RTU32M
Guideline for Safety

Version 0.10 - June 2020

BRODERSEN
simplifying systems

Table of Contents

List of Figures

1. Customer Information

1.1 Copyright Notice

1.2 Trademark Acknowledgement

Brand and product names are trademarks or registered trademarks of their respective owners.

1.3 Disclaimer

Brodersen A/S reserves the right to make changes, without notice, to any product, including circuits and/or software described or contained in this manual in order to improve design and/or performance. Brodersen A/S assumes no responsibility or liabilities for the use of the described product(s), conveys no license or title under any patent, copyright, or mask work rights to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described in this manual are for illustration purposes only.
Brodersen A/S makes no representation or warranty that such application will be suitable for the specified use without further testing or modification.

1.4 Life Support Policy

BRODERSEN A/S's PRODUCTS ARE NOT FOR USE AS CRITICAL COMPONENTS, IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE PRIOR WRITTEN APPROVAL OF BRODERSEN A/S.
As used herein:
1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into body, or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

1.5 Brodersen Customer Services

Your satisfaction is our primary concern. Here is a guide to Brodersen customer services. To ensure you get the full benefit of our services, please follow the instructions below carefully.

1.6 Technical Support

We want you to get the maximum performance from your products. So if you run into technical difficulties, we are here to help. For the most frequently asked questions, you can easily find answers in the product documentation. These answers are normally a lot more detailed than the ones we can give over the phone. So please consult this manual first.
To receive the latest version of the user manual, please visit our Web site at:

http://www.brodersen.com,

Choose the product in question under product search and under each product you will find accompanying data sheets, manuals, and user guides etc.
If you still cannot find the answer, gather all the information or questions that apply to your problem, and with the product close at hand, call your dealer. Our distributors are well trained and ready to give you the support you need to get the most from your Brodersen products. In fact, most problems reported are minor and are able to be easily solved over the phone.
In addition, technical support is available from Brodersen engineers every business day. We are always ready to give advice on application requirements or specific information on the installation and operation of any of our products. Please do not hesitate to call or e-mail us on support@brodersen.com .

Brodersen A/S
Islevdalvej 187
DK-2610 Roedovre
Tel.: +45 45 35 26 27
sales@brodersen.com
www.brodersen.com

2 Overview

This safety reference manual is intended to do the following:

- Describe the Brodersen RTU32M components available that are suitable for use in low demand and high demand safety-related control, up to and including SIL 2 applications
- Provide safety-related information specific to the use of RTU32M modules in SIL 2 systems - including PFD calculations that must be considered for SIL 2-certified systems
- Explain some possible SIL 2-certified system configurations
- Describe basic programming techniques for the implementation of RTU32M SIL 2-certified systems with references and links to more-detailed programming and implementation techniques

IMPORTANT This manual describes typical SIL 2 implementations using Brodersen equipment. Keep in mind that the descriptions presented in this manual do not preclude other methods of implementing a SIL 2-compliant system by using Brodersen equipment.
Make sure that other methods are reviewed and approved by a recognized certifying body.

3 Terminology

Abbreviations Used throughout this Reference Manual:

Abbreviation	Full Term	Definition
EN	European Norm.	The official European Standard.
MTBF	Mean Time Between Failures	Average time between failure occurrences.
MTTR	Mean Time to Restoration	Average time that is needed to restore normal operation after a failure has occurred.
PFD	Probability of Failure on Demand	The average probability of a system to fail to perform its design function on demand.
PFH	Probability of Failure per Hour	The probability of a system to have a dangerous failure occur per hour.
SFF	Safe Failure Fraction	The ratio of safe failure plus dangerous detected failure to total failures.
SIL	Safety Integrity Level	A discrete level for specifying the safety integrity requirements of the safety functions allocated to the electrical/electronic/programmable electronic (E/E/PE) part of the safety system.
STR	Spurious Trip Rate	That part of the overall failure rate that does not lead to a dangerous undetected failure.

4 SIL Policy

4.1 About the RTU32M

RTU32M is a modular programmable RTU with the ability to pre-configure outputs and other responses to fault conditions. As such, a system can be designed to meet requirements for 'hold last state' in the event of a fault so that the system can be used in up to, and including, SIL 2-level applications that require that output signals to actuators remain in a preconfigured fail-safe condition. By understanding the behavior of the RTU32M in an emergency shutdown situation, you can incorporate appropriate system design measures to meet other application requirements. These measures relate to the control of outputs and actuators, which must go to a safe state. Other requirements for SIL 2 (inputs from sensors, software that is used, and so on) must also be met.

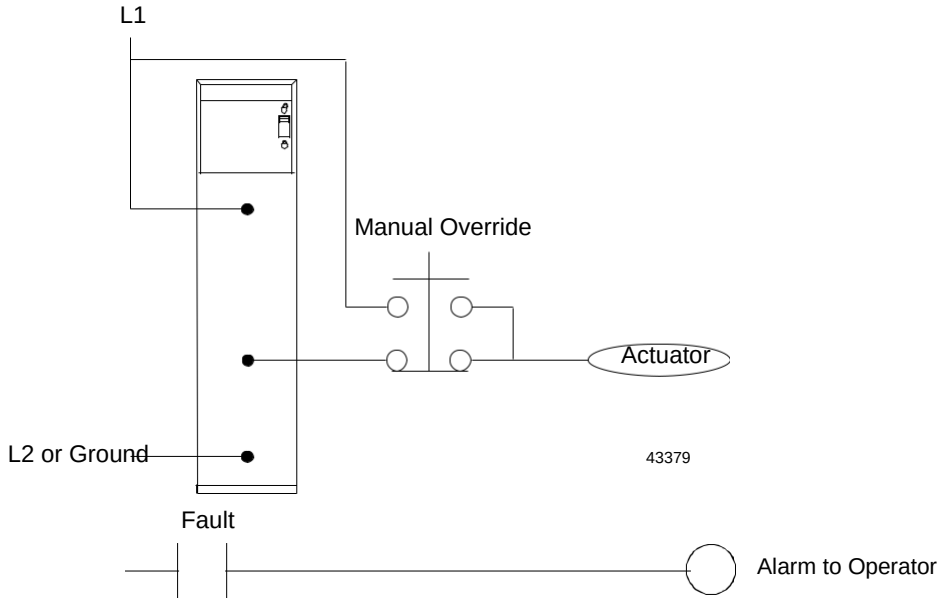
4.2 Gas and Fire Considerations

These are the measures and modifications that are related to the use of the RTU32M in Gas and Fire applications.

- The use of a manual override is necessary to make sure that the operator can maintain the desired control in the event of a controller failure. This is similar in concept to the function of the external relay or redundant outputs that are required to make sure that a de-energized state is achieved for an ESD system when a failure occurs (for example, a shorted output driver) that prevents this from normally occurring. The system knows that it has a failure, but the failure state requires an independent means to maintain control and either remove power or provide an alternate path to maintain power to the end actuator.
- If the application cannot tolerate an output that can fail shorted (energized), then an external means such as a relay or other output must be wired in series to remove power when the fail shorted condition occurs. See Wiring LB2 Digital Output Modules for more information.
- If the application cannot tolerate an output that fails open (de-energized), then an external means such as a manual override or output must be wired in parallel. See **Figure 1**. You must supply alternative means and develop the application program to initiate the alternate means to remove or continue to supply power in the event the main output fails.

- This manual override circuit is shown in **Figure 1**. It is composed of a hard-wired set of contacts from a selector switch or push button. One normally open contact provides for the bypass of power from the controller output directly to the actuator. The other is a normally closed contact to remove or isolate the controller output.
- Generate an application program to monitor the diagnostic output modules for dangerous failures such as shorted or open-output driver channels. Diagnostic output modules must be configured to hold last state in the event of a fault.
- A diagnostic alarm must be generated to inform the operator that manual control is required.
- The faulted module must be replaced within the Mean Time to Restoration (MTTR).
- Any time a fault is detected, the system must annunciate the fault to an operator by some means (for example, an alarm light).

Figure 1: Manual Override Circuit



4.3 Boiler and Combustion Considerations

If your SIL 2-certified RTU32M is used in combustion-related applications, you are responsible for meeting appropriate safety standards including National Fire Protection Association (NFPA) standard NFPA 85 and 86. In addition, you must provide a documented lifecycle-system safety analysis that addresses the requirements of NFPA 85 related to Burner Management System Logic.
To comply with the requirements of IEC 61508, the safety demand rate must be no more than 10 demands per year.
You must also consider system reaction capability.
If your system must meet standard EN 50156, then you must also meet the requirements that are identified in the current version of EN 50156.

4.4 Typical SIL 2 Configurations

SIL 2-certified RTU32M can be used in standard (simplex) or high-availability (duplex) configurations. For the purposes of documentation, the various levels of availability that can be achieved by using various RTU32M configurations are referred to as simplex or duplex. When using a duplex RTU32M configuration, the RTU32M remains simplex (1oo1) from a safety perspective.

IMPORTANT The system operator is responsible for the following tasks when any of the RTU32M configurations are used:

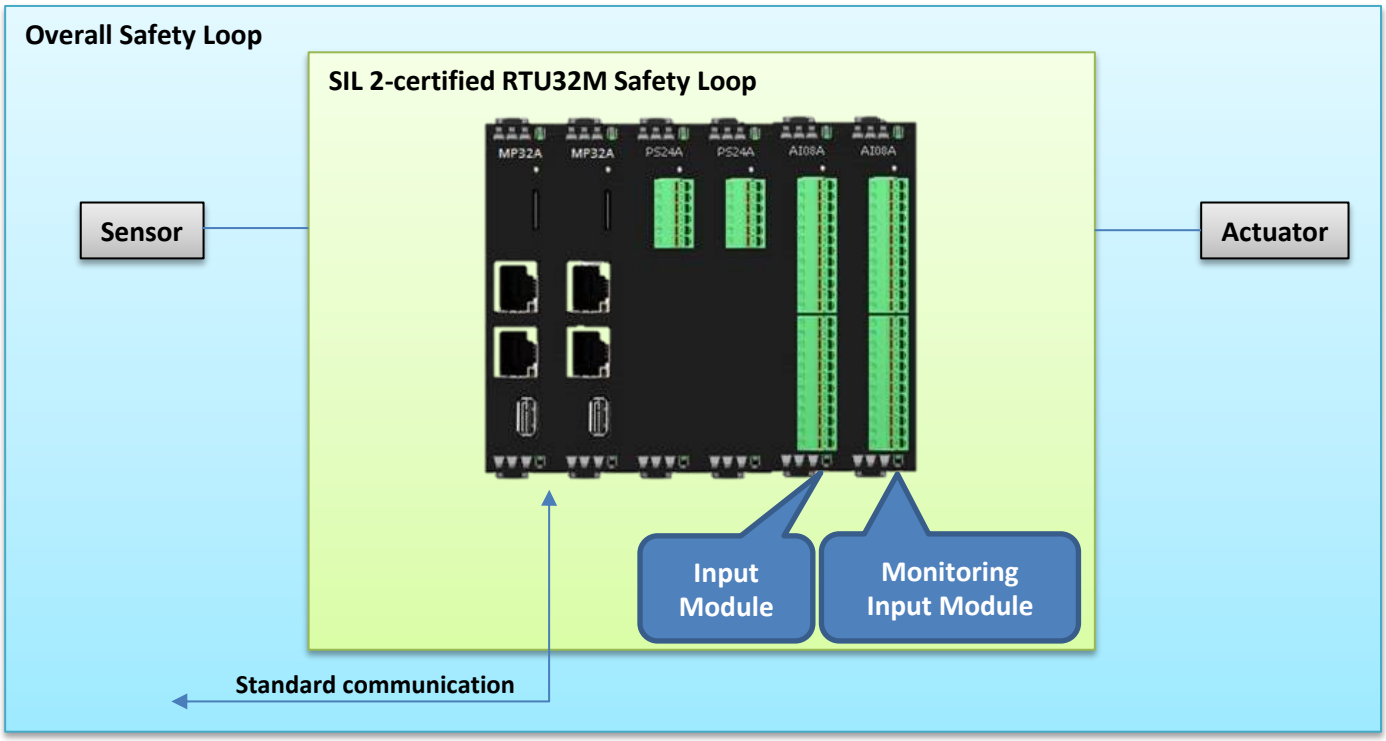
- The setup, SIL rating, and validation of any sensors or actuators that are connected to the RTU32M control system
- Project Management and functional testing
- Programming the application software and the module configuration according to the descriptions in this manual

The SIL 2 portion of the certified system excludes the development tools and display/human machine interface (HMI) devices; these tools and devices must not be part of the safety loop.

4.4.1 Simplex Configuration

In a simplex configuration, the hardware that is used in the safety loop is programmed to fail to safe. The failure to safe is typically an emergency shutdown (ESD) where outputs go to fail-safe state.

Figure 2: Single-chassis Configuration



4.5 Proof Tests

IEC 61508 requires that you perform various proof tests of the equipment that is used in the system. Proof tests are performed at user-defined times (for example, proof test intervals can be once a year, once every 2 years or whatever time frame is appropriate based on the SIL verification calculation) and could include some of the following tests:

- Test all safety application-fault routines to verify that process parameters are monitored properly and the system reacts properly when a fault condition arises.
- Test all digital input or output channels to verify that they are not stuck in the ON or OFF state.
  - Manually cycle inputs to make sure that all inputs are operational and not stuck in the ON state.
  - Manually test outputs that do not support runtime pulse testing.
  - You can automatically perform proof tests by switching ground open on input modules and check to make sure that all input points go to zero (turn OFF.).
- The relays in the redundant power supplies must be tested to make sure that they are not stuck in the closed state.
- Calibrate analog input and output modules to verify that accurate data is obtained from and used on the modules.

IMPORTANT Each specific application has its own time frame for the proof test interval.

4.6 Proof Testing with Redundancy Systems

A RTU32M redundancy system uses an identical pair of RTU32M chassis to keep your process running if a problem occurs with one of the chassis. When a failure occurs in the primary chassis, control switches to the secondary controller in the secondary chassis. The switchover can be monitored so that the system notifies the user when it has occurred. In this case (that is, when a switchover takes place), we recommend that you replace the failed controller within the mean time to restoration (MTTR) for your application. If you are using controller redundancy in a SIL 2 application, you must perform the proof test on the primary controller and on the secondary controller.

TIP If you are concerned about the availability of the secondary controller if the primary controller fails, it is good engineering practice to implement a switchover periodically (for example, once per proof test interval).

4.7 Reaction Times

The response time of the system is defined as the amount of time it takes for a change in an input condition to be recognized and processed by the controller's logic program, and then to initiate the appropriate output signal to an actuator.
The system response time is the sum of the following:

- Input hardware delays
- Input filtering
- I/O and communication module RPI settings
- Controller program scan times
- Output module propagation delays
- Redundancy system switchover times (applicable in duplex systems)

Each of the times listed is variably dependent on factors such as the type of I/O module and instructions used in the logic program.

4.8 Reaction Times in Redundancy Systems

The worst-case reaction time of a duplex system is different than a simplex system. The redundancy system has a longer reaction time because of the following:

- There are a series of crossloading operations that continuously occur between the primary and secondary controllers so it increases scan time.
- The switchover between controllers slows system response.

IMPORTANT To avoid nuisance trips, you must account for the additional cross checking time of a duplex system when setting the watchdog time.

4.1 Safety Watchdog

Configure the properties of the SIL 2 safety task correctly for your application.

- Priority: must be the highest-priority task in the application (lowest number)
- Watchdog: the value that is entered for the SIL 2 safety task must be large enough for all logic in the task to be scanned

If the task execution time exceeds the watchdog time, a major fault occurs on the controller. You must monitor the watchdog and program the system outputs to transition to the safe state in the event of a major fault occurring on the controller.

4.2 Safety Certifications and Compliance

Diagnostic hardware and firmware functions, and how you apply RTU32M components, enable the system to achieve CL SIL 2 compliance.

IMPORTANT You must implement these requirements, or at a minimum the intent of the requirements that are defined in this manual, to achieve CL (claim limit) SIL 2.

5 Features of the RTU32M SIL 2 System

The diagnostic methods and techniques that are used in the RTU32M platform let you configure and program RTU32M controllers to perform checks on the total system. The checks include configuration, wiring, and performance, monitoring input sensors and output devices. Timestamping of I/O and diagnostic data also aid in diagnostics.

Examples of these methods and techniques include the following:

- If an anomaly (other than automatic shutdown) is detected, the system can be programmed to initiate user-defined fault handling routines.
- Output modules can turn OFF selected outputs in the event of a failure.
- Diagnostic I/O modules self-test to make sure that field wiring is functioning.
- Output modules use pulse testing to make sure that output switching devices are not shorted.

5.1 Module Fault Reporting

Every module in the system is 'owned' by one controller. Multiple controllers can share data, and consume data from non-owned modules. When a controller 'owns' an I/O module, that controller stores the module's configuration data, which you define; this data dictates how the module behaves in the system. Inherent in this

configuration and ownership is the establishment of a 'heartbeat' between the controller and module, which is known as the requested packet interval (RPI).
The RPI defines a time interval in which the controller and I/O module must communicate with each other. If, communication cannot be established or maintained, for example, the I/O module has failed, the communication path is unavailable, the system can be programmed to run specialized routines. These specialized routines can determine whether the system can continue functioning or whether the fault condition warrants a shutdown of the application. For example, the system can be programmed to retrieve the fault code of the failed module. It can also make a determination, which is based on the type of fault, whether to continue operating.
The controller can monitor the health of I/O modules in the system. The controller can take appropriate action that is based on the severity of a fault condition and gives you complete control of the application. It is your responsibility to establish the course of action appropriate to your safety application.
For more information on Fault Handling, see Faults in the RTU32M System.

5.2 Data Echo Communication Check

Output data echo allows you to verify that the correct output module received the ON/OFF command from the controller was received and that the module attempts to execute the command to the field device.
During normal operation, when a controller sends an output command, the output module that receives the command will 'echo' the output command back to the controller upon its receipt. This verifies that the module has received the command and tries to execute it. By comparing the requested state from the controller to the data echo received from the module, you can validate that the signal has reached the correct module. You can also verify that the module attempts to activate the appropriate field-side device. The echo data is technically input data from the output module and is located with the other input module data. Again, it is your responsibility to establish the course of action appropriate for your safety application.
When used with LB2 output modules, the data echo validates the integrity of communication up to the system-side of the module, but not to the field-side. When you use this feature, you can verify the integrity from the controller to the output terminal on the module.
You must verify the ON and OFF state. This verification must be accomplished by monitoring the output command from the output module in an input module or validation by alternative methods. Approve all methods according to IEC 61508. A separate input module is required for an output module.

5.3 Pulse Test

Discrete diagnostic output modules contain a feature called a pulse test. A pulse test can verify the output circuit functionality without actually changing the state of the actuator connected to the output. A short-duration pulse is directed to a particular output on the module. The output circuitry momentarily changes its state long enough to verify that it can change state on demand. The test pulse is fast (milliseconds), and typically does not affect actuators. Some actuators can have electronic front ends and can detect these fast pulses. You can disable pulse testing, if necessary.

6 RTU32M IO Modules

6.1 Using Digital Input Modules

To achieve SIL 2, two digital input modules must be used, with field sensors wired to channels on each module. The software must compare the two channels before reconciling the data.

6.1.1 Requirements When Using Any RTU32M Digital Input Module

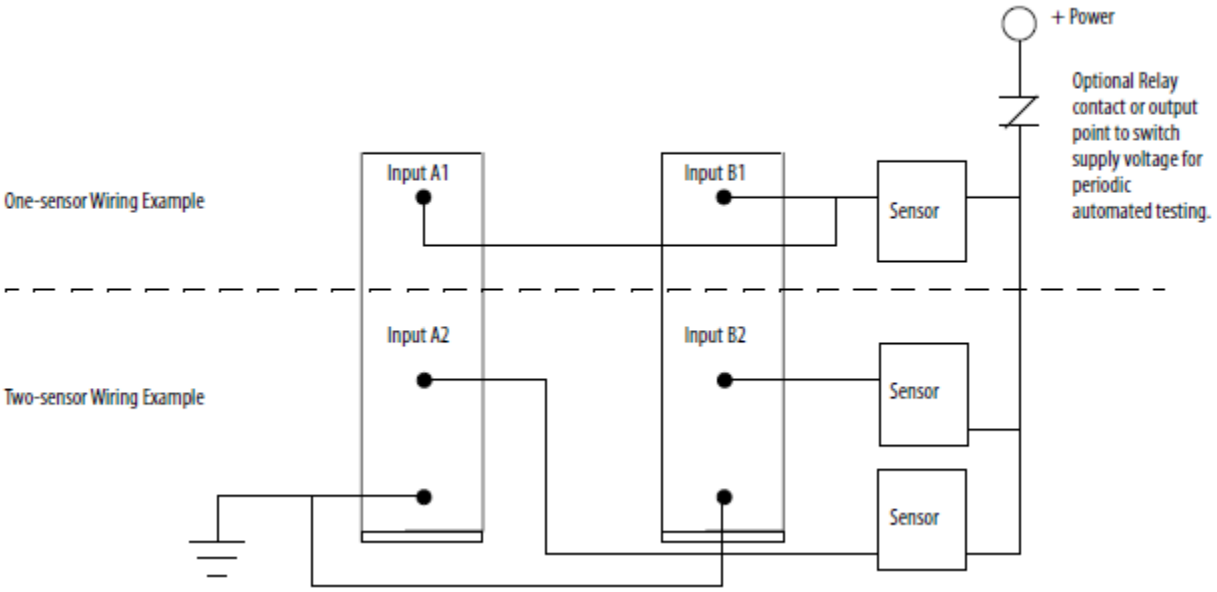
Regardless of the type of RTU32M input module that is used, you must follow these general application requirements when applying these modules in a SIL 2 application:

- **Ownership** – The same controller must own both modules.
- **Separate input points** – Wire sensors to separate input points on two separate modules. The use of two digital input modules is required, regardless of the number of field sensors.
- **Field device testing** – Test field devices by cycling them. The closer you can get to the device being monitored to perform the test, the more comprehensive the test is.
- **Proof tests** – Periodically perform a system validation test. Manually or automatically test all inputs to make sure that they are operational and not stuck in the ON or OFF state. Inputs must be cycled from ON to OFF or OFF to ON. For more information, see Proof Tests.

6.1.2 Wiring Digital Input Modules

This diagram shows two examples of wiring digital inputs. In either case, the type of sensors being used determines whether the use of 1 or 2 sensors is appropriate to fulfill SIL 2 requirements.

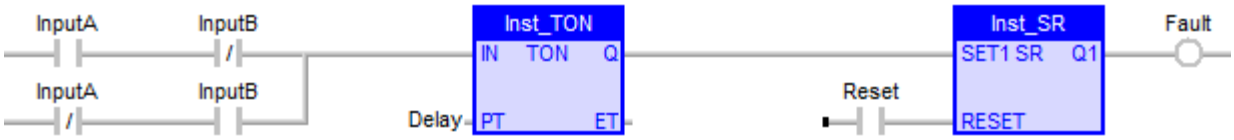
Figure 3: Digital Input Module Wiring Example



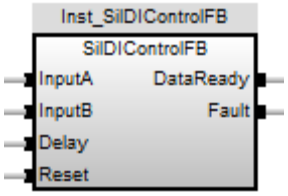
6.1.3 Application logic for Digital Input Modules

Application logic is used to compare input values for concurrence.

Figure 4: Logic-comparing Input Values or States



You can also use a Brodersen predefined function block for this purpose:



The user program must also contain rungs to annunciate a fault in the event of a sustained miscompare between two points.

Figure 5: Rungs Annunciating a Fault

The control, diagnostics, and alarming functions must be performed in sequence. For more information on faults, see Faults in the RTU32M System.

6.2 Using Digital Output Modules

To achieve SIL 2, an output module must be wired back to an input module for monitoring.

6.2.1 Requirements When Using LB2 Digital Output Modules

You must follow these general application requirements when applying these modules in a SIL 2 application:

- **Proof tests -** Periodically perform a system validation test. Manually or automatically test all outputs to make sure that they are operational and not stuck in the ON or OFF state. Outputs must be cycled from ON to OFF or OFF to ON. For more information, see Proof Tests.

- **Examination of output data echoes signal in application logic** – The application logic must examine the Data Echo value that is associated with each output point to make sure that the requested ON/OFF command from the controller was received and acted upon by the module.
  In Figure 21, a timer begins to increment for any miscompare between the controller's output and the module's Data Echo feedback. The discrepancy timer must be set to accommodate the delay between the controller output data and the module's Data Echo response. The time value that is chosen must consider various system RPIs and network latency. If a miscompare exists for longer than that time, a fault bit is set.

The control, diagnostics, and alarm functions must be performed in sequence. For more information on faults, see Faults in the RTU32M System.

- **Use of external relays to disconnect module power if output de-energized state is critical**. To verify that outputs de-energize, you must wire an external relay or other measure, that can remove power from the output module if a short or other fault is detected. See Figure 22 on page 56 for an example method of wiring an external relay.
- **Test outputs at specific times to make sure that they are operating properly**. The method and frequency of testing is determined by the requirements of the safety application.
- **For typical emergency shutdown (ESD) application outputs must be configured to de-energize**: When configuring any LB2 output module, each output must be configured to de-energize in the event of a fault.
- When wiring two digital output modules in series so that one can break source voltage one controller must own both modules.

6.2.2 Wiring LB2 Digital Output Modules

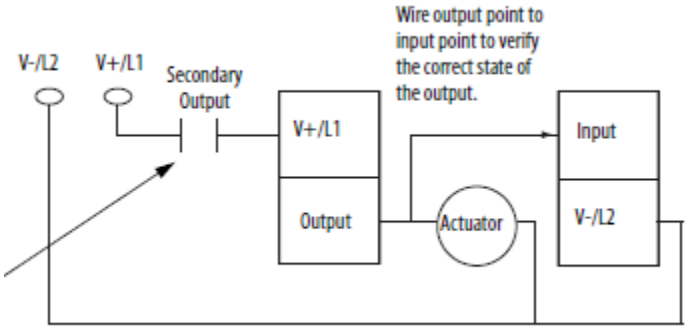
When using output modules, you must wire each output to its field device and also to a system input to monitor the performance. To verify output performance, use one of these methods:

- Write logic to test the ability of the output to turn ON and OFF at power-up.
- At the proof test interval, force the output ON and OFF and use a voltmeter to verify output performance.

For limited high demand applications, test the output modules (that is, you turn the outputs ON and OFF to verify proper operation) once every 8 hours. High demand applications are limited to 10 demands per year for RTU32M SIL 2 systems.

Figure 6: Digital Output Module Wiring

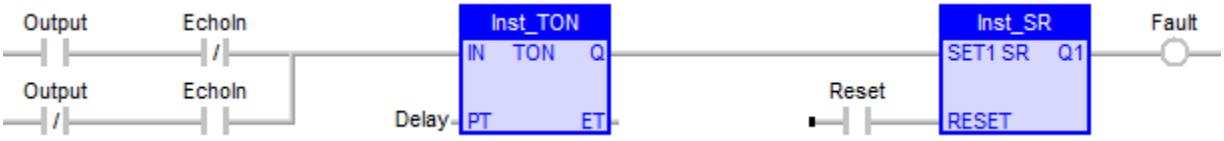
This normally open contact (held closed) must represent the healthy operation of the controller and safety I/O modules. Safety I/O status can be restricted to inputs that affect outputs directly on the specific module. This contact can represent the healthy status of all safety inputs and the controller. The module that is used to control this relay must follow SIL 2 output guidelines. This module must be also considered during PFD analysis for each safety function



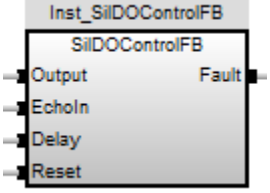
6.2.3 Application logic for Digital Output Modules

Write the application logic to generate a fault in the event of a miscompare between the controller, the actual output state, and the monitored input.

Figure 7: Comparison Logic for Requested Versus Actual Output



You can also use a Brodersen predefined function block for this purpose:

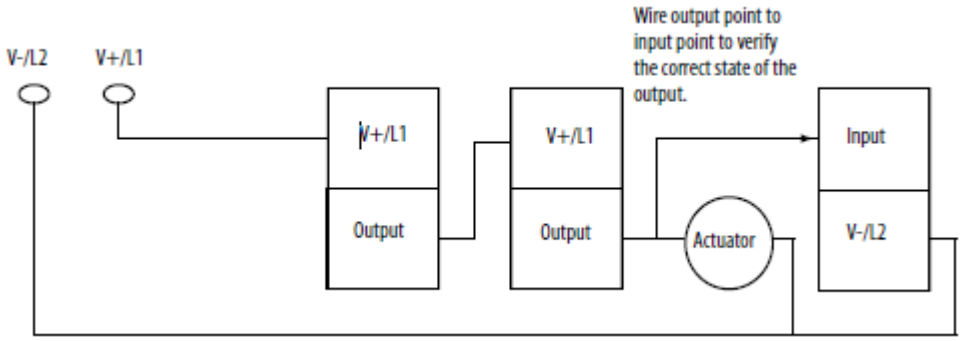


Output Fault contact must represent module and channel diagnostics.

The control, diagnostics, and alarm functions must be performed in sequence. For more information on faults, see Faults in the RTU32M System.

You can also wire two output modules in series to critical actuators. If a failure is detected, the outputs from each of the output modules must be set to OFF to make sure that the field devices de-energize.
Figure 8 shows how to wire two isolated, standard outputs in series to critical actuators.

Figure 8: LB2 Output Module Wiring with Two Modules



6.3 Using Analog Input Modules

There are a number of general application considerations that you must make when using analog input modules in a SIL 2 application. The following section describes those considerations specific to the use of analog input modules.

To achieve SIL 2, two analog input modules are required. Field sensors must be wired to channels on each module and compared within a deadband. Whether one or two field sensors are required is dependent on the Probability of Failure on Demand (PFD) value of the sensor.

6.3.1 Conduct Proof Tests

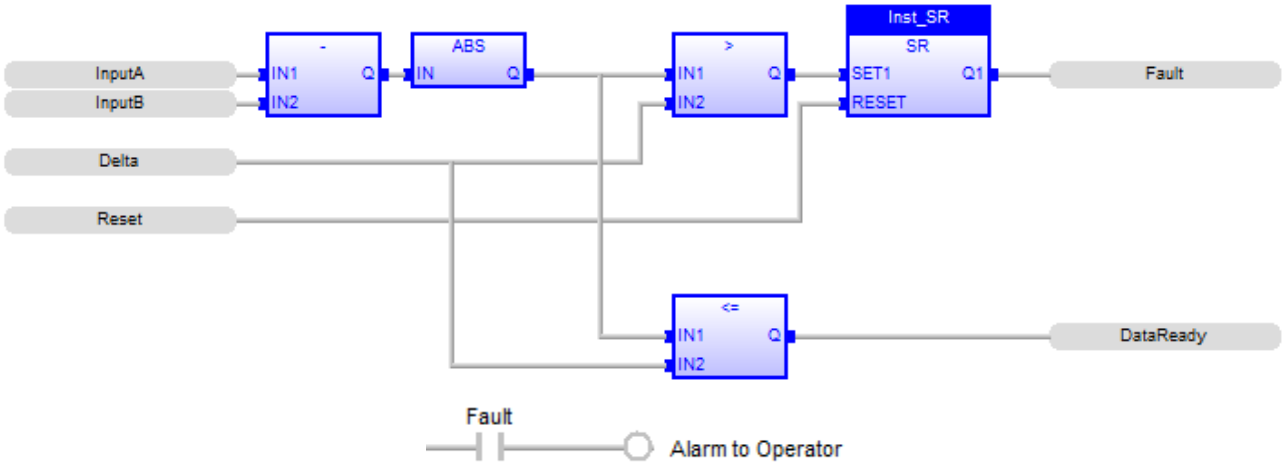
Periodically perform a system validation test. Manually or automatically test all inputs to make sure that they are operational. Field signal levels must be varied over the full operating range to make sure that the corresponding channel data varies accordingly. For more information, see Proof Tests.

6.3.2 Program to Respond to Faults Appropriately

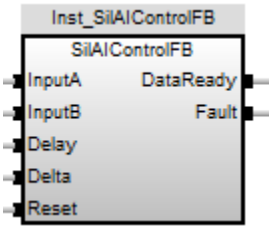
When programming the SIL 2 system, verify that your program examines the appropriate module fault, channel fault, and channel status and responds by initiating the appropriate fault routine.
Each module communicates the operating status of each channel to the controller during normal operation. Application logic must examine the appropriate bits to initiate a fault routine for a given application. For more information on faults, see Faults in the RTU32M System.

6.3.3 Program to Compare Analog Input Data

When wiring sensors to two input channels on different modules, the values from those channels must be compared to each other within the program for concurrence within an acceptable range for the application, before an output is actuated. Any miscompare between the two inputs outside the programmed acceptable range must be annunciated as a fault.
In **Figure 9**, a user-defined percentage of acceptable deviation (that is, tolerance) is applied to the configured input range of the analog inputs (that is, range) and the result is stored (that is, delta). This delta value is then added to and subtracted from one of the input channels; the results define an acceptable High and Low limit of deviation. The second input channel is then compared to these limits to determine if the inputs are working properly.
The input's OK bit preconditions a Timer run that is preset to accommodate an acceptable fault response time and any communication filtering lags in the system. If the inputs miscompare for longer than the preset value, a fault is registered with a corresponding alarm.
Figure 9: Comparison Logic for Two Analog Inputs



You can also use a Brodersen predefined function block for this purpose:



The control, diagnostics, and alarming functions must be performed in sequence. For more information on faults, see Faults in the RTU32M System.

6.3.4 Configure Modules

Configure the modules identically, that is, by using the same RPI, filter values, and so on.
The same controller must own both analog input modules.
You must use Analog Inputs Faulted as a safety status/permissive in respective safety-related outputs.

6.3.5 Wiring LB2 Analog Input Modules

The wiring diagrams that are shown in this section apply to applications that require two transmitters. The type of transmitter along with the application requirements determine whether one or two transmitters are required.
Good design practice dictates that each of the two transmitters must be wired to input terminals on separate modules such that the channel values can be validated by comparing the two within an acceptable range. Special consideration must be given when you apply this technique, depending on the type of module being used.

Wiring the Single-Ended Input Module in Voltage Mode

Make sure you:

- Review the considerations in Using Analog Input Modules on page 60.
- Tie all (-) leads of the transmitters together when operating in single-ended Voltage mode.

Figure 10 shows how to wire an analog input for use in Voltage mode.

Figure 10: LB2 Analog Input Module Wiring in Voltage Mode

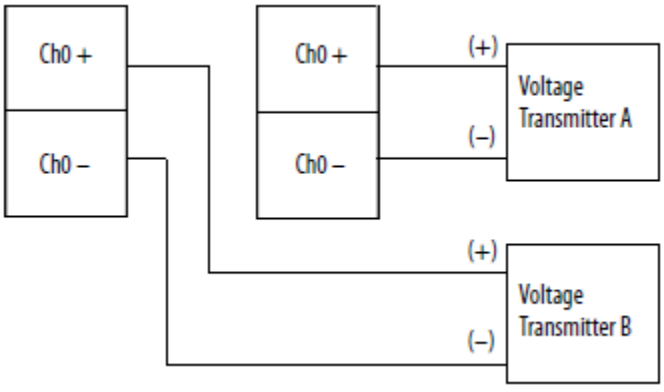
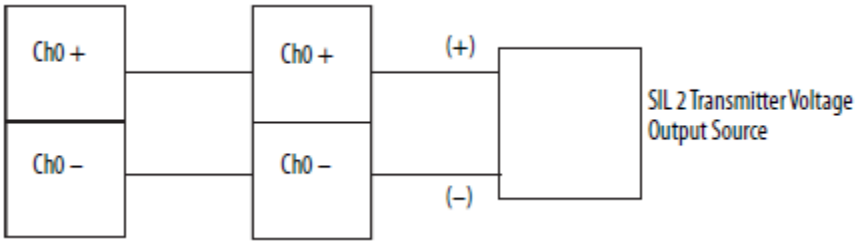


Figure 11 shows how to wire a SIL 2 transmitter to two analog input modules configured for voltage mode.

Figure 11: LB2 Analog Input Module Wiring in Voltage Mode



Wiring the Single-ended Input Module in Current Mode

Make sure you:

- Review the considerations in Using Analog Input Modules on page 60.
-
-
- Place devices correctly in the current loop. You can locate other devices in the current loop of an input channel anywhere as long as the current source can provide sufficient voltage to accommodate all voltage drops. Each module input is 250 Ω.

Figure 12 and **Figure 13** show how to wire an analog input for use in Current mode.

Figure 12: RTU32M Analog Input Module Wiring in Current Mode

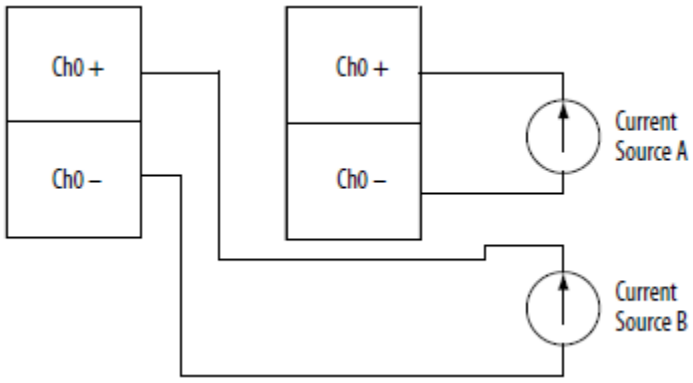
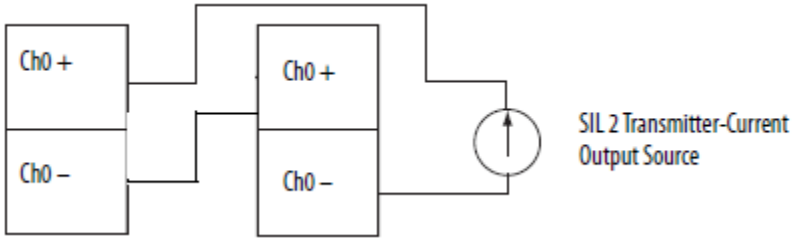


Figure 13: LB2 Analog Input Module Wiring for Isolated Channels (in Current mode)



6.4 Using Analog Output Modules

There are a number of general application considerations that you must make when using analog output modules in a SIL 2 application. An analog output module, along with an analog input module is required to monitor to achieve SIL 2. The following sections describe those considerations specific to the use of analog output modules.

6.4.1 Considerations for Using Analog Output Modules

IMPORTANT It is strongly recommended that you do not use analog outputs to execute the safety function that results in a safe state. Analog output modules are slow to respond to an ESD command and are therefore not recommended for use ESD output modules.

The use of digital output modules and actuators to achieve the ESD de-energized state is recommended.

Conduct Proof Tests

Periodically perform a system validation test. Manually or automatically test all outputs to make sure that they are operational. Field signal levels should be varied over the full operating range to make sure that the corresponding channel data varies accordingly. For more information, see Proof Tests.

Program to Respond to Faults Appropriately

When programming the SIL 2 system, verify that your program examines the appropriate module fault, channel fault, and channel status and responds by initiating the appropriate fault routine.

Each module communicates the operating status of each channel to the controller during normal operation. Application logic must examine the appropriate bits to initiate a fault routine for a given application. For more information on faults, see Faults in the RTU32M System.

Configure Outputs to De-energize in ESD Applications

For typical emergency shutdown (ESD) applications, outputs must be configured to de-energize. When configuring any LB2 output module, each output must be configured to de-energize in the event of a fault.

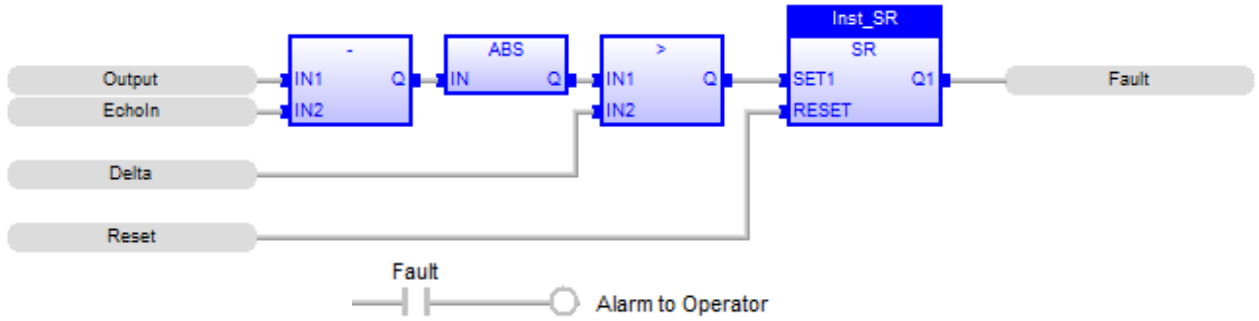
Monitor Channel Status

You must wire each analog output to an actuator and then back to an analog input to monitor the performance of the output, as shown in **Figure 15**. The application logic must examine the analog input (feedback value) associated with each analog output to make sure that the output from the controller was received correctly at the actuator. The analog output value must be compared to the analog input that is monitoring the output to make sure that the value is within an acceptable range for the application.

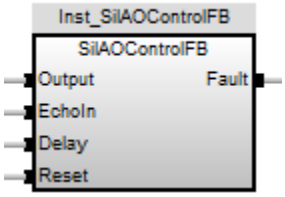
In the function block diagram in **Figure 14**, a user-defined percentage of acceptable deviation (that is, tolerance) is applied to the configured range of the analog input and output and the result is stored (that is, delta). This delta value is then added to and subtracted from the monitoring analog input channel; the results define an acceptable high and low limit of deviation. The analog Output Echo is then compared to these limits to determine if the output is working properly.

The OK output bit preconditions or the Timer run is preset to accommodate an acceptable fault response time and any communication filtering, or output, lags in the system. If the monitoring input value and the Output Echo miscompare for longer than the preset value, a fault is registered with a corresponding alarm.

Figure 14: Monitoring an Analog Output with an Analog Input



You can also use a Brodersen predefined function block for this purpose:



-

The control, diagnostics, and alarm functions must be performed in sequence.

The same controller must own both analog modules.

6.4.2 Wiring RTU32M Analog Output Modules

In general, good design practice dictates that each analog output must be wired to a separate input terminal to make sure that the output is functioning properly.

Wiring the Analog Output Module in Voltage Mode

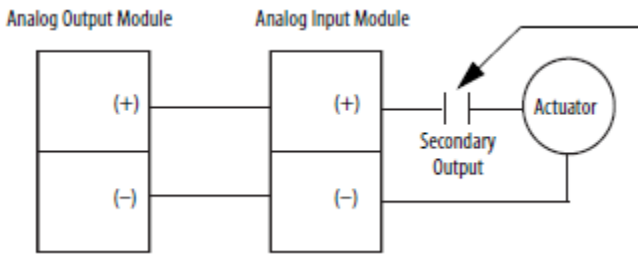
Make sure you:

- Review the considerations in Considerations for Using Analog Output Modules on page 69.

- Use the correct documentation (listed in Additional Resources on page 9) to wire the module.

Figure 15 shows how to wire the output module for use in Voltage mode.

Figure 15: RTU32M Analog Output Module Wiring in Voltage Mode



This normally open relay is controlled by the status of the rest of the RTU32M system. If a short-circuit or fault occurs on the module, the relay can disconnect power to the module. The module that is used to control this relay must follow SIL 2 output guidelines. This module must also be considered during PFD analysis for each safety function.
Use a signal-grade relay using bifurcated or similar grade contacts. The relay can be located in a position to remove power to an actuator, or can remove power to multiple actuators depending on the granularity needed.

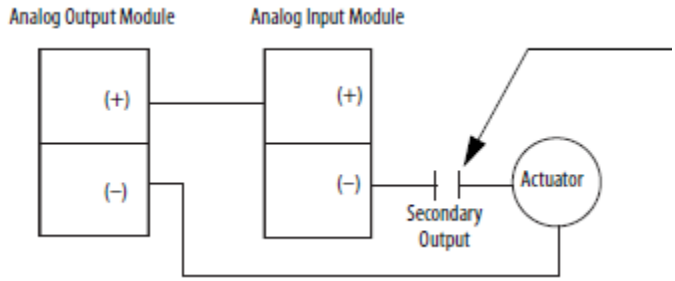
Wiring the Analog Output Module in Current Mode

Make sure you:

- Review the considerations in Considerations for Using Analog Output Modules on page 69.
- Use the correct documentation (listed in Additional Resources on page 9) to wire the module.
- Place devices correctly in the current loop. You can locate other devices in a current loop of the output channel anywhere as long as the current source can provide sufficient voltage to accommodate all voltage drops (each module output is 250 Ω).

Figure 16 shows how to wire the -OF8 module for use in Current mode.

Figure 16: RTU32M Analog Output Module Wiring in Current Mode



This normally open relay is controlled by the status of the rest of the RTU32M system. If a short-circuit or fault occurs on the module, the relay can disconnect power to the module. The module that is used to control this relay must follow SIL 2 output guidelines. This module must also be considered during PFD analysis for each safety function.
Use a signal-grade relay using bifurcated or similar grade contacts. The relay can be located in a position to remove power to an actuator, or can remove power to multiple actuators depending on the granularity needed.

7 Requirements for Application Development

7.1 Software for SIL 2-Related Systems

The application software for the SIL 2-related automation system is created using the Brodersen WorkSuite, according to IEC 61131-3.

The application program has to be created by using the programming tool. Parameters for the operating function are also entered into the system with the programming software.

7.2 SIL 2 Programming

The safety concept of the SIL 2 RTU32M system assumes the following:

- The user who is responsible for creating, operating, and maintaining the application is fully qualified, specially trained, and experienced in safety systems.
- The programming software is installed correctly.
- Control system hardware is installed in accordance with product installation guidelines.
- User application code (user program) uses common and good design practices.
- A test plan is documented and adhered to, including well-understood proof test requirements and procedures.
- A well-designed validation process is defined and implemented.

For the initial startup of a safety-related RTU32M system, the entire system must successfully complete a functional test. After a modification of the application program, the modified program or logic must be checked.

7.3 Programming Languages

It is good engineering practice to keep safety-related logic as simple and easy to understand as possible. The preferred language for safety-related functions is ladder logic, followed by function block. Structured text and sequential function chart are not recommended for safety-related functions.

7.4 Programming Options

Pre-programmed SIL 2 I/O Add-On Instructions can be used in WorkSuite.
SIL 2 Add-On Instructions simplify the programming that is required for a SIL 2 system. However, these instructions are not necessarily suitable for use in all SIL 2 applications and system configurations. You must evaluate the suitability of a SIL 2 Add-On Instruction that is used in a safety-related function.

IMPORTANT If Program Parameters are used, safety-related tags can be read by either standard or safety-related logic or other communication devices, but can be written by only safety-related logic.

7.5 Basics of Application Program Development and Testing

A system integrator develops the application program. The developer must consider general procedures for programming RTU32M SIL 2 applications. (does not require independent third-party review).

- Specification of the SIL 2 safety control function, including the following:
  - Specifications
  - Flow and timing charts
  - Engineering diagrams
  - Sequence charts
  - Program description
  - Program review process
- Writing the application program
- Checking by independent reviewer
- Verification and validation

All application logic must be independently reviewed and tested. To facilitate reviews and reduce unintended responses, limit the set of instructions to basic Boolean/ladder logic (such as examine On/Off, timers, counters) whenever possible. Include instructions that can be used to accommodate analog variables, such as the following:

- Limit tests
- Comparisons
- Math instructions

For more information, see Proof Tests.

7.6 Functional Specification Guidelines

You must create a specification for your control function. Use this specification to verify that program logic correctly and fully addresses the functional and safety control requirements of your application. The specification can be in various formats, depending on your application.
The specification must include a detailed description that includes the following (if applicable):

- Sequence of operations
- Flow and timing diagrams
- Sequence charts
- Program description
- Program print-out
- Written descriptions of the steps with step conditions and actuators to be controlled, including the following:
  - Input definitions
  - Output definitions
  - I/O wiring diagrams and references
  - Theory of operation
- Matrix- or table form of stepped conditions and the actuators to be controlled, including the sequence and timing diagrams
- Definition of marginal conditions, for example, operating modes, EMERGENCY STOP, and others

The I/O-portion of the specification must contain the analysis of field circuits, that is, the type of sensors and actuators.

7.6.1 Sensors (digital or analog)

- Signal in standard operation (dormant current principle for digital sensors, sensors OFF means no signal)
- Determination of redundancies that are required for SIL levels
- Discrepancy monitoring and visualization, including diagnostic logic

7.6.2 Actuators

- Position and activation in standard operation (normally OFF)
- Safe reaction or positions when switching OFF
- Discrepancy monitoring and visualization, including diagnostic logic

7.7 Creating the Application Program

Consider the following when developing the application program logic.

7.7.1 Logic and Instructions

The logic and instructions that are used in programming the application must be:

- Easy to understand.
- Easy to trace.
- Easy to change.
- Easy to test.
- Well-documented.

7.7.2 Program Language

You must implement simple, easy to understand:

- Ladder.
- Other IEC 61131-3-compliant language.
- Function blocks with specified characteristics.

We use ladder, for example, because it is easier to visualize and make partial program changes with this format.

7.7.3 Program Identification

Identify the application program by one of the following:

- Name
- Date
- Revision
- Any other user identification information

7.7.4 SIL Task/Program Instructions

Include a single SIL task composed of programs and routines in the user application. The SIL 2 task must be the top priority task of the controller and the user-defined watchdog must be set to accommodate the SIL 2 task.

IMPORTANT You must dedicate a specific task for safety-related functions and set that task to the highest priority (1). SIL 2 safety logic and logic that is intended for use in non-SIL 2 functions must be separate, or everything in the task containing safety must be treated as safety-related.

7.8 Forcing

The following rules apply to forcing in a project:

- You must remove forces on all SIL 2 tags before beginning normal operation for the project.
- You must not force SIL 2 tags after validation is performed and during controller operation in Run mode.

IMPORTANT Forcing must not be used during normal operation, during final system test, and validation.

7.9 Checking the Application Program

To check safety-related application logic for adherence to specific safety functions, you must generate a suitable set of test cases that cover the safety specification. The set of test cases must be well-written and filed as the test specification.

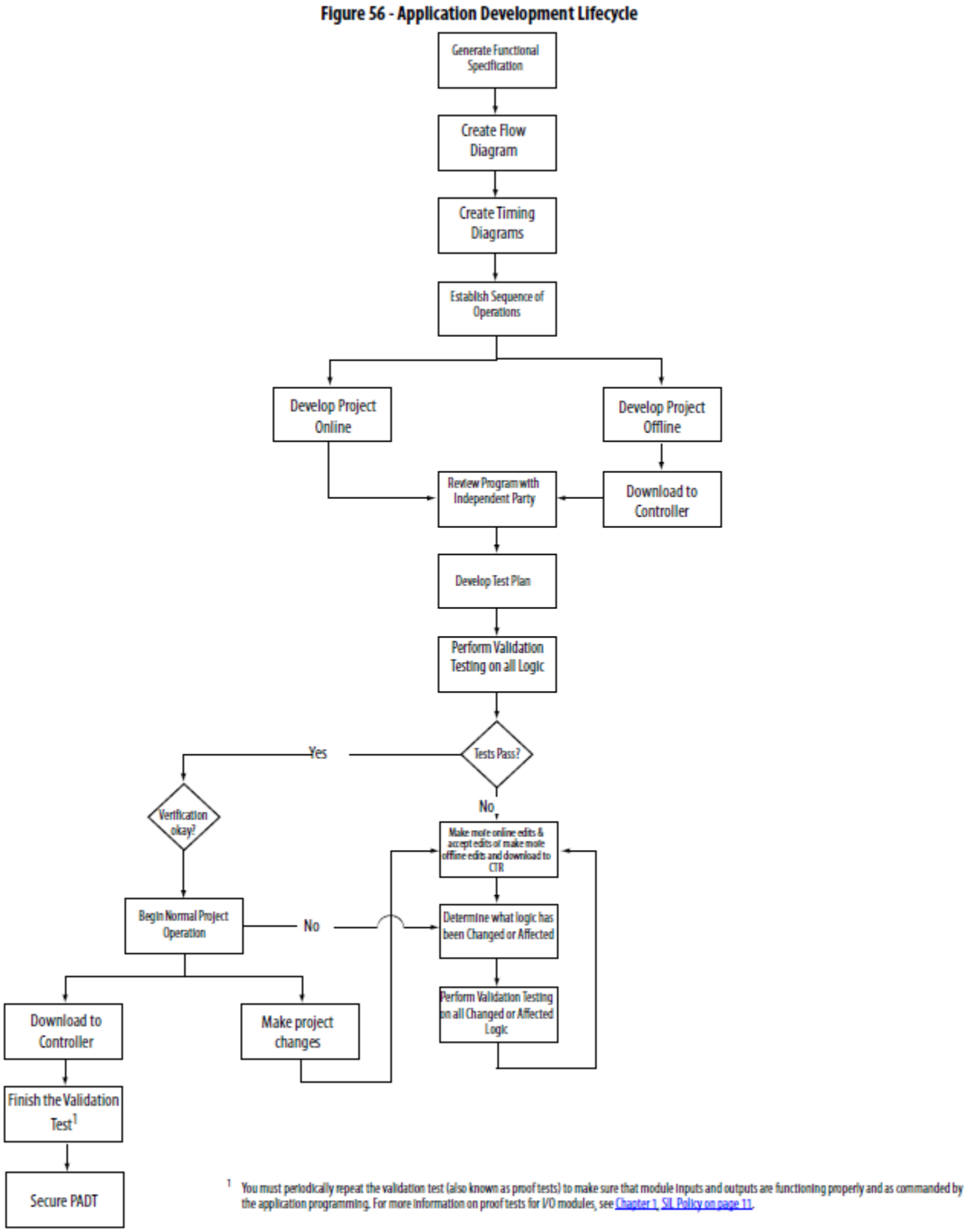
Suitable tests must also be generated for the numeric evaluation of formulas. Equivalent range tests are acceptable. Suitable tests are tests within defined value ranges, at the limits, and outside the defined value ranges. The test cases must be selected to prove the correctness of the calculation. The necessary number of test cases depends on the formula that is used and must comprise critical value pairs.

However, active simulation with sources cannot be omitted. It is the only means to detect the correct wiring of the sensors and actuators to the system. Furthermore, active simulation is the only means to test the system configuration. You must verify the correct programmed functions by forcing I/O or by manual manipulation of sensors and actuators.

7.10 Commissioning Lifecycle

Figure 17 shows the steps that are required to develop, debug, and commission an application program.

Figure 17: Application Development Lifecycle



7.11 Changing Your Application Program

The following rules apply when you change your application program in Brodersen WorkSuite:

IMPORTANT You cannot make program edits while the program is online if the changes prevent the system from executing the safety function or if alternative protection methods are not in place.

- Program edits are not recommended and must be limited. For example, minor changes such as changing a timer preset or analog setpoint are allowed.
- Only authorized, specially trained personnel can make program edits. These personnel must use all supervisory methods available.
- Anyone making data or programming edits to an operational system assumes the central safety responsibility while the changes are in progress. These personnel must also maintain safe application operation.
- Before you make any program edits, perform an impact analysis by following the safety specification and other lifecycle steps that are described in **Figure 17** as if the edits were an entirely new program.
- Sufficiently document all program edits, including:
  - Authorization.
  - Impact analysis.
  - Execution.
  - Test information.
  - Revision information.
- Multiple programmers cannot edit a program from multiple programming terminals simultaneously.

8 Faults in the RTU32M System

8.1 Detect and React to Faults

The RTU32M architecture provides many ways to detect and react to faults in the system.

- Various device objects can be interrogated to determine the current operating status.
- Modules provide runtime status of their operation and of the process that is executing.
- You can configure a RTU32M system to identify and handle faults, including such tasks as:
  - Developing a fault routine.
  - Creating a user-defined major fault.
  - Monitoring minor faults.
  - Developing a power-up routine.

  See RTU32M User Manual, for more information.

It is your responsibility to determine what data is most appropriate for your application to initiate a shutdown sequence.

8.2 Module Fault Reporting for any LB2 I/O Module

You must verify that all components in the system are operating properly. Verification can be accomplished in ladder logic by checking the System Status Function Blocks for a running condition.

9 Use of Human-to-Machine Interfaces

9.1 Precautions

You must exercise precautions and implement specific techniques on HMI devices. These precautions include, but are not restricted to the following:

- Limited access and security
- Specifications, testing, and validation
- Restrictions on data and access
- Limits on data and parameters

Use sound techniques in the application software within the HMI and controller.

IMPORTANT If any changes are needed to the program in the safety loop, they must be done in accordance with IEC 61511-1, paragraph 11.7.1.5, which states:
'The Safety Instrumentation System (SIS) operator interface design shall be such as to prevent changes to SIS application software. Where safety information needs to be transmitted from the basic process control system (BPCS) to the SIS then systems should be used that can selectively allow writing from the BPCS to specific SIS variables. Equipment or procedures should be applied to confirm the proper selection has been transmitted and received by the SIS and does not compromise the safety function of the SIS.'

9.2 Accessing Safety-related Systems

HMI- related functions consist of two primary activities: reading and writing data.

9.2.1 Reading Parameters in Safety-related Systems

Reading data is unrestricted because reading doesn't affect the operation or behavior of the safety system. However, the number, frequency, and size of the data being read can affect controller performance. To avoid safety-related nuisance trips, use good communication practices to limit the impact of communication processing on the controller. Do not set read rates to the fastest rate possible.

9.2.2 Changing Safety-related Parameters in SIL-rated Systems

A parameter change in a safety-related loop via an external (that is, outside the safety loop) device (for example, an HMI) is allowed only with the following restrictions:

- Only authorized, specially trained personnel (operators) can change the parameters in safety-related systems via HMIs.
- The operator who changes a safety-related system via an HMI is responsible for the effect of those changes on the safety loop.
- You must clearly document variables that need changed.
- You must use a clear, comprehensive, and explicit operator procedure to make safety-related changes via an HMI.
- Changes can only be accepted in a safety-related system if the following sequence of events occurs.
    a. The new variable must be sent twice to two different tags; that is, both values must not be written to with one command.
    b. Safety-related code that executes in the controller, must check both tags for equivalency and make sure that they are within range (boundary checks).
    c. Both new variables must be read back and displayed on the HMI device.
    d. Trained operators must visually check that both variables are the same and are the correct value.
    e. Trained operators must manually acknowledge that the values are correct on the HMI screen that sends a command to the safety logic, which allows the new values to be used in the safety function.

In every case, the operator must confirm the validity of the change before they are accepted and applied in the safety loop.

- Test all changes as part of the safety validation procedure.
- Sufficiently document all safety-related changes that are made via HMI, including the following:
    - Authorization
    - Impact analysis
    - Execution
    - Test information
    - Revision information
- Changes to the safety-related system, must comply with IEC 61511 standard on process safety section 11.7.1 Operator Interface requirements.
- The developer must follow the same sound development techniques and procedures that are used for other application software development, including the verification and testing of the operator interface and its access to other parts of the program. The controller application software builds a table that is accessible by the HMI and limits access to required data points only.
- Similar to the controller program, you must secure and maintain the HMI software for SIL-level compliance after the system has been validated and tested.