WOODWARD

Application Note 51620
(Revision -, 3/2020)
Original Instructions

Flex500 DR
Redundant GAP Application Design

General Precautions

Read this entire manual and all other publications pertaining to the work to be performed before installing, operating, or servicing this equipment.

Practice all plant and safety instructions and precautions.

Failure to follow instructions can cause personal injury and/or property damage.

Revisions

This publication may have been revised or updated since this copy was produced. To verify that you have the latest revision, check manual *26455*, *Customer Publication Cross Reference and Revision Status & Distribution Restrictions*, on the *publications page* of the Woodward website:

http://www.woodward.com

The latest version of most publications is available on the *publications page*. If your publication is not there, please contact your customer service representative to get the latest copy.

Proper Use

Any unauthorized modifications to or use of this equipment outside its specified mechanical, electrical, or other operating limits may cause personal injury and/or property damage, including damage to the equipment. Any such unauthorized modifications: (i) constitute "misuse" and/or "negligence" within the meaning of the product warranty thereby excluding warranty coverage for any resulting damage, and (ii) invalidate product certifications or listings.

Translated Publications

If the cover of this publication states "Translation of the Original Instructions" please note:

The original source of this publication may have been updated since this translation was made. Be sure to check manual *26455*, *Customer Publication Cross Reference and Revision Status & Distribution Restrictions*, to verify whether this translation is up to date. Out-of-date translations are marked with ⚠. Always compare with the original for technical specifications and for proper and safe installation and operation procedures.

Revisions— A bold, black line alongside the text identifies changes in this publication since the last revision.

Application Note 51620

Contents

The following are trademarks of Woodward, Inc.:
ProTech
Woodward

The following are trademarks of their respective companies:
Modbus (Schneider Automation Inc.)
Pentium (Intel Corporation)

Illustrations and Tables

Warnings and Notices

Important Definitions

This is the safety alert symbol used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

- **DANGER** - Indicates a hazardous situation, which if not avoided, will result in death or serious injury.
- **WARNING** - Indicates a hazardous situation, which if not avoided, could result in death or serious injury.
- **CAUTION** - Indicates a hazardous situation, which if not avoided, could result in minor or moderate injury.
- **NOTICE** - Indicates a hazard that could result in property damage only (including damage to the control).
- **IMPORTANT** - Designates an operating tip or maintenance suggestion.

⚠WARNING

Overspeed / Overtemperature / Overpressure

The engine, turbine, or other type of prime mover should be equipped with an overspeed shutdown device to protect against runaway or damage to the prime mover with possible personal injury, loss of life, or property damage.

The overspeed shutdown device must be totally independent of the prime mover control system. An overtemperature or overpressure shutdown device may also be needed for safety, as appropriate.

⚠WARNING

Personal Protective Equipment

The products described in this publication may present risks that could lead to personal injury, loss of life, or property damage. Always wear the appropriate personal protective equipment (PPE) for the job at hand. Equipment that should be considered includes but is not limited to:

- **Eye Protection**
- **Hearing Protection**
- **Hard Hat**
- **Gloves**
- **Safety Boots**
- **Respirator**

Always read the proper Material Safety Data Sheet (MSDS) for any working fluid(s) and comply with recommended safety equipment.

⚠WARNING

Start-up

Be prepared to make an emergency shutdown when starting the engine, turbine, or other type of prime mover, to protect against runaway or overspeed with possible personal injury, loss of life, or property damage.

Electrostatic Discharge Awareness

NOTICE

Electrostatic Precautions

Electronic controls contain static-sensitive parts. Observe the following precautions to prevent damage to these parts:

- **Discharge body static before handling the control (with power to the control turned off, contact a grounded surface and maintain contact while handling the control).**
- **Avoid all plastic, vinyl, and Styrofoam (except antistatic versions) around printed circuit boards.**
- **Do not touch the components or conductors on a printed circuit board with your hands or with conductive devices.**

To prevent damage to electronic components caused by improper handling, read and observe the precautions in Woodward manual *82715*, *Guide for Handling and Protection of Electronic Controls, Printed Circuit Boards, and Modules*.

Follow these precautions when working with or near the control.

1. Avoid the build-up of static electricity on your body by not wearing clothing made of synthetic materials. Wear cotton or cotton-blend materials as much as possible because these do not store static electric charges as much as synthetics.
2. Do not remove the printed circuit board (PCB) from the control cabinet unless absolutely necessary. If you must remove the PCB from the control cabinet, follow these precautions:
   - Do not touch any part of the PCB except the edges.
   - Do not touch the electrical conductors, the connectors, or the components with conductive devices or with your hands.
   - When replacing a PCB, keep the new PCB in the plastic antistatic protective bag it comes in until you are ready to install it. Immediately after removing the old PCB from the control cabinet, place it in the antistatic protective bag.

Chapter 1.
General Information

Introduction

The primary focus of this Application Note is to provide some guidance to GAP application programmers on how to design a redundant GAP application using the Flex500 redundant control. For details on the correct hardware part numbers, chassis configuration settings and required control interlock wiring, refer to the Flex500 manual (**26838**). The GAP Help Files (Software p/n **9927-2520**) also contain a large amount of helpful information. In the **Flex500 GAP Programmer Block Help** – See "**FLEX DR Getting Started**" section.

This manual is intended for users that are reasonably familiar with GAP programming. They may or may not be familiar with the elements of redundancy programming. The implementation of redundancy in the Flex500 platform is considerably different than that of a Micronet Plus control. While some items like CAN communication links are handled the same, many other components of the logic are not handled the same.

Two Flex500 controllers can be applied together to function in a redundant manner to increase overall system reliability and availability. In such applications, one Flex500 functions as the SYSCON (In-Control) unit and controls all outputs of the system. The second Flex500 functions as a BACKUP unit and tracks the SYSCON unit's operating parameters to ensure a smooth transfer if the SYSCON unit fails.

The Flex500 uses the term Primary to describe the unit with DIP Switch position 0001 and the term Secondary to describe the unit with DIP Switch position 0002 (please refer to Appendix A in the Flex500 hardware manual **26838** for DIP switch configuration instructions). The Primary and Secondary unit designations allow the system to identify each unit specifically. The term SYSCON is used to describe the unit that is currently in-control of the system and the term BACKUP to describe the tracking unit. Either of the Primary or Secondary units can become the SYSCON unit, but in a healthy system, the Primary unit will always boot up initially as the SYSCON.

The Flex500 operating system continuously keeps the BACKUP unit in-sync with the current control state of the SYSCON. On a control transfer, the BACKUP unit becomes the new SYSCON in the exact same state as the previous unit just prior to the transfer. The previous SYSCON will then become the BACKUP unit and begin tracking the SYSCON in the same way. Once the transfer occurs, the new SYSCON begins controlling the system processing its local IO. The system is designed to have identical IO signals between both the Primary and Secondary units such that either unit can become SYSCON with no change in the system control state. In the case of an IO signal discrepancy between the SYSCON and BACKUP units, application logic is required to annunciate these conditions.

The Operating System (OS) will transfer SYSCON control upon the following conditions:

- SYSCON unit failure (CPU or internal problem, OS)
- Loss of power to the SYSCON unit

Other References

9927-2520	**GAP Help File**
26838 Rev D or later	**Flex500 Digital Control – Installation and Operation Manual**
35018V3	**Redundant 505XT Digital Control for Steam Turbines**

Chapter 2.
Application Requirements

Hardware Interlocks

There are three required hardware interlocks to allow two Flex500 units operate as a redundant control.

- Setting the chassis configuration DIP switches on the top of the unit to assign 1 unit as Primary (0001) and the other as Secondary (0010)
- Connect an Ethernet cable between the Ethernet 4 ports of each unit (DR_COMMS link)
- Connect Discrete power and Discrete Input (DI) 20 of each unit through the NO & COM terminals of Relay 8 on the other unit (CRISSCROSS)

NOTICE

As shipped from the factory, the default setting of the DIP switches on the Flex500 hardware will be 0000 which supports simplex GAP applications.

DIP Switch Settings

In regards to the bullet points above, Ethernet port 4, DI channel 20 and Relay 8 are therefore not available for use in the GAP application. The DI and Relay however will need to be programmed in the application, as show in the information in this chapter.

New Application Software Blocks (Coder 1.04)

There are five new blocks used in the design of a redundant application. The first three blocks are required, the last two are required only if the analog outputs or actuator outputs of the Flex500 hardware are programmed in the application.

Once these first three blocks are added to the application, the GAP will become a redundant application and require the hardware interlocks described above. It will function in a “Run Alone” mode, but will not run as a simplex application, it will require the DIP switch settings be adjusted for Primary or Secondary identity.

Below is a brief description of each.

FLEX_DR

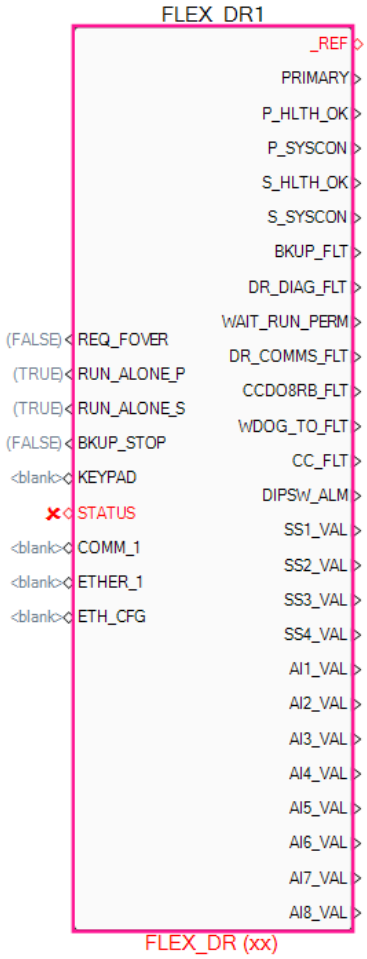

Fig 2-1. Second Chassis Block FLEX_DR

This block defines the second chassis in the system, and is required to be connected to the FLEX_DR input field on the FLEX_500 block (it is a child block requiring the FLEX_500 block as a parent). This block will require that an additional STATUS_FLEX block be added and connected to the STATUS input field.

This block has inputs & outputs to provide:

- Outputs of all the health status information of the I/O on the Backup unit
- Outputs of the status information related to the redundant operation and OS detected errors
- Inputs to trigger transfers of SYSCON control, resetting of the Backup and allowing solo operation
- Inputs for assigning the serial and Ethernet communication ports on the Backup unit

It is important to keep in mind that the FLEX_DR block always shows outputs from the Backup unit, which can be either the Primary or the Secondary chassis at any given time.

BI_FLEX_DR and RELAY_FLEX_DR Blocks –

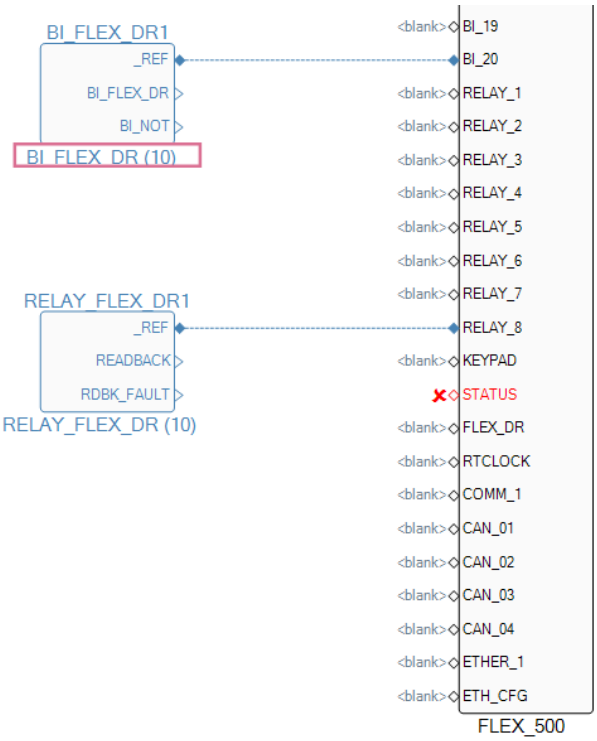

Fig 2-2. Criss-Cross DI to DO Interlock Blocks

These blocks are required as placeholders for the Criss-Cross DI-to-DO interlock and provide status information of these channels to the GAP application. In a redundant application, these are the only block types that will be accepted by BI channel 20 and Relay 8 on the FLEX_500 chassis block.

ACT_FLEX_DR (If These Output Channels are Used)

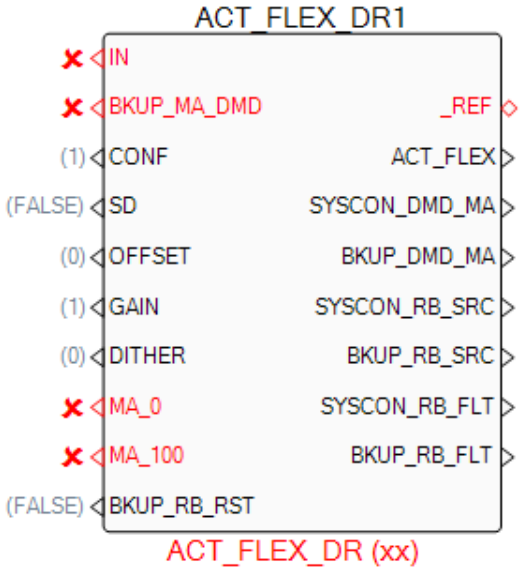

Fig 2-3. Actuator Driver Block ACT_FLEX_DR

For redundant applications a new actuator block was created to provide programming handles for the output demand current that will be supplied by the Backup unit in order to allow the OS to monitor this channel on the Backup unit and provide health status information. The typical setting of the BKUP_MA_DMD input is to be one-half of the MA_0 value. This block will control the output demands of both of the circuits for this channel on both control units. This block will provide the correct output from the SYSCON dependent on the health of the Backup circuit which is reflected by the BKUP_RB_FLT field. The programmer will need to use the SYSCON_RB_FLT output to trigger a SYSCON transfer in the GAP application.

AO_4_20_FLEX_DR (If These Output Channels are Used) –

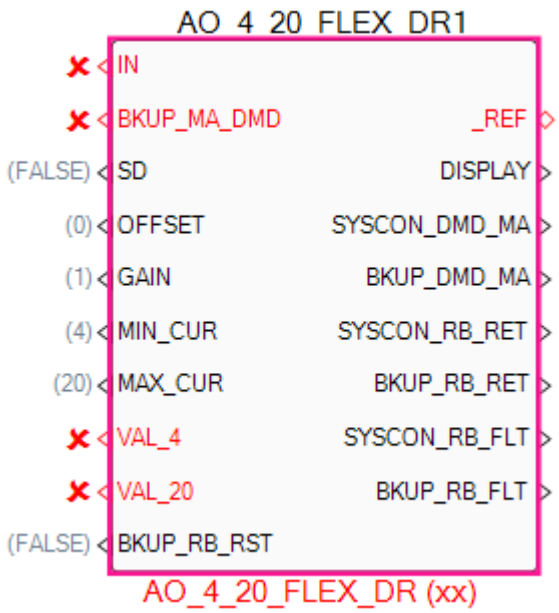

Fig 2-4. Analog Output Block AO_4_20_FLEX_DR

For redundant applications a new analog output block was created to provide programming handles for the output demand current that will be supplied by the Backup unit in order to allow the OS to monitor this channel on the Backup unit and provide health status information. The typical setting of the BKUP_MA_DMD input is to be one-half of the MIN_CUR value. This block will control the output demands of both of the circuits for this channel on both control units. This block will provide the correct output from the SYSCON dependent on the health of the Backup circuit which is reflected by the BKUP_RB_FLT field. The programmer will need to use the SYSCON_RB_FLT output to trigger a SYSCON transfer in the GAP application.

Application Software Logic

To design a redundant application in the Flex500 platform the following items need to be addressed:

1. Adding the required blocks from above
2. Adding logic to provide annunciation and detect faults on the Backup controller
3. Adding logic to trigger transfers of SYSCON control upon certain faults
4. Create logic to inhibit any transfer of SYSCON when the Backup unit is not healthy
5. Adding logic to allow either unit to run in ‘solo’ mode when other unit is not available
6. Consider how Ethernet and serial communication ports will be assigned, if redundancy is required
7. Consider how best to handle I/O (local Flex500 channels or Distributed I/O or some of both)

The following sections below will describe each of these in a little more detail.

Adding Required Blocks

This is pretty straight forward. If starting with an existing, functional simplex application, it is good to create a new module in the application as a container for the redundancy logic. In the 505DR application we created a module named REDUN.

Below is an example of the FLEX_DR block and the associated STATUS_FLEX block in this module. On the chassis block, you can see that almost all of the outputs are used by other logic to determine the health of the chassis.

In regards to the STATUS_FLEX block, the logic on the 2 CPU LED inputs was designed to provide a quick visual reference on the control as to which unit is the SYSCON and which is the Backup. The comment above the block describes the resulting annunciation of these LED's.

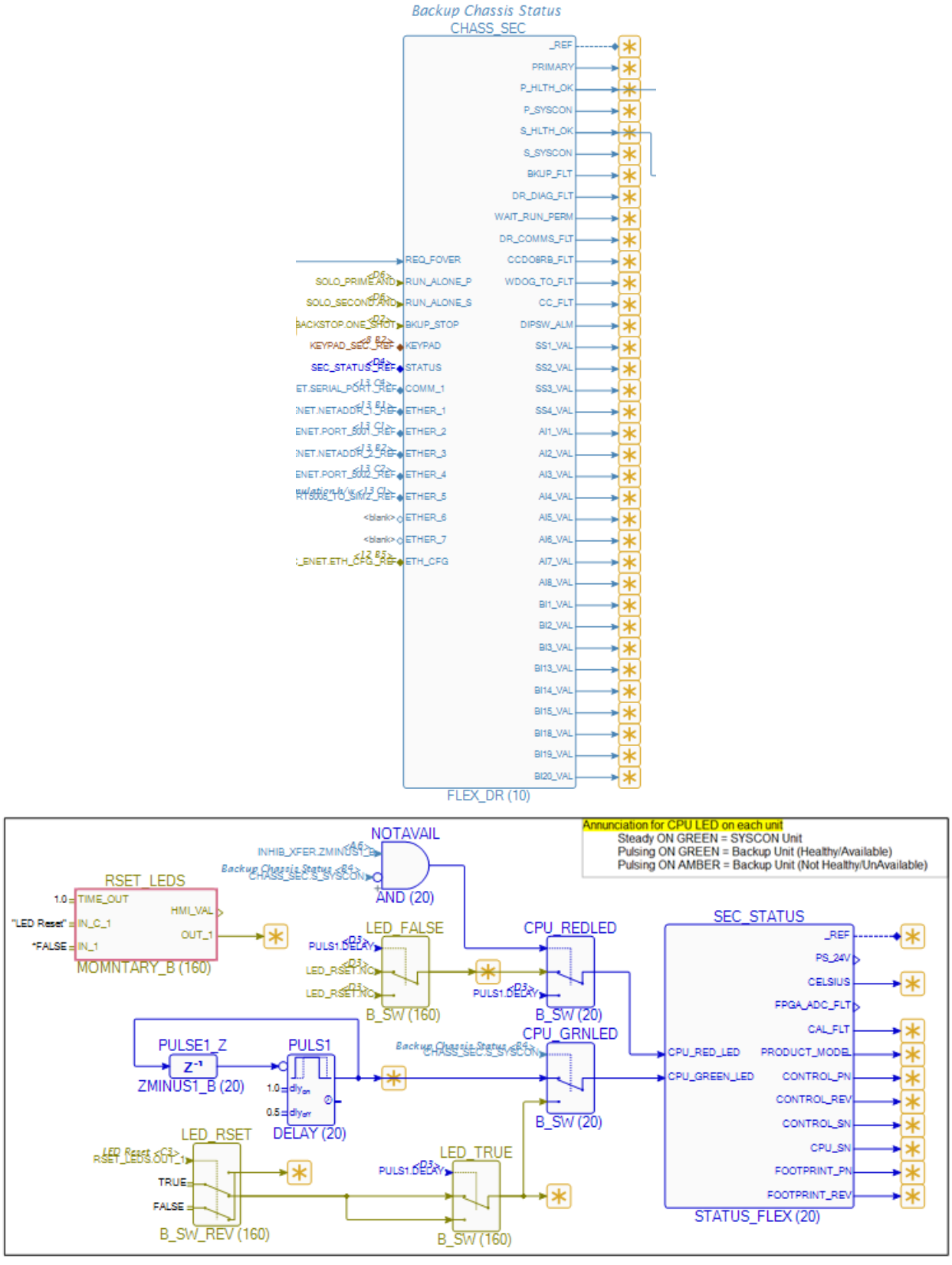

Fig 2-5. Secondary Chassis Status Block & LED Logic

Backup Chassis Health

The majority of the outputs of the FLEX_DR block are used to determine the condition of the Backup unit I/O and Operating System indications like the DR interlocks, DIP switch checks and identification of whether the Primary or the Secondary unit is the SYSCON. The block does not contain status of the analog or actuator output channels that information is found on the new channel blocks that were created in this template.

Below are some examples of fault detection logic on analog input, discrete inputs and relay outputs.

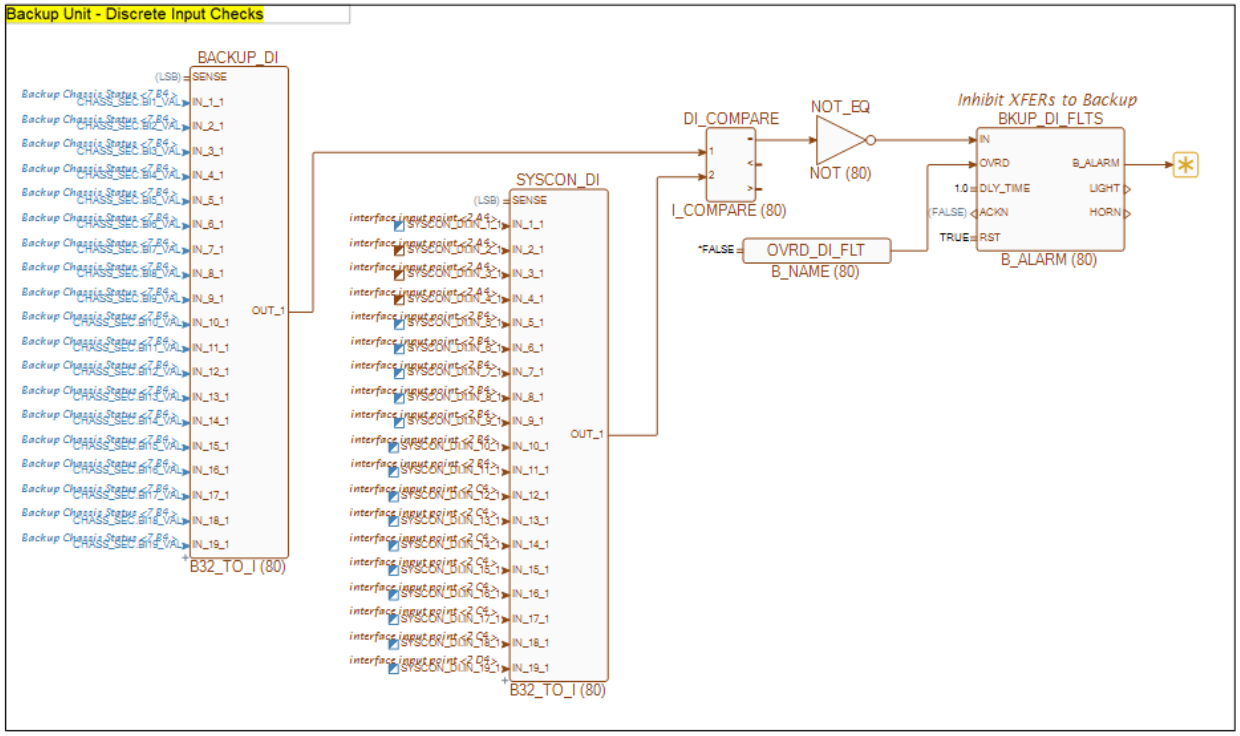

Fig 2-6. Discrete Input difference detection

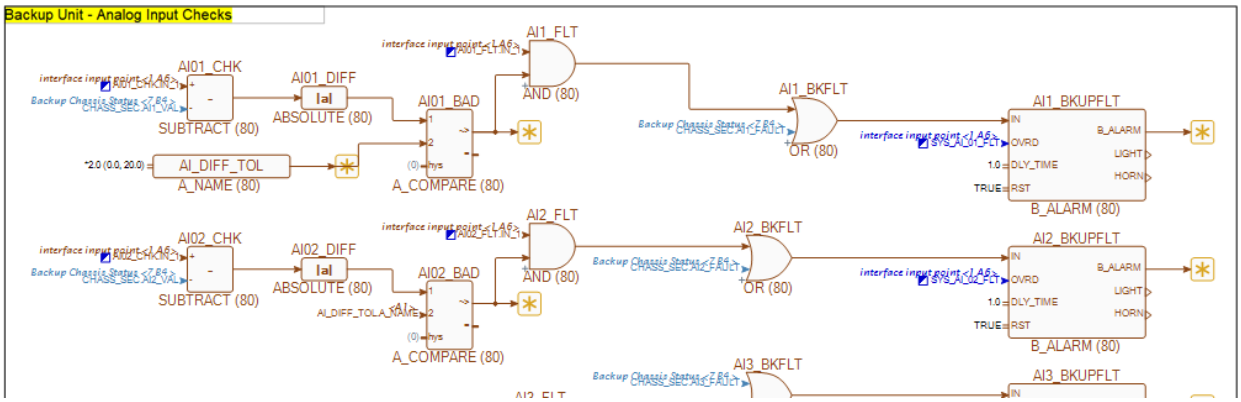

Fig 2-7. Detecting a Fault or Signal Difference on Analog Inputs

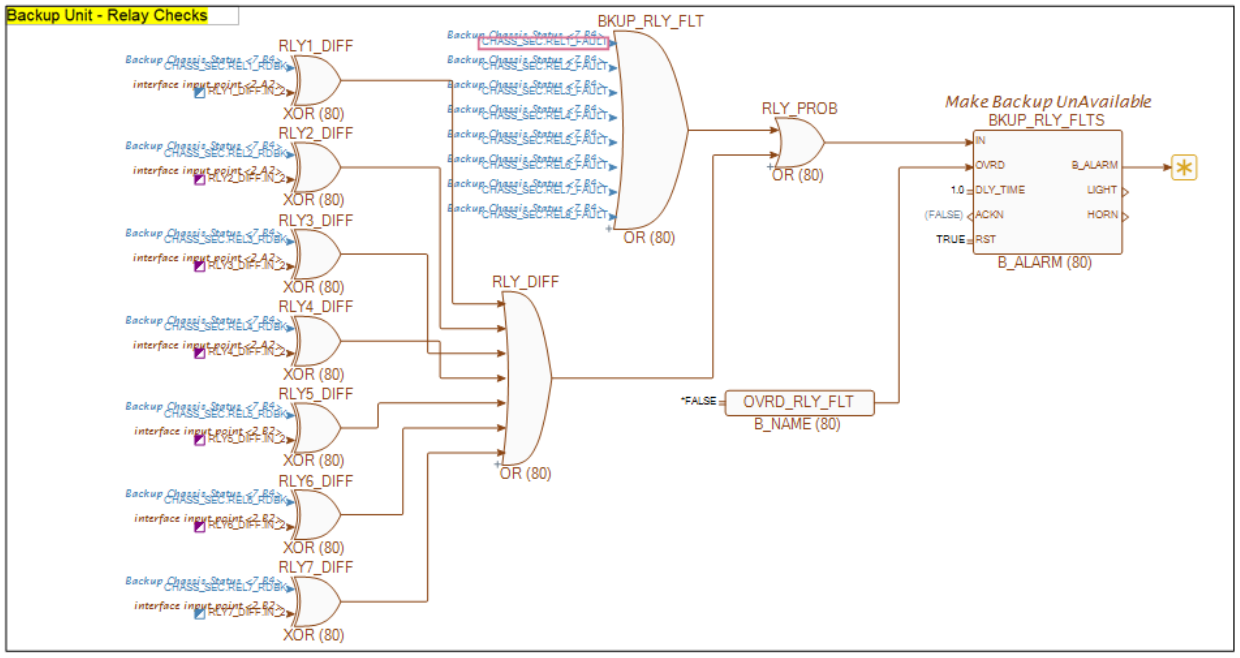

Fig 2-8. Relay Output Difference Detection

Detection of a problem on the speed inputs is a little more complicated since the speed seen on the Backup unit will be lagging the actual speed on the SYSCON unit. Below is an example of what we chose to do in the 505DR application.

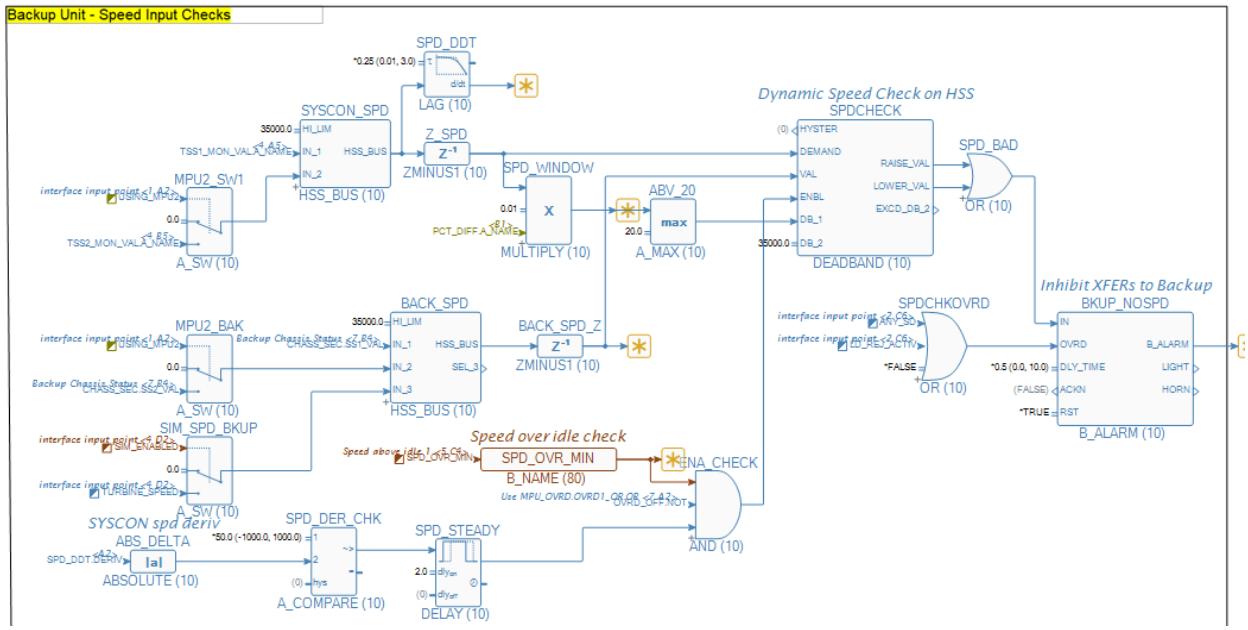

Fig 2-9. Speed Input Difference or Signal Fault Detection

Application Triggered Transfer of SYSCON Control

It is important to emphasis that successful redundancy requires careful consideration of what events, that only the application is aware of, must trigger a transfer of SYSCON control (XFER) to the Backup unit. For example – pulling the power input off of the SYSCON unit is handled by the OS, not the application, so that XFER will occur without any logic added to the GAP. However, an actuator connector (or loose wire) pulled off of the SYSCON, must be detected by the application and an XFER initiated.
Two items that must always initiate an XFER are analog outputs (4-20 or Actuators) or any CAN port connection to the SYSCON controller.

Actuator (or Analog Output) Signal Fault –
In the logic below the SYSCON_RB_FLT is used to trigger a SYSCON XFER, when the actuator connector (or a wire) is pulled from the SYSCON. Once the XFER is complete, the new SYSCON will not have a SYSCON_RB_FLT, but it will show the BKUP_RB_FLT as TRUE. If the actuator signal was actually failed at the field end then the control will annunciate a complete signal loss.

This same logic is applied to analog output signals using the AO_4_20_FLEX_DR block.

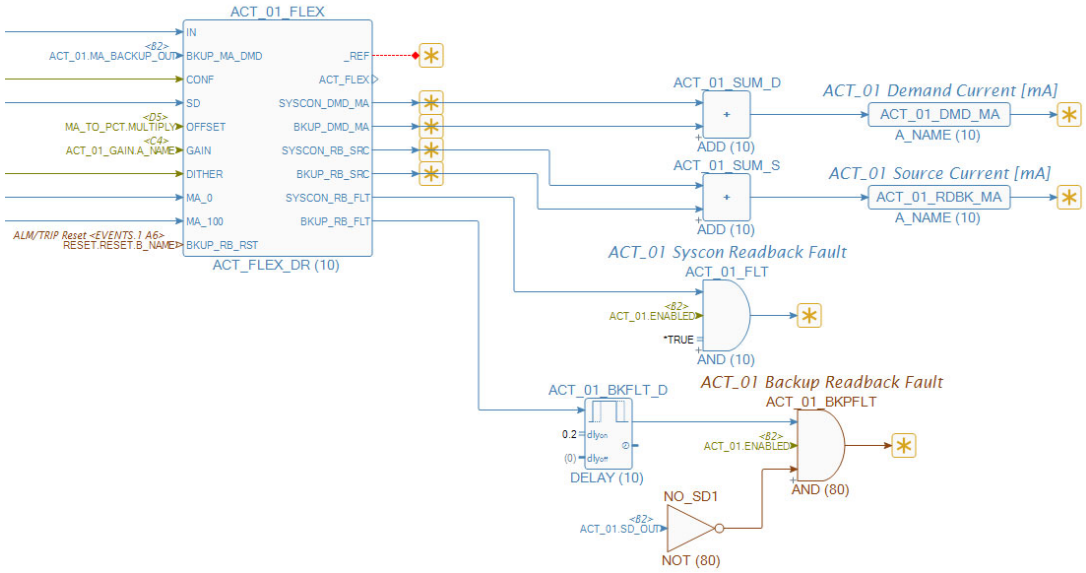

Fig 2-10. Actuator SYSCON_RB_FLT to Trigger an XFER

CAN Port Fault –
The logic below is an example of how to initiate a SYSCON XFER using the LNK_ALM output of the CAN_P_STAT block. This is an RTC Node network with some nodes in 10ms Rate Group. It is critical that the A_TIMEOUT be set so that a LNK_ALM comes in when a single message is lost so that an XFER can be completed prior to the E_TIMEOUT which would cause a LNK_ERR (which could cause a TRIP).

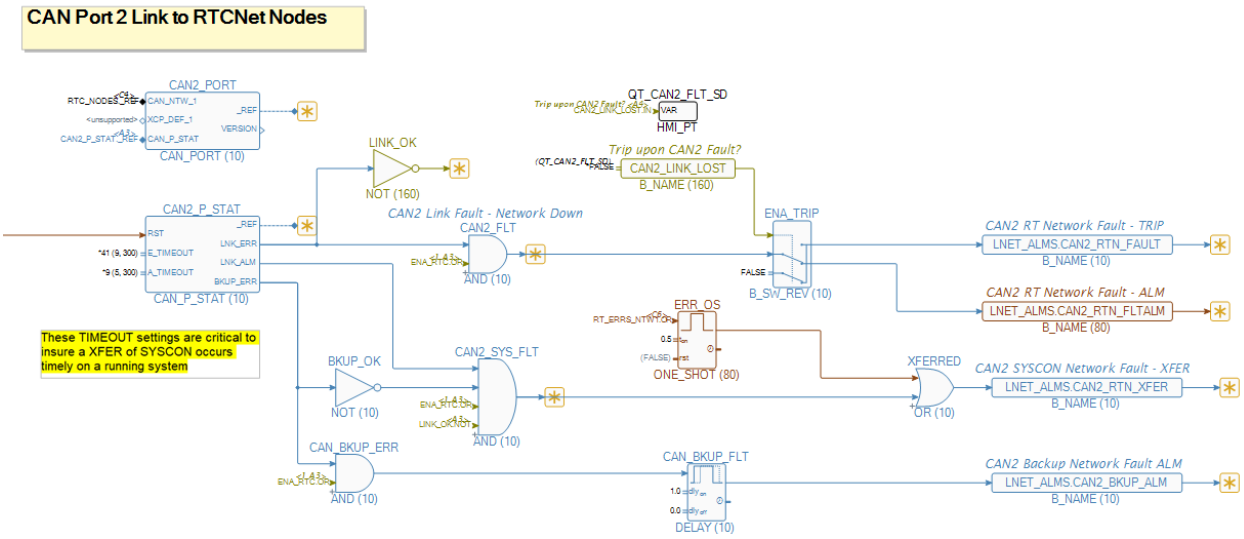

Fig 2-11. CAN port LNK_ALM to trigger an XFER

When using CAN:
– The LNK_ALM (related to A_TIMEOUT) output should be used to trigger an application XFER of SYSCON control to the other unit. This must happen before the LNK_ERR (related to E_TIMEOUT) output goes TRUE

– The settings of the E_TIMEOUT and A_TIMEOUT fields of the CAN_P_STAT block are critical to surviving CAN communication faults, these must be set related to the Rate Group settings of the NODES. Example for a network with nodes in RG 10 – set A_TIMEOUT = 9 to x and E_TIMEOUT = 39

- The default boot-up/initialization of the control is to make the Primary unit the SYSCON and the Secondary unit the Backup. If there is a fault in the CAN link on the Primary unit at boot-up – then application logic or user intervention will be required to get the controls up to a healthy state

Application Transfer Triggers –
This is the summary logic of all the events that can trigger a SYSCON XFER. In this logic speed, analog inputs, operator and CORE logic triggers were added to the trigger list. The APPL_XFER.AND block is used as the handle to inhibit an XFER if the Backup unit is not available. When the REQ_FOVER is pulsed (edge triggered input), the OS will immediately transfer control of SYSCON to the Backup unit.

Any Application XFER's should be ONE_SHOTS and go nowhere else in logic

Blocks with "states" that are involved with XFER's should also go nowhere else in logic (Example: If the APPL_XFER.AND block below is used to trigger a ONE_SHOT or a DELAY block the durations will not be followed since this unit will stop being SYSCON and Backup will not have this block TRUE)

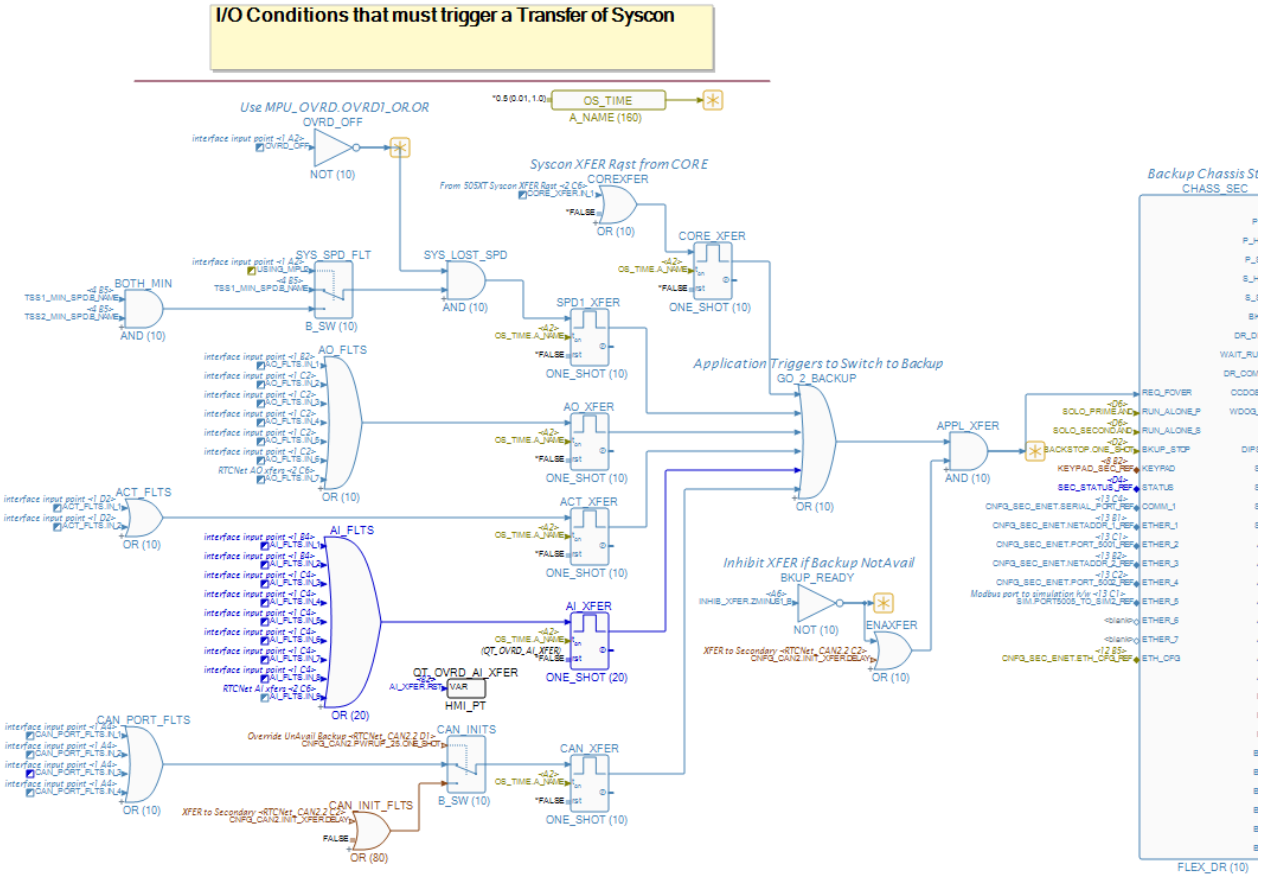

Fig 2-12. Summary of Application Events to Trigger an XFER

Ability to Run a Single Unit When the Other is Unavailable

The FLEX_DR block has two inputs that are available to allow either the Primary or the Secondary unit to assume the authority of the SYSCON controller when the OS does not detect the other unit. The main conditions that cause this are the DR_COMMS_FLT (ENET 4 link faulted) or the CC_FLT (Criss-Cross DI-to-DO link faulted).

The intended use for these triggers are when the control has booted up and has not successfully found the other chassis. During normal operation the control will handle

When the OS detects this condition:

- the GAP and GUI applications will initialize
- the unit will be the SYSCON
- the OS will hold the unit in IOLOCK (front LED will be RED)
- the WAIT_RUN_PERM output on the FLEX_DR block will be TRUE

When the control is in this condition and is configured as the Primary chassis, pulsing the RUN_ALONE_P input will remove IOLOCK. If the unit is configured as the Secondary chassis, pulsing the RUN_ALONE_S input will remove IOLOCK. These inputs should never be pulsed if the ENET4 and Criss-Cross links are healthy.

The logic below is an example of what conditions were used in the 505DR to run alone on 1 unit.

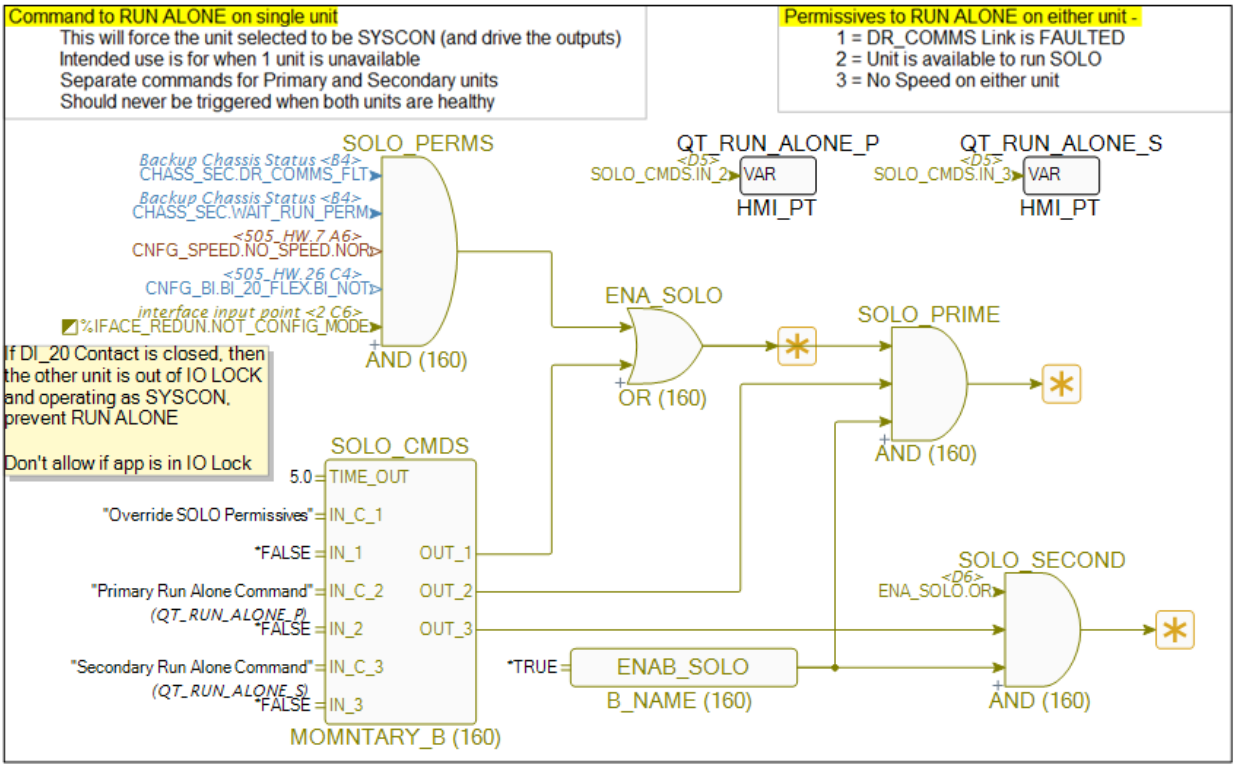

Fig 2-13. Permissive Logic to Trigger 1 Unit to Run Alone as SYSCON

Other Communication Links and Distributed I/O

The communications to other devices (typically Modbus or EGD) can be handled similar to way we do this in MicroNet redundant and MicroNet TMR systems. Often the master or consumer end can switch between IP addresses when a fault is detected. The data from either control will be the same since all output states are determined by the SYSCON controller. Thus there is no need to worry about switching communication links dependent on which unit is SYSCON and which is Backup. The other device only needs to switch upon a failure of a communication link. For example, if an HMI is connected to a controller that gets powered-down, then it will need to switch to the IP address of the other unit.

Using Woodward distributed I/O via CAN in redundant applications provides some extra benefits other than just expanding the number of I/O channels. The CAN links of the controls are forced redundancy, meaning that in the application there are only 4 CAN ports. Nodes must be linked to the same CAN port on both units. Using CAN port 1 as an example, recommended wiring is Primary CAN1 with termination resistor to CAN1 port on Secondary unit (no term resistor) then the first node, next node then last node with a termination resistor.

Using RTCNet or LinkNet HT nodes has these advantages:

- Field analog signals will not require isolation diodes (unlike any signals added directly to the Flex500 controls that are not coming thru the DR-FTM).
- Performance of I/O on the CAN network is completely bumpless, in terms of when switching of SYSCON control occurs between the units (this is also true for digital drivers on CAN networks)

Redundant communication links can be supported by adding this to the GUI main.qml file. If this is done then RemoteView will allow a second link to be configured when the tool properties are setup from the PC.

```
/*****************************************************************************************************************************
 *Device Connection Settings
 *****************************************************************************************************************************/
                devices: [
                    Device {
                        name: "MyDev"
                        id: mydev
                        sid:
                            SidRemote{                          //Use when using a remote sid definition.
                                cached:true

                            }

//                            SidFile{                           //Use when using a local sid file.
//                                cached: true
//                                fileName: "redSID.txt" //Use for HW connection.
//                            }

                        connection: RedundantConnection{
                            id: redund

                            ServlinkTcp {
                                id:remlink1
                                ip: "127.0.0.1"
                                port:666
                                updateRate: 200
                                readMultiple: true
                            }
                            ServlinkTcp {
                                id:remlink2
                                ip: "127.0.0.2"
                                port:666
                                updateRate: 200
                                readMultiple: true

                            }
                            useAlwaysFirst: (ApplicationType == "ControlSimulator") ? false : true

                            onStateChanged: {
                                if(redund.state === 3  ){
                                    if(last_login_name !== ""){
                                        auto_reconnect.start()
                                    }
                                }
                            }
                        }
                    }
                ]
```

Fig 2-14. GUI main.qml Device Declaration for Redundant Links

Other Helpful Hints

- We found the CPU LED annunciation logic (described above) to be extremely helpful as a visual indication of which unit was in control (SYSCON) – works on units with or with front panel display
- It is extremely important to test and verify the correct TIMEOUT settings needed on the CAN port blocks to insure successful transfers of SYSCON when the unit or port connection fails
- Using the DR-FTM & cable harnesses greatly simplifies things, the hand wiring with diodes was very annoying
- The Revision F version of RemoteView supports redundant connections to the control and also supports having an audible alarm (enabled in Display Properties) on the PC side (must be enabled in Service Menu/Alarms on the control)
- Once an application has been made redundant, the Online Change block (OLC_STAT) can be added if that is required for the target application
- The OS enforces a 12 second duration between application XFER's, triggered by the REQ_FOVER command
- The DR-DTM requires input power (24vdc) to power the Relays #6 and #7. If this power is lost, or is not provided these 2 relays will drop out (de-energize). All analog and discrete input signals will be unaffected by loss of power.

Revision History

New Manual—

-

Released

We appreciate your comments about the content of our publications.

Send comments to: icinfo@woodward.com

Please reference publication 51620.

B 5 1 6 2 0 : -

PO Box 1519, Fort Collins CO 80522-1519, USA
1041 Woodward Way, Fort Collins CO 80524, USA
Phone +1 (970) 482-5811

Email and Website—www.woodward.com

Woodward has company-owned plants, subsidiaries, and branches, as well as authorized distributors and other authorized service and sales facilities throughout the world.

Complete address / phone / fax / email information for all locations is available on our website.