2500M

I/O Module Redundant Configuration

Handbook

HA029462 Issue 3

June 2015

Contents 2500M REDUNDANT CONFIGURATION HANDBOOK

CHAPTER 1 INTRODUCTION

The 2500M (I/O Modules) provide the basic hardware interface to plant transducers and sensors. The Terminal Units provide the physical wiring connections, while the I/O modules perform the data conversions required - analogue or digital, current or voltage, input or output. The modules also offer an electrical isolation barrier that ensures safety and simplifies system wiring reducing interference and cross-talk effects. All working from a 24Vdc supply.

1.1 MANUAL CONTENTS

This manual is divided into the following chapters:

1.2 OTHER INFORMATION SOURCES

For details of Local Instrument Network (LIN) based Function Blocks, their parameters and input/output connections refer to the *LIN Blocks Reference Manual* (Part no. HA 082 375 U003) which explains how Control Strategy function blocks are selected and interconnected etc. The creation, monitoring and On-line Reconfiguration of LIN Databases and LIN Sequential Function Charts (SFCs) is described in the *LINtools Help* (Part no. RM 263 001 U055). SFCs do not support On-line Reconfiguration, but can be stopped via commands wired to the SFC_CON block. When the SFC has stopped it can be edited as usual, downloaded to the instrument and restarted on-line via the Network . The *ELIN User Guide* (Part no. HA 082 429) gives full details of installation, and how to configure an ELIN network, including setting the IP address using the instruments internal configurator.

Note If you do not possess any documents stated please contact your distributor.

1.3 THE INSTRUMENT

The 2500 and T2550 instruments comprises the Base Unit (2500B or T2550B), holding up to 16 I/O modules (2500M), and either a single (Simplex) IOC module (2500E or T2550S) or a pair of (Duplex) modules (T2550R).

Note Simplex configuration uses 1 IOC module that can be fitted to either a Simplex (single width) or Duplex (double width) Terminal Unit. A blanking plate should be fitted in the vacant right-hand position. Redundant configuration uses 2 IOC modules.

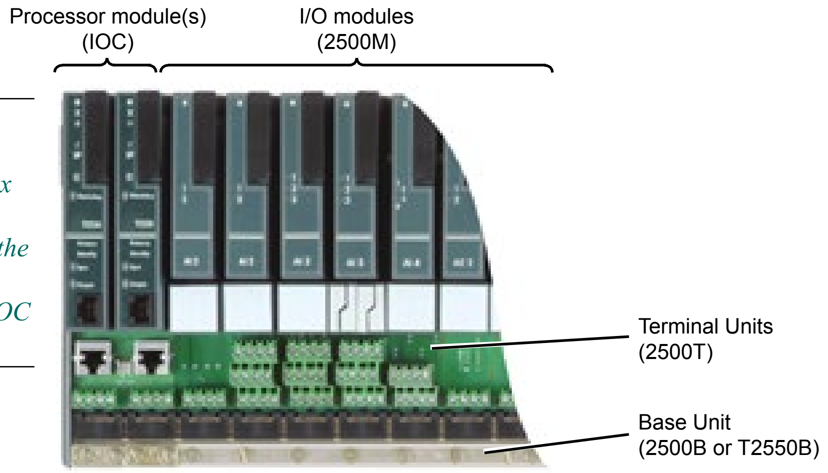


Figure 1.1 Redundant (T2550R) Modules configuration with I/O Modules (16) on the Base Unit

1.3.1 Typical applications

The instrument is designed for process control, recording and automation applications using local input/output modules. A number of these instruments can be networked together, allowing thousands of I/O points to be monitored and controlled.

Type	Description	Slow I/O Task (110ms)	Fast I/O Task (10ms)
AI2	Analogue I/P 2 channels (universal; 3 Terminal Unit options)	✓	-
AI3	Analogue I/P 3 channels (4-20mA, with transmitter PSU)	✓	-
AI4	Analogue I/P 4 channels (TC, mV, mA Terminal Unit options)	✓	-
AO2	Analogue O/P 2 channels (0-20mA or 0-10V output)	✓	-
DI4	Digital I/P 4 channels (logic)	✓	-
DI8_LG*	Digital I/P 8 channels (logic)	✓	✓
DI8_CO*	Digital I/P 8 channels (contact closure)	✓	✓
DI6_MV	Digital I/P 6 channels (ac mains input, 115V rms)	✓	-
DI6_HV	Digital I/P 6 channels (ac mains input, 230V rms)	✓	-
DO4_LG*	Digital O/P 4 channels (externally powered, 10mA)	✓	✓
DO4_24*	Digital O/P 4 channels (externally powered, 100mA)	✓	✓
DO8	Digital O/P 8 channels	✓	✓
RLY4*	Relay O/P 4 channels (2 amp; 3 n/o, 1 change-over)	✓	✓
FI2	Frequency I/P 2 channels (logic, magnetic, and contact closure)	✓	✓
ZI	Zirconia I/P 2 channels (mV (TC), and 2V range (Zirconia))	✓	-

*Note. * indicates the Module upgraded, refers to Version 2 modules.*

Table 1.3.1 Module compatibility

1.3.2 Features

The main features of the 2500M I/O Modules are as follows:

LIN

The Local Instrument Network (LIN) is a collection of LIN instruments, and LIN communications, etc. that together form the control system.

LIN COMMUNICATION

The Local Instrument Network (LIN) communications is our proprietry communications system used to connect each LIN instrument in the network.

ELIN COMMUNICATION

ELIN communication is the LIN communications system transported via Ethernet. It allows peer-to-peer communications between T2550s and the wider network via a standard Ethernet infastructure.

REDUNDANT I/O MODULE CONNECTION

Most I/O Modules can be set up for redundant (duplex) or non-redundant (simplex) operation. When operating in redundant mode, both modules supply the control database operating in the IOC Modules with the requested values, allowing bumpless changeover by the secondary I/O Module should the primary module fail. The wiring configuration of most I/O module redundant connections apply to most Channels. The modules can be located within the same Base Unit or on completely separate Base Units.

HEALTH MONITORING

Automatic health checks, self-testing, and initialisation on power-up, with continuous checking of I/O status.

FRONT PANEL ANNUNCIATION

Instrument LEDs are provided for module and channel status.

1.3.2 Features (Cont.)

CONFIGURATION

Each I/O Module is controlled via the configuration, Strategy, held in the local IOC Module. Most standard LIN Database function blocks are supported in redundant mode. Special diagnostic blocks are available for hardware and software status reporting, refer to the *LIN Blocks Reference Manual* (Part no. HA 082 375 U003) for a full description of each individual block.

Continuous strategies and Sequences are configured/downloaded/monitored using LINtools, the recommended configuration tool. Continuous strategies are built up by interconnection of fixed function blocks from a comprehensive library of analogue and logic elements, common to all LIN based instruments. Special ACTION blocks, i.e. ACT15A3W block, support user-algorithms written in Structured Text (ST) and are well-suited to implement plant logical devices.

Note The I/O Module blocks, <block type>_UIO, relating to each of the I/O Modules on the Base Unit must be configured to operate on the same Task as the strategy.

A Sequence can be employed when the process being controlled by the LIN Database (.dbf) can adopt several distinct states - e.g. 'Starting Up', 'Full Running', 'Shutting Down', etc. A LIN Sequence is a program that runs in a LIN instrument, in conjunction with a LIN Database. It interacts with its associated LIN Database by writing new values to specified LIN Database fields, in response to changes in the values of other specified LIN Database fields.

A ladder diagram is a type of Action represented graphically by a column of 'rungs'. Rungs are equivalent to program statements, with icons along them representing digital or analog fields, constants, and logical or arithmetic functions. Each rung has only one 'output' or 'objective' - at its right-hand end - which is either a coil (digital field), variable (analogue field), or a 'jump' to another labelled rung.

Note A single rung that evaluates TRUE or FALSE can also be used for a Sequence Transition.

Rungs can include any number of input elements and use any complexity of wired or explicit functions to perform the rung operation - subject only to screen space limitations.

CHAPTER 2 ANALOGUE INPUT MODULES

The 2500M (I/O Modules) provide the basic hardware interface to plant transducers and sensors. The Terminal Units provide the physical wiring connections, while the I/O modules perform the data conversions required - analogue or digital, current or voltage, input or output. The modules also offer an electrical isolation barrier that ensures safety and simplifies system wiring reducing interference and cross-talk effects. All working from a 24Vdc supply.

The following information provides the suggested means of wiring a single plant device (sensor, transmitter, etc.) to two I/O channels in order to reduce down-time should one of those I/O channels fail. For obvious reasons it is assumed that each I/O channel is operating on different modules. It is possible that they are installed in different bases or instruments (unless otherwise noted). Obviously there are alternative strategies for some problems, the use of a single duplex or paralleled thermocouples (not recommended) for example. In some cases it is not sensible to connect the sensor to two I/O Modules, as in the case of RTD sensors.

Note It is the user's duty to assess the risk and adopt the necessary wiring strategy.

In order for the Instruments to operate correctly when the modules are wired for redundant operation, as shown, it is necessary to configure a Strategy using LINtools Engineering Studio, see **I/O Module Control Strategy**.

2.1 AI2 - TWO CHANNEL ANALOGUE INPUT MODULE

The AI2, 2 Channel Analogue Input Module supports redundant wiring of Thermocouple plant devices, Voltage or current inputs, and 3- or 4-wire Platinum Resistance Thermometer plant devices. Once wired, the Database file, .dbf, resident in the IOC Module, must be correctly configured via LINtools Engineering Studio Studio using an appropriate number of AI_UIO blocks, one per channel, and additional Function Blocks, Sequences and Actions.

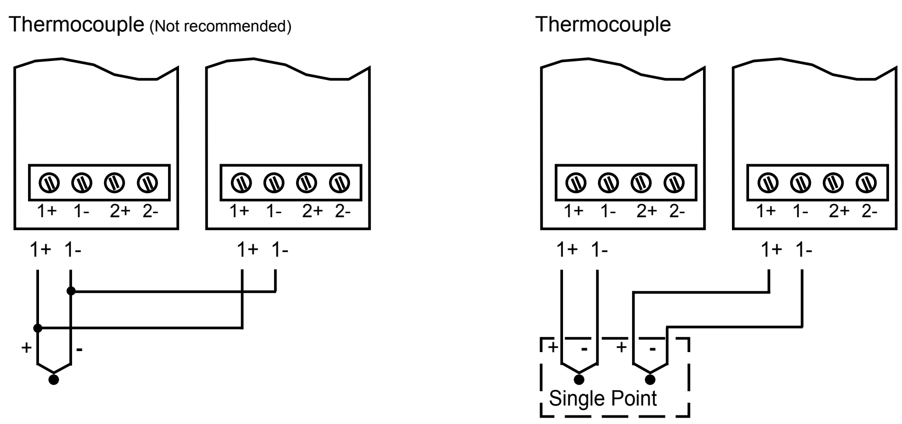
2.1.1 Redundant Thermocouple

Using a duplex thermocouple input redundancy can be achieved by wiring two I/O modules (AI2 T/C terminal units) in parallel. These modules can be located on the same or different bases.

Notes

1. *If using channel 2 of an AI2 module, channel 1 MUST be configured for T/C input in order to provide cold junction compensation. Otherwise any combination of channels may be used.*
2. *Both channels should be set for the same sensor-break response, e.g. up or down scale drive. The channel fault reaction detects any overload, due to sensor break, and implements a fault reaction. Where the fault reaction can be modified, set an appropriate strategy.*

REDUNDANT THERMOCOUPLE WIRING



2.1.2 Redundant Resistance (RTD)

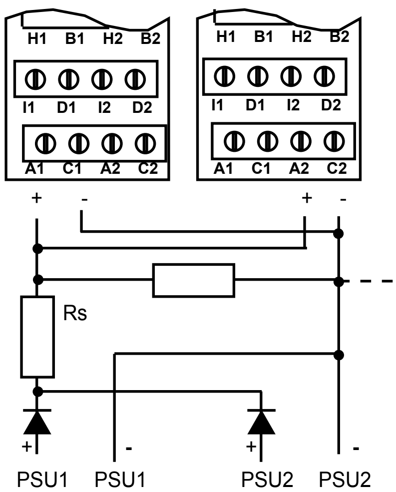
Single RTD probes can not be wired to provide dual redundancy.

One possible solution is to measure the voltage across the bulb, and then map the value to a PT100 Linearisation curve, but this solution would only offer the ability to alarm on a fault condition.

Another solution is to provide an external redundant supply to the sensor and using two mV inputs (via different AI2 modules) measure the resistance PD as shown below.

REDUNDANT RESISTANCE (RTD) WIRING

Dual 3-, 4-Wire Platinum Resistance Thermometer (PRT) Only



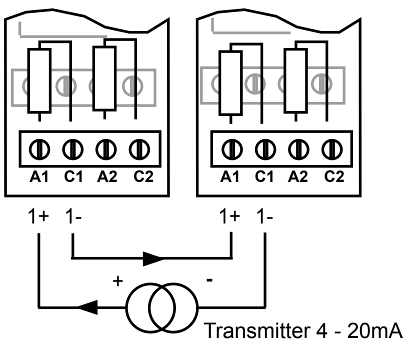
2.1.3 Redundant Current

The AI2 provides an isolated mV circuit with a 5 Ω (ohm) shunt resistor (100mV at 20mA). For single source wiring the only redundant connection is series as shown in the diagram below.

Note An open circuit fault condition would cause loss of input signal to both modules; however this wiring strategy suppports faulty modules and short circuit inputs. If necessary, faulty modules can be removed and replaced as the burden (shunt) resistor is located in the terminal unit.

REDUNDANT CURRENT WIRING

Dual milli-amps (mA) Shunt Option, *See Note above*



The following input types are not suitable for redundant wiring

- Zirconia Probe input is possible but the high impedance and leakage currents could adversely affect the sensor.
- Pyrometer inputs could be wired as shown for Thermocouple inputs, but the increase in leakage current may affect initial calibration accuracy.

2.2 AI3 - THREE CHANNEL ANALOGUE INPUT MODULE

The AI3, 3 Channel Analogue Input Module does not support redundant wiring. This module provides a burden and a Power Supply Unit, PSU, that prevents sensibly wiring the PSU's and the burdens so as to provide a valid PV reading under fault conditions.

Note The burden resistor, as mounted in the module, and not the terminal unit, so removing the module will cause the second channel to fail.

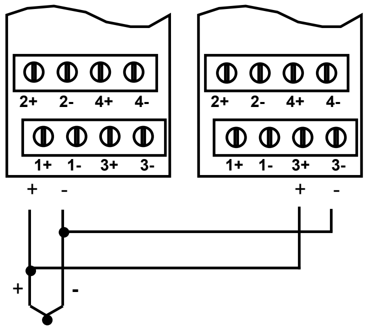
2.3 AI4 - FOUR CHANNEL ANALOGUE INPUT MODULE

The AI4, 4 Channel Analogue Input Module can be wired for mV or Thermocouple redundancy as per the AI2 module. However, the channels are not completely isolated from each other, which may introduce problems. When wiring two AI4 modules for redundant Thermocouple (T/C) or milli-volt (mV) inputs, ALL four channels must be set to operate in the same mode.

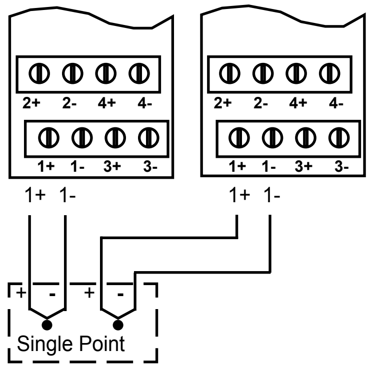
Note The AI4 Milli-amps (mA) Shunt Option does not support redundant wiring operation, use AI2 modules for milli-amp redundant wiring operation.

REDUNDANT THERMOCOUPLE WIRING

Thermocouple (Not recommended)

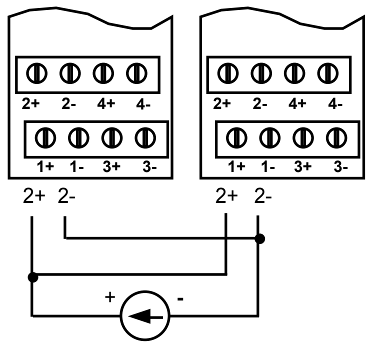


Thermocouple



REDUNDANT VOLTAGE WIRING

Voltage (milli-volt)



2.4 ZI - ZIRCONIA ANALOGUE INPUT MODULE

The ZI, Zirconia Analogue Input Module does not support redundant wiring. The module provides high impedance and leakage currents that can adversely affect the sensor causing a Zirconia Probe input to be not suitable for redundant wiring.

Intentionally left blank

CHAPTER 3 ANALOGUE OUTPUT MODULES

The 2500M (I/O Modules) provide the basic hardware interface to plant transducers and sensors. The Terminal Units provide the physical wiring connections, while the I/O modules perform the data conversions required - analogue or digital, current or voltage, input or output. The modules also offer an electrical isolation barrier that ensures safety and simplifies system wiring reducing interference and cross-talk effects. All working from a 24Vdc supply.

The following information provides the suggested means of wiring a single plant device (sensor, transmitter, etc.) to two I/O channels in order to reduce down-time should one of those I/O channels fail. For obvious reasons it is assumed that each I/O channel is operating on different modules. It is possible that they are installed in different bases or instruments (unless otherwise noted). Obviously there are alternative strategies for some problems, in some cases it is not sensible to connect to two I/O Modules.

Note It is the user's duty to assess the risk and adopt the necessary wiring strategy.

In order for the Instruments to operate correctly when the modules are wired for redundant operation, as shown, it is necessary to configure a Strategy using LINtools Engineering Studio, see **I/O Module Control Strategy**.

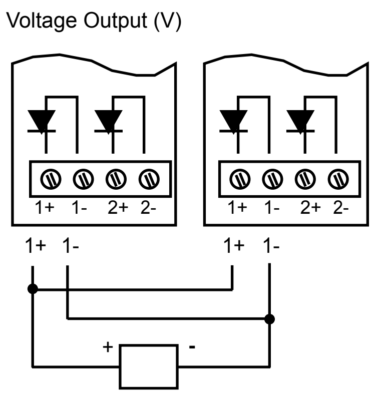
3.1 AO2 - TWO CHANNEL DIGITAL OUTPUT MODULE

The AO2, 2 Channel Analogue Output Module supports redundant wiring of Voltage or Current outputs. Once wired, the Database file, .dbf, resident in the IOC Module, must be correctly configured via LINtools Engineering Studio using appropriate Function Blocks, Sequences and Actions.

Note The principal method of this is based on the highest output wins. This means that in the event of a failure it is expected that the primary output will drive to 0%, and the secondary (redundant) module having a higher output will provide the signal. However, under certain conditions a module could fail with the output stuck at say 90%. In this scenario the secondary (redundant) module would be in operation when the output is between 90-100%. If this output is deemed critical an Analogue Input, in conjunction with a suitable Strategy can be used to measure the output signal, to ensure correct operation of the module.

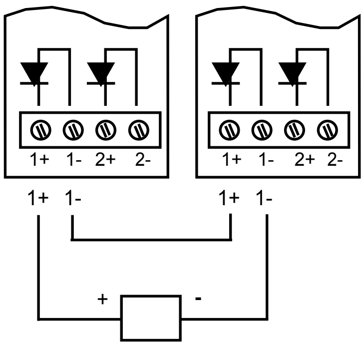
3.1.1 Redundant Voltage Output

For voltage output the circuitry for ensuring a smooth takeover is contained within the module where the module outputs are wired in parallel.



3.1.2 Redundant Current Output

For current outputs there are diodes fitted on the terminal assembly for ensuring circuit completion on failure where the module outputs are wired in series.



CHAPTER 4 DIGITAL INPUT MODULES

The 2500M (I/O Modules) provide the basic hardware interface to plant transducers and sensors. The Terminal Units provide the physical wiring connections, while the I/O modules perform the data conversions required - analogue or digital, current or voltage, input or output. The modules also offer an electrical isolation barrier that ensures safety and simplifies system wiring reducing interference and cross-talk effects. All working from a 24Vdc supply.

The following information provides the suggested means of wiring a single plant device (sensor, transmitter, etc.) to two I/O channels in order to reduce down-time should one of those I/O channels fail. For obvious reasons it is assumed that each I/O channel is operating on different modules. It is possible that they are installed in different bases or instruments (unless otherwise noted). Obviously there are alternative strategies for some problems, in some cases it is not sensible to connect the sensor to two I/O Modules.

Note It is the user's duty to assess the risk and adopt the necessary wiring strategy.

In order for the Instruments to operate correctly when the modules are wired for redundant operation, as shown, it is necessary to configure a Strategy using LINtools Engineering Studio, see **I/O Module Control Strategy**.

4.1 DI4 - FOUR CHANNEL DIGITAL INPUT MODULE

The DI4, 4 Digital Input Module supports Contact Input and Logic Input redundant wiring from plant devices. Once wired, the Database file, .dbf, resident in the IOC Module, must be correctly configured via LINtools Engineering Studio using an appropriate number of DI_UIO blocks, one per channel, and additional Function Blocks, Sequences and Actions.

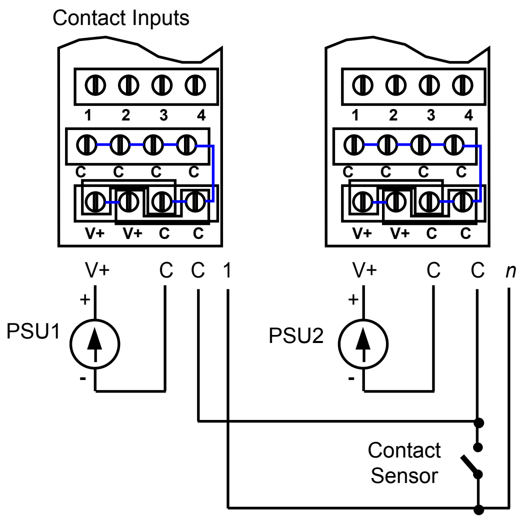
4.1.1 Redundant Contact Inputs

Using a redundant Contact Input requires an external Power Supply. The modules can be located on the same or different bases.

The following diagram shows how the DI4 Modules can be linked together using the 'C' terminals. The diagram below shows two PSU's independently connected.

Note The PSU's can be wired together (in parallel) if capable of redundant connection, such as the 2500P range.

CONTACT INPUT WIRING

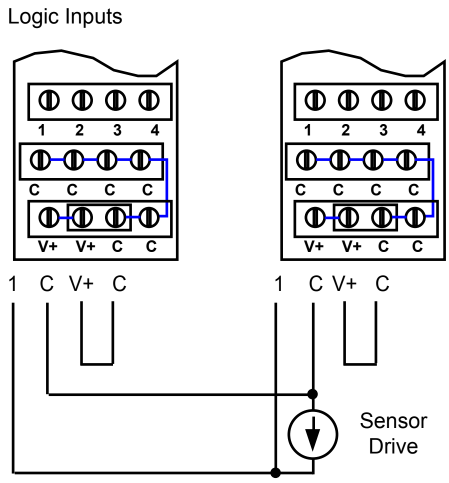


Note Contact Input wiring configuration can be to any channel, 1, 2, 3, or 4 (n).

4.1.2 Redundant Logic Input

The DI4, 4 Channel Digital Input Module can be wired for Logic Input redundancy. However, the channels are not completely isolated from each other, but do remain isolated from the system. The following diagram shows how two DI4 modules can be wired to operate in redundant mode.

LOGIC INPUT WIRING



Note Always link the V+ and C terminals, as shown above. ON state is >10.8V for 'I' – either polarity.

4.2 DI6 - SIX CHANNEL DIGITAL INPUT MODULE

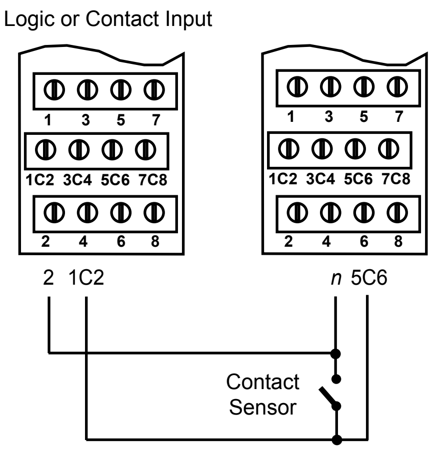
The DI6, 6 Channel Digital Input Module does not support redundant wiring.

4.3 DI8 - EIGHT CHANNEL DIGITAL INPUT MODULE

The DI8, 8 Channel Digital Input Module supports Contact Input and Logic Input redundant wiring from plant devices. Once wired, the Database file, .dbf, resident in the IOC Module, must be correctly configured via LINtools Engineering Studio Studio using an appropriate number of DI_UIO blocks, one per channel, and additional Function Blocks, Sequences and Actions.

The use of isolated PSU's and diodes allow channels to be wired in parallel as shown below. Any channel pair can be used, but there are four channels with a common connection, similar to a DI4 module. When wiring two DI8 modules for redundant Contact Input or Logic Inputs, ALL modules must be the same type.

4.3.1 Redundant Contact Inputs and Logic Inputs



Note Contact Input wiring configuration can be to any channel, 1, 2, 3, or 4 (n).

Intentionally left blank

CHAPTER 5 DIGITAL OUTPUT MODULES

The 2500M (I/O Modules) provide the basic hardware interface to plant transducers and sensors. The Terminal Units provide the physical wiring connections, while the I/O modules perform the data conversions required, analogue or digital, current or voltage, input or output. The modules also offer an electrical isolation barrier that ensures safety and simplifies system wiring reducing interference and cross-talk effects. All working from a 24Vdc supply.

There is no information for the means of wiring a single plant device (sensor, transmitter, etc.) to two I/O channels in order to reduce down-time should one of those I/O channels fail.

Note It is the user's duty to assess the risk and adopt the necessary wiring strategy.

5.1 DO4 - FOUR CHANNEL DIGITAL OUTPUT MODULE

The DO4, 4 Channel Digital Output Module is not recommended for redundant operation. The output is a transistor switch, which could get reverse biased (one output on, one off) and could cause damage to the switch via the -24V VBE. Also the semiconductor switch common failure mode is short circuit, thus preventing a parallel switch turning off.

5.2 DO8 - EIGHT CHANNEL DIGITAL OUTPUT MODULE

The DO8, 8 Channel Digital Output Module is not recommended for redundant operation. The output is a transistor switch, which could get reverse biased (one output on, one off) and could cause damage to the switch via the -24V VBE. Also the semiconductor switch common failure mode is short circuit, thus preventing a parallel switch turning off.

5.3 RLY4 - FOUR CHANNEL RELAY OUTPUT MODULE

The RLY4, 4 Channel Relay Output Module is not recommended for redundant operation. The most likely failure mode for a Relay module is a burnt contact. Therefore, although wiring contacts in parallel is beneficial, it does not support the possibility of contacts that are welded closed, short cicuits or faults leaving a relay permanently ON.

Note It is possible to wire normally closed (N/C) contacts in series in order to provide an alarm or reaction if one contact drops out, for example power failure indication.

Intentionally left blank

CHAPTER 6 SPECIALISED MODULES

The 2500M (I/O Modules) provide the basic hardware interface to plant transducers and sensors. The Terminal Units provide the physical wiring connections, while the I/O modules perform the data conversions required, analogue or digital, current or voltage, input or output. The modules also offer an electrical isolation barrier that ensures safety and simplifies system wiring reducing interference and cross-talk effects. All working from a 24Vdc supply.

The following information provides the suggested means of wiring a single plant device (sensor, transmitter, etc.) to two I/O channels in order to reduce down-time should one of those I/O channels fail. For obvious reasons it is assumed that each I/O channel is operating on different modules. It is possible that they are installed in different bases or instruments (unless otherwise noted). Obviously there are alternative strategies for some problems, in some cases it is not sensible to connect the sensor to two I/O Modules, as in the case of RTD sensors.

Note It is the user's duty to assess the risk and adopt the necessary wiring strategy.

In order for the Instruments to operate correctly when the modules are wired for redundant operation, it is necessary to configure a Strategy using LINtools Engineering Studio, see **I/O Module Control Strategy**.

6.1 FI2 - TWO CHANNEL FREQUENCY INPUT MODULE

The FI2, 2 Channel Frequency Input Module supports redundant wiring of Magnetic inputs, Voltage inputs, and Contact input devices. The wiring configuration of all FI2 redundant connections apply to both Channel 1 and Channel 2. The modules can be located within the same Base Unit or on completely separate Base Units. Once wired, the Database file, .dbf, resident in the IOC Module, must be correctly configured via LINtools Engineering Studio using an appropriate number of FI_UIO blocks, one per channel, and additional Function Blocks, Sequences and Actions.

Note The module does not support redundant wiring of a Current input device, because of the Namur compliance required.

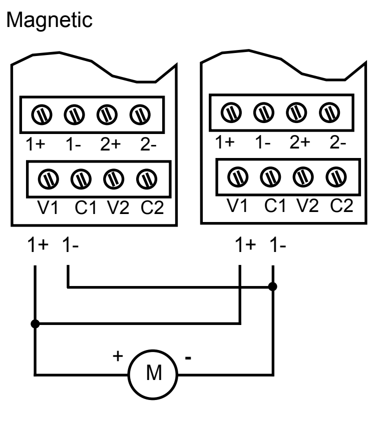
6.1.1 Redundant Magnetic

The FI2, 2 Channel Frequency Input Module can be wired for Magnetic input redundancy. Magnetic input redundancy can be achieved by wiring two I/O modules (FI2 terminal units) in parallel.

Note The Links on both Terminal Units in the redundant pair MUST be set to position C. The InType parameter in both blocks must be set to Magnetic. If required, an impedance matching resistor can be fitted across the input of one module.

REDUNDANT MAGNETIC WIRING

Note When using a Magnetic Input type, it is common practice to fit a parallel resistor, Termination Resistor, between the 1+ and 1- terminals. Make sure the correct Termination Resistor is used when wiring a single input to the two FI2 Modules operating as a redundant pair.

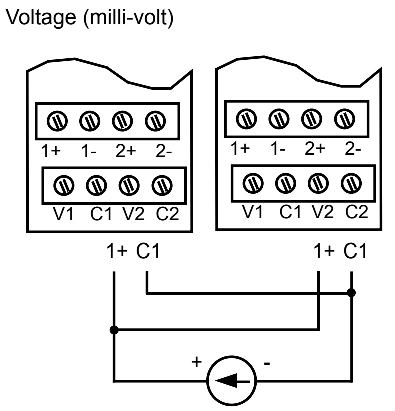


6.1.2 Redundant Voltage

The FI2, 2 Channel Frequency Input Module can be wired for Voltage (V) redundancy. Voltage input redundancy can be achieved by wiring two I/O modules (FI2 terminal units) in parallel.

Note The Links on both Terminal Units in the redundant pair MUST be set to position C. The InType parameter in both blocks must be set to Volts. The Threshold parameter must also be configured the same in both blocks.

REDUNDANT VOLTAGE WIRING



6.1.3 Redundant Current

The FI2, 2 Channel Frequency Input Module is not suitable for redundant wiring when using a Current input configuration.

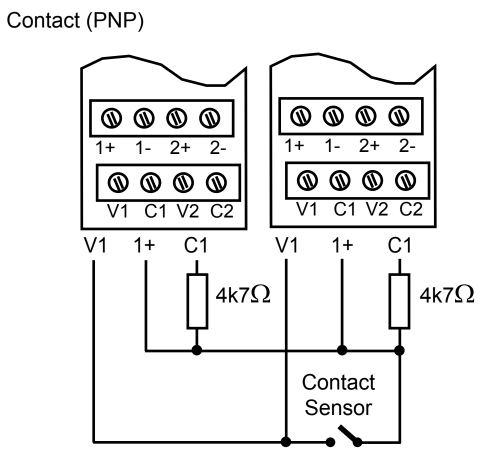
Note The module does not support redundant wiring of a Current input device, because of the Namur compliance required.

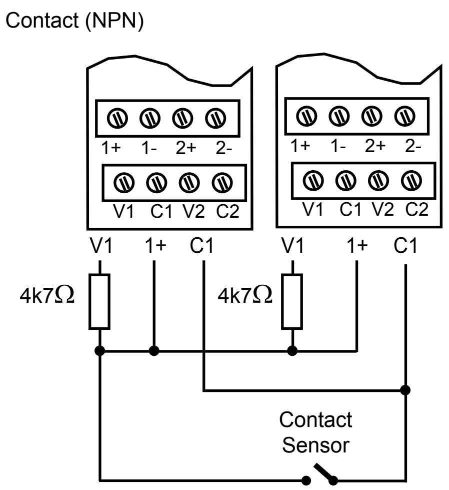
6.1.4 Redundant Contact

The FI2, 2 Channel Frequency Input Module can be wired for Contact (PNP, and NPN) input redundancy. Contact device input redundancy can be achieved by wiring two I/O modules (FI2 terminal units) in parallel.

Note The Links on both Terminal Units in the redundant pair MUST be set to position C. The InType parameter in both blocks must be set to Volts. The Threshold, and PSU parameters must also be configured the same in both blocks.

REDUNDANT CONTACT WIRING





CHAPTER 7 I/O MODULE CONTROL STRATEGY

In order for the Instrument to operate correctly when I/O Modules, specifically Input modules, are wired for redundant operation, it is necessary to configure a strategy using LINtools Engineering Studio, that will ensure consistent operation and select the appropriate process variable. The strategy may include alarm conditions for events such as Primary/Secondary input failure as well as a deviation alarm.

This section describes the basic principles for configuring the Database file, .dbf, and Structured Text, .stx, of a strategy using LINTools Engineering Studio to ensure a basic redundant I/O system is operational and provides examples of each.

IMPORTANT ***It is recommended the User ensures that redundant inputs are wired to two different I/O modules, remembering to consider failure modes for each specific application and add appropriate logic as required.***

Note The I/O Module blocks, <block type>_UIO, must be configured to operate on the same Task as the strategy.

7.1 ANALOGUE MODULE STRATEGY

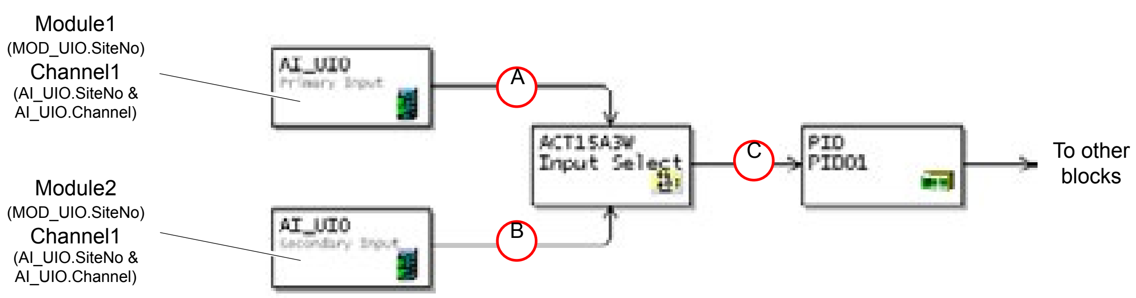
The deviation feature for analogue inputs can be used to test if the input values deviate by more that a predefined value.

Note Some function blocks have been omitted for clarity. However, a LIN database may be reliant on these function blocks to ensure correct operation of the LIN Instrument.

7.1.1 Function Block Configuration

The following example shows the relevant function blocks in the Database file, .dbf, of a strategy based on the use of two 2 Channel Analogue Input modules, represented by the AI_UIO blocks. A thermocouple input is the source PV to a PID control block, the ACT15A3W Control Module block and an appropriate Action file, .stx, see **Structured Text Configuration**, are used to determine which PV should be forwarded to the PID.

Note This principle can also be applied to the AI4, 4 Channel Analogue Input Module.



Wire	From	To
A	Primary Input.PV	Input Select.A0
	Primary Input.Alarms.PVError	Input Select.Byte0.Bit0
	Primary Input.Alarms.OutRange	Input Select.Byte0.Bit1
B	Secondary Input.PV	Input Select.A1
	Secondary Input.Alarms.PVError	Input Select.Byte0.Bit2
	Secondary Input.Alarms.OutRange	Input Select.Byte0.Bit3
C	Input Select.A6	PID01.PV
	Input Select.Byte0.Bit6	PID01.SelMode.SelFMan

Note Multiple wires between the same two blocks are shown as a single line. A tooltip listing each connection appears in LINtools when the cursor hovers over a line.

7.1.2 Structured Text Configuration

The following examples show various sections of Structured Text configured in the <filename>.stx file for an ACT15A3W block, from the Maths category of the Control Modules Template Library.

Accept PV from other Module - Example

Structured Text code can be generated to determine which PV value will be passed to the PID block, PID01, if one input should fail.

Note Alarms are dependant on the application, e.g. CharErr Alarm is not applicable to a Flow strategy. The relevant alarm and status bits should be wired to both Primary and Standby Health inputs.

```
(*
The Following is executed to swap between the inputs in case of either input failure
A0         = PRIMARY INPUT.PV
A1         = SECONDARY INPUT.PV
A4         = Limit defined for discrepancy alarm to be generated
A6         = RESULTANT PV
A8         = MODE (Operator Input)
             0 NORMAL MODE
             1 SELECT PRIMARY
             2 SELECT SECONDARY
             3 COMMISSIONING
Byte0.Bit0 = PRIMARY INPUT.Health1          F = HEALTHY,   T = FAILED
Byte0.Bit1 = PRIMARY INPUT.Health2          F = HEALTHY,   T = FAILED
Byte0.Bit2 = SECONDARY INPUT.Health1        F = HEALTHY,   T = FAILED
Byte0.Bit3 = SECONDARY INPUT.Health2        F = HEALTHY,   T = FAILED
Byte0.Bit4 = Discrepancy Alarm
Byte0.Bit5 = Mode Alarm
Byte0.Bit6 = Total Input Failure Alarm
*)

IF A8 = 0 THEN (* Normal Mode *)
  IF NOT(Byte0.Bit0 OR Byte0.Bit1) AND NOT(Byte0.Bit2 OR Byte0.Bit3) THEN (* Primary OK,
Secondary OK *)
    A6:=MAX(A0,A1); (* This accounts for downscale break*)
    Byte0.Bit6:=0;
  ELSIF NOT(Byte0.Bit0 OR Byte0.Bit1) THEN (* Primary OK, Secondary Failed *)
    A6:=A0;
    Byte0.Bit6:=0;
  ELSIF NOT(Byte0.Bit2 OR Byte0.Bit3) THEN (* Primary Failed, Secondary OK *)
    A6:=A1;
    Byte0.Bit6:=0;
  ELSE (* Primary Failed, Secondary Failed - Hold Output & Raise Alarm Flag *)
    Byte0.Bit6:=1;
  END_IF;
ELSE
  Byte0.Bit6:=0;
  IF A8 = 1 THEN (* Primary Override *)
    A6:=A0;
  ELSIF A8 = 2 THEN (* Secondary Override *)
    A6:=A1;
  ELSE (* Commissioning - Hold Output *)
  END_IF;
END_IF;
```

7.1.2 Structured Text Configuration (Cont.)

<u>Mode Alarm - Example</u>

An alarm signal can be generated (*Byte0.Bit5*) if an operator chooses to use a mode other than Normal.

```
                                                        (*
                                        Raise Alarm Flag if not in NORMAL MODE *)
Byte0.Bit5:=A8<>0;


  (*
```

This Structured Text sets the specified Bit, *Byte0.Bit5*, TRUE when Mode is not in Normal. This could be wired to a DIGALARM block to indicate an alarm, or linked to a display screen.

<u>Discrepancy Alarm - Example</u>

An alarm signal can be generated (*Byte0.Bit4*) if the difference between the two inputs is greater than the value specified in field A4.

```
                                                              Raise
                                                   Discrepancy Alarm Flag if healthy
                               inputs deviate *)
IF NOT(Byte0.Bit0 OR Byte0.Bit1) AND NOT(Byte0.Bit2 OR Byte0.Bit3) THEN
   Byte0.Bit4:=Abs(A0 - A1) > A4;
ELSE
   Byte0.Bit4:=0;
END_IF;
```

This Structured Text sets the specified Bit, *Byte0.Bit4*, TRUE when a discrepancy between values exist and both inputs are healthy, otherwise the bit is FALSE. This could be wired to a DIGALARM block to indicate an alarm, or linked to a display screen.

7.2 DIGITAL MODULE STRATEGY

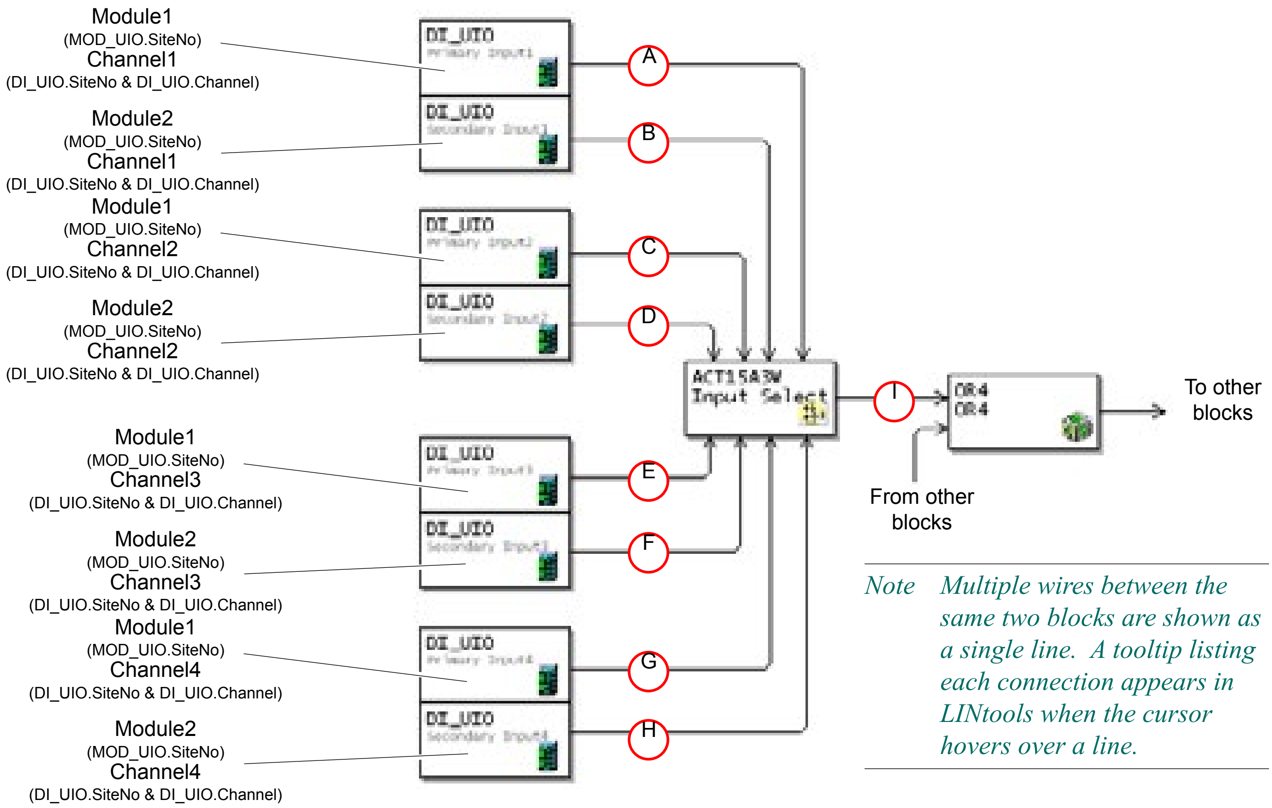
The deviation feature for digital inputs can be used to test if the input values differ.

Note Some function blocks have been omitted for clarity. However, a LIN database may be reliant on these function blocks to ensure correct operation of the LIN Instrument.

7.2.1 Function Block Configuration

The following example shows the relevant function blocks in the Database file, .dbf, of a strategy based on the use of two 4 Channel Digital Input modules, represented by the DI_UIO blocks. The ACT15A3W Control Module block and an appropriate Action file, .stx, see **Structured Text Configuration**, are used to determine which input should be used.

Note This principle can also be applied to the DI8, 8 Channel Digital Input Module.



Note Multiple wires between the same two blocks are shown as a single line. A tooltip listing each connection appears in LINtools when the cursor hovers over a line.

Wire	From	To
A	Primary Input1.In	Input Select.Word0.Bit0
	Primary Input1.Alarms.Hardware	Input Select.Word0.Bit1
B	Secondary Input1.In	Input Select.Word1.Bit0
	Secondary Input1.Alarms.Hardware	Input Select.Word1.Bit1
C	Primary Input2.In	Input Select.Word0.Bit2
	Primary Input2.Alarms.Hardware	Input Select.Word0.Bit3
D	Secondary Input2.In	Input Select.Word1.Bit2
	Secondary Input2.Alarms.Hardware	Input Select.Word1.Bit3
E	Primary Input3.In	Input Select.Word0.Bit4
	Primary Input3.Alarms.Hardware	Input Select.Word0.Bit5
F	Secondary Input3.In	Input Select.Word1.Bit4
	Secondary Input3.Alarms.Hardware	Input Select.Word1.Bit5
G	Primary Input3.In	Input Select.Word0.Bit6
	Primary Input3.Alarms.Hardware	Input Select.Word0.Bit7
H	Secondary Input4.In	Input Select.Word1.Bit6
	Secondary Input4.Alarms.Hardware	Input Select.Word1.Bit7
I	Input Select.Byte0.Bit4	OR4.In_1

7.2.2 Structured Text Configuration

The following shows various sections of Structured Text configured in the <filename>.stx file.

Accept PV from other Module - Example

Structured Text code can be generated to determine which PV value will be passed, if one input should fail.

Note Alarms are dependant on the application, e.g. CharErr Alarm is not applicable to a Flow strategy. The relevant alarm and status bits should be wired to both Primary and Standby Health inputs.

```
                                                                                   (* CHANNEL 1
                                                        The following is executed to swap between the
inputs in case of either input failure
Word0.Bit0 = PRIMARY INPUT1.In
Word0.Bit1 = PRIMARY INPUT1.Health            F = HEALTHY,   T = FAILED
Word1.Bit0 = SECONDARY INPUT1.In
Word1.Bit1 = SECONDARY INPUT1.Health          F = HEALTHY,   T = FAILED
Byte0.Bit0 = RESULTANT INPUT
Byte0.Bit1 = Discrepancy Alarm
Byte0.Bit2 = Mode Alarm
Byte0.Bit3 = Total Input Failure Alarm
A0     = MODE (Operator Input)
           0 NORMAL MODE
           1 SELECT PRIMARY
           2 SELECT SECONDARY
           3 COMMISSIONING
A8     = DISCREPANCY ALARM COUNTER
*)

IF A0 = 0 THEN (* NORMAL MODE *)
  IF NOT(Word0.Bit1) THEN (* Primary OK, Secondary Don't Care *)
    Byte0.Bit0:=Word0.Bit0;
    Byte0.Bit3:=0;
  ELSIF NOT(Word1.Bit1) THEN (* Primary Failed, Secondary OK *)
    Byte0.Bit0:=Word1.Bit0;
    Byte0.Bit3:=0;
  ELSE (* Primary Failed, Secondary Failed - Hold Output & Raise Alarm Flag *)
    Byte0.Bit3:=1;
  END_IF;
ELSE
  Byte0.Bit3:=0;
  IF A0 = 1 THEN (* Primary Override *)
    Byte0.Bit0:=Word0.Bit0;
  ELSIF A0 = 2 THEN (* Secondary Override *)
    Byte0.Bit0:=Word1.Bit0;
  ELSE (* Commissioning - Hold Output *)
  END_IF;
END_IF;
```

Mode Alarm - Example

An alarm signal can be generated (*Byte0.Bit2*) if an operator chooses to use a mode other the Normal.

```
                                                                  (*
R a i s e              Alarm Flag if not in NORMAL MODE *)
Byte0.Bit2:=A8<>0;
```

This Structured Text sets the specified Bit, *Byte0.Bit2*, TRUE when Mode is not in Normal. This could be wired to a DIGALARM block to indicate an alarm, or linked to a display screen.

7.2.2 Structured Text Configuration (Cont.)

Discrepancy Alarm - Example

An alarm signal can be generated (*Byte0.Bit1*) if a difference between the two inputs is detected.

The Structured Text is configured not to generate an alarm immediately, but after two cycles. This avoids generating nuisance alarms as the digital inputs change. Nuisance alarms are generated if the value read by one input changes before it is read by the other input, because the order in which the inputs are scanned and the I/O blocks, DI_UIO, are executed. A simple counter delay ST configuration is used to eliminate these nuisance alarms. This discrepancy alarm checking is only appropriate for digital inputs that change on an occasional basis, e.g. pressure or temperature switches etc.

```
(* Raise Discrepancy Alarm Flag if healthy inputs deviate *)
IF NOT(Word0.Bit1 OR Word1.Bit1) THEN
  IF Word0.Bit0 <> Word1.Bit0 THEN
    A8:=MIN(2,A8+1); (* A8 = Discrepancy Alarm Counter *)
  ELSE
    A8:=0;
  END_IF;
ELSE
  A8:=0;
END_IF;
Byte0.Bit1:=(A8=2); (* Discrepancy exists for 2 cycles *)
```

This Structured Text sets the specified Bit, *Byte0.Bit5*, TRUE when a discrepancy between values exist and both inputs are healthy, otherwise the bit is FALSE. This could be wired to a DIGALARM block to indicate an alarm, or linked to a display screen.

7.2.2 Structured Text Configuration (Cont.)

The following example of redundant Structured Text, ST, configuration can be used directly in a strategy.

```
(* CHANNEL 1
The following is executed to swap between the inputs in case of either input failure
Word0.Bit0 = PRIMARY INPUT1.In
Word0.Bit1 = PRIMARY INPUT1.Health              F = HEALTHY,   T = FAILED
Word1.Bit0 = SECONDARY INPUT1.In
Word1.Bit1 = SECONDARY INPUT1.Health            F = HEALTHY,   T = FAILED
Byte0.Bit0 = RESULTANT INPUT
Byte0.Bit1 = Discrepancy Alarm
Byte0.Bit2 = Mode Alarm
Byte0.Bit3 = Total Input Failure Alarm
A0     = MODE (Operator Input)
           0 NORMAL MODE
           1 SELECT PRIMARY
           2 SELECT SECONDARY
           3 COMMISSIONING
A8     = DISCREPANCY ALARM COUNTER
*)

IF A0 = 0 THEN (* NORMAL MODE *)
  IF NOT(Word0.Bit1) THEN (* Primary OK, Secondary Don’t Care *)
    Byte0.Bit0:=Word0.Bit0;
    Byte0.Bit3:=0;
  ELSIF NOT(Word1.Bit1) THEN (* Primary Failed, Secondary OK *)
    Byte0.Bit0:=Word1.Bit0;
    Byte0.Bit3:=0;
  ELSE (* Primary Failed, Secondary Failed - Hold Output & Raise Alarm Flag *)
    Byte0.Bit3:=1;
  END_IF;
ELSE
  Byte0.Bit3:=0;
  IF A0 = 1 THEN (* Primary Override *)
    Byte0.Bit0:=Word0.Bit0;
  ELSIF A0 = 2 THEN (* Secondary Override *)
    Byte0.Bit0:=Word1.Bit0;
  ELSE (* Commissioning - Hold Output *)
  END_IF;
END_IF;

(* Raise Alarm Flag if not in NORMAL MODE *)
Byte0.Bit2:=A0<>0;

(* Raise Discrepancy Alarm Flag if healthy inputs deviate *)
IF NOT(Word0.Bit1 OR Word1.Bit1) THEN
  IF Word0.Bit0 <> Word1.Bit0 THEN
    A8:=MIN(2,A8+1); (* A8 = Discrepancy Alarm Counter *)
  ELSE
    A8:=0;
  END_IF;
ELSE
  A8:=0;
END_IF;
Byte0.Bit1:=(A8=2); (* Discrepancy exists for 2 cycles *)
```

7.2.2 Structured Text Configuration (Cont.)

```
                                 (* CHANNEL 2
The following is executed to swap between the inputs in case of either input failure
Word0.Bit2 = PRIMARY INPUT2.In
Word0.Bit3 = PRIMARY INPUT2.Health            F = HEALTHY,   T = FAILED
Word1.Bit2 = SECONDARY INPUT2.In
Word1.Bit3 = PRIMARY INPUT2.Health            F = HEALTHY,   T = FAILED
Byte0.Bit4 = RESULTANT INPUT
Byte0.Bit5 = Discrepancy Alarm
Byte0.Bit6 = Mode Alarm
Byte0.Bit7 = Total Input Failure Alarm
A1     = MODE (Operator Input)
           0 NORMAL MODE
           1 SELECT PRIMARY
           2 SELECT SECONDARY
           3 COMMISSIONING
A9     = DISCREPANCY ALARM COUNTER
*)

IF A1 = 0 THEN (* NORMAL MODE *)
  IF NOT(Word0.Bit3) THEN (* Primary OK, Secondary Don't Care *)
    Byte0.Bit4:=Word0.Bit2;
    Byte0.Bit7:=0;
  ELSIF NOT(Word1.Bit3) THEN (* Primary Failed, Secondary OK *)
    Byte0.Bit4:=Word1.Bit2;
    Byte0.Bit7:=0;
  ELSE (* Primary Failed, Secondary Failed - Hold Output & Raise Alarm Flag *)
    Byte0.Bit7:=1;
  END_IF;
ELSE
  Byte0.Bit7:=0;
  IF A1 = 1 THEN (* Primary Override *)
    Byte0.Bit4:=Word0.Bit2;
  ELSIF A1 = 2 THEN (* Secondary Override *)
    Byte0.Bit4:=Word1.Bit2;
  ELSE (* Commissioning - Hold Output *)
  END_IF;
END_IF;

(* Raise Alarm Flag if not in NORMAL MODE *)
Byte0.Bit6:=A1<>0;

(* Raise Discrepancy Alarm Flag if healthy inputs deviate *)
IF NOT(Word0.Bit3 OR Word1.Bit3) THEN
  IF Word0.Bit2 <> Word1.Bit2 THEN
    A9:=MIN(2,A9+1); (* A9 = Discrepancy Alarm Counter *)
  ELSE
    A9:=0;
  END_IF;
ELSE
  A9:=0;
END_IF;
Byte0.Bit5:=(A9=2); (* Discrepancy exists for 2 cycles *)
```


HA029462
Issue 3 June 2015

7.2.2 Structured Text Configuration (Cont.)

```
                                                        (* CHANNEL 3
                                               The following is executed to swap
                        between the inputs in case of either input failure
Word0.Bit4 = PRIMARY INPUT3.In
Word0.Bit5 = PRIMARY INPUT3.Health            F = HEALTHY,   T = FAILED
Word1.Bit4 = SECONDARY INPUT3.In
Word1.Bit5 = PRIMARY INPUT3.Health            F = HEALTHY,   T = FAILED
Byte1.Bit0 = RESULTANT INPUT
Byte1.Bit1 = Discrepancy Alarm
Byte1.Bit2 = Mode Alarm
Byte1.Bit3 = Total Input Failure Alarm
A2     = MODE (Operator Input)
          0 NORMAL MODE
          1 SELECT PRIMARY
          2 SELECT SECONDARY
          3 COMMISSIONING
A10    = DISCREPANCY ALARM COUNTER
*)

IF A2 = 0 THEN (* NORMAL MODE *)
  IF NOT(Word0.Bit5) THEN (* Primary OK, Secondary Don't Care *)
    Byte1.Bit0:=Word0.Bit4;
    Byte1.Bit3:=0;
  ELSIF NOT(Word1.Bit5) THEN (* Primary Failed, Secondary OK *)
    Byte1.Bit0:=Word1.Bit4;
    Byte1.Bit3:=0;
  ELSE (* Primary Failed, Secondary Failed - Hold Output & Raise Alarm Flag *)
    Byte1.Bit3:=1;
  END_IF;
ELSE
  Byte1.Bit3:=0;
  IF A2 = 1 THEN (* Primary Override *)
    Byte1.Bit0:=Word0.Bit4;
  ELSIF A2 = 2 THEN (* Secondary Override *)
    Byte1.Bit0:=Word1.Bit4;
  ELSE (* Commissioning - Hold Output *)
  END_IF;
END_IF;

(* Raise Alarm Flag if not in NORMAL MODE *)
Byte1.Bit2:=A2<>0;

(* Raise Discrepancy Alarm Flag if healthy inputs deviate *)
IF NOT(Word0.Bit5 OR Word1.Bit5) THEN
  IF Word0.Bit4 <> Word1.Bit4 THEN
    A10:=MIN(2,A10+1); (* A10 = Discrepancy Alarm Counter *)
  ELSE
    A10:=0;
  END_IF;
ELSE
  A10:=0;
END_IF;
Byte1.Bit1:=(A10=2); (* Discrepancy exists for 2 cycles *)
```

7.2.2 Structured Text Configuration (Cont.)

```
                                                    (* CHANNEL 4
                                 The following is executed to swap between the inputs in case
of either input failure
Word0.Bit6 = PRIMARY INPUT4.In
Word0.Bit7 = PRIMARY INPUT4.Health            F = HEALTHY,   T = FAILED
Word1.Bit6 = SECONDARY INPUT4.In
Word1.Bit7 = PRIMARY INPUT4.Health            F = HEALTHY,   T = FAILED
Byte1.Bit4 = RESULTANT INPUT
Byte1.Bit5 = Discrepancy Alarm
Byte1.Bit6 = Mode Alarm
Byte1.Bit7 = Total Input Failure Alarm
A3     = MODE (Operator Input)
           0 NORMAL MODE
           1 SELECT PRIMARY
           2 SELECT SECONDARY
           3 COMMISSIONING
A11    = DISCREPANCY ALARM COUNTER
*)

IF A3 = 0 THEN (* NORMAL MODE *)
  IF NOT(Word0.Bit7) THEN (* Primary OK, Secondary Don't Care *)
    Byte1.Bit4:=Word0.Bit6;
    Byte1.Bit7:=0;
  ELSIF NOT(Word1.Bit7) THEN (* Primary Failed, Secondary OK *)
    Byte1.Bit4:=Word1.Bit6;
    Byte1.Bit7:=0;
  ELSE (* Primary Failed, Secondary Failed - Hold Output & Raise Alarm Flag *)
    Byte1.Bit7:=1;
  END_IF;
ELSE
  Byte1.Bit7:=0;
  IF A3 = 1 THEN (* Primary Override *)
    Byte1.Bit4:=Word0.Bit6;
  ELSIF A3 = 2 THEN (* Secondary Override *)
    Byte1.Bit4:=Word1.Bit6;
  ELSE (* Commissioning - Hold Output *)
  END_IF;
END_IF;

(* Raise Alarm Flag if not in NORMAL MODE *)
Byte1.Bit6:=A3<>0;

(* Raise Discrepancy Alarm Flag if healthy inputs deviate *)
IF NOT(Word0.Bit7 OR Word1.Bit7) THEN
  IF Word0.Bit6 <> Word1.Bit6 THEN
    A11:=MIN(2,A11+1); (* A11 = Discrepancy Alarm Counter *)
  ELSE
    A11:=0;
  END_IF;
ELSE
  A11:=0;
END_IF;
Byte1.Bit5:=(A11=2); (* Discrepancy exists for 2 cycles *)
```

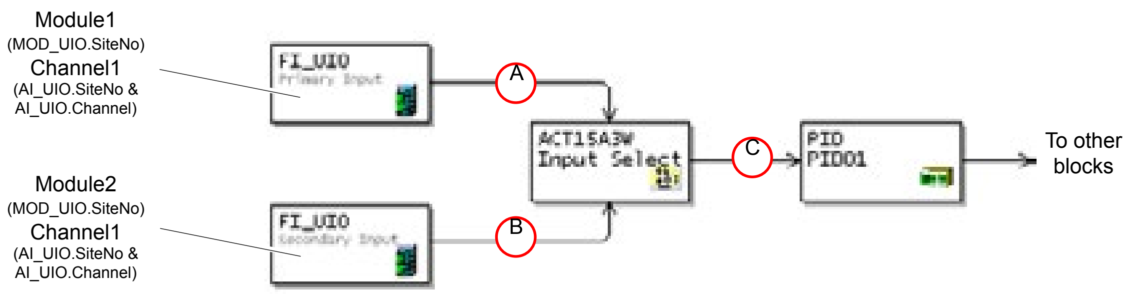
7.3 FREQUENCY MODULE STRATEGY

The deviation feature for analogue inputs can be used to test if the input values deviate by more that a predefined value.

Note Some function blocks have been omitted for clarity. However, a LIN database may be reliant on these function blocks to ensure correct operation of the LIN Instrument.

7.3.1 Function Block Configuration

The following example shows the relevant function blocks in the Database file, .dbf, of a strategy based on the use of two 2 Channel Frequency Input modules, represented by the FI_UIO blocks. The voltage input is the source PV to a PID control block, the ACT15A3W Control Module block and an appropriate Action file, .stx, see **Structured Text Configuration**, are used to determine which PV should be forwarded to the PID.



Wire	From	To
A	Primary Input.PV	Input Select.A0
	Primary Input.Alarms.PVError	Input Select.Byte0.Bit0
	Primary Input.Alarms.OutRange	Input Select.Byte0.Bit1
B	Secondary Input.PV	Input Select.A1
	Secondary Input.Alarms.PVError	Input Select.Byte0.Bit2
	Secondary Input.Alarms.OutRange	Input Select.Byte0.Bit3
C	Input Select.A6	PID01.PV
	Input Select.Byte0.Bit6	PID01.SelMode.SelFMan

Note Multiple wires between the same two blocks are shown as a single line. A tooltip listing each connection appears in LINtools when the cursor hovers over a line.

7.3.2 Structured Text Configuration

The following shows various sections of Structured Text configured in the <filename>.stx file.

Accept PV from other Module - Example

Structured Text code can be generated to determine which PV value will be passed to the PID block, PID01, if one input should fail.

> *Note Alarms are dependant on the application, e.g. CharErr Alarm is not applicable to a Flow strategy. The relevant alarm and status bits should be wired to both Primary and Standby Health inputs.*

```
(*
The Following is executed to swap between the inputs in case of either input failure
A0         = PRIMARY INPUT.PV
A1         = SECONDARY INPUT.PV
A4         = Limit defined for discrepancy alarm to be generated
A6         = RESULTANT PV
A8         = MODE (Operator Input)
             0 NORMAL MODE
             1 SELECT PRIMARY
             2 SELECT SECONDARY
             3 COMMISSIONING
Byte0.Bit0 = PRIMARY INPUT.Health1           F = HEALTHY,   T = FAILED
Byte0.Bit1 = PRIMARY INPUT.Health2           F = HEALTHY,   T = FAILED
Byte0.Bit2 = SECONDARY INPUT.Health1         F = HEALTHY,   T = FAILED
Byte0.Bit3 = SECONDARY INPUT.Health2         F = HEALTHY,   T = FAILED
Byte0.Bit4 = Discrepancy Alarm
Byte0.Bit5 = Mode Alarm
Byte0.Bit6 = Total Input Failure Alarm
*)

IF A8 = 0 THEN (* Normal Mode *)
  IF NOT(Byte0.Bit0 OR Byte0.Bit1) AND NOT(Byte0.Bit2 OR Byte0.Bit3) THEN (* Primary OK,
Secondary OK *)
    A6:=MAX(A0,A1); (* This accounts for downscale break*)
    Byte0.Bit6:=0;
  ELSIF NOT(Byte0.Bit0 OR Byte0.Bit1) THEN (* Primary OK, Secondary Failed *)
    A6:=A0;
    Byte0.Bit6:=0;
  ELSIF NOT(Byte0.Bit2 OR Byte0.Bit3) THEN (* Primary Failed, Secondary OK *)
    A6:=A1;
    Byte0.Bit6:=0;
  ELSE (* Primary Failed, Secondary Failed - Hold Output & Raise Alarm Flag *)
    Byte0.Bit6:=1;
  END_IF;
ELSE
  Byte0.Bit6:=0;
  IF A8 = 1 THEN (* Primary Override *)
    A6:=A0;
  ELSIF A8 = 2 THEN (* Secondary Override *)
    A6:=A1;
  ELSE (* Commissioning - Hold Output *)
  END_IF;
END_IF;
```

7.3.2 Structured Text Configuration (Cont.)

Mode Alarm - Example

An alarm flag can be raised (*Byte0.Bit5*) if an operator chooses to use a mode other than Normal.

```
(*
Raise Alarm Flag if not in NORMAL MODE *)
Byte0.Bit5:=A8<>0;
```

This Structured Text sets the specified Bit, *Byte0.Bit5*, TRUE when Mode is not in Normal. This could be wired to a DIGALARM block to indicate an alarm, or linked to a display screen.

Discrepancy Alarm - Examlpe

An alarm flag can be raised (*Byte0.Bit4*) if a difference between the two inputs is detected.

```
(*
Raise Discrepancy Alarm Flag if
healthy inputs deviate *)
IF NOT(Byte0.Bit0 OR Byte0.Bit1) AND NOT(Byte0.Bit2 OR Byte0.Bit3) THEN
   Byte0.Bit4:=Abs(A0 - A1) > A4;
ELSE
   Byte0.Bit4:=0;
END_IF;
```

This Structured Text sets the specified Bit, *Byte0.Bit5*, TRUE when a discrepancy between values exist and both inputs are healthy, otherwise the bit is FALSE. This could be wired to a DIGALARM block to indicate an alarm, or linked to a display screen.

Intentionally left blank

International sales and support
Eurotherm: www.eurotherm.com

Contact Information

Eurotherm Head Office
Faraday Close
Durrington
Worthing
West Sussex
BN13 3PL

Sales Enquiries
T +44 (01903) 695888
F 0845 130 9936

General Enquiries
T +39 031 975 111
F +39 031 977 512

Worldwide Offices
www.eurotherm.com/global

Scan for local contacts