SEMAPHORE
a CSE-Global Group Company

TECHNICAL BRIEF

Configuring Semaphore's Kingfisher PLUS+ RTU as a DNP3 Slave

INTRODUCTION

This technical brief explains in detail the steps involved in configuring the Kingfisher CP-30 and G30 RTUs as a DNP3 Slave device. This document is intended to be used in union with the product's hardware and software manuals. An assumption has been made that the person reading this document has an understanding of the Toolbox PLUS+ software.

Before we configure the RTU as a DNP3 Slave, a quick explanation on DNP3 protocol and the main features has been provided below. DNP3 or Distributed Network Protocol (Level 3) is an open protocol and was created by Westronic (now GE Harris) in 1990. It was primarily developed for the electrical power industry. However, the protocol has gained wide acceptance in other industries from 1993 when the first version of the protocol document was publicly released and DNP3 user groups were setup.

The main features of the DNP3 protocol are as follows:

- Open protocol — a system may contain devices from multiple manufacturers and will be able to communicate with a single top end DNP3 master
- It is an object oriented protocol rather than register based
- Supports poll only or exception based systems and on different communications media
- Supports four (0, 1,2 and 3) different classes of data — allows for easy categorization of the different types of objects (AI, BI, AO, BO, Counters (frozen counters))
- Time stamped protocol
- Access to diagnostic information such as I/O point online or offline for all object types
- Supports peer-to-peer communications, secure authentication for control functions and File transfer

Kingfisher's DNP3 protocol implementation is among the strongest in the industry. The DNP3 protocol is supported on processor (CP30, G30) and communication (MC30/MC-31) modules. The protocol is also supported on different media types — RS232, 485, TCP/IP, Fiber, Radio, PSTN, Satellite, GSM/GPRS to name a few. Subsets of complete, DNP3 functionality are allowed and are defined in various levels. The Kingfisher implementation is higher than level 3 and supports both DNP3 Master and DNP3 Slave functionality in the RTU and includes capabilities such as secure authentication and file transfer. DNP3 Master and Slave device profile documents, which include complete implementation tables detailing all of the features supported, are available from Semaphore.

GETTING STARTED

- Ensure Kingfisher Toolbox PLUS+ version 3.8.5 or newer is installed
- Ensure ISaGRAF version 5.13.309 is installed

- Ensure the ISaGRAF license key in connected to the PC's USB port*
- Ensure firmware version 2540 or newer is installed on the CP-30 or G30 processors
- Ensure firmware version 272 or higher is installed on the MC-30 modules (if being used for DNP3 protocol communications)*

The main steps involved in configuring the Kingfisher PLUS+ RTU as a DNP3 Slave is as follows:

1. Ensure Toolbox PLUS+ is successfully installed
2. Create a Project, Group (not always required) and a RTU Configuration in Toolbox PLUS+
3. Add I/O modules as appropriate to the RTU Configuration
4. Ensure DNP3 protocol is added to the configuration
5. Edit the settings of the DNP3 protocol to ensure it is compatible with the top end
6. Ensure the port is configured correctly and that DNP3 protocol is selected on the port
7. Create the required DNP3 variables from the Dictionary folder in Toolbox PLUS+
8. Edit the properties of the DNP3 variables to ensure settings like the class and variation is set correctly
9. Map physical I/O points to DNP3 variables
10. Save, compile, and download logic into the RTU and test. The test should be to change the status or value of the I/O points and confirm that appropriate logs have been created. Logs can be uploaded into Toolbox PLUS+ for viewing and verification. As an alternative, a DNP3 master device can be configured to poll the Kingfisher PLUS+ DNP3 Slave RTU and confirm the behaviour.
11. Finally, if variables other than physical wired points need to be mapped to DNP3, then ISaGRAF will need to be used. Appropriate logic will be required to be written mapping the points to DNP3 variables and manually log the variables. Pre-defined function blocks are available in ISaGRAF to assist in this task.

CREATING THE BASIC RTU CONFIGURATION

Follow the steps below to create a basic RTU configuration:

- Open Toolbox PLUS+ software
- Create a new Project, Group, and RTU. A "Group" is not always required but a "Project" and "RTU" is always required. The "Project" and "Group" are management tools and are not downloaded into the RTU.

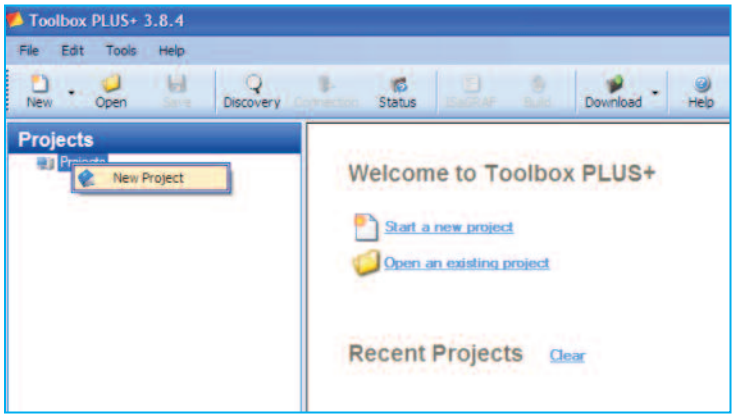


Figure 1: Creating a New Project in Toolbox PLUS+. Refer to the software manual for additional information on creating a "Project" and "Group."

* *For additional information on how to install the software, license protection key or downloading firmware into a processor communications module, please refer to the Toolbox PLUS+ manual.*

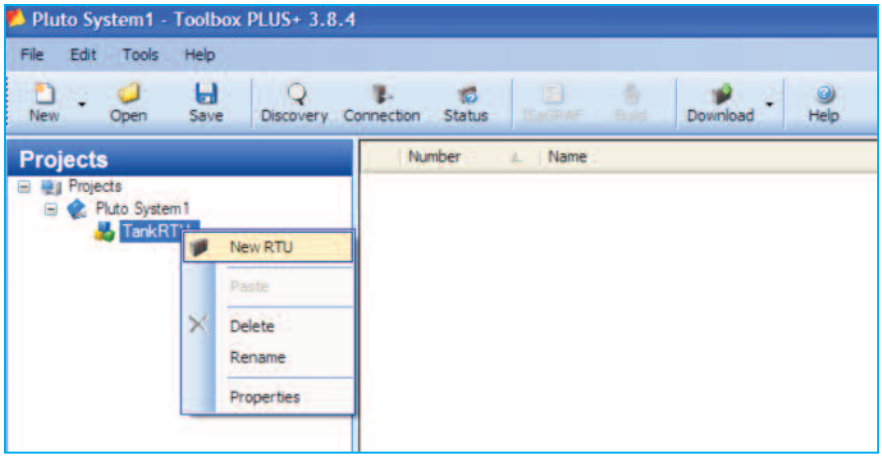


Figure 2: Screen dump after project Pluto System 1 and group TankRTUs have been created.

- After the project and group have been created, add RTU. An RTU can be added to a group or project. To add, right-click on the project name or group name and select New RTU as shown above. This will bring up a window as shown below (Figure 3).

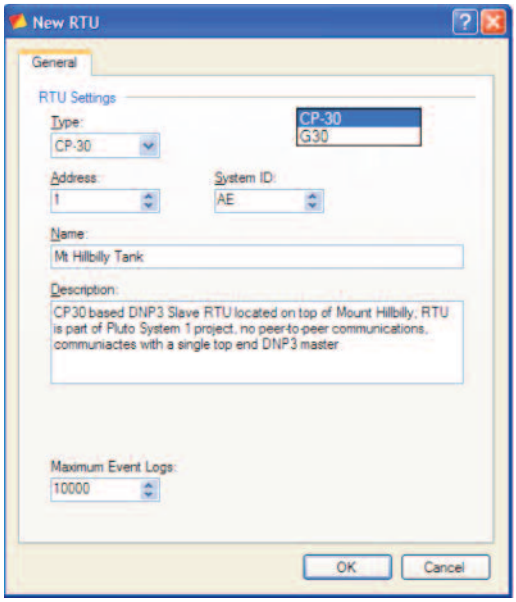


Figure 3: Pop-up when creating a new RTU. Processor type can be selected as CP30 (modular) or G30 (all-in-one). The "Maximum Event Logs" field is explained further below.

- When an RTU is added, by default the Power Supply and Processor modules are added. Additional modules can be added as required. This may include additional Power Supply or Processor (CP30) modules to accommodate a redundant configuration or standard I/O modules. If the processor type is G30, then I/O daughter cards can be added. The G30 is an all-in-one RTU. See Figures 4 and 5 below.

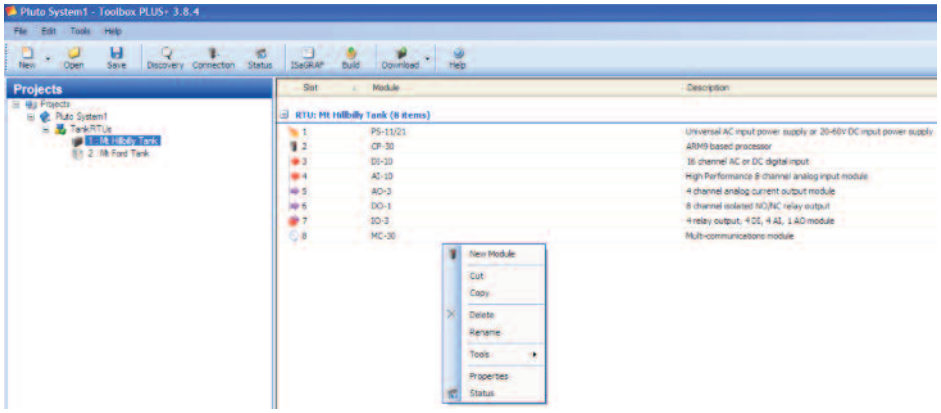


Figure 4: Screen shot after two RTUs have been added to the group TankRTUs. RTU1, Mt. Hillbilly Tank, is a CP30 based RTUs. Additional modules can be added by right-clicking on the project workspace window.

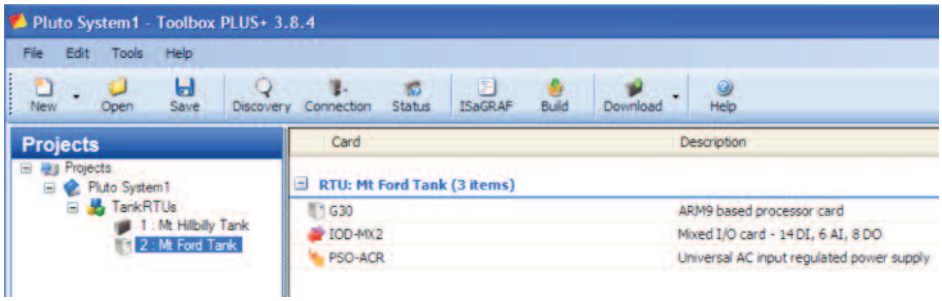


Figure 5: Screen shot of RTU2, Mt. Ford Tank, belonging to group TankRTUs. The G30 is an all-in-one RTU.

- Save the project by clicking on File → Save As. A folder will be created in Windows Explorer that will have the same name as the project. Sub folders are automatically created for each RTU in the project.

Note: From this point onward, the document will focus on the complete configuration of RTU1, Mt. Hillbilly Tank.

ADDING DNP3 PROTOCOL TO THE RTU

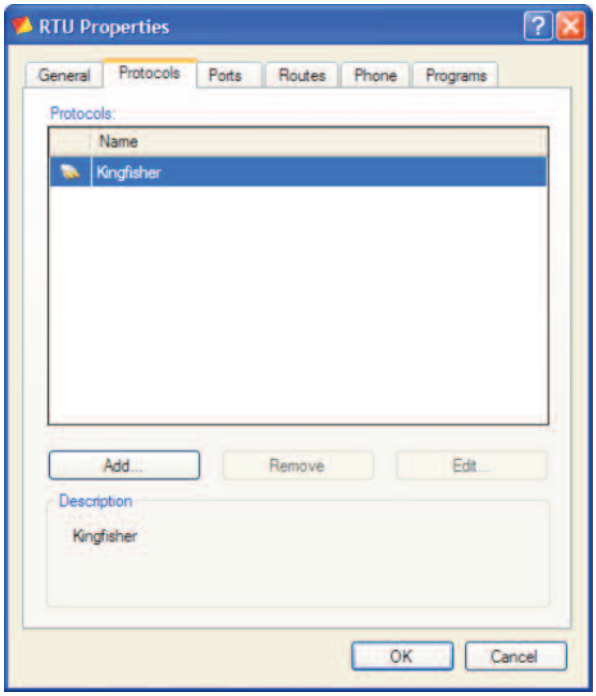
DNP3 protocol or any other protocol supported by the Kingfisher RTUs is part of the base firmware, but is not enabled. As stated previously each RTU can support up to 32 different communication ports. Each port can use a different protocol — DNP3, Modbus, AB DF1 et al.

Once the protocol has been added to the RTUs configuration, it can be selected for us on a particular communications port. This part of the configuration is described in detail in the section, Configuring Communication Ports.

Follow the steps listed below to add the DNP3 protocol to the RTU.

- Right click on the RTU and select Properties. Select the protocols tab, as shown below.

Figure 6: The protocols tab for an RTU.



- Click on the Add button and the following pop-up will appear, select DNP3 and click on OK.

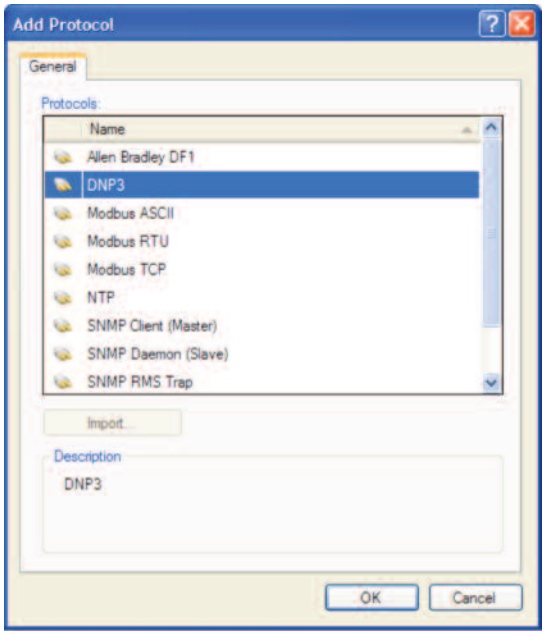


Figure 7: Adding DNP3 protocol to the RTU so that it can be selected on a communications port.

- Once the protocol has been added, the Protocols tab will be as shown below.

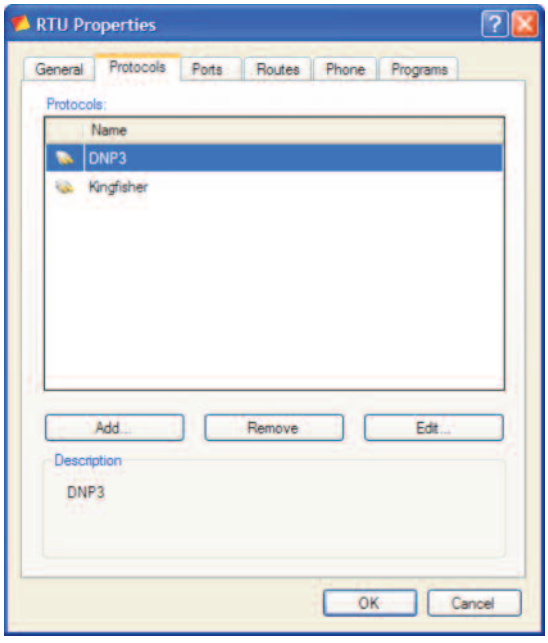


Figure 8: Screen shot of the Protocols tab after DNP3 has been selected.

- The "DNP3" protocol will work on Serial or Ethernet ports. The properties of the DNP3 protocol can be edited by clicking on the "Edit" button (see Figure 8, above)

When the RTU is being configured as a DNP3 Slave, it may not be required to edit the properties of the DNP3 protocol in the RTU. Assume the protocol properties will have to be edited. This is explained in the following section.

DNP3 PROTOCOL PROPERTIES

DNP3 protocol has a number of properties which the user can configure. In many cases, default settings may be the best option.

A detailed explanation of all parameters is not provided in this document. Please refer to the manual. This document simply shows what parameters can be edited by the user.

From figure 8 (above), when the Edit button is clicked, the following window will appear.

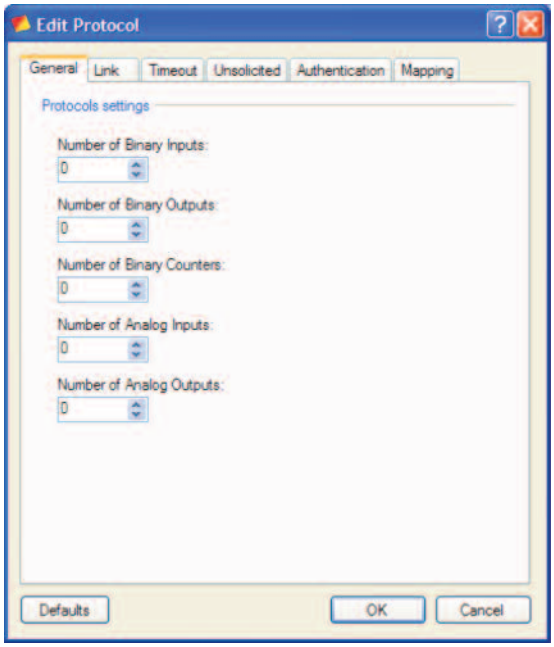


Figure 9: Properties of the DNP3 Protocol that can be edited by the user.

In Figure 9 (above), the different tabs can be explained as follows.

- General — This lists the number of objects of each type of DNP3 variable. The objects or variables can also be created manually via the Dictionary. The process of creating DNP3 variables is explained further below in this document.
- The "Link" tab contains several low level protocol parameter settings. These include configuring the message Fragment and Frame sizes as well as link layer settings specific to DNP3 protocol. Default settings are automatically displayed. Care should be given when changing the default settings as they will have to match the top end.
- The "Timeout" tab allows the user to configure the global "select before operate" timeout as well as the application layer timeout. It also allows you to configure how often the processor will scan through the DNP variables.
- The "Unsolicited" tab allows the user to configure unsolicited reporting from the DNP3 Slave (the RTU) to a DNP3 Master device. The RTU can send unsolicited messages to multiple DNP3 master devices and has the ability to send only a particular class type (1, 2 or 3). Unsolicited reporting is similar to exception reporting — the slave device initiates a message to a master upon occurrence of a certain event such as an alarm condition becoming true.

- The “Authentication” tab allows the user to program in security features. DNP3 Secure is a complete topic by itself and will not be discussed in this technical brief.
- The “Mapping” tab allows the user to map remote DNP3 objects to local DNP3 objects. This is not a function of a typical Slave DNP3 device and as such will not be discussed in this technical brief. Mapping is a function typically associated with a DNP3 Data Concentrator device.

The “Defaults” button located on the bottom left hand corner (see figure 9) allows the user to set the defaults for the different types of DNP3 variables.

For each of the different DNP3 object types the class number (0, 1, 2, 3) and the type of variation (static and event) can be set. By default, the object will be treated as a Class 0 object and no event log will be created when there is a change of state or a percentage change in value.

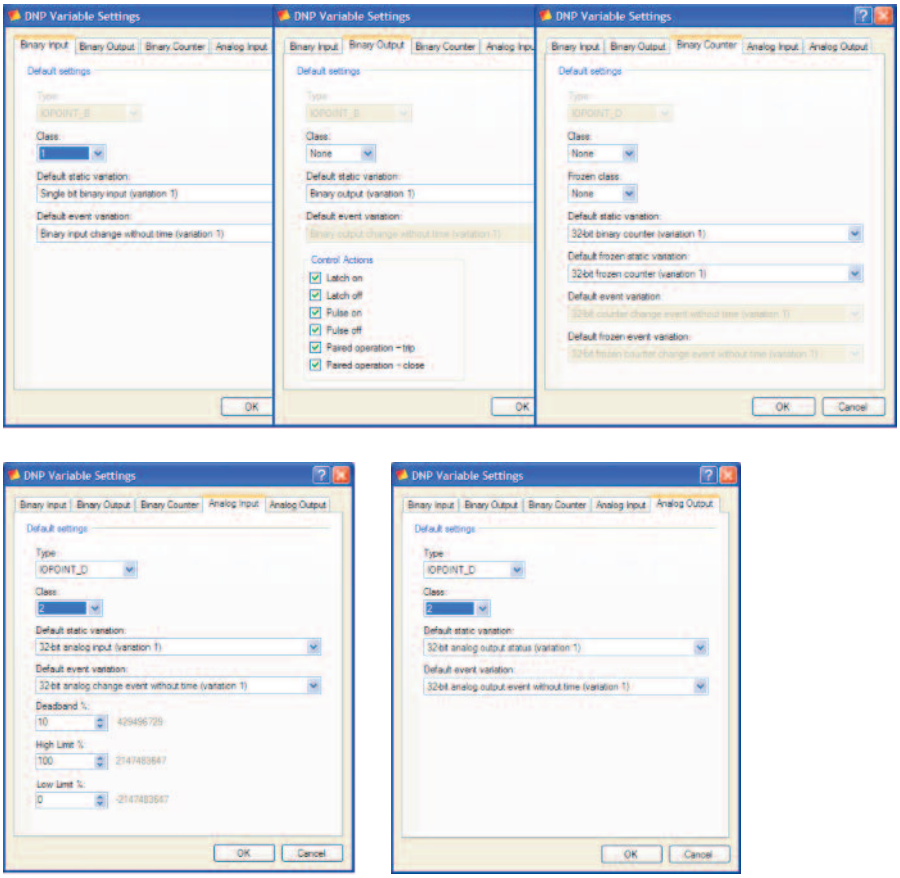
For the Binary Output (BO) objects, the default control action — latch ON/OFF, pulse ON/OFF, TRIP or CLOSE can be configured. It should be noted that this setting will apply for all BO objects.

For analog input (AI) objects the dead band, high and low limits can be set. The can be configured as a percentage or as raw vales. Again, this setting will apply for all AI objects.

Each DNP3 variable can be configured with its own settings for class number and variation. This is done by editing each variable from the Dictionary. This process is explained further below in this document.

Once all of the DNP3 protocol parameters have been configured, click on OK to return to the RTU Properties window (see figure 8).

Figure 10: Configuration of default settings for each type of DNP3 variable.



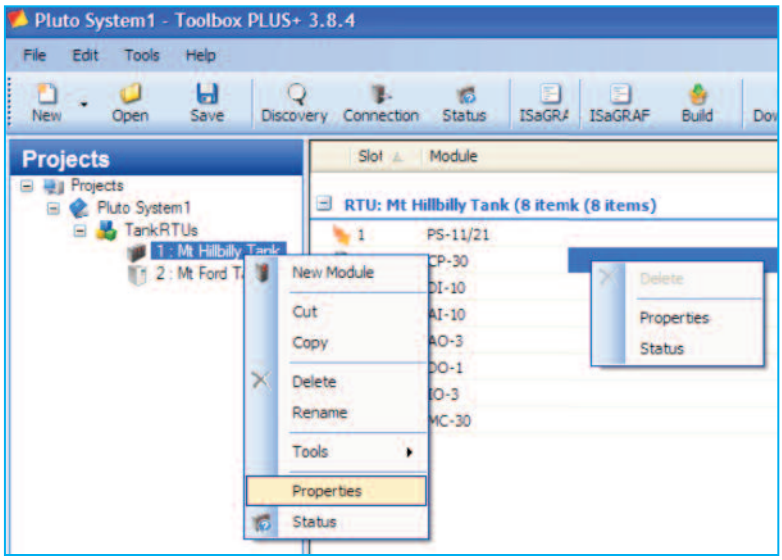
CONFIGURING COMMUNICATION PORTS

RTU1. Mt. Hillbilly Tank has a processor (CP-30) and multi communications (MC-30) module. The CP-30 can have a maximum of three communication ports. Port 1 of the CP-30 is a fixed 10/100MBit Ethernet port. The communications module, MC-30, can also have a maximum of three communications ports. Port 1 of the MC-30 is a fixed RS232 port.

An RTU can have up to 32 communication ports of different types — RS232, RS485, PSTN, Fiber Optic, Spread Spectrum, FSK, PLINE and Ethernet.

Communication ports for a module can be accessed by right-clicking on the module and selecting properties. To access all communication ports for a RTU, right click on the RTU name and select properties.

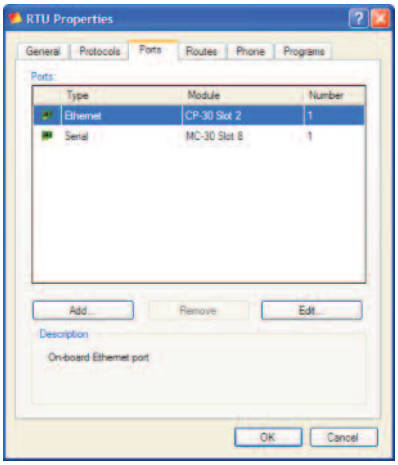
Figure 11: Accessing the ports for a Module or an RTU.



Assuming the communication ports of the RTU are accessed by right-clicking on the RTU name, follow the steps listed below to configure the communication ports.

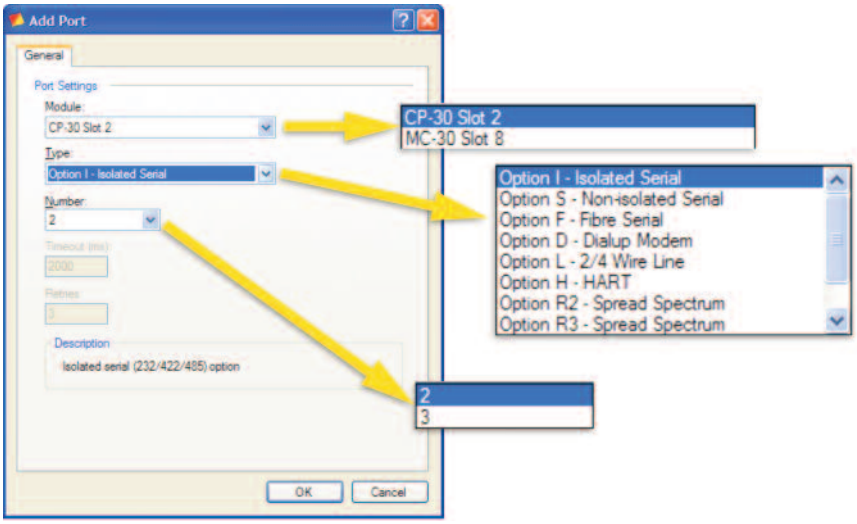
- Select the "Ports" tab. The default ports of the CP-30 and MC-30 modules will be displayed, as shown below.

Figure 12: Default ports of the CP-30 and MC-30 modules.



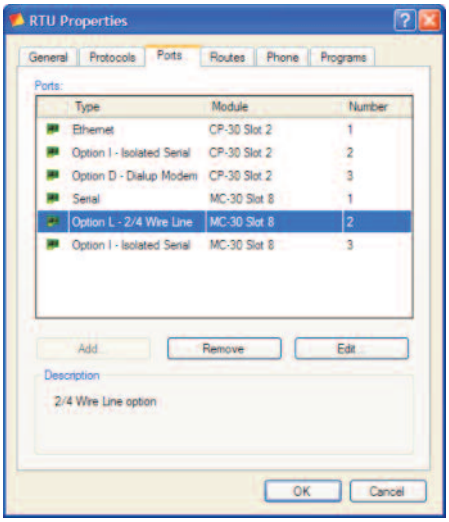
- The CP-30 and MC-30 can have a maximum of three ports each. To add additional ports that may belong to the CP-30 or MC-30 module, click on the Add button and the following screen will appear (Figure 12, above). Port 1 of the CP-30 and MC-30 is fixed — Ethernet and Serial respectively. A RTU may contain several MC-30 modules. The option ports in a CP-30 or MC-30 is either Port 2 or 3. The port number, module type and port type can be selected as shown below.

Figure 13: Adding a new port to a CP-30 or MC-30 Module.



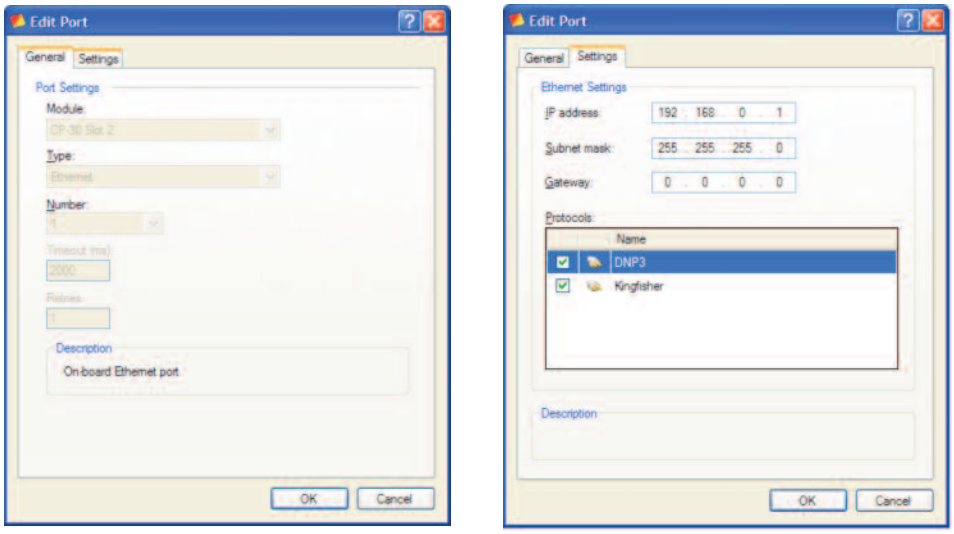
- Once all of the ports have been added, the screen should appear as follows:

Figure 14: Screen shot after all communication ports have been added.



- Now, select a port and click on the Edit button. For the purpose of this technical brief, we will select Port1 of the CP-30 module in Slot 2 — the default Ethernet port. The primary Ethernet port on the CP-3 module supports multiple protocols. Go to the Settings tab and select DNP3 Protocol. Ensure IP address settings are correct and then click OK.

Figure 15: Editing the Ethernet port of the CP-30 module.



DNP3 protocol will also work on the following ports:

- RS232, RS485
- Spread Spectrum Radio
- Integrated PSTN Modem
- FSK/PLINE
- External Modems connected to the Serial port. This includes GSM, GPRS, Radio, and Satellite.

Please refer to the manual for additional information on configuring the ports.

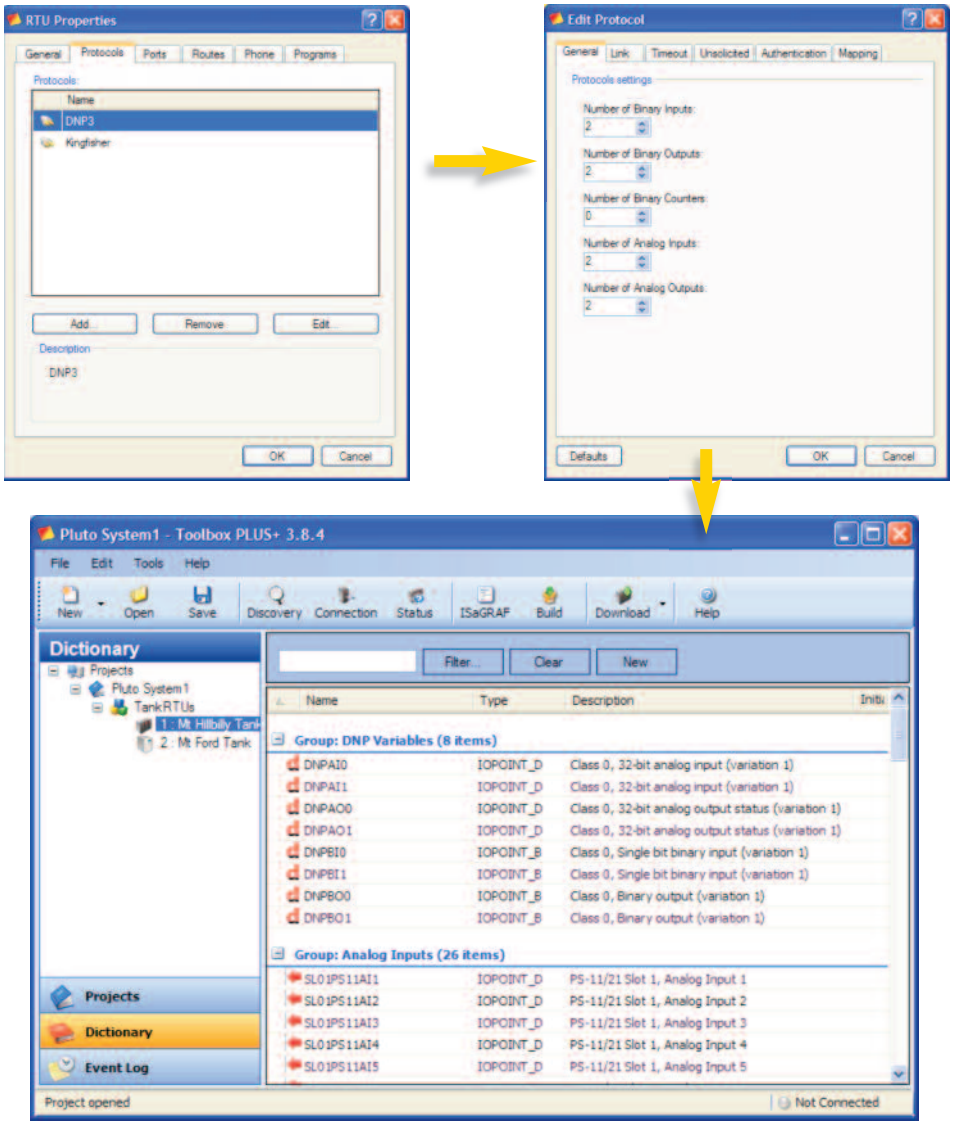
CREATING DNP3 VARIABLES AND MAPPING

We have looked at creating a basic project, RTU, adding modules, protocols, and then configuring the communications port. The next step is to create DNP3 variables and map them, if required.

DNP3 variables can be created in one of two ways. Method one is as described below.

Right click on the RTU Name, go to Properties. Then select the Protocols tab and ensure DNP3 is on the list. If no, click on Add and select DNP3 protocol. Click OK. Then, select DNP3 and click on Edit. A pop up window will appear as shown in Figure 9. Enter the number of objects of each type and click OK. The DNP3 objects or variables will appear in the Dictionary folder. See screen dumps below.

Figure 16: Creating DNP3 Variables automatically by editing the properties of the DNP3 protocol.



An alternative way to create DNP3 variables is to go to the Dictionary and manually create the required DNP3 variables. This process is described below.

- Click on Dictionary from the Wonder Bar (bottom left hand corner), see below:

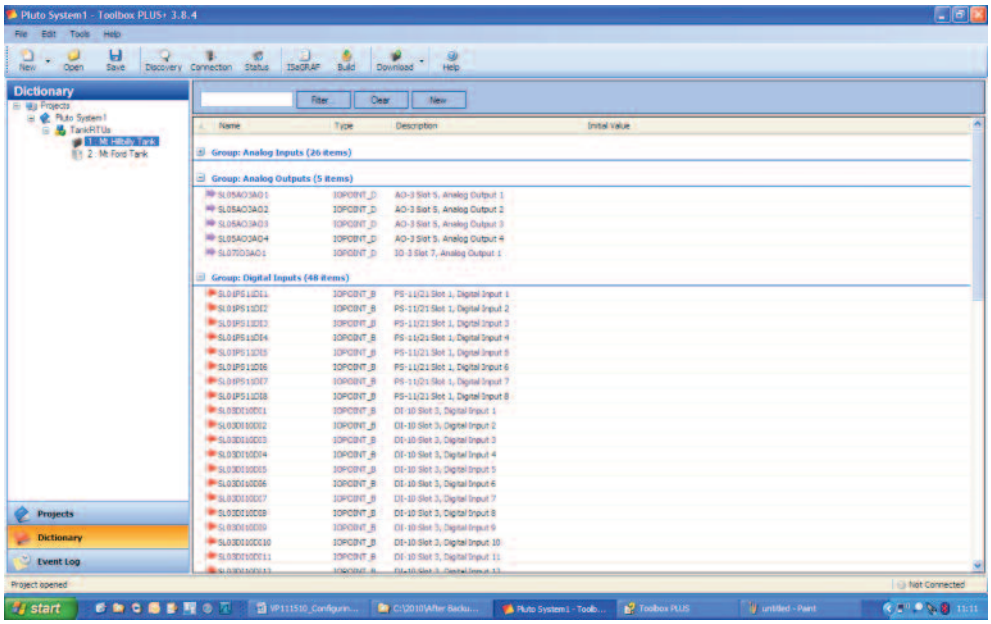


Figure 17: The Dictionary folder within Toolbox PLUS+. DNP3 variables (or other protocol variables) can be created by going to this folder. The different groups listed above are for the different types of real variables. Real variables are automatically created and are based on I/O modules added to the RTU. The naming convention for a real variable is based on the I/O module slot address, module type and channel address.

- Click on New and a window will popup. Select the Multiple tab. See figure 18 below. The correct settings to create DNP3 variables should be as follows:
  a. Format — Must select DNP3.
  b. Prefix — This is automatically filled in and is based on the selected format.
  c. Group — Must select DNP Variables. If the format or the group is selected incorrectly, then, variables will be created but they will not be DNP3 variables.
  d. RTU Address — This is the same as the address of the DNP3 Slave RTU. The correct RTU address will appear by default, but can be changed.
  e. Type — Refers to the type of DNP3 variables. The different types of DNP3 variables are as follows:
     i. BI = Binary Inputs (same as DI)
     ii. BO= Binary Output (same as DO)
     iii. AI = Analog Input
     iv. AO = Analog Output
     v. BC = Binary Counter
     vi. FC = Frozen Counter
  f. Register Range — This is the DNP3 object range. The range can be from 0 to 65535. That is, 65535 objects of each type can be created and is supported by the Kingfisher RTU.
  g. Data Type — This field is filled in automatically and is based on the Type field.
  h. Description — provides a common description but is also dynamically lists the variable range that is being created.

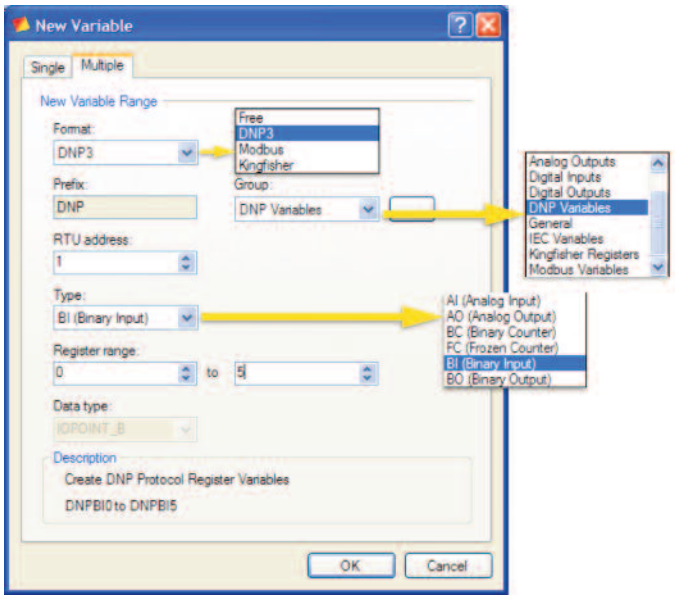


Figure 18: Creating multiple DNP3 variables of BI Type

Click OK once all of the fields have been completed. Repeat process if other types of DNP3 variables need to be created.

- Once the variables have been created, they will appear in the Dictionary. See figure below.

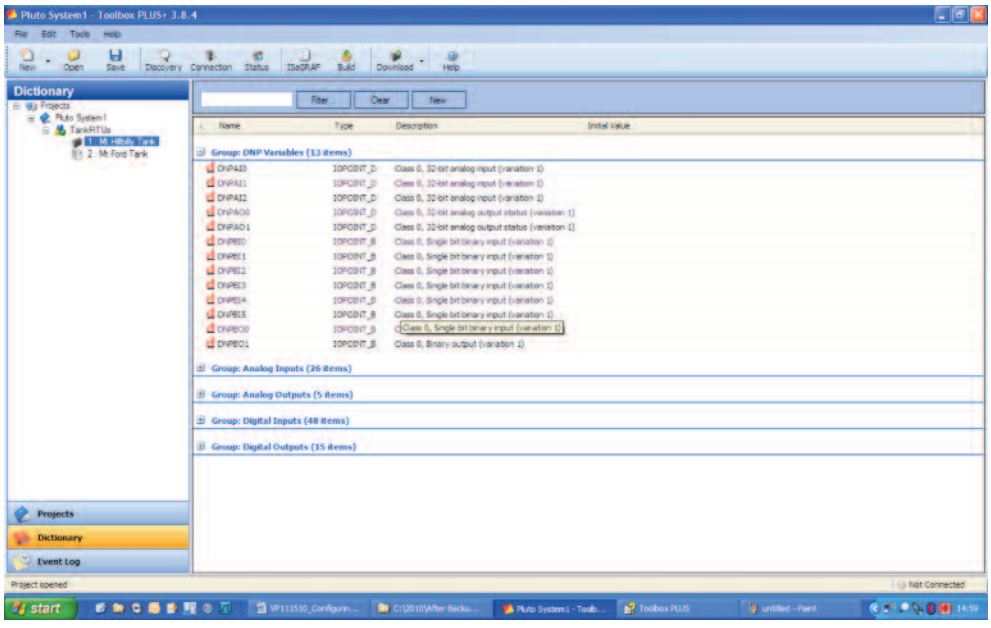


Figure 19: DNP3 variables of different types that were created using the process described above and as viewed in the Dictionary.

If the variables are not listed in the Group: DNP Variables, then the variable is not a DNP3 variable.

After the DNP3 variables have been created, each variable can be edited and certain features configured. This includes mapping a real I/O point to the DNP3 variable — obviously the types will have to match.

But, before we look at the properties of each variable, let's look at the format of the DNP3 variable itself. The DNP3 variable is a structure. This is not evident when viewing the variables via the Dictionary in Toolbox PLUS+. One will have to access the Dictionary in ISaGRAF to understand that a DNP3 variable is a structure. Figure 20 below is a screen shot of the Dictionary from ISaGRAF.

DNP3 variables can have one of two formats:

- DNPTTx where TT is the object type (AI, BI, BO, AO, BC) and x is the object number (0 to 65535)
- DNPrTTx where r is the remote RTU address, TT is the object type and x is the object number

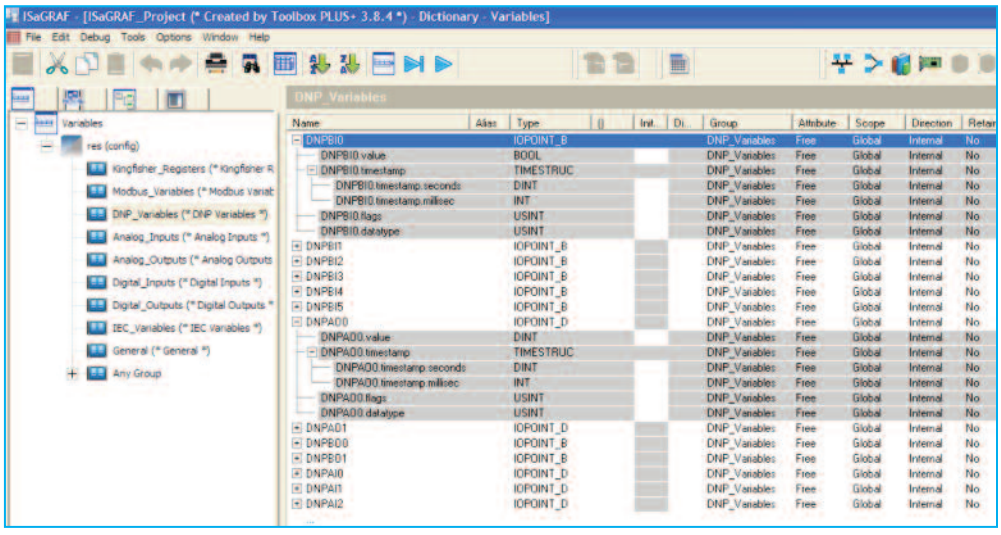
The DNPrTTx variables are only used when the Kingfisher RTU is configured as a DNP3 Master and is storing information from other DNP3 slave devices.

The DNP variable is a structure made up of the following parameters:

- DNPTTx.value — This contains the actual value of the variable and may either be an integer value or a Boolean value
- DNPTTx.timestamp — This contains an additional two attributes:
  - DNPTTx.timestamp.seconds — if the variable is of class 1, 2 or 3 then a log is created when there is a change in value. This field contains the number of seconds since 1971.
  - DNPTTx.timestamp.millisec — As above but contains the millisecond value.
- DNPTTx.flags — contains diagnostics information such as variable online/offline.
- DNPTTx.datatype — contains an integer value representing the type of data.

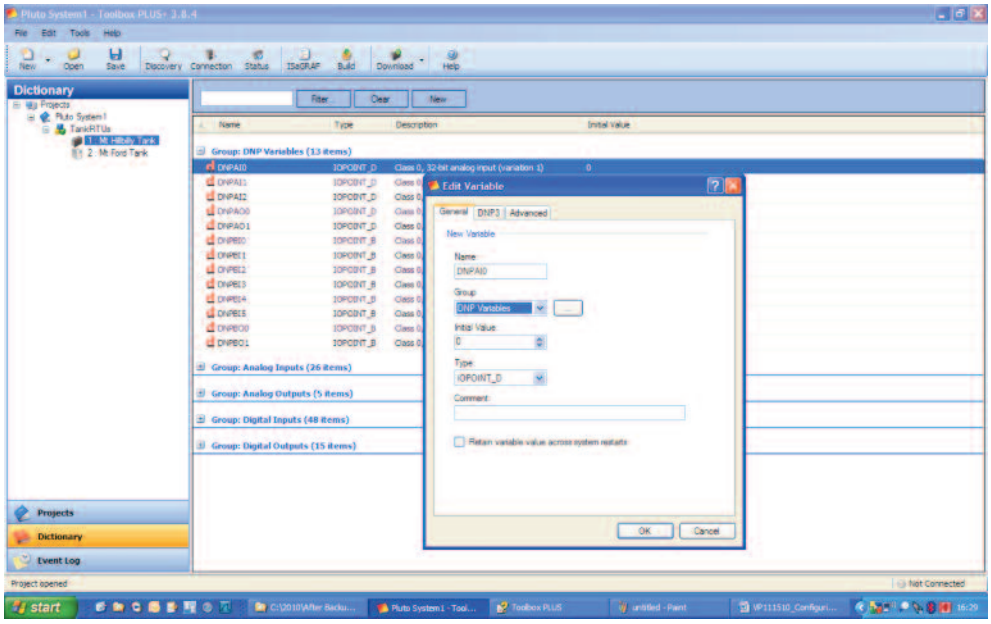
In most cases a thorough understanding of the DNP3 variable structure is not required; especially if the system is simple and only real I/O points are being monitored and/or controlled. An understanding of the DNP3 variable structure will be required if internal variables are mapped to DNP3 variables through ISaGRAF and then logged.

Figure 20: Dictionary as viewed in ISaGRAF.



To edit a DNP3 variable, be in Dictionary view in Toolbox PLUS+ and then double click on the DNP3 variable to edit. A window showing all of the editable attributes will appear, as shown below (figure 21).

Figure 21: Editing a DNP3 variable in the Toolbox PLUS+ environment.



In the **General** tab:

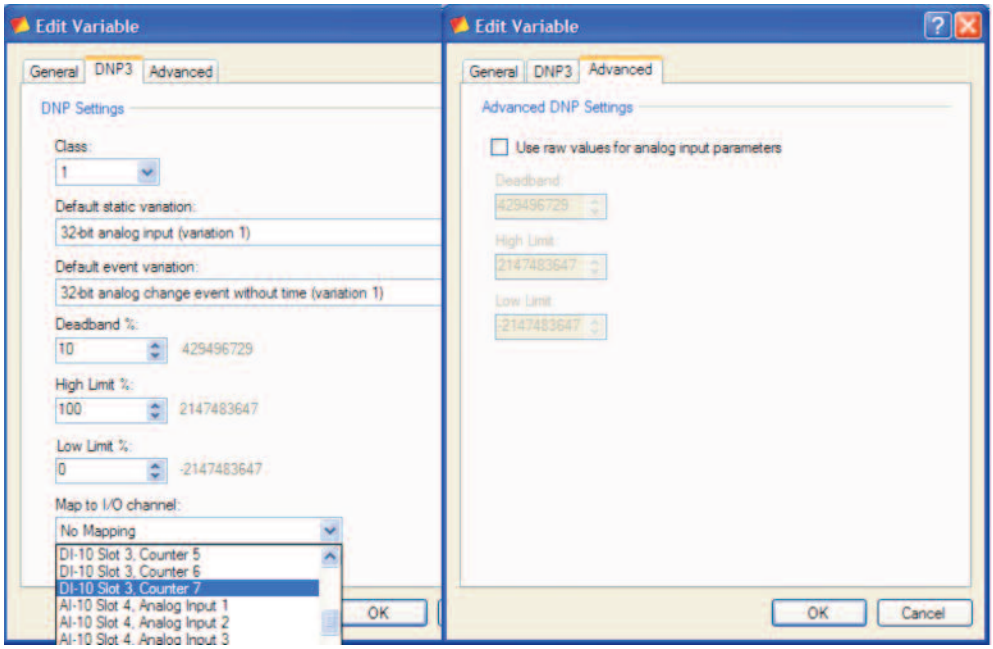
- The Name, Group and Type fields should not be changed as these attributes were selected when creating the variable. A useful description for the variable can be entered in the Comment field.
- If "Retain variable value across system restarts" is selected, the value of the variable will be maintained during power loss or RTU reconfiguration.

In the **DNP3** tab (figure 22), the default settings as configured when editing the protocol properties will be displayed. This can be changed.

- Class — select the class the DNP3 variable will belong to. This is typically 0, 1, 2 or 3. Class0 refers to static data and no time stamped logs will be created. If class 1, 2 or 3 is selected then time stamped logs will be created. The class is simply a method to differentiate between the different types of DNP3 objects and prioritize the importance of data to be sent back to the DNP3 Master.
- Default Static Variation — different types allowing the user to select sending the value of the DNP3 object or value and diagnostic information.
- Default Event Variation — Configurable if Class 1, 2 or 3 is selected. Allows the user to record the event with or without time.
- Dead band % (Applicable only for AI objects) — Specify a dead band value as a percentage.
- High Limit % (Applicable only for AI objects) — Specify the high limit value as a percentage.
- Low Limit % (Applicable only for AI objects) — Specify the low limit value as a percentage.
- Map to I/O channel — Allows the user to map a physical I/O point to a DNP3 variable.

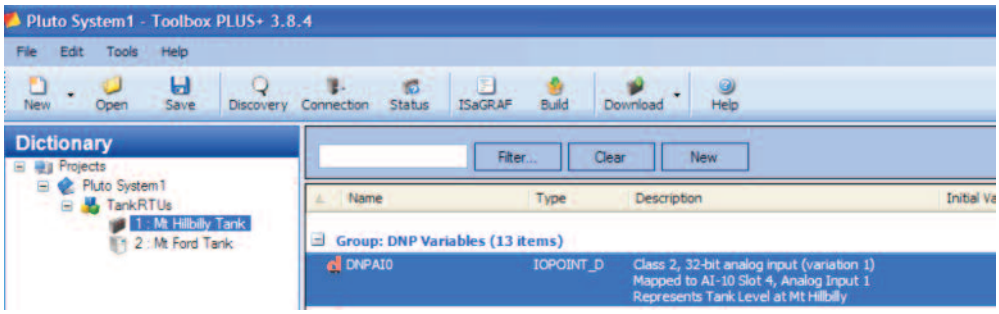
The **Advanced** tab allows the user to use raw values instead of a percentage value for the AI deadband, high limit and low limit.

Figure 22: Editing a single DNP3 variable.



Click OK once all of the parameters have been configured correctly. The Dictionary in Toolbox PLUS+ will look as follows:

Figure 23: Toolbox PLUS+ Dictionary view of the DNP3 variables when a physical I/O point has been mapped.



Repeat process to continue mapping DNP3 objects to physical I/O points as required.

DOWNLOADING TO THE RTU

The next steps are to save, build/compile and download the configuration into the RTU and then verify that the configuration works.

To save the configuration — In Toolbox PLUS+, click on File → Save As

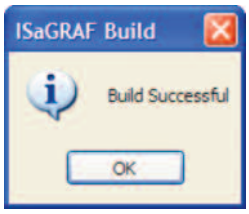
A window will appear and the project name will be displayed as the default file name to be saved. Click OK. A folder will be created with the same name as the project name and all of the information will be saved.

To Build/compile the configuration:

- Ensure the ISaGRAF dongle is connected to the PS USB port
- Make sure the correct RTU configuration is selected within the Project and then press F9 which is the short-cut key to "Build" the project. The "build" function can also be accessed by clicking on Tools → Build or from the toolbar within Toolbox PLUS+

During the build process, Toolbox PLUS+ will open ISaGRAF automatically and compile the project in full. If successful, the following message will appear:

Figure 24: Message displayed in Toolbox PLUS+ when the project is "Build" successfully.

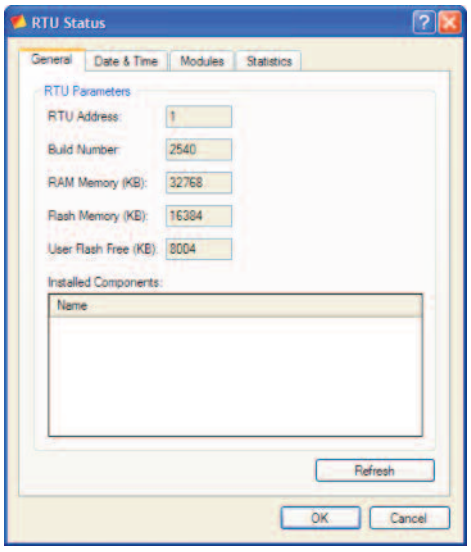


If the build process is not successful, then open ISaGRAF to check the error messages. However, if the RTU is configured as described above then a build should always be successful as there is no ISaGRAF logic.

The configuration is now ready to be downloaded into the RTU. To download:

- Verify PC LAN address is compatible with RTU IP Address. Default RTU IP address is 192.168.0.1 The PC LAN address can be accessed as follows: Start → Settings →Network Connections, double-click on the LAN connection. In the "General" tab, select "Internet Protocol (TCP/IP)" and click on Properties, specify the LAN address compatible to the RTU.
- In Toolbox PLUS+, make sure the correct project and RTU is selected.
- Click on Tools → Connection; Make sure Toolbox PLUS+ is connecting to the correct IP Address and click on OK.
- Then, click on Tools → Status; If Toolbox PLUS+ is communicating with the RTU, then the following screen will appear, click OK after verifying the information.

Figure 25: RTU Status screen when Toolbox PLUS+ is communicating with the CP-30 successfully.



Note: If Toolbox PLUS+ is not communicating with the RTU, then check the following – IP Address settings of the PC and other peripheral devices (is any), firewall settings, diagnostic LEDs on the CP-30 and the cable itself. A cross over cable should be used if connecting directly between Toolbox PLUS+ and the RTU. If the connection is via a hub/switch/router, then standard straight through Ethernet cables must be used.

- Click on Tools → Download → Configuration and Logic; Follow prompts to download the configuration into the RTU. Important: Once the file has been downloaded, do not recycle power to the RTU, allow the RTU to restart automatically.
- If the RTU has restarted successfully, the OK and F3 LEDs on the CP-30 will come on. If they are not ON or any other LEDs on the top row are ON, there is an error in Port configuration. Verify the configuration and re-download into the RTU, if required.
- Reconfirm communications between Toolbox PLUS+ and the RTU using the Status (Tools → Status) feature. See figure 25, above.

TESTING THE CONFIGURATION

The final STEP is to verify that the configuration downloaded into the RTU is working. This is done in several steps. The first step is to ensure that time stamped logs is being created in the RTU - the logging will be based on the mapping of physical I/O points to DNP3 variables.

The Analog and Digital Inputs will have to be connected to simulators so that the status and values can be changed and logs generated.

To upload the logs into Toolbox PLUS+, follow the steps below:

- Confirm communications between Toolbox PLUS+ and CP-30 RTU (Tools → Status).
- Click on Event Log in the Wonder Bar (bottom left hand corner when in Toolbox PLUS+).
- Click on the "Retrieve" button and the Select from the available options as appropriate and click OK. The events will be uploaded and displayed, as shown below (figure 26).

The logs can be exported to Excel if required using the Export button.

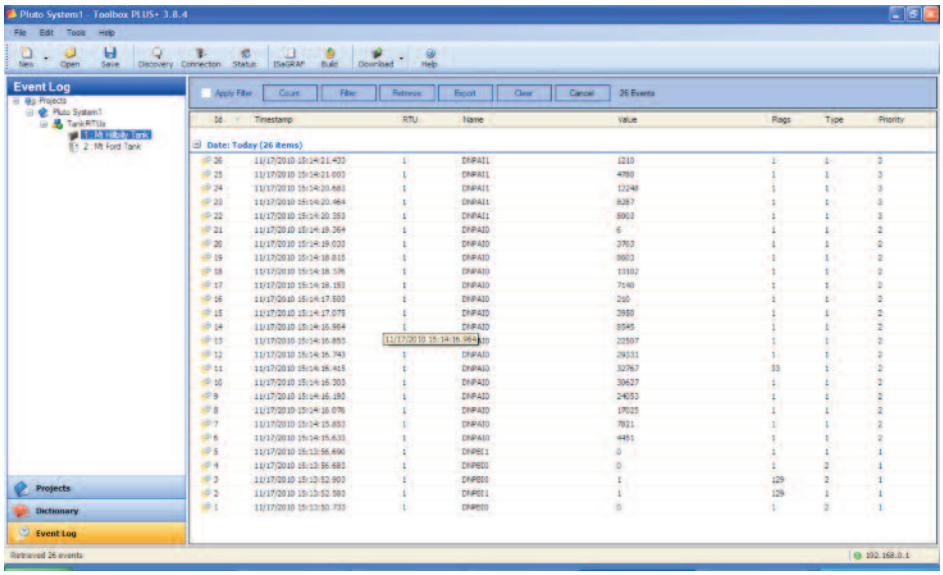


Figure 26: Viewing uploaded logs in Toolbox PLUS+.

The second step in verifying the configuration works, if to interface the RTU to a DNP3 Master device that is capable of polling the DNP3 Slave CP-30 RTU for static and event data. The DNP3 master is typically a SCADA system or a simulator (Triangle Microworks, ASC) of some sort. Certain SCADA systems also offer a OPC/DNP3 Driver interface that is segregated from the main SCADA system to test.

ISaGRAF FUNCTIONS IN A DNP3 SLAVE RTU

The Kingfisher PLUS+ RTUs include embedded ISaGRAF as part of the firmware. ISaGRAF is compliant to process programming standard IEC61131-3. ISaGRAF differs from Toolbox PLUS+ in that ISaGRAF is used to create the process logic while Toolbox PLUS+ is used to create the RTU Configuration – name, address, ports, route list, modules list, protocols etc., CSE-Semaphore has created a number of pre defined function blocks to work in ISaGRAF. This is in additional to a large number of pre defined function blocks that comes standard with ISaGRAF. These functions will help the user to program efficiently.

When the CP-30 or G30 RTU is being used as a DNP3 Slave, in most applications any additional ISaGRAF logic will not be required. But there are also several instances where additional programming may be required. In such instances there may be internal parameters that will be required to be mapped to DNP3 variables and logged. When internal parameters are mapped to DNP3 variables, then time-stamped logs will not be created automatically. This is only applicable for physical I/O points.

As a DNP3 Slave, the following functions are available:

- KF_EVENT_LOG: This function can be used to manually create time stamped event logs.
- DNPS_NEED_TIME: The outstation DNP device sets the Request Time Sync bit in the IIN register, so that when the DNP3 Master device next performs any sort of Poll, the IIN will indicate the outstation device requires a time sync.
- DNPS_UNSOL_ENABLE: Enables unsolicited reporting from the outstation DNP device to a DNP master.
- DNPS_UNSOL_DISABLE: Disables unsolicited reporting from the outstation DNP device to a DNP master.

Refer to the online help for additional information on how to use these functions correctly.

The simple example shown below explains how an internal Boolean variable, CabinetDoorOpen, is mapped to DNP3 BI object DNPBI3.value. The DNPBI3.value changes state (positive or negative edge), then the DNP3 variable DNPBI3 is logged. The result is shown in Figure 28.

In the function KF_EVENT_LOG, "DoorOpen" is a STRING variable that is mapped to DNPBI3.

Please refer to the online help for additional information.

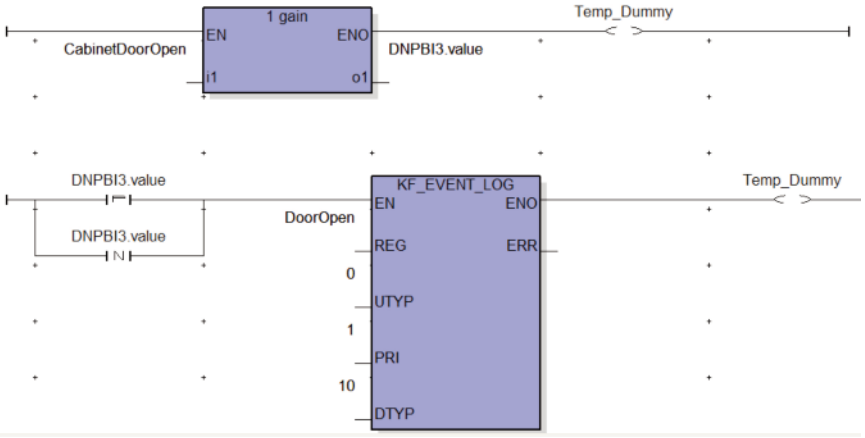


Figure 27: Simple ISaGRAF example of creating a DNP3 time stamped log manually.

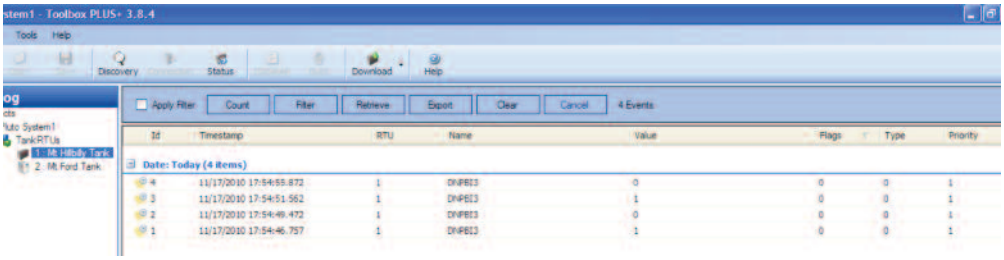


Figure 28: Viewing of logs in Toolbox PLUS+.

SUMMARY

The aim of this technical brief was to provide a detailed step-by-step guide on configuring the Kingfisher CP-30 or G30 RTUs as a DNP3 Slave. It is indented that a user of this document will use it in conjunction with the Toolbox PLUS+ software manual. Online help is another source of information. General information about the DNP3 protocol can be found by searching the web. The Wikipedia article on DNP3 provides a good general overview of the protocol. Device Profile Documents which explain in detail the functions supported by the Kingfisher RTUs are available from CSE-Semaphore.

Configuring a Kingfisher RTU to be a DNP3 slave device is simple and intuitive. The main steps in this process are:

- Adding the DNP3 protocol
- Configuring the communication ports with the correct parameters and linking the protocol to the port
- Creating the required DNP3 Protocol variables
- Mapping the DNP3 variables to physical I/O points and configuring the class and variation type
- If needed, use the pre-defined ISaGRAF functions to enhance the DNP3 Slave functionality of the Kingfisher RTU

SEMAPHORE
Worldwide contact information

U.S.A.
CSE Semaphore Inc.
1200 Chantry Place
Lake Mary, FL 32746
U.S.A.
P +1 (407) 333 3235
F +1 (407) 386 6284
Days@cse-semaphore.com

Australia
CSE-Semaphore
Unit 8, 3-5 Gilda Crt
Mulgrave, Victoria 3170
Australia
P +61 (03) 8544 8544
F +61 (03) 8544 8555
Info.kingfisher@cse-semaphore.com

Europe
CSE-Semaphore Begium
Waterloo Office Park — Building "M"
Dreve Richelle, 161
B-1410 Waterloo
Belgium
P +32 (2) 387 42 59
F +32 (2) 387 42 75
info.tbox@cse-semaphore.com