

How to Configure Redundancy

April 28, 2010

The built in redundancy features were added to make redundancy easy in the IPms. Very minimal programming is required to make redundant controllers. There are some other features to carry redundancy into other aspects of the application. There are two different typical scenarios meant to be used with Sixnet redundancy. Figure 1 shows redundant controllers using the ST-IPM-6350 controllers, EtherTRAK-2 modules and redundant SCADA computers respectively. Figure 2 is a redundancy configuration with VT-IPM-X410 controllers, EtherTRAK-2 and redundant SCADA.

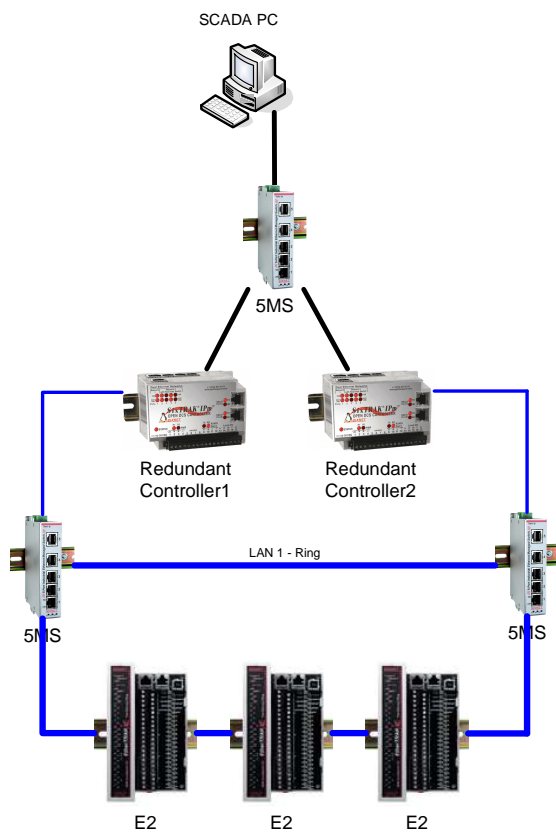


Figure 1

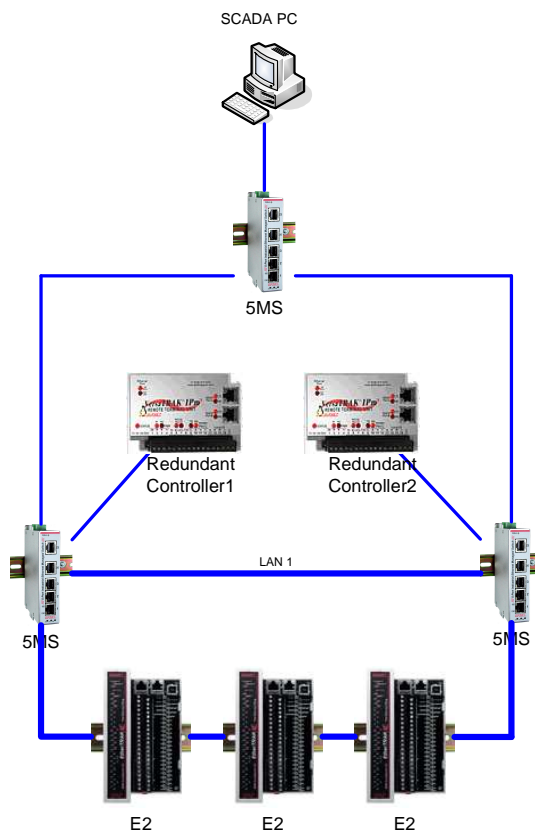


Figure 2

Industrial & Commercial Networking

These redundant systems provide fail-over for 5 key failure conditions:

1. Software failure or program maintenance in the IPm – This condition accounts for the logic code (ISaGRAF or C) stopping in one controller. In this condition the system will still continue to function. The cause of this may be program crash or deleting the program for maintenance.
2. IPm hardware failure – If one IPm losses power or fails to continue to operate the system will continue to run.
3. SCADA failure – The support for multiple SCADA computers means that one computer could be rebooted or taken off line and the system will continue to operate.
4. Power Supply failure – Redundant power inputs on I/O modules and controllers means if a power supply fails or there is a power outage an alternative power (ex: Battery back-up) supply can be used to power the system.
5. Ethernet connection failure – Any one Ethernet connection can be broken on the system and the system will be fully functional.

The next 5 sections will describe in detail how the SIXNET redundant system handles the fail-over conditions and which features were added to make it work.

Software Failure or program maintenance in the IPm

The built-in redundancy features are the most significant and important changes required to make redundancy work. The system always assumes there are two redundant controllers in the system.

Station Status Module variables

There are set of key variables are in each controller and is monitored by the redundancy code to ensure the controllers can determine which is in control and if the logic code is running. The variables are separated into “self” and “other”. The controller uses the “self” as its own variables and the “other” as the other controller. The built-in redundancy configuration automatically set a write transfer the “self” bits to the other stations “other” bits.

The variables are as follows:

Self_Heartbeat – Discrete Output variable that is pulsed by the ISaGRAF or C program. This variable is used by the built-in redundancy code to determine whether the station should be in Active or Back-up mode.

Self_Active – Status-bit generated by the built-in redundancy feature. When the bit is on the controller is in Active mode. On indicates that the controller. The built-in redundancy will set this bit when all the active criteria are met.

Self_Backup – Status-bit generated by the built-in redundancy feature. When this bit is ON the station is in back-up mode.

Other_Heartbeat – This bit is the pulsing heartbeat Discrete output of the other station generated by the ISaGRAF or C program of the other station.

Flexible. Reliable. Powerful.

Other_Active – This bit is the “Self_Active” bit from the other station. This is used by the built-in redundancy feature to help determine whether the station should be in Active or Back-up mode.

Other_Status – This bit is the “Self_Back-up” bit from the other station. This is used by the built-in redundancy feature to help determine whether the station should be in Active or Back-up mode.

Self_Status – This is a simple user bit to quickly determine if the station is in Active or Back-up mode. ON is Active and OFF is Back-up.

Fail-over code in the IPm

Active mode – Reading all inputs and writing outputs.

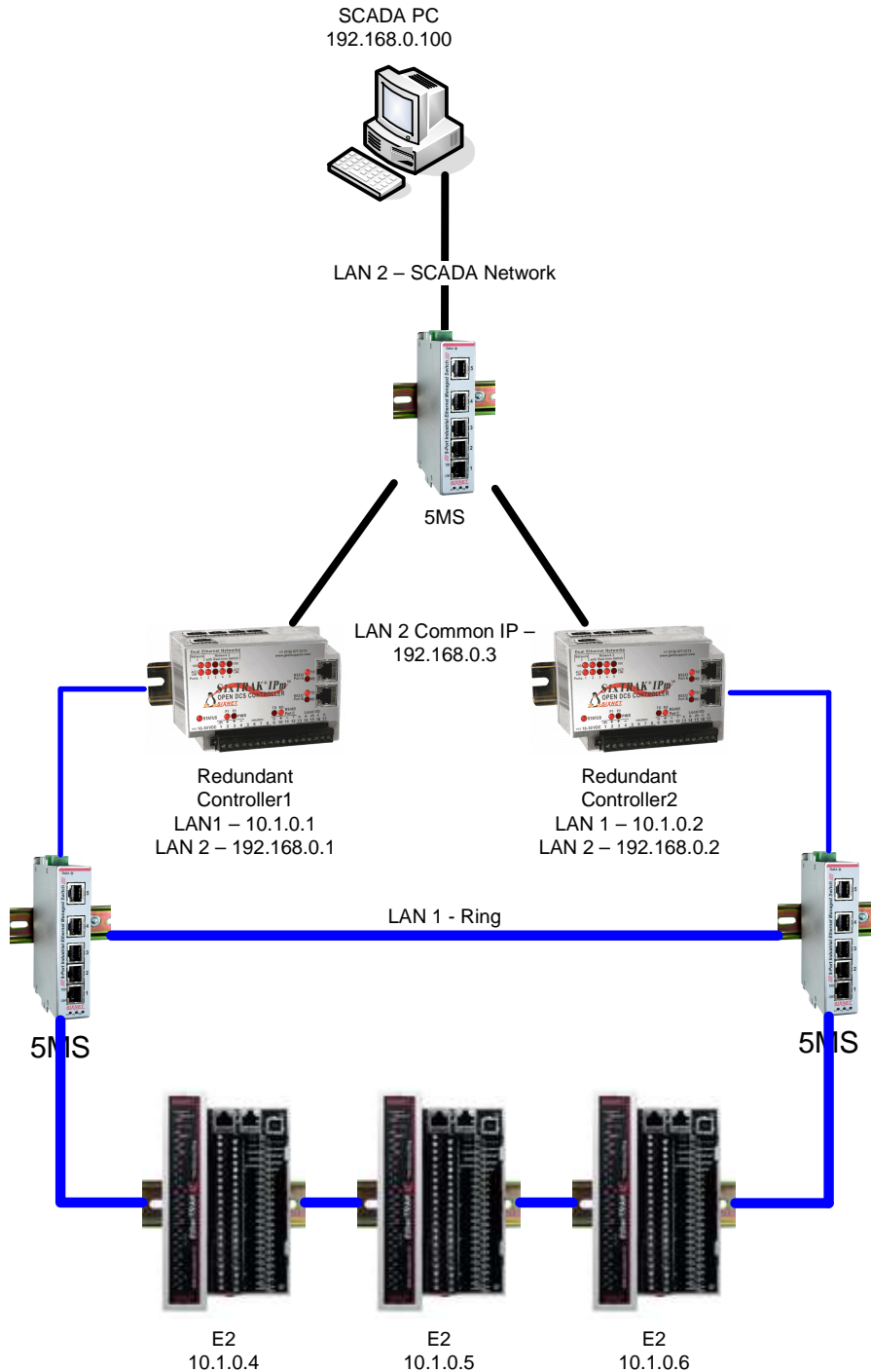
Back-up mode – Reading all inputs and outputs.

Back-up versus Active – When redundancy is enabled the controller is in one of the aforementioned states. By default the controller starts in back-up mode. The code that controls the will keep the controller in this state until all of the following conditions are met: (1) The LINK state on Ethernet port 1 is ON. (2) The Self_Heartbeat is pulsing within the tolerance of the “allowable loss period” setting. (3) The “Other_Active” variable is not ON.

Flexible. Reliable. Powerful.

Step-by-step guide to configure redundancy

It is important to start the configuration with a network plan. First, plan out the network including the IP address of the SCADA PC, controllers, EtherTRAK-2 and switches used. In this case the network is being designed as shown below.



Flexible. Reliable. Powerful.

Start with a blank configuration or a configuration without EtherTRAK-2 modules. To make configuration easier an abbreviated EtherTRAK-2 configuration wizard has been added. Configure the Controller (ST-IPM-6350 in this case) selecting the redundancy options.

IPm Remote Terminal Unit Station Configuration

Define General Properties:

Station Name:

Station number:

Serial number:

Station Type: Use customized product code:

Automatically configure a secondary controller
Secondary station name:

Enable built-in redundancy

SIXTRAK IPm Open DCS Controller
Features:

Next, Configure the LAN 1 IP address and LAN 2 back-up IP address.

Ports

Define Communications Port Properties:

Port	Settings
Ethernet1	Ethernet: 10.1.0.1
Ethernet2	Ethernet: 192.168.0.1
RS232 Port A	SIXNET Universal,9600,N,8,1,RS232
RS232 Port B	SIXNET Universal,9600,N,8,1,RS232
RS485 Port C	SIXNET Universal,9600,N,8,1,RS485

Flexible. Reliable. Powerful.

In the the I/O Modules tab the “redundancy module” will appear. This module holds all the station status module variables that the redundancy program in the controller firmware uses to determine what state it should be in. The starting address can be modified to whatever register is convenient. Keep in mind that the “Self_Heartbeat” variable needs to be pulsed on and off by an ISaGRAF program. The screen shot below shows the status bits with a starting address of 100.

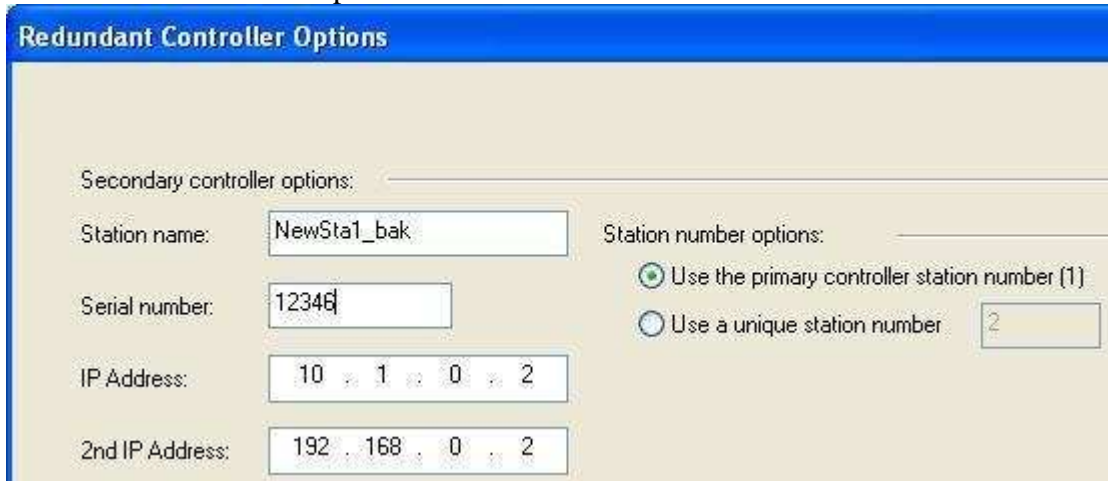
I/O Type	I/O Address	I/O Tag Name	OFF Message	ON Message	Feature	Tag
D OUT	Y100	Self_Heartbeat	Diastole	Systole	Self Heartbeat	
D OUT	Y101	Self_Active	Idle	Running	Self Active	
D OUT	Y102	Self_Backup	Idle	Backup	Self Backup	
D OUT	Y103	Other_Heartbeat	Diastole	Systole	Other Heartbeat	
D OUT	Y104	Other_Active	Idle	Running	Other Active	
D OUT	Y105	Other_Status	Alive	Dead	Other Status	
D OUT	Y106	Self_Status	Alive	Dead	Self Status	

Notice in the I/O Transfer window an automatic I/O Transfer was created by the Tool Kit so the Controllers would pass the station status module information to each other using the LAN 1 network.

Action Name	Defined Action	Local Type	Local Start	Remote Type	Remote Start	Number of Registers	Load Options
StaStatWrite	Write SIXNET I/O	From Discrete Out	100	To Discrete Out	103	2	both

Flexible. Reliable. Powerful.

In the redundant controller options tab configure the LAN 1 IP address of the second controller and the Back-up IP address of the LAN 2 interface.



Redundant Controller Options

Secondary controller options:

Station name:

Serial number:

IP Address:

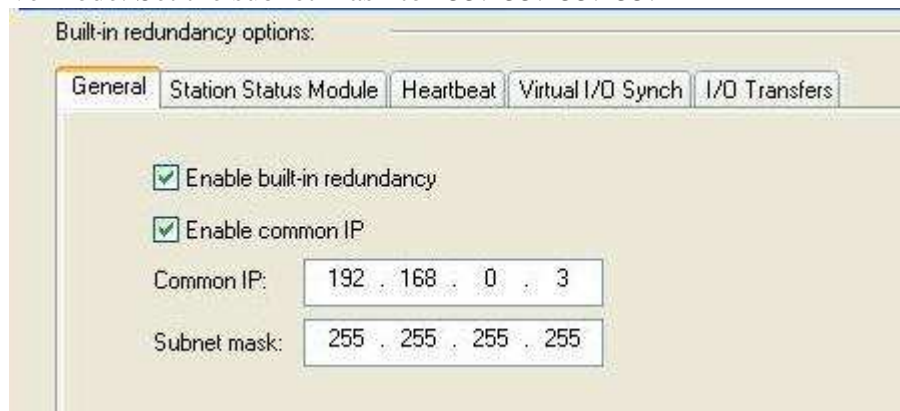
2nd IP Address:

Station number options:

Use the primary controller station number (1)

Use a unique station number

Next, in the same tab configure the common IP address if desired. It must be a third IP address that is unique on the LAN 2 network. This is the IP that the controllers when they are in active mode. Set the subnet mask to 255.255.255.255.



Built-in redundancy options:

General Station Status Module Heartbeat Virtual I/O Synch I/O Transfers

Enable built-in redundancy

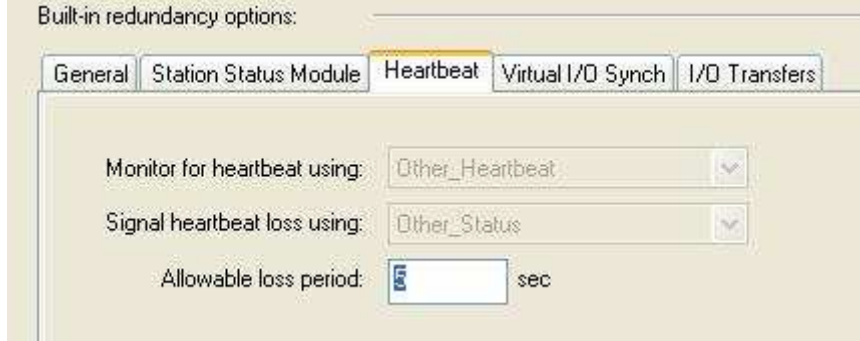
Enable common IP

Common IP:

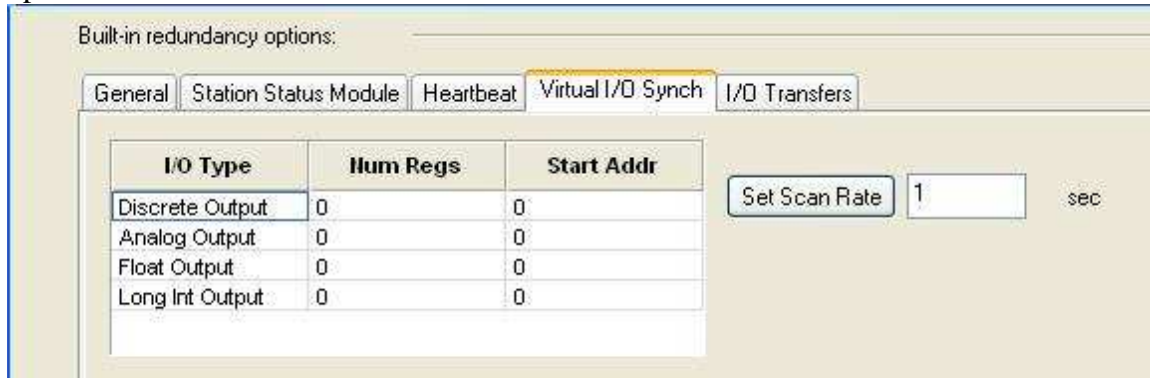
Subnet mask:

Flexible. Reliable. Powerful.

In the heartbeat tab select the allowable loss time. This is the time the redundancy code will use to detect the loss of the other stations heartbeat. This time should be greater than the pulse time of the heartbeat in ISaGRAF.

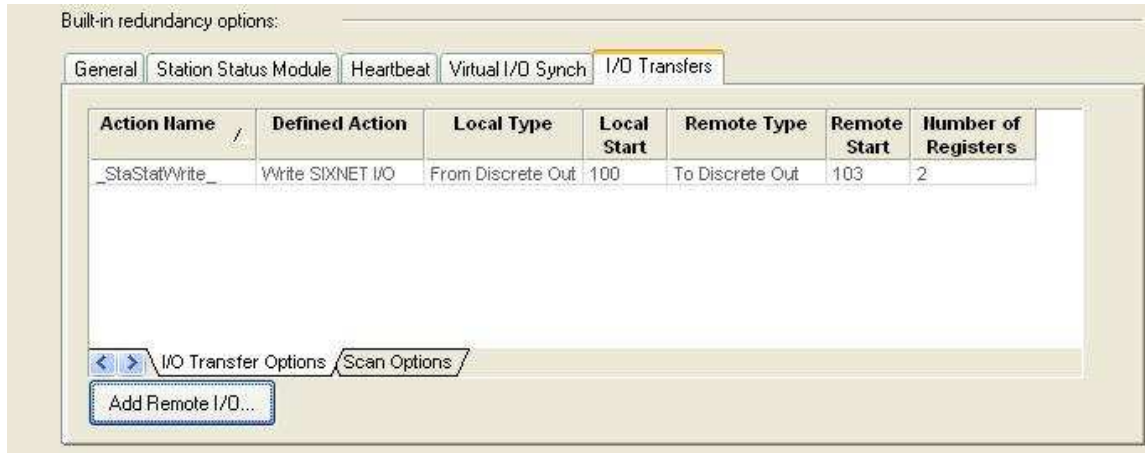


The Virtual I/O Synch tab can optionally set a series of write transfer between the two controllers active only when the controller is in the active mode. This allows any virtual outputs used in the ISaGRAF program to be synchronized between the active and back-up controller.

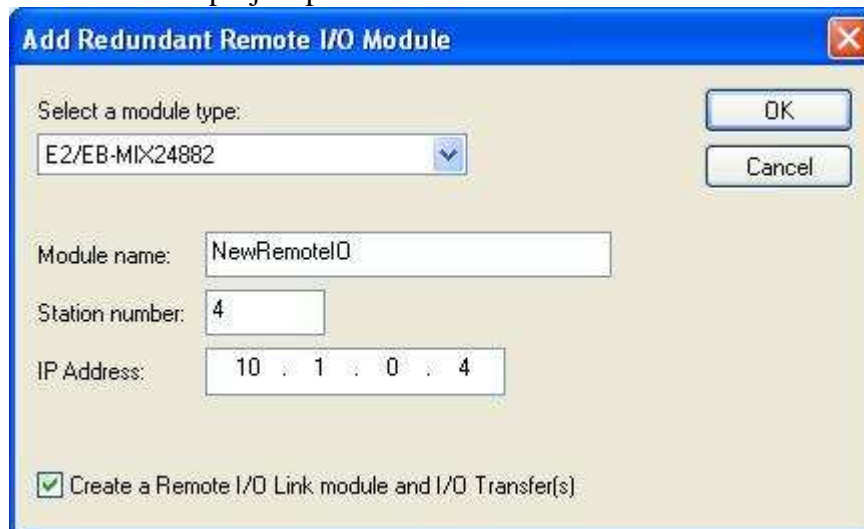


Flexible. Reliable. Powerful.

The I/O transfers tab shows a summary of the I/O transfers relating to redundancy. This is also the place to add EtherTRAK-2 I/O modules. Adding them here simplifies configuration by adding the necessary redundancy options. Simply click the “add remote I/O” button.

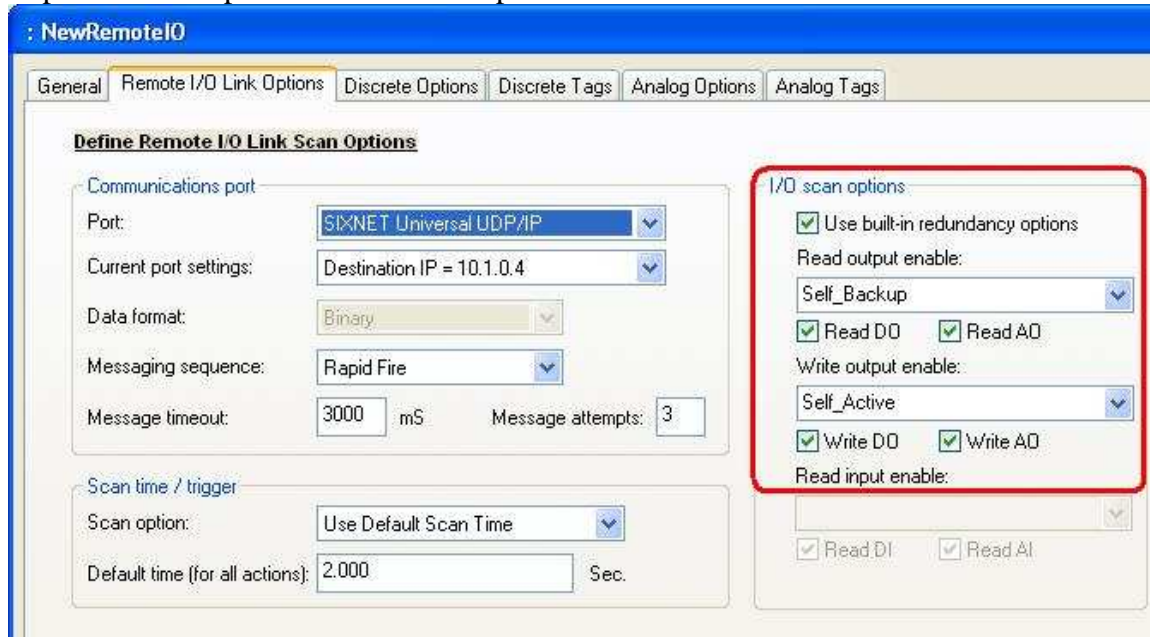


When the button is pressed the add redundant remote I/O configuration screen appears. Select the EtherTRAK-2 modules in the project and specify the unique station number and IP addressed from the project plan.

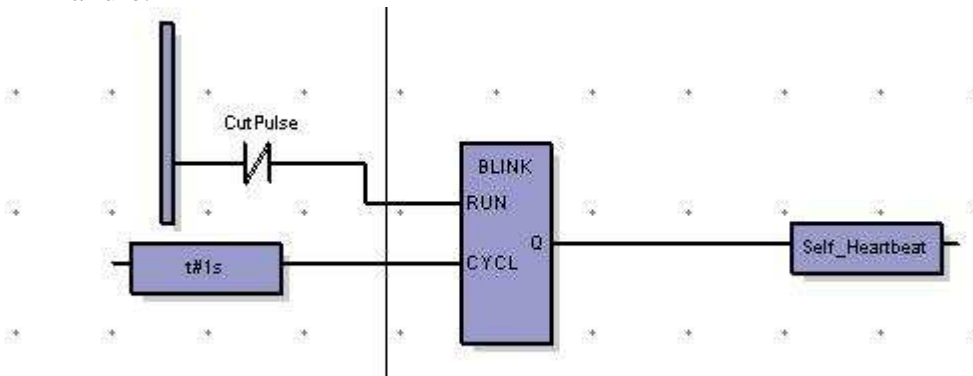


Flexible. Reliable. Powerful.

To view the redundant options automatically added to the EtherTRAK-2 module click finish. Then, go to the I/O transfer tab in the controller configuration and double-click on the remote I/O link transfer. In the remote I/O link options the I/O scan options has automatically been filled in. This allows both controllers to read inputs at all times, read outputs in back-up mode and write outputs in the active mode.



As mentioned previously in this document very little programming is required. An example of the ISaGRAF program required is shown below in FBD language. The program provides the heartbeat for the redundancy program to know whether the ISaGRAF program is running or not. One blink function block is required with a time variable input to determine the blink rate. The output of the block goes to the self_heartbeat variable in the redundancy module. In this example it is DO:100. An optional normally closed contact “CutPulse” variable can be added to simulate an ISaGRAF failure.



Flexible. Reliable. Powerful.