associated CSV tag information, this will allow the Security or Automation Professional to quickly get an operating system.

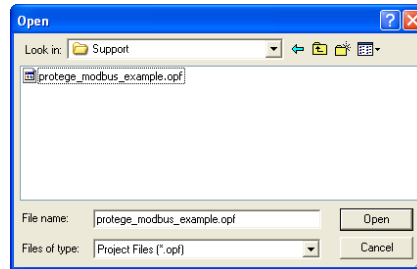
? The project files are specific to the Topserver Application however the tag databases included with the application note are available independently and can be imported in to any suitable application, they are provided in CSV format.

Getting Started

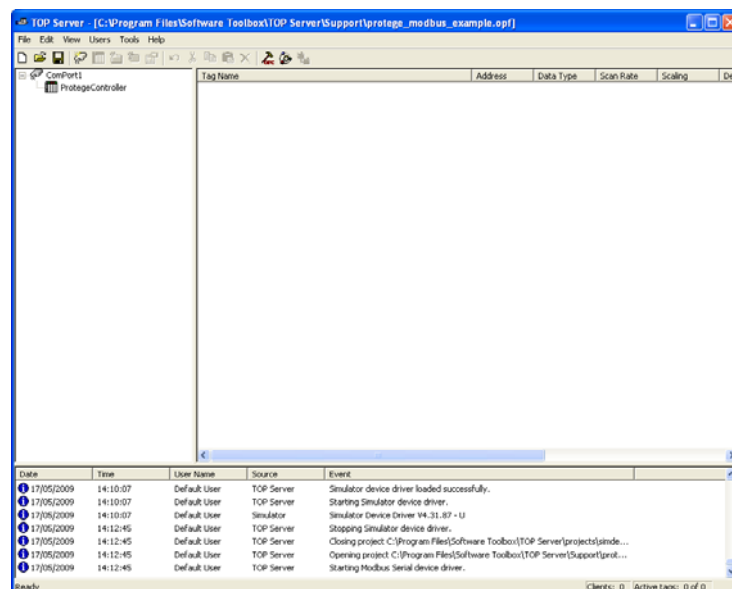
Now it is time to get connected with Top Server, make sure that you have started the Modbus service and it is operating as per Application Note 23. Once you have confirmed this follow the steps below ensuring your screen shows a similar set of information.



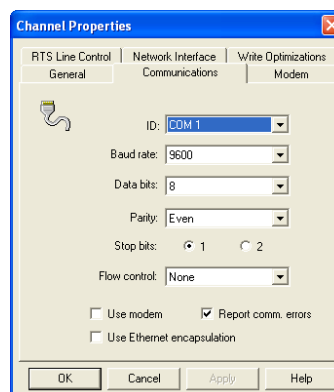
1. Start the topserver application by browsing to the location from your start menu and selecting the topserver application. If this is the first time you have started topserver then it will show by default the sample application project.



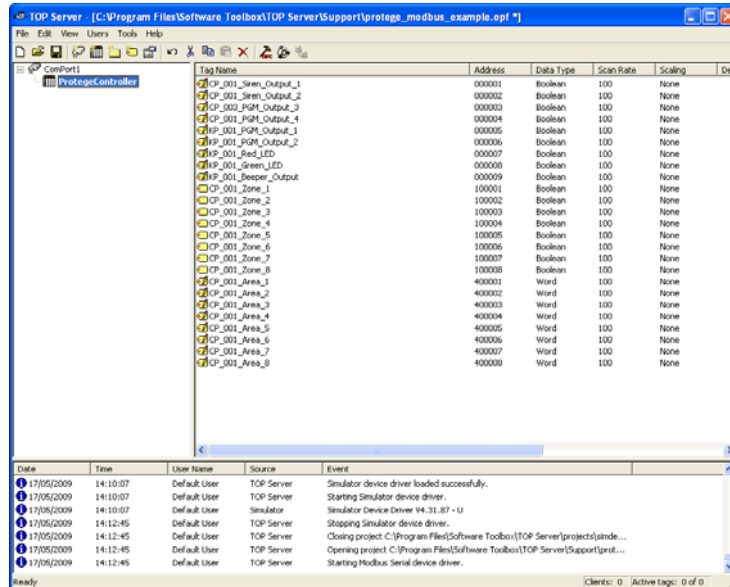
2. Select file from the menu and open, browse to the location that you saved the example topserver project that you downloaded with the application.
This will be a zip file with the same name as this application note's PDF.
3. Once you have found the file select the open button.



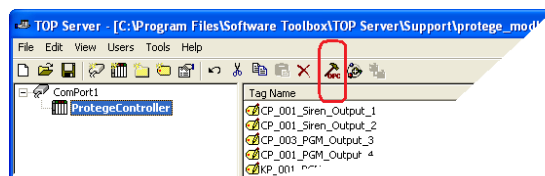
4. You should see the window above with the '*CommPort1*' and '*ProtegeController*' entries in the browsing window.



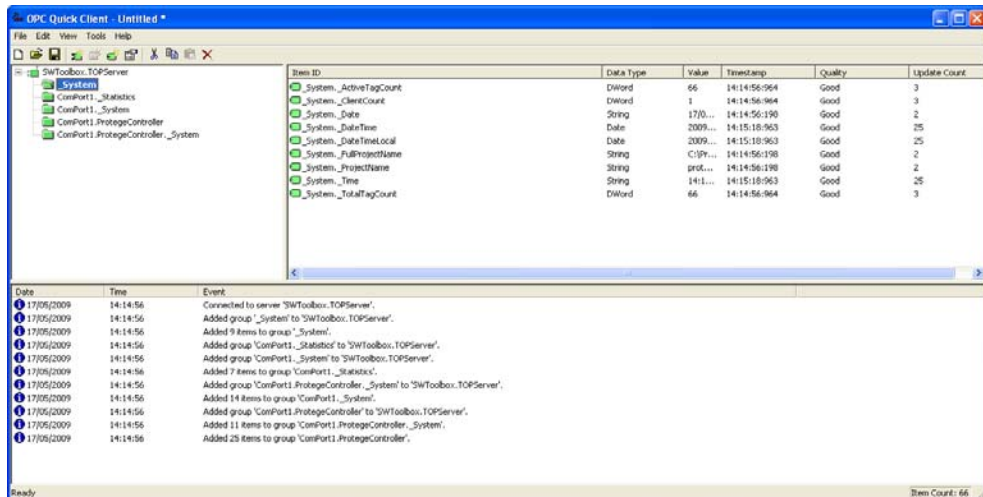
5. In many cases your communications port may not be the same port that is used in the sample application and in this case you can double click the ComPort1 Entry in the browsing window and change the communication port settings. If you check you will see the settings here match the settings that are configured for the ModBUS service, this allows your Protege System Controller and the Modbus driver to communicate successfully. Once your changes are made select the OK button.



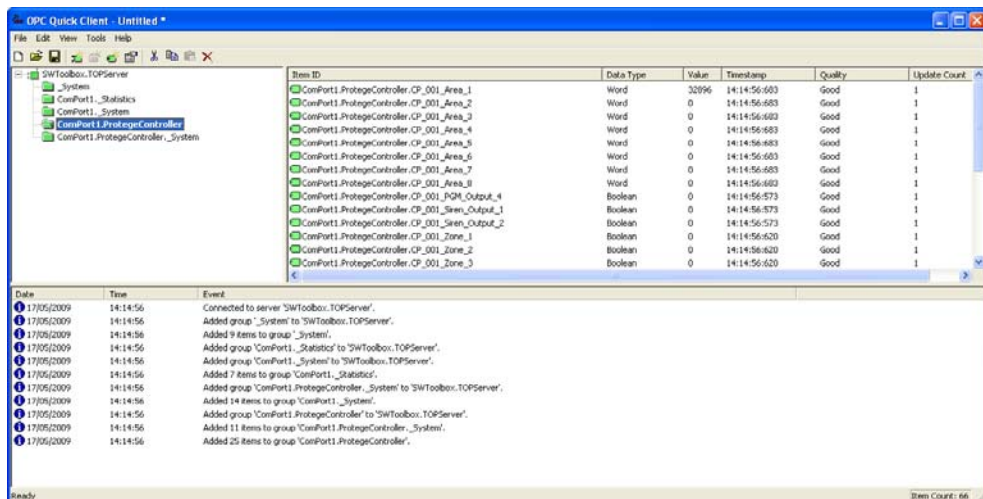
6. If you select the ProtegeController entry from the browsing window you will then see a list of tags shown in the tag window. You should see a similar list to the above however the project may contain more or less depending on which example you opened, in this case these are the tags for the Protege_Modbus_Example.opf project file. If you want to change or add the tags we recommend that you save the project as another name so you always have a reference project.



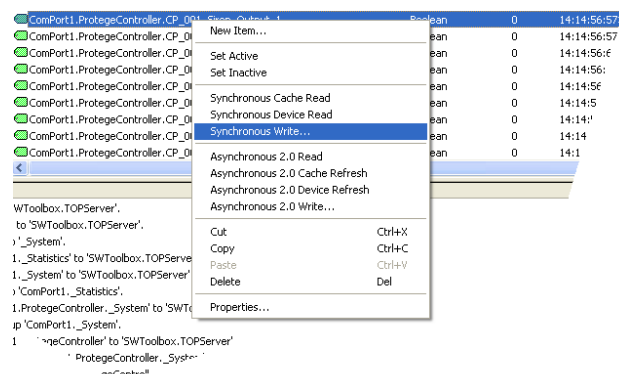
7. At this point you are now ready to communicate with the Protege Controller. Make sure that your Serial Cable is connected to the PRT-COMM on the Protege System Controller and your service is started with the correct parameters as per Application Note 23. When you click on the OPC quick connect button as shown (Circle Above) the system will start the ModBUS communication engine in the Topserver application, you will see the RX and TX lights flash rapidly as the communication takes place.



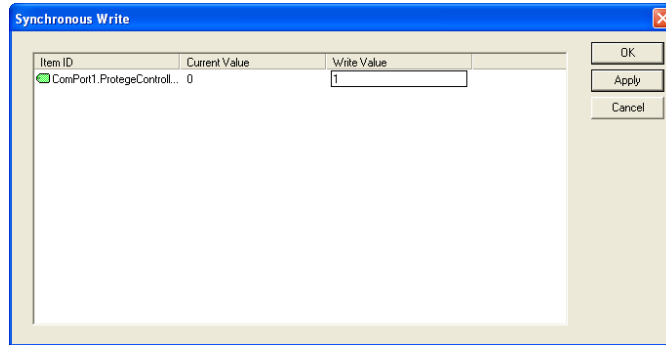
8. When you have selected the OPC button another window will open and show you the communication status. This can be seen to the right in the above window.



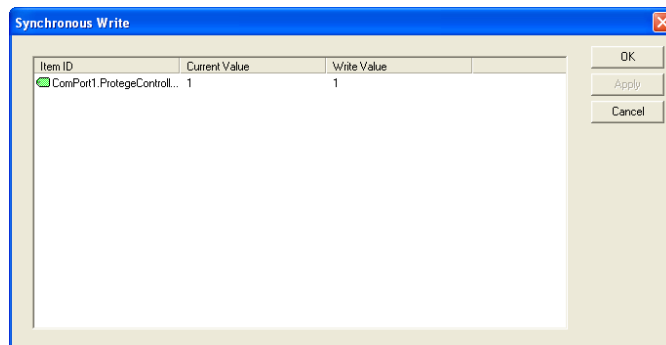
9. We are more interested in the actual values and the ability to change the values over the Modbus interface and to do this we need to select the 'ComPort1.ProtegeController' entry. This will show the available tags in the system. A tag is the configuration of a Modbus register or point that links to an object (Zone, PGM etc...) in the Protege Controller.



10. You are now able to control the items that can be written (tags are automatically updated so you do not need to specifically read a register). To control an output, for example we are going to control the first PGM Output on the controller (CP001:01). Right click on the tag in the item window that is labelled *'ComPort1.ProtegeController.CP001_Siren_Output_1'* this will provide you a menu and select synchronous write from the context menu. This will open the next window as shown below.



11. Once this is open you can select the Write Value location and put a one. You can then select the apply button which will 'Write' this value to the modbus register in the Protege System Controller which will in turn send this to the correct output.



12. As you can see in the next window when you write to the output the current value changes this is because the system reads back the value so you are able to see the status.

In the examples that follow extensive use of the synchronous write function is used to exploit the programming in the examples allowing the Modbus solution to control the arming and disarming of an area so it follows a point and so that there is a specific arm and specific disarm allowing the area to be jointly controlled by multiple locations.

CONTROLLING AREAS

Introduction

The following programming shows how a Modbus application can control the arming and disarming of an area so it follows the status of a coil (Modbus term for a single output). In many cases the follow function is not viable because an area may need to be controlled from another source (keypad, touch screen or reader). It may also be required that there is an arm coil and a disarm coil so that if either of these are activated that the area will arm or disarm but will also allow the area to be controlled by a user on a keypad, touch screen or card reader.

This section will cover programming from the Protege System Management Suite only and will not show keypad programming.

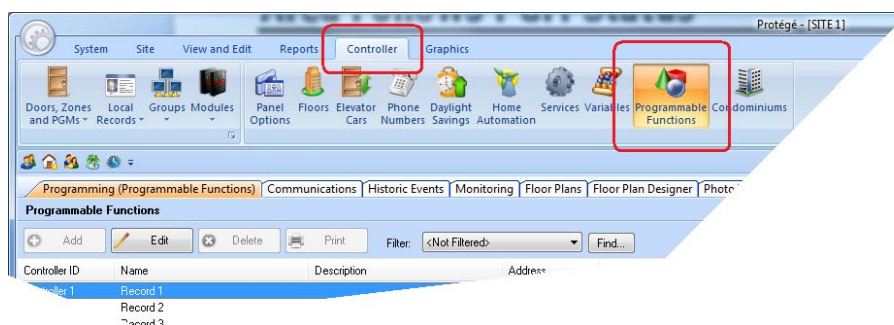
The PGM's used for the following exercise will be AE001:01 and AE001:02 these are commonly spare unless analog expanders are used.

Programmable functions are used extensively in the following examples to link the Modbus PGM being controlled to the resulting activation in the area. It is recommended that the section in the Protege Controller Reference Manual and the Online help system is reviewed in detail.

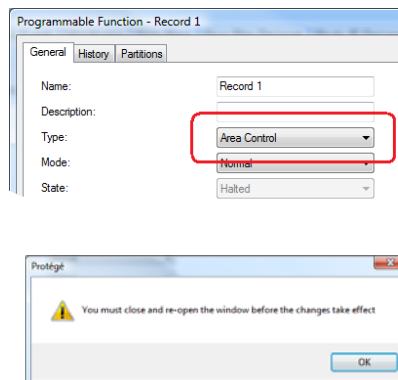
Area Follows PGM Status

When an area is used in a programmable function in the follow state the area will 'Follow' the PGM it is linked to, this will mean that any changes in the PGM will be mimicked by the area. Any changes to the area status (Disarm or Arm from a keypad) will be effective until the programmable function checks the state of the area against the PGM (approximately 30 seconds). In many cases this is not the desired action. If external control is also required please see the area pulse on arming and area pulse on disarming. This allows the area to be controlled by two PGMs each having a specific function (Arming/Disarming).

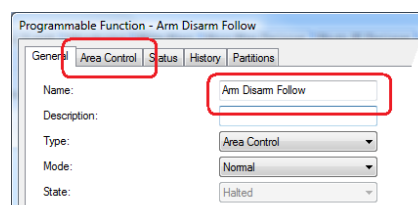
Open the Protege System Management Suite. Select the communications tab and connect to your controller.



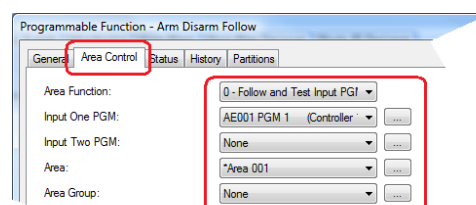
1. From the controller tab select the programmable functions. In this example we will use programmable function 1 for the follow arming to arm area 1 and programmable function 2 and 3 for the pulse on/off arming of area 2. DO NOT USE AREA 1 for both programmable functions as they will not work and the programmable functions will interfere with the area control.



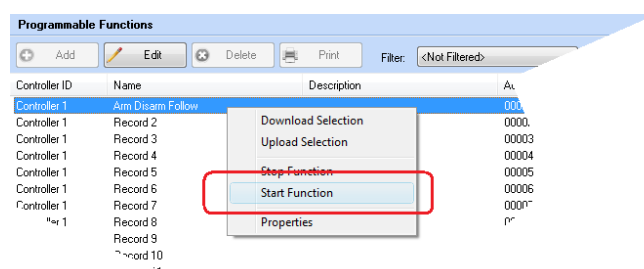
2. Select the first programmable function. Change the type to 'Area Control' and select the OK button, you will be prompted by the software to close the screen, this allows the system to redraw the configuration settings. Once this is completed open the programmable function again.



3. When you open the function again you will see the Area Control tab is now available. Before selecting the tab complete the name with a descriptive name of the programmable functions action.



4. Select the Area Control tab as shown and complete the configuration parameters above ensuring that Input Two PGM and Area Group are set to NONE. Once you have completed the programming select the OK button.



5. Once the programming for the function is completed you need to start the function. Starting a function allows the PLC internal to the controller to run the programmed settings, much like running a program, once it is run it will remain running until it is stopped and will automatically be started if it was stopped when the controller is restarted for any reason.

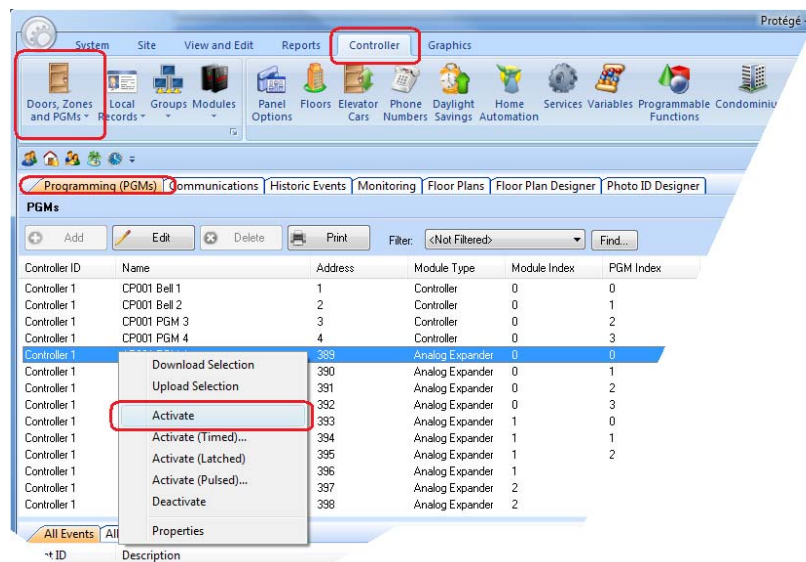
Start the Programmable Function by right clicking the function from the list and select start function.



Once you have started a programmable function YOU CAN NOT edit the settings in the controller and must STOP the function download any changes and then restart the function.

Testing Your Programming

Once you have your programmable function running successfully you will be able to test the operation very easily by controlling the PGM you have used (AE001:01) for the Input PGM One in the programmable function.



Select the controller tab and then PGMs from the Doors, Zones and PGM's icon. From the PGM programming list right click on the AE001:01 PGM and select activate. You will see that your area will begin to arm a short time later. Allow the area to complete the arming cycle and then select deactivate from the menu and you will observe the area begin a disarm cycle.



It is recommended that when you are testing to select the Log Events option in the PGM configuration that will allow the activation and deactivation of a PGM to be presented in the event window.

Testing From Modbus

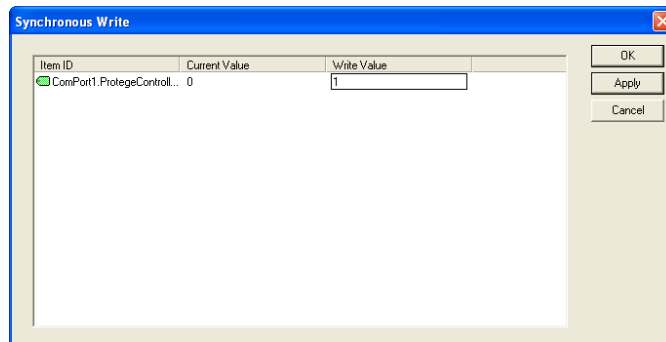
Once you have completed the internal tests on the PGM as shown previous you can now link this to the Modbus interface that you programmed previously.

Complete the steps in the Modbus Section up to Step 8 (Page 6) for controlling a device on the protégé controller. Then when presented with the tags select the tag CommPort1.ProtegeController.AE001_PGM_Output_1 and select synchronous write. Follow the next steps to control the PGM.

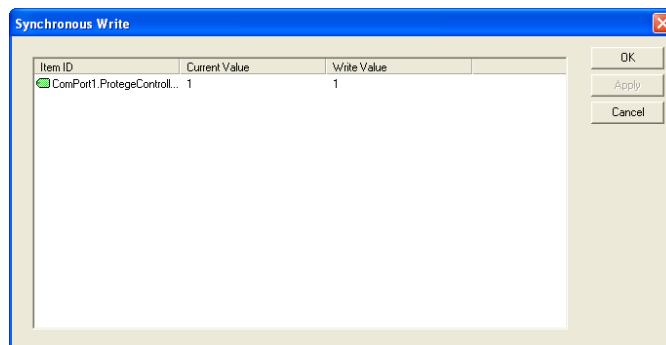


It is recommended that when you are testing to select the Log Events option in the PGM configuration you will see the PGM state change with the appropriate event log being generated.

1. To control the output AE001:01. Right click on the tag in the item window that is labelled 'ComPort1.ProtegeController. AE001_PGM_Output_1' this will provide you a menu and select synchronous write from the context menu. This will open the next window as shown below.



2. Once this is open you can select the Write Value location and put a one. You can then select the apply button which will 'Write' this value to the Modbus register in the Protege System Controller which will in turn send this to the PGM output. You should now also see your area change state from Disarmed to Armed.



3. As you can see in the next window when you write to the output the current value changes this is because the system reads back the value so you are able to see the status.

You should now be able to control the output that controls the area using the programmable function allowing any Modbus enabled device to control the area.

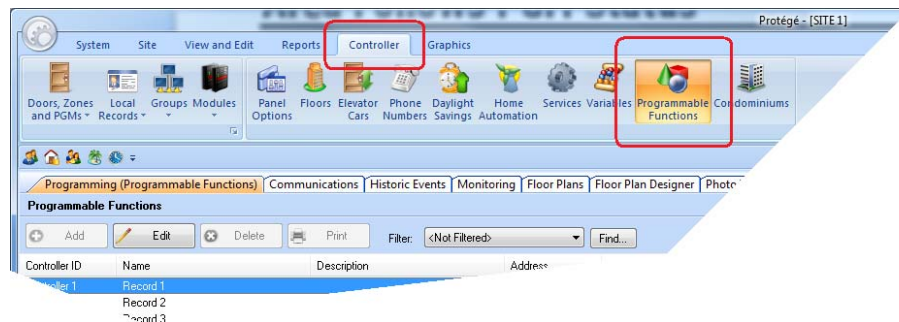


You can obtain the status of the area by reading the area register. This will present the information on the area's arming progress and can be used to show the current status of the area. It is not likely needed in the follow application as the area will be either on or off as a result of the PGM.

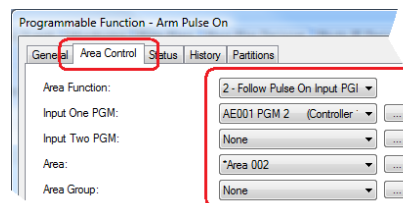
Area Arm On Pulse PGM

The following two programmable function assignments allow you to control the area from two separate PGM's. One used to arm the Area and the other used to disarm the Area. Activating either one when it is in the resulting state will yield no change in the area status. This level of control is ideal for when an area needs to be controlled from the Modbus system using an operator panel.

Open the Protege System Management Suite. Select the communications tab and connect to your controller.



1. From the controller tab select the programmable functions. In this example we will use programmable function 2 for the pulse on arming to arm area 2 and programmable function 3 for the pulse on disarming. DO NOT USE AREA 2 for both programmable functions as they will not work and the programmable functions will interfere with the area control.



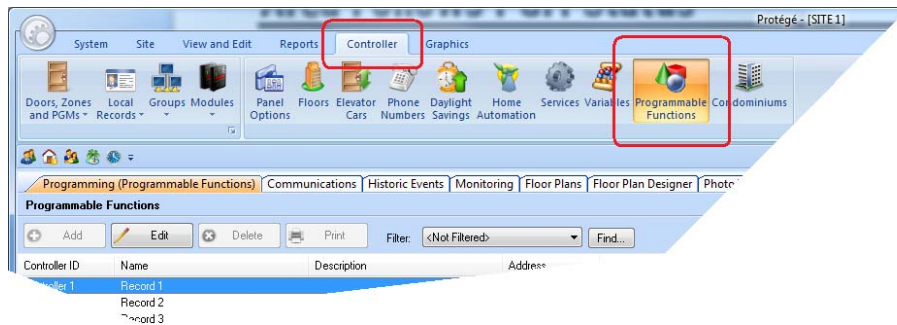
2. Follow the steps to add a new function as shown in the previous section. In this example we have named this Arm Pulse On. Complete the entries above paying special attention to the function which is Follow Pulse On Input PGM meaning that the area will arm (turn on) when the PGM turns on.
3. Once programmed select ok and start the function as shown in the previous example.
4. Now before you test your application open the PGM output AE001:02 and change the activation time to 3 seconds. This will mean that when you arm the area the pgm will only turn on for this period of time allowing you to turn it on again to re-activate an arming cycle.



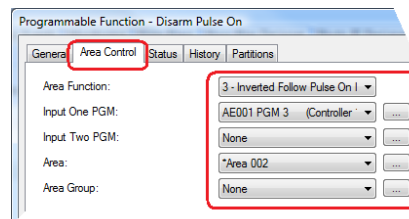
Now test your configuration by activating the PGM using the timed option from the menu for 3 seconds. If your programming is correct the area will arm however when the PGM deactivates will remain armed.

Area Disarm On Pulse PGM

The disarm programming is once again another programmable function however the operation is inverted. Instead of arming the area the function needs to disarm the area when the PGM is activated.



1. From the controller tab select the programmable functions. In this example we will use programmable function 3 for the pulse on disarming to disarm area 2. Area 2 should only be assigned to programmable function 2 and 3.



2. Follow the steps to add a new function as shown in the previous section. In this example we have named this Disarm Pulse On. Complete the entries above paying special attention to the function which is Inverted Follow Pulse On Input PGM meaning that the area will arm (turn on) when the PGM turns on.
3. Once programmed select ok and start the function as shown in the previous example.
4. Now before you test your application open the PGM output AE001:03 and change the activation time to 3 seconds. This will mean that when you arm the area the PGM will only turn on for this period of time allowing you to turn it on again to re-activate an arming cycle.



Now test your configuration by activating the PGM using the timed option from the menu for 3 seconds. If your programming is correct the area will Disarm however when the PGM deactivates will remain Disarmed. Ensure the area is already armed.

CONTROLLING AUTOMATION

Introduction

Automation points can be controlled from the Modbus interface and are available in the example projects and tag databases. The following steps should be taken to test the automation point.



An automation point can be further linked to an external lighting control system such as C-Bus, Lutron, AMX and other systems. If you activate a automation point that is not connected to a PGM it will turn on but no associated PGM will activate.

You can view an automation points state from the keypad by enabling offline automation view under the keypad options. Now you will be able to view the state of the automation points as you control them from Modbus regardless if the automation point is linked to a PGM output.

Controlling Automation From Modbus

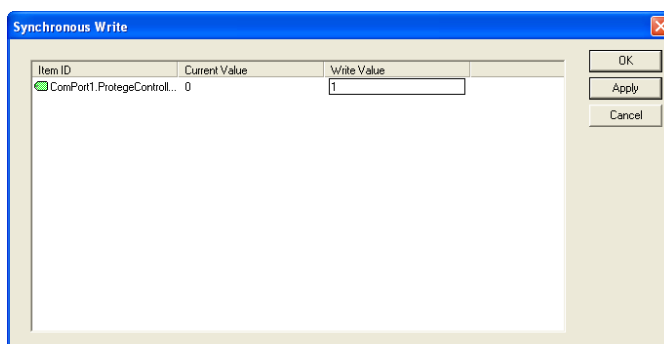
From the example application a single tag has been added for automation point 1. In the protégé system assign PGM AE001:04 to this automation point. This will allow you to see the activation and deactivation. Enable event logging for the PGM to see the information in the event logs.

Complete the steps in the Modbus Section (Page 6) up to step 8 for controlling a device on the protégé controller. Then when presented with the tags select the tag CommPort1.ProtegeController.AE001_PGM_Output_1 and select synchronous write. Follow the next steps to control the PGM.

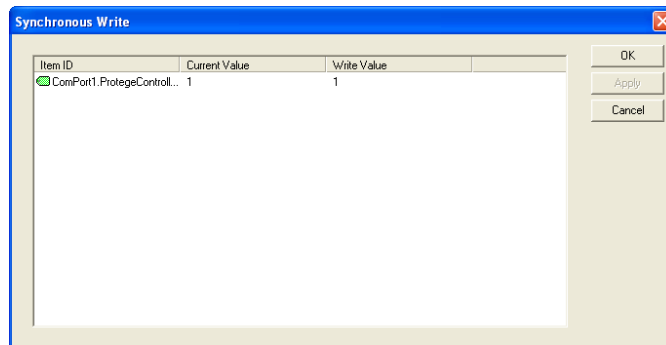


It is recommended that when you are testing to select the Log Events option in the PGM configuration you will see the PGM state change with the automation point change.

1. To control the automation point linked to AE001:04. Right click on the tag in the item window that is labelled 'ComPort1.ProtegeController. AT001_Automation_Point_1' this will provide you a menu and select synchronous write from the context menu. This will open the next window as shown below.



2. Once this is open you can select the Write Value location and put a one. You can then select the apply button which will 'Write' this value to the Modbus register in the Protege System Controller which will in turn send this to the automation point.



3. As you can see in the next window when you write to the output the current value changes this is because the system reads back the value so you are able to see the status of the item.

You should now be able to control the automation point and any subsequent links to C-Bus, Lutron or other third party applications. This process is called bridging.

Bridging Modbus and Other Protocols

The Protege System Controller will allow you to bridge other communication protocols to Modbus that are supported by the Protégé System Controller. For example if you wanted to control a C-Bus output from Modbus you could create a bridge using the Automation Points and then control the associated PGM which would result in the C-Bus point activating.

INFORMATION

Introduction

Application notes are provided as is without any implied warranty or intended use and are written by Integrated Control Technology as an aid in programming and connecting devices to the Protégé System.

Contact

Integrated Control Technology welcomes all feedback. Please visit our website or use the information below.

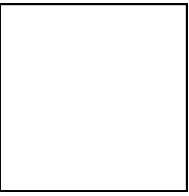
Integrated Control Technology

P.O. Box 302-340
North Harbour Post Centre
Auckland
New Zealand

11 Canaveral Drive
Albany
North Shore City
Auckland
New Zealand

Phone: +64-9-476-7124
Fax: +64-9-476-7128

www.integratedcontroltchnology.com



11 Canaveral Drive, Albany, North Shore City, P.O. Box 302-340
North Harbour, Auckland, New Zealand.
Phone: +64 (9) 476 7124 • Fax: +64 (9) 476 7128
www.integratedcontroltechnology.com