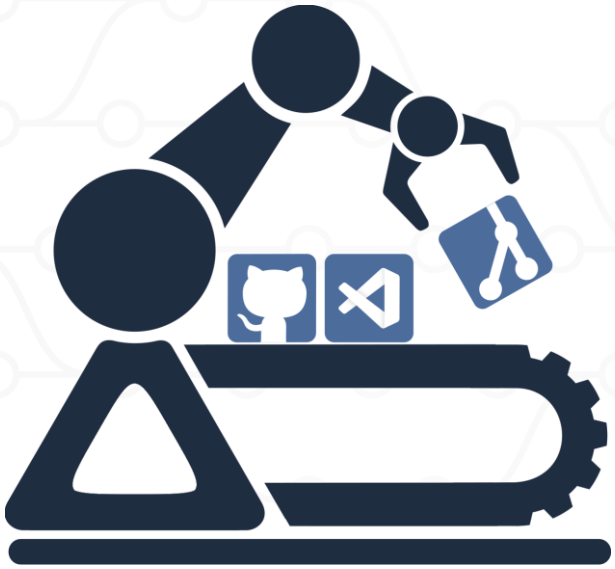
Title Page

MASTERING GITHUB



FROM FIRST COMMIT TO EXPERT USER

YOUR EASY GUIDE TO UNDERSTANDING GIT AND GITHUB WITH VS CODE

MICHAEL GLEDHILL

PUBLISHER

This document and associated materials are published as follows:

Published by: Practical Series of Publications
Published in the United Kingdom

Document number: PS2231-5-2101-001

Document revision: Rel 000.001

Document filename: PS2231-5-2101-001 R000.001 FS Mastering GitHub.docm (File size: 11MB)

Document template: PS0001-5-220101-001 R001.022 Eqty Template.dotm

Based on template: PS0001-5-220101-001 R001.022 Eqty Template.dotm WebIndex: edy

Contents

CONTENTS

PAGE INTENTIONALLY LEFT BLANK

Presenting GitHub

This is a complete guide to using GitHub and its annoying, provincial, command line cousin, Git *(the one that nobody likes or understands)*.

GitHub is the online version of Git and both are version control systems for software (websites, applications, programming, documentation, &c. pretty much anything you can develop on a computer).

They both *(but especially Git)* have a fairly steep learning curve. GitHub is based upon Git and they both use the same convoluted terminology and have the same underlying software engine — what works with Git will work with GitHub.

This site is intended to be a simple, straightforward instruction manual for using Git and GitHub. It is based around Visual Studio Code *(also called VS Code*), a free and powerful text editor from Microsoft which is used to manage the Git and GitHub interfaces.

What is this site about?

WHAT IS THIS SITE ABOUT?

Quite simply it is a guide, tutorial and reference for everything to do with GitHub and Git.

If you are not familiar with Git and GitHub, they are version control systems for software applications.

Version control systems record changes made to any file within a software project; they document why those changes were made, when they were made and by whom.

They not only keep track of every change made to every file within the project, but allow any modified file to be reverted back to a previous state (you can always go back to an earlier version).

Version control systems allow multiple people to work on the project at the same time, even to work on the same file at the same time and they provide mechanisms for

resolving conflicts (where two different people have modified the same section of a file for example).

Git is predominantly a local version control system (it runs on your PC). GitHub is its online partner in crime and allows projects to be viewed by multiple people *(if it is a public repository, it can be viewed by anyone with internet access)* and allows specific people to work on the project simultaneously. GitHub is generally considered to be a software development platform projects can be developed, published and shared with other people.

GitHub is probably the most widely used version control system for software development. There are alternatives: BitBucket, GitLAB, Gitea, Gogs, Google Cloud Source Repostitory, Amazon Code Commit &c. *Just Google it*.

What all these applications have in common is that they all use Git as their underlying version control system engine. There is no getting away from Git.

A complete website that teaches you how to use Git and GitHub and manage the whole thing with Visual Studio Code.

From first commit to expert, your easy guide to understanding Git and GitHub with VS Code.

The website is available here:

1 Functional overview

0

Mastering GitHub

This is a complete guide to using GitHub and its annoying, provincial, command line cousin, Git *(the one that nobody likes or understands)*.

GitHub is the online version of Git and both are version control systems for software (websites, applications, pro-

FC11001 is the *Isolating Valve* block, it is a comprehensive block designed to operate and monitor a standard isolating valve.

1.2 Defining an isolating valve

1.2 Defining an isolating valve

An isolating valve is any valve that has only two states, it is either open or closed and cannot sustain any intermediate position. Isolating valves are generally spring-return type valves that adopt a particular state when energised, and return to the other state when deenergised. Such valves are generally operated directly via an electronic solenoid. Some are pneumatically operated, again by energising a solenoid that when energised applies air to one side to the valve, driving it to the energised state, the solenoid then vents the air when deenergised allowing the valve to return to its deenergised state via the spring-return mechanism within the valve.

Isolating valves have a single flow-path through the valve, and this path is either open or closed depending on the state of the valve.

From the software perspective, there are two types of isolating valve: *normally closed* (the most common type) and *normally open*:

1.2.1 Normally open and normally closed valves

A normally closed valve is closed in the deenergised state and will open when an energising signal is applied — normally closed valves operate in the manner most commonly associated with how valves are envisaged to work. I.e. they open when a voltage is applied to them and close when that voltage is removed.

A normally open valve operates in the reverse arrangement, the valve is open in the deenergised state and will only close when the energising signal is present. Normally open valves are used in fail-safe applications where a power failure causes the valve to open by default, usually to vent pressure or release some other motive force from the vessel to which it is connected.

Both normally open and normally closed valves require the continuous application of the energising signal to maintain the energised state, the valves have no latching characteristic (*cf.* a bi-stable valve that can maintain either state without the continuous application of the energising signal). For example, to maintain an open state with a normally closed valve, the energising signal must be constantly present; if it is removed, the valve will close, and will only re-open when the energising signal is once more present.

1.2.2 Valve position feedback arrangements

1.2.2 Valve position feedback arrangements

Isolating valves can be equipped with limit switch signals that positively identify the current position of the valve. There are four combinations of limit switches:

1	No limit switches	There is no positive indication of the current state of the valve, it is assumed to be in the expected state
2	Closed limit switch only	Positive indication is given for the closed state only. The valve is assumed to be open (after a predetermined time) if the closed signal is inactive
3	Open limit switch only	Positive indication is given for the open state only. The valve is assumed to be closed (after a predetermined time) if the open signal is inactive
4	Open and closed limit switches	Positive indication is given for both states. This allows the true state of the valve to be determined under all circumstances

Table 1.1 Valve position feedback arrangements

Isolating valve limit switches are always linked to the true state of the valve (open or closed, not energised or deenergised).

I.e. a closed limit will always indicate when the valve is closed (preventing flow through the valve); for a normally closed valve, the closed limit will activate when the valve is deenergised. For a normally open valve, the closed limit will activate when the valve is energised.

Note: *Positive indication of a valve state need not be determined by a limit switch (though this is by far the most common arrangement), it could for example, be determined by some process value: if a flow is detected after the opening of a valve, for example.*

In practice, limit switches are the preferred mechanism for the positive indication of a given valve state.

1.3 Module operation

FC11001 is used to control either a normally closed or normally open isolating valve, the module also monitors the various configurations of valve position feedback signals to determine if the valve is in a fault condition.

The module will drive the valve to the requested state by energising or deenergising a digital signal (usually a digital output signal).

The state of the digital output signal for a given state of the valve is dependent on the type of valve in use (normally open or normally closed), the following table shows the state of the output signal for each type of valve:

REQUESTED STATE	NORMALLY OPEN OUTPUT	NORMALLY CLOSED OUTPUT
Closed	Energised	Deenergised
Open	Deenergised	Energised

Table 1.2 Output signal for normally open and normally closed valves

If the requested state of the valve requires the output to be energised, the output will remain energised until the requested state changes.

If the requested state signal changes (either from closed to open, or from open to closed), the module will start either an opening operation timer (request has changed from closed to open) or a closing operation timer (request has changed from open to closed); these timers can have different values.

The valve will be expected to reach the requested state before the relevant operation timer expires; if it does not, the valve will enter a fault state and an appropriate alarm will be generated when the operation timer expires.

Once the valve has reached its requested state, the operation time will reset and the module will continue to monitor the state of the valve.

If the request signal changes whilst the valve is in the process of changing state (e.g. the valve is requested to change from closed to open and whilst the opening operation timer is running, the request is changed back to closed), the running operation timer will be reset and the operation timer for the second request will start from its specified value. I.e. no allowance is made for how far the valve may have moved following the first request change.

If the valve is stable, in a given state (i.e. is not in the process of changing state) and the valve position signals no longer match the required state (e.g. the valve had correctly opened, but at some time after this, the open position feedback is lost), the valve will instantly generate an alarm condition.

1.4 Safety and interlocks

The module, optionally, accommodates a single safety signal (emergency stop), and the standard arrangement of up to three interlock signals (interlock, permissive and trip) as follows:

1.4.1 Safety (emergency stop) signal

Safety or emergency stop signals are hardwired signals that remove power from a device forcing it to a safe state. Such signals act independently of the software (the pressing of an emergency stop button will physically remove power from the affected devices).

The emergency stop signal passed to the module is used to allow the state of the valve to be correctly interpreted within the software under emergency stop conditions.

When the emergency stop signal is removed, the valve will resume the requested state (with the appropriate operation timer).

The presence of an emergency stop signal is indicated via the block icon status signals generated by the module (see §§ 1.8 and 3.2).

Emergency stop signals cannot be bypassed under any circumstances *(cf. interlock signals, below)*.

1.4.2 Interlock signals

There are three types of interlock signal:

① Interlock: a simple interlock that is active whenever a set of conditions is true. If active, it will force the devices to a specified state.

② Permissive: designates a specified "permitted" state for the device. A permissive interlock takes no action if the device is in the non-permitted state, but once the device changes to the permitted state, it will be prevented from changing back to the non-permitted state (e.g. if the permitted state was closed and the valve was currently open when the permissive signal activated, it would not force the valve to close, but once it was closed, it would prevent it from re-opening).

③ Trip: a latching interlock. It activates whenever a set of events are true (like an interlock), but will not deactivate until the triggering conditions are removed and a reset signal has been given (effectively a latching interlock). If active, it will force the device to a specified state. The resetting of the trip is handled by the trip logic itself (and not within this module).

Note: *The trip reset signal is handled by the trip logic, the valve module will simply respond to the presence or absence of a trip signal.*

Each type of interlock has its own individual signal passed to the module (three signals in total).

Interlock signals can be optionally bypassed (effectively disabled) under specific, manually selected operations.

1.4.3 Priority of safety and interlock signals

1.4.3 Priority of safety and interlock signals

The presence of any interlock stop signal is indicated via the block icon status signals generated by the module (see § 3.2). The priority order of the various types of interlock (if more than one is active at the same time) is as follows:

SIGNAL TYPE	PRIORITY	CAN BE BYPASSED
Emergency Stop	High	NO
Trip	↑	YES
Interlock	↑	YES
Permissive	Low	YES

Table 1.3 Safety and interlock signal priority — High priority signal take precedence over lower priority signals

1.5 Alarms and warnings

1.5 Alarms and warnings

The module can generate up to 5 distinct alarms:

1.	Failed to open	Indicates that the valve did not open when requested to do so. The alarm is generated when the opening operation timer expires without the corresponding position feedback activating.
2.	Failed to close	Indicates that the valve did not close when requested to do so. The alarm is generated when the closing operation timer expires without the corresponding position feedback activating.
3.	Failed while open	Indicates that the valve opened correctly, but then at some later point, the position feedback signals, indicating the open state of the valve, were lost
4.	Failed while closed	Indicates that the valve closed correctly, but then at some later point, the position feedback signals, indicating the closed state of the valve, were lost
5.	External fault	External fault is a separate (and optional) hardwired signal from the valve that indicates the valve itself has detected a problem (e.g. a trip signal from a motorised valve)

The module does not generate any warnings.

1.5.1 Alarm — Failed to open/close

1.5.1 Alarm — Failed to open/close

Failed to open and failed to close alarms are generated if the valve is required to change state and fails to do so within the time set by the relevant operation timer.

For example, if the valve were in a closed state and was requested to open, the *failed to open* alarm would be generated if the valve had not received positive indication that the valve was correctly open (from the valve limit switches) by the time the opening operation timer expired.

If the valve is not equipped with any limit switches, then there are no circumstances under which the failed to open and failed to close alarms can be generated (the valve is always assumed to operate correctly and be in the required position).

1.5.2 Alarm — Failed while open/closed

1.5.2 Alarm — Failed while open/closed

Failed while open and failed while closed (see § 1.5.6 for a note on language usage) closed are generated if the valve is in a steady state (not in the process of changing from open to closed or vice versa) and the positive indication of the valve position changes to indicate the steady state position has changed.

An example would be for a normally closed valve, with both open and closed limit switches, to be in the open (energised) state, where the open limit is active and closed limit is inactive (positively indicating the valve is open). If the power to the valve failed (a blown fuse or broken wire, for example), the valve would deenergise and return to the closed state and this would be reflected in the state of the limit switches. The module, knowing nothing of the power failure, would expect the valve to still be open and would instantly generate a *failed while open* alarm.

If the valve is not equipped with any limit switches, then there are no circumstances under which the failed while open and failed while closed alarms can be generated (the valve is always assumed to operate correctly and be in the required position).

1.5.3 Alarm — External fault

1.5.3 Alarm — External fault

The external fault alarm is activated whenever the external fault signal provided to the module is active. The external fault signal is an optional signal that indicates that the valve itself is experiencing a problem. The signal is more commonly associated with motorised valves where thermal overload signals are used to indicate that the valve motor has experienced a fault, it could also be used to indicate the loss of pneumatic pressure for pneumatic solenoid valves.

1.5.4 Resetting alarm conditions

The four alarms derived from valve position feedback signals:

① Failed to open

② Failed to close

③ Failed while open

④ Failed while closed

Are latching alarms, if the alarm conditions occur, the valve will be placed in a fault condition (see § 1.6) and will no longer be operable, the valve will remain in a deenergised condition until the fault is corrected.

The alarm conditions are only reset when the module receives a `RESET` command.

The fifth alarm:

⑤ External fault

Is a non-latching alarm that is reset automatically whenever the external fault signal is deactivated. The cause of the external fault may however require some form of physical reset.

1.5.5 Open and closed, a note on grammar and usage

1.5.5 Open and closed, a note on grammar and usage

There are some subtle differences between the words *open* and *closed* that reflect how these words are used within the context of this document. These differences are grammatical in nature, and whilst not wishing to cast aspersions upon anyone reading this, the following explanation is included to clarify the usage for those, of say, a more grammatically innocent nature.

The word *open* is both a verb (to *open* the door) and an adjective (the door is *open*), this is not true of the word *close* (in the sense of shutting something rather than in the sense of nearness — to be *close* to something), this is only a verb (to *close* the door), the adjective is *closed* (the door is *closed*, not the door is *close* — the latter would indicate the door is nearby). Conversely, the word *opened* is the past participle of the verb open and is not an adjective (the door is *open* not the door is *opened*).

This leads to some inconsistences in the use of *open*, *opened* and *close*, *closed*. This can be seen in the alarms: "Failed while *open*" and "failed while *closed*", in the first case the word is *open* not *opened* (without the *ed*) and in the second case it is *closed* (with the *d*) not *close*. This is the correct English usage of these words.

In this document, the usage of *open*, *opening* and *opened*, and the alternatives, *close*, *closing* and *closed* are as follows (the differences are with the adjectives, highlighted):

	WORD	MEANING	EXAMPLE
VERB	Open	To open something	A signal to open the valve
ADJECTIVE	**Open**	Not closed	The valve is **open** (cf. the valve is closed)
PAST PARTICIPLE	Opened	It opened	It opened the valve
PRESENT PARTICIPLE	Opening	It is opening	The valve is in the process of opening
VERB	Close	To close something	A signal to close the valve
ADJECTIVE	**Closed**	Not open	The valve is **closed** (cf. the valve is open)
PAST PARTICIPLE	Closed	It closed	It closed the valve
PRESENT PARTICIPLE	Closing	It is closing	The valve is in the process of closing

Table 1.4 The usage of the words open and closed

1.5.6 A note on the use of "while" in alarms

1.5.6 A note on the use of "while" in alarms

The wording used in the *failed while open/closed* alarms: "**while** open" and "**while** closed" is not standard for British English; this would use the word "**whilst**" in place of "**while**" (failed **whilst** open/closed). However, **whilst** and **while** have precisely the same meaning in this context (where the word is used as a conjunction), and since **while** is more commonly used in other versions of English (American English for example), and is generally better understood, it has been decided to use the more common form "**while**" in place of the more esoteric "**whilst**".

This change can be made because both **whilst** and **while** are acceptable in British English and can be freely interchanged (even if the former is more common than the later), it is not incorrect to use **while** in place of **whilst**.

This curtesy does not extend to other linguistic irregularities entertained by the Americans; the American spellings of colour (*color*), centre (*center*), litre (*liter*) &c. along with their very many other philological shortcomings will not be adopted.

Apologies to the more pedantic users of British English and also to Americans in general.

1.6 Fault conditions

The valve is in a fault condition if any alarm conditions is active:

① Failed to open

② Failed to close

③ Failed while open

④ Failed while closed

⑤ External fault

A valve cannot operate if any fault condition is active and it will return to its deenergised state.

For a normally closed valve, the deenergised state is closed and for a normally open valve, the deenergised state is open.

A valve that is in a fault state will ignore any request that would move it to an energised state, irrespective of the source of the command: automatic operation, manual operation or even an interlock condition.

1.7 Operating modes

The module supports the following optional modes of operation:

- Automatic/manual mode (allows the operator to take manual control of the valve and open or close it irrespective of the control system requirements)
- Interlock bypass mode (allows the operator to override permissive, interlock and trip signals
- Simulation mode (allows the operator to override the current position feedback signals with simulated values)
- Remote/local/all selection (allows control of the device to be restricted to a specific SCADA/HMI device)
- Faceplate disabled (informs the supervisory system not to display any faceplates for the device, preventing graphical operator selections)
- By default, the module will be in automatic mode; this means that any open or close instruction received by the module from the control software (continuous or sequential control) will cause the valve to adopt the requested state (if the valve is not in fault and is free of interlock and safety conditions).
- If the module is switched to manual mode, the automatic control signals will be ignored and the module will adopt the state required by the manual request signals set by the operator (again, the valve must not be in fault and must be free of interlock and safety conditions).

- Selecting automatic or manual mode is an operator action and can be implemented at any time. Changing from automatic mode to manual mode is a *bumpless* operation, in which the initial manual state of the valve is set to match the current automatic state. For example, if a valve were open under automatic control, switching the valve to manual would cause the initial manual state to be set to open (matching the automatic state), ensuring that the valve does not initially change state when changing to manual mode.

- Once manual mode is established, the operator is free to open and close the valve as required.

- There is no bumpless transfer back to automatic mode, the valve will simply adopt the state required by the control software.

- Interlocks and safety signals operate in exactly the same way for a valve in manual mode as they do for a valve in automatic mode and will take priority over any command issued in either automatic or manual mode.

By default, the module will be in automatic mode. This means that any open or close instruction received by the module from the control software (continuous or sequential control) will cause the valve to adopt the requested state (if the valve is not in fault and is free of interlock and safety conditions).

If the module is switched to manual mode, the automatic control signals will be ignored and the module will adopt the state required by the manual request signals set by the operator (again, the valve must not be in fault and must be free of interlock and safety conditions).

Selecting automatic or manual mode is an operator action and can be implemented at any time. Changing from automatic mode to manual mode is a *bumpless* operation, in which the initial manual state of the valve is set to match the current automatic state. For example, if a valve were open under automatic control, switching the valve to manual would cause the initial manual state to be set to open (matching the automatic state), ensuring that the valve does not initially change state when changing to manual mode.

Once manual mode is established the operator is free to open and close the valve as required.

There is no bumpless transfer back to automatic mode, the valve will simply adopt the state required by the control software.

Interlocks and safety signals operate in exactly the same way for a valve in manual mode as they do for a valve in automatic mode and will take priority over any command issued in either automatic or manual mode.

1.7.1 Interlock bypass mode

Interlock bypass mode allows the three interlock types (trip, interlock and permissive) to be overridden (*bypassed*) by the operator.

If bypass mode is activated, any interlock signal will be ignored by the module.

Bypass mode is an entirely manual operation that can only be initiated by an operator, usually via a supervisory system (SCADA or HMI).

1.7.2 Simulation mode

Simulation mode

The module can be switched to a *simulation* mode, under which any positive position feedback signals from valve limit switches are replaced by simulated signals selected by the operator. Simulation mode can be used during testing and can also be applied in a process environment if a fault condition is detected within the valve.

The operator can select any one of the following possible simulation arrangements (only one can be active at any given time):

- Follow demand — The simulation will generate open and closed signals that match the requested state of the valve
- Open — The simulation will generate signals to give a permanently open state for the valve, irrespective of any valve state request (allows failed open conditions to be simulated)
- Closed — The simulation will generate signals to give a permanently closed state for the valve, irrespective of any valve state request (allows failed closed conditions to be simulated)
- No limits — No position feedback signals will be simulated

Simulation mode is a manual operation that can only be initiated by an operator.

1.7.3 Remote/local

1.7.3 Remote/local mode

Remote and local operating modes refer to the device that has control of the module. Where a plant has more than one supervisory system, say a SCADA in a control room and a field mounted HMI local to the device, it is possible to restrict the operator control of the module to one, or the other, or both supervisor systems as follows:

- Remote — Only the remote system in the control room can control the device
- Local — A local system has taken control of the device and the remote system can no longer issue commands to it
- All — Any system can control the device

The default arrangement is for all systems to have access to the device.

1.7.4 Faceplate disabled

1.7.4 Faceplate disabled

Prevents the supervisory system from displaying the faceplate from which the above options are selected. If the faceplate is disabled, the operator will not be able to select or change any of the various operating modes.

1.8 Supervisory system interface

Isolating valves are represented on supervisory systems (SCADA and HMI) by both a symbol that displays the current state of the valve and a block icon that provides additional information about the condition of the valve, examples of each are shown below:

STANDARD NC VALVE	STANDARD NO VALVE	MOTORISED NC VALVE	MOTORISED NO VALVE	DESCRIPTION
				Closed
				Opening
				Open
				Closing
				Fault (valve body shows state)
				Loss of communications

Table 1.5 FC11001 symbols NC — Normally closed NO — Normally open

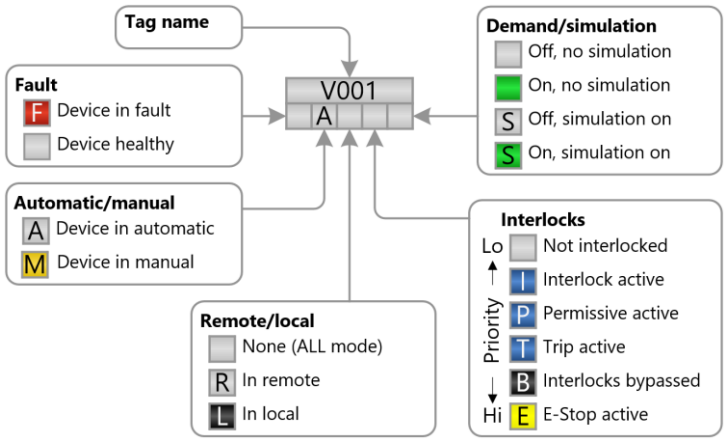


Figure 1.1 FC11001 block icon

All the information displayed in the symbol and block icon is provided as status signals that can be read by a supervisory system.

2 Detailed block description

(1) This module, *FC11001_StdDevValveIsol* uses user data types to provide the primary interface to the module, it also uses direct digital signals to provide an interface for the position feedback signals and for the fault, safety and interlock signals. It also has a single output signal used to operate the valve.

(2) These are the formal module parameters:

PARAMETER	FUNCTION	TYPE	IN/OUT
SYS_SIGNALS	Common system signals logic and timing signals for parametric access	UT21000	In
FBK_OPEN	Optional open state positive feedback (1 = open, 0 = not open) If not required, set permanently to _False	Bool	In
FBK_CLOSED	Optional closed state positive feedback (1 = closed, 0 = not closed) If not required, set permanently to _False	Bool	In
EXT_FAULT	Optional external fault signal for the device (1 = fault, 0 = healthy) If not required, set permanently to _False	Bool	In
E_STOP	Optional emergency stop (safety) signal (1 = emergency stop active, 0 = healthy). If not required, set permanently to _False	Bool	In
TRIP	Optional trip signal (1 = trip condition active, 0 = healthy) If not required, set permanently to _False	Bool	In
INTERLOCK	Optional interlock signal (1 = interlock condition active, 0 = healthy) If not required, set permanently to _False	Bool	In
PERMISSIVE	Optional permissive signal (1 = permissive condition active, 0 = healthy) If not required, set permanently to _False	Bool	In

CONTINUED ON NEXT PAGE

PARAMETER	FUNCTION	TYPE	IN/OUT
STATIC_DATA	Data structure that holds the configuration data for the device (normally open/closed, limit switch and interlock configuration &c.)	UT11001	In
DYNAMIC_DATA	Data structure that holds the live data for the device (its mode, status and messages and any other dynamic information required by the module)	UT31001	InOut
OPERATE	Output to drive the valve to a particular state (1 = energised state, 0 = deenergised state)	Bool	Out

Table 2.1 Module parameters

(3) FBK_OPEN, FBK_CLOSED and EXT_FAULT parameters are input signals provided directly from the device itself (via digital input cards). The OPERATE signal is an output signal (via a digital output card) that causes the valve to open and close.

(4) The E-STOP parameter is the interface to any safety signal that might apply to the valve; similarly, the PERMISSIVE, INTERLOCK and TRIP parameters are the links to any such interlock signals that may be applied to the device.

(5) The STATIC_DATA and DYNAMIC_DATA parameters are UDTs that hold (respectively) the configuration information and the live data for the device.

PARAM	ASSOCIATED UDT	ASSOCIATED DATA BLOCK
STATIC_DATA	*UT11001_St_DevValveIsol*	*DB11001_St_DevValveIsol*
DYNAMIC_DATA	*UT31001_St_DevValveIsol*	*DB31001_St_DevValveIsol*

Table 2.2 Associated UDTs and data blocks

(6) Section 5 contains a full list of all variables within these UDTs.

(7) This arrangement conforms with the PAL philosophy where IO card connections, safety signals and interlocks have unique formal parameters; all remaining signals are passed as variables within UDTs, see the Software Design Specification *[Ref 008]*.

2.1 Typical wiring arrangement

(1) The typical wiring arrangements for a valve with all signals is as follows:

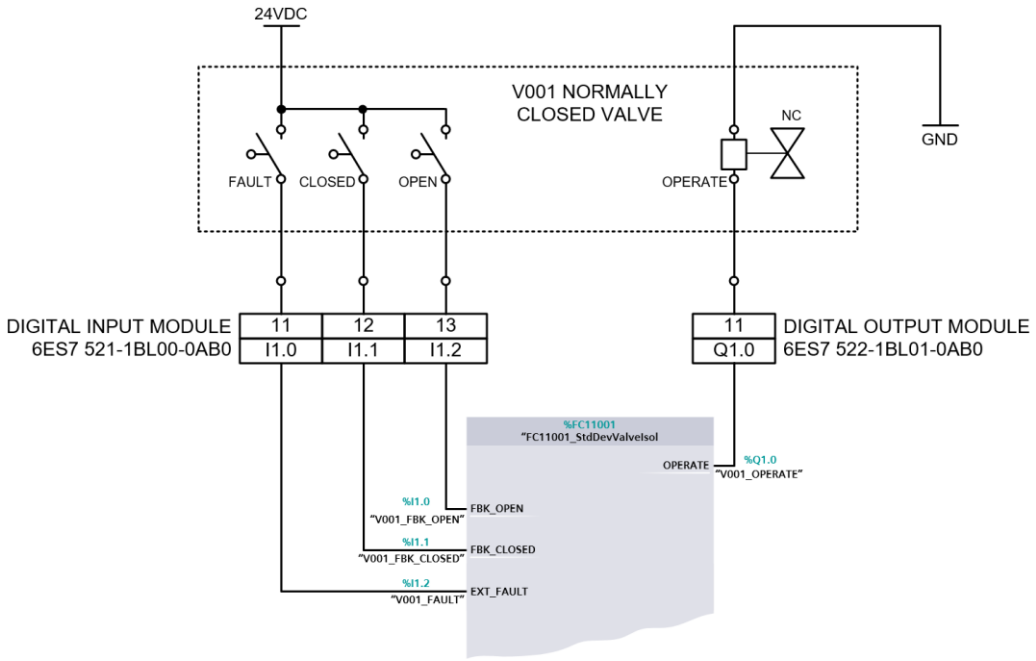


Figure 2.1 Electrical wiring arrangements

Note: *The digital input and output cards shown above are typical 24V DC IO cards, in this case 32 channel cards. Terminal allocations are correct examples for the cards indicated.*

2.2 Normal operation

(1) Normal operation can generally be considered the operation of both normally open and normally closed valves under automatic or manual control (see § 2.6.1 for a full explanation of how automatic and manual modes are selected). And the mechanisms used to determine the state of the valve.

2.2.1 Opening and closing the valve

(1) Under automatic mode, the valve is controlled by separate software within the Controller (this is usually derived from either continuous control logic or sequential control logic and implemented by the command handling application module).

(2) The result of this software will be to either open the valve or close the valve.

(3) Under manual mode, the valve is controlled directly by the operator via some form of supervisory system (SCADA or HMI). Again, the result of the manual operator action will be to either open or close the valve.

(4) There are two signals that, under normal operation, can cause the valve to move to a particular state, one for automatic mode and one for manual mode, both signals are passed parametrically to the module as part of the DYNAMIC_DATA parameter (*UT31001_Dy_DevValveIsol*):

DATA STRUCTURE SIGNAL	*UT31001_Dy_DevValveIsol* FUNCTION	TYPE	DETAILS
ctrl_Aut_OpenClose	Causes the valve to open and close if automatic mode is active (mode_AutMan = 1) If automatic mode is not active, the signal is ignored	Bool	1 = open the valve, 0 = close the valve
ctrl_Man_OpenClose	Causes the valve to open and close if manual mode is active (mode_AutMan = 0) If manual mode is not active, the signal is ignored	Bool	1 = open the valve, 0 = close the valve

Table 2.3 Signals to operate the valve

Note: *The above signals specify the required physical state of the valve (either open or closed), i.e. it specifies the true state required of the valve irrespective of whether the valve is normally open or normally closed.*

The module will determine whether the OPERATE *signal should be activated or deactivated to achieve the required state depending on the configuration of the valve (either normally open or normally closed).*

(5) The operating mode (either automatic or manual) is specified within the DYNAMIC_DATA parameter UDT (*UT31001_Dy_DevValveIsol*) with the variable mode_AutMan as follows:

DATA STRUCTURE	*UT31001_Dy_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
Mode_AutMan	Identifies the operating mode of the valve. If manual mode is disabled (see below), then Mode_AutMan is set permanently to 1 (automatic mode)	Bool	1 = automatic mode, 0 = manual mode

Table 2.4 Automatic/manual mode selection

(6) It is possible to configure the module such that manual mode is permanently disabled; this uses the following STATIC_DATA parameter:

DATA STRUCTURE	*UT11001_St_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
CONFIG_MAN_DIS	Prevents manual mode being activated under all circumstances; the device is permanently in auto, Mode_AutMan is set permanently to 1	Bool	1 = no manual mode, 0 = manual permitted

Table 2.5 Manual mode configuration

(7) If manual mode is disabled (configuration signal CONFIG_MAN_DIS is set to 1), then manual mode cannot be turned on under any circumstances, the option will be greyed out on the supervisory system faceplate and the module will constantly set the mode_AutMan signal to 1.

(8) By default the device will be in automatic mode.

(9) The state of the OPERATE parameter is determined by either the ctrl_Aut_OpenClose or ctrl_Man_OpenClose (depending on whether automatic mode or manual mode is active) and the type of valve in use (either normally open or normally closed).

(10) The type of valve is determined by the CONFIG_NO_NC signal specified within the STATIC_DATA parameter UDT (*UT11001_St_DevValveIsol*) as follows:

DATA STRUCTURE SIGNAL	*UT11001_St_DevValveIsol* FUNCTION	TYPE	DETAILS
CONFIG_NO_NC	Identifies the type of the valve as either normally open or normally closed. A normally closed valve is energised to open, deenergised to close. A normally open valve is deenergised to open and energised to close.	Bool	1 = Normally open, 0 = Normally closed

Table 2.6 Valve type — normally open or normally closed

(11) The various states of the valve for both automatic/manual and normally open/closed are given in the following table:

MODE	Automatic mode (mode_AutMan = 1)			
ctrl_Aut_OpenClose	ctrl_Man_OpenClose	CONFIG_NO_NC	OPERATE	VALVE STATE
0	N/A	0 (normally closed)	0	Closed
1	N/A	0 (normally closed)	1	Open
0	N/A	1 (normally open)	1	Closed
1	N/A	1 (normally open)	0	Open
MODE	Manual mode (mode_AutMan = 0)			
ctrl_Aut_OpenClose	ctrl_Man_OpenClose	CONFIG_NO_NC	OPERATE	VALVE STATE
N/A	0	0 (normally closed)	0	Closed
N/A	1	0 (normally closed)	1	Open
N/A	0	1 (normally open)	1	Closed
N/A	1	1 (normally open)	0	Open

Table 2.7 Automatic/manual and normally open/close valve states

(12) The module will determine the correct state of the OPERATE parameter based on the operating mode, required state in that mode and the type of valve in use.

(13) If the valve is in a fault state (see § 2.3), the OPERATE parameter will be set to 0 (deenergised), irrespective of any other signals.

2.2.2 Operation timers

(1) The module has an operation timer that is used for both opening and closing the valve.

(2) The timer is loaded with either the opening operation time or closing operation time whenever the valve is requested to change state. The timer counts down from the loaded value to zero (the valve can have different times for opening and closing).

(3) The specific operating times are given within the STATIC_DATA parameter UDT (*UT11001_St_DevValveIsol*) as follows:

DATA STRUCTURE SIGNAL	*UT11001_St_DevValveIsol* FUNCTION	TYPE	UNITS
TIME_OPENING_MAX	The maximum time required for the valve to travel from a closed state to an open state	Real	Seconds
TIME_CLOSING_MAX	The maximum time required for the valve to travel from an open state to a closed state	Real	Seconds

Table 2.8 Opening and closing timer preset values

(4) If the timer is running, the elapsed (counting down) time is stored within the DYNAMIC_DATA parameter UDT (*UT31001_Dy_DevValveIsol*) as follows:

DATA STRUCTURE SIGNAL	*UT31001_Dy_DevValveIsol* FUNCTION	TYPE	UNITS
actual_Operating_Timer	Actual (elapsed) value of the operating opening/closing timer	Real	Seconds

Table 2.9 Elapsed opening/closing timer

(5) There is a single operating timer for both opening and closing (opening and closing are mutually exclusive operations, hence only one timer is needed. The timer is preloaded with either the opening time preset or the closing time preset prior to the timer starting).

(6) The preset signals TIME_OPENING_MAX and TIME_CLOSING_MAX are the maximum possible times it can take for the valve to reach the expected position. This should be set to between 110% and 120% of the normal opening and closing times specified for the valve. For example, if a valve has a normal opening time of 5 seconds and a normal

closing time of 3 seconds, sensible values for TIME_OPENING_MAX and TIME_CLOSING_MAX would be between 5.5-6.0 s and 3.3-4.0 s respectively.

(7) A valve will enter a fault state (see § 2.3) if the operation timer counts down to a value of zero without confirmation that the valve has reached the required state. Such confirmation is provided by positive feedback from the valve limit switches (or other indicating devices) or by simulated signal within the module itself (see § 2.2.3).

(8) The operation timer always counts down from a preset value to zero, this is in accordance with the general requirements and practices of the PAL software.

2.2.3 Position feedback signal

(1) The module can accommodate either none, one or two positive position indication signals:

1	No limit switches	There is no positive indication of the current state of the valve, it is assumed to be in the expected state
2	Closed limit switch only	Positive indication is given for the closed state only. The valve is assumed to be open (after a predetermined time) if the closed signal is inactive
3	Open limit switch only	Positive indication is given for the open state only. The valve is assumed to be closed (after a predetermined time) if the open signal is inactive
4	Open and closed limit switches	Positive indication is given for both states. This allows the true state of the valve to be determined under all circumstances

Table 2.10 Possible feedback signal arrangements

(2) There are two configuration signals that determine the position feedback arrangements, these are stored within the STATIC_DATA parameter UDT as follows:

DATA STRUCTURE	*UT11001_St_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	UNITS
CONFIG_FBK_OPEN_EN	Indicates that positive feedback for the open state is present	Bool	1 = open feedback in use, 0 = no open feedback
CONFIG_FBK_CLOSED_EN	Indicates that positive feedback for the open state is present	Bool	1 = closed feedback in use, 0 = no closed feedback

Table 2.11 Possible feedback signal configuration

(3) If the CONFIG_FBK_OPEN_EN signal is activated (set to 1), the module will use the state of the signal connected to the FBK_OPEN parameter to determine the open position of the valve.

(4) If the CONFIG_FBK_OPEN_EN signal is deactivated (set to 0), the module will derive the open position of the valve such that it matches the requested state of the valve.

(5) Similarly, if the CONFIG_FBK_CLOSED_EN signal is activated (set to 1), the module will use the state of the signal connected to the FBK_CLOSED parameter to determine the closed position of the valve.

(6) If the CONFIG_FBK_CLOSED_EN signal is deactivated (set to 0), the module will derive the closed position of the valve such that it matches the requested state of the valve.

(7) Where a CONFIG_FBK_OPEN/CLOSE_EN signal is activated (set to 1), a `true` state on the associated parameter FBK_OPEN/CLOSED indicates that the valve is in that position, a `false` state indicates that the valve is not in that position as follows:

OPEN SIGNALS		
CONFIG_ FBK_OPEN_EN	FBK_OPEN	MEANING
1	0	Not open
1	1	Confirmed open
0	N/A	Internally derived open signal matches requested valve state, FBK_OPEN is ignored
CLOSED SIGNALS		
CONFIG_ FBK_CLOSED_EN	FBK_CLOSED	MEANING
1	0	Not closed
1	1	Confirmed closed
0	N/A	Internally derived closed signal matches requested valve state, FBK_CLOSED is ignored

Table 2.12 Association between configuration signals and position feedback signals

Position status signals

Position status signals

(8) The position and condition of the valve is reflected in various status signals, these are as follows:

DATA STRUCTURE SIGNAL	*UT31001_Dy_DevValveIsol* FUNCTION	TYPE	DETAILS
status_Closed	Indicates the valve is in the closed state	Bool	1 = valve is closed, 0 = valve is not closed
status_Open	Indicates the valve is in the open state	Bool	1 = valve is open, 0 = valve is not open
status_Closing	Indicates the valve is in the process of closing (i.e. closing operation timer is running)	Bool	1 = valve is closing, 0 = valve is not closing
status_Opening	Indicates the valve is in the process of opening (i.e. opening operation timer is running)	Bool	1 = valve is opening, 0 = valve is not opening
status_Operating	Indicates the valve is opening or closing (i.e. either status_Closing or status_Opening is active – set to 1)	Bool	1 = valve is changing state, 0 = valve is static
status_Fault	The valve is in a fault condition (at least one of the five alarm conditions is active)	Bool	1 = valve is in fault, 0 = valve is healthy

Table 2.13 Symbolic status information

Understanding derived position signals

Understanding derived position signals

(9) Where ether the CONFIG_FBK_OPEN_EN or CONFIG_FBK_CLOSED_EN signal is deactivated (set to 0), the associated position feedback is internally determined by the module according to the actual state of the valve (determined by the OPERATE parameter and the type of valve in use).

(10) These internally derived signals are themselves configurable and can be set such that the position feedback signal occurs after a particular percentage of the associated operation timer has expired.

(11) For the derived FBK_OPEN signal, the configurable percentage signal is TIME_DER_OPEN stored within the STATIC_DATA parameter. Similarly, for the derived FBK_CLOSED signal, the configurable percentage signal is TIME_DER_CLOSED, again, stored within the STATIC_DATA parameter.

(12) The operation of these parameters can be seen in Figure 2.2:

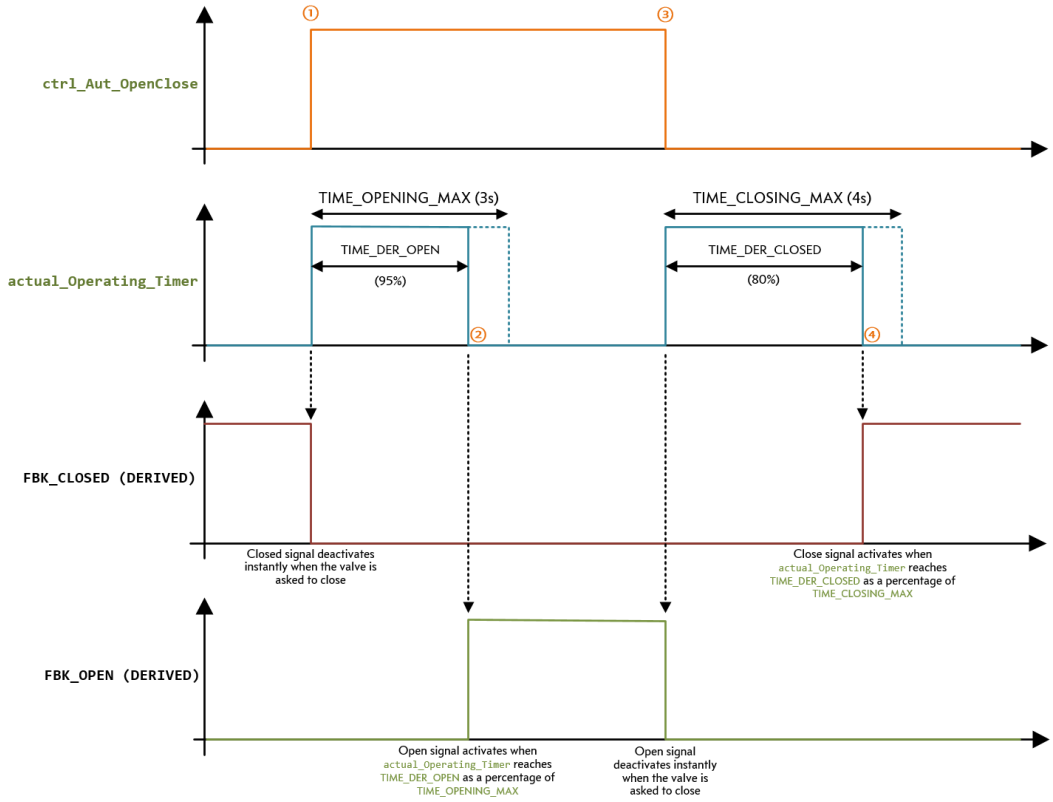


Figure 2.2 Deriving open and closed signal when positive feedback is not used

(13) Figure 2.2 assumes a normally closed valve with no position feedback signals (both CONFIG_FBK_OPEN_EN and CONFIG_FBK_CLOSED_EN are set to 0) operating in automatic mode. It also assumes that the valve is free to operate without restriction (there are no active interlock or safety signals).

(14) The valve is initially closed with the derived FBK_CLOSED signal active (1) and the derived FBK_OPEN signal deactivated (0). At some point ①, the module receives a request to open the valve.

(15) At this point two things happen, firstly the operation timer starts, with a countdown from the value TIME_OPENING_MAX, which in this case is 3 seconds. Secondly, the derived FBK_CLOSED signal is deenergised (indicating the valve is no longer closed).

(16) The simulated opening time (TIME_DER_OPEN) is set to 95%, this means that the derived FBK_OPEN signal will activate when the actual_Operating_Timer value has counted down for 95% of the TIME_OPENING_MAX value (i.e. when the timer has counted down to 0.15s or less).

(17) The exact formula used is:

$$\text{Derived FBK\_OPEN} = 1 \text{ when}$$
$$\text{actual\_Operating\_Timer} \leq \text{TIME\_OPENING\_MAX} - \left(\text{TIME\_OPENING\_MAX} \times \frac{\text{TIME\_DER\_OPEN}}{100}\right)$$

(18) This can be seen at point ②, where the derived FBK_OPEN signal activates. The valve now remains in the open state (FBK_OPEN = 1. FBK_CLOSED = 0) until there is a request to close the valve, point ③.

(19) Again, at point ③, the module receives a request to close the valve.

(20) At this point two things happen, firstly the operation timer starts, with a countdown from the value TIME_CLOSING_MAX, which in this case is 4 seconds. Secondly, the derived FBK_OPEN signal is deenergised (indicating the valve is no longer open).

(21) The simulated closing time (TIME_DER_CLOSED) is set to 80%, this means that the derived FBK_CLOSED signal will activate when the actual_Operating_Timer value has counted down for 80% of the TIME_CLOSING_MAX value (i.e. when the timer has counted down to 0.80s or less).

(22) The exact formula used is:

$$\text{Derived FBK\_CLOSED} = 1 \text{ when}$$
$$\text{actual\_Operating\_Timer} \leq \text{TIME\_OPENING\_MAX} - \left(\text{TIME\_OPENING\_MAX} \times \frac{\text{TIME\_DER\_OPEN}}{100}\right)$$

(23) This can be seen at point ④, where the derived FBK_CLOSED signal activates. The valve now remains in the closed state (FBK_OPEN = 0. FBK_CLOSED = 1) until there is a request a subsequent request to reopen the valve.

(24) If only one position feedback is derived (the other being an actual signal), then the derived signal will be generated in exactly the same way as above, the non-derived signal will simply use the true status of the signal passed as a parameter to the module.

(25) The simulated time signals are passed to the module within the STATIC_DATA parameter UDT (*UT11001_St_DevValveIsol*) as follows:

DATA STRUCTURE SIGNAL	*UT11001_St_DevValveIsol* FUNCTION	TYPE	UNITS
TIME_DER_OPEN	Specifies the percentage of the TIME_OPENING_MAX value that must expire before the derived FBK_OPEN signal activates. A value of 0, means the FBK_OPEN signal will activate the instant the valve is requested to open. Not used if CONFIG_FBK_OPEN_EN = 1.	Real	Percentage
TIME_DER_CLOSED	Specifies the percentage of the TIME_CLOSING_MAX value that must expire before the derived FBK_CLOSED signal activates. A value of 0, means the FBK_CLOSED signal will activate the instant the valve is requested to close. Not used if CONFIG_FBK_CLOSED_EN = 1.	Real	Percentage

Table 2.14 Derived open and closed signal activation times

(26) Both configuration signals are restricted to the range 0.00-100.00, the default value is 90% for both open and closed.

(27) If the time at which the derived position feedback is activated has an elapsed time of close to or less than a single timing pulse (200ms), a minimum value is imposed on the above calculations where the open or closed position feedback activates if:

$$\text{actual\_Operating\_Timer} \leq \text{K\_TIME\_OP\_MIN}$$

(28) Where K_TIME_OP_MIN is a constant within the FC11001 module. It has a predefined value of 0.21 seconds (210ms).

2.2.4 Recording the last operation time

(1) The module records the last complete operation time value, it records the actual time taken for the valve to reach the correct designated position.

(2) For example if the valve received a request to open, and it required 4.8 seconds for the position feedback to report the correct (open) state, the module would record a value of 4.8 seconds in the actual_Last_Op_time value (passed as part of the DYNAMIC_DATA parameter).

(3) Similarly, if the valve were requested to close, and required 6.2 seconds for the position feedback to report the correct (closed) state, the module would record a value of 6.2 seconds in the actual_Last_Op_time value.

(4) The actual_Last_Op_time value forms part of the DYNAMIC_DATA parameter UDT as follows:

DATA STRUCTURE	*UT31001_Dy_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
actual_Last_Op_time	Stores the actual time taken for the valve to achieve the correct open or closed state. Returns a value of `-1.0` if the valve did not achieve the correct position in the specified operation time	`Real`	Time to achieve position in seconds

Table 2.15 Symbolic status information

(5) If the valve does not achieve the requested position (within either the TIME_OPEN_MAX or TIME_CLOSED_MAX limits), a value of `-1.0` is returned as the actual_Last_Op_time value, indicating a fault occurred.

2.2.5 Detecting position faults

(1) There are two types of position faults: *dynamic* and *static*.

(2) Dynamic faults occur when the valve is changing state (from open to closed or closed to open) and fails to achieve the requested position within the relevant operation time.

(3) The dynamic faults are:

- Failed to open
- Failed to close

(4) Static faults occur when a valve is stable in the requested state (either open or closed) and is not in the process of moving between states, and some fault occurs such that the position feedback no longer matches the requested state (e.g. the valve was correctly in the open state, and then at some point, the open feedback signal was lost without any change being made to the state of the valve by the module. Such a fault could be a power failure at the valve).

(5) The static faults are:

- Failed while open
- Failed while closed

(6) If the valve is not equipped with any limit switches (both CONFIG_FBK_OPEN_EN and CONFIG_FBK_CLOSED_EN are set to 0), then there are no circumstances under which any position fault can be generated (the valve is always assumed to operate correctly and be in the required position).

(7) The following table indicates the conditions under which the various alarms can be generated for the various position feedback combinations:

LIMIT SWITCH ARRANGEMENT	FAILED TO OPEN Opening operation timer expired and:	FAILED TO CLOSE Closing operation timer expired and:
No feedback CONFIG_FBK_OPEN_EN = 0 CONFIG_FBK_CLOSED_EN = 0	Not applicable	Not applicable
Open feedback only CONFIG_FBK_OPEN_EN = 1 CONFIG_FBK_CLOSED_EN = 0	Open feedback is **INACTIVE**	Open feedback is **ACTIVE**
Closed feedback only CONFIG_FBK_OPEN_EN = 0 CONFIG_FBK_CLOSED_EN = 1	Closed feedback is **ACTIVE**	Closed feedback is **INACTIVE**
Closed feedback only CONFIG_FBK_OPEN_EN = 0 CONFIG_FBK_CLOSED_EN = 1	Closed feedback is **ACTIVE** OR Open feedback is **INACTIVE**	Closed feedback is **ACTIVE** OR Open feedback is **INACTIVE**
LIMIT SWITCH ARRANGEMENT	FAILED WHILE OPEN Opening operation timer not running and:	FAILED WHILE CLOSE Closing operation timer not running and:
No feedback CONFIG_FBK_OPEN_EN = 0 CONFIG_FBK_CLOSED_EN = 0	Not applicable	Not applicable
Open feedback only CONFIG_FBK_OPEN_EN = 1 CONFIG_FBK_CLOSED_EN = 0	Open feedback is **INACTIVE**	Open feedback is **ACTIVE**
Closed feedback only CONFIG_FBK_OPEN_EN = 0 CONFIG_FBK_CLOSED_EN = 1	Closed feedback is **ACTIVE**	Closed feedback is **INACTIVE**
Closed feedback only CONFIG_FBK_OPEN_EN = 0 CONFIG_FBK_CLOSED_EN = 1	Closed feedback is **ACTIVE** OR Open feedback is **INACTIVE**	Closed feedback is **ACTIVE** OR Open feedback is **INACTIVE**

Table 2.16 Position faults and position feedback configurations conditions

(8) The OR in the alarm conditions is a logical or, it covers the conditions in which both or neither limit switches are active; consider the *failed to open alarm*, if both limits were active, the alarm would be generated because the closed limit switch is active (irrespective of the state of the open limit). Similarly, if neither switch were active, the alarm would be generated because the open limit is inactive (irrespective of the state of the closed limit).

(9) All alarms are reported as message signals, these form part of the DYNAMIC_DATA parameter UDT as follows:

DATA STRUCTURE	*UT31001_Dy_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
msg_FailToOpen	Valve failed to open in the specified opening operation time	Bool	1 = alarm active 0 = alarm inactive
msg_FailToClose	Valve failed to close in the specified closing operation time	Bool	1 = alarm active 0 = alarm inactive
msg_FailWhileOpen	Valve suffered position failure while in a static open state	Bool	1 = alarm active 0 = alarm inactive
msg_FailWhileClosed	Valve suffered position failure while in a static closed state	Bool	1 = alarm active 0 = alarm inactive
msg_Flt_External	Valve is reporting a (hardware) fault condition, see § 2.3.2	Bool	1 = alarm active 0 = alarm inactive

Table 2.17 Module alarm signals

2.3 Fault handling

(1) There are five fault conditions, four are derived from position faults and one from an external fault signal:

FAULT	DESCRIPTION	LATCHING
Failed to open	Valve failed to open in the specified opening operation time	Yes
Failed to close	Valve failed to close in the specified closing operation time	Yes
Failed while open	Valve suffered position failure while in a static open state	Yes
Failed while closed	Valve suffered position failure while in a static closed state	Yes
External fault	The EXT_FAULT signal is active	No

Table 2.18 Module faults

(2) If any of the above faults are active, the valve will be deenergised (the OPERATE parameter will be set to *0*) and the module will ignore all commands either automatic or manual and will ignore all interlock conditions.

2.3.1 Position faults

(1) All position faults (derived from position feedback signals) are latching fault conditions that require the operator to issue a reset command, this is usually via a faceplate available at the supervisory system (SCADA/HMI), but could also be via a hardwired (pushbutton) signal.

(2) The reset signal is specified within the DYNAMIC_DATA parameter UDT as follows:

DATA STRUCTURE	*UT31001_Dy_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
mode_Reset	A reset pulse (active for one Controller cycle)	Bool	1 = reset latching faults, 0 = no action

Table 2.19 Module reset signal

(3) If a position fault has occurred, and the fault is then reset, the appropriate opening or closing operation timer will be retriggered, allowing the valve time to achieve the required position.

2.3.2 External fault

(1) The external fault is an optional hardwired signal from the valve, it indicates that the device itself is registering that there is some problem. Such external fault signals are more commonly associated with motorised valves (equipped with thermal overload detectors), but may also be associated with pneumatic or hydraulic operated valves to indicate that the is a pneumatic/hydraulic system failure.

(2) Such fault signals are generally indicated via a hardwired digital input to the Controller. The external fault signal (if used) is passed to the module as the parameter: EXT_FAULT:

PARAMETER	FUNCTION	TYPE	DETAILS
EXT_FAULT	Hardwired signal indicating the valve is in a fault condition	Bool	1 = fault, 0 = healthy

Table 2.20 External fault parameter

(3) Any signal connected to the EXT_FAULT parameter must be active high (i.e. the signal must be set to 1 to indicate the fault condition is present).

(4) Where a failsafe response signal is used (active when set to 0), it must be inverted at the EXT_FAULT parameter:

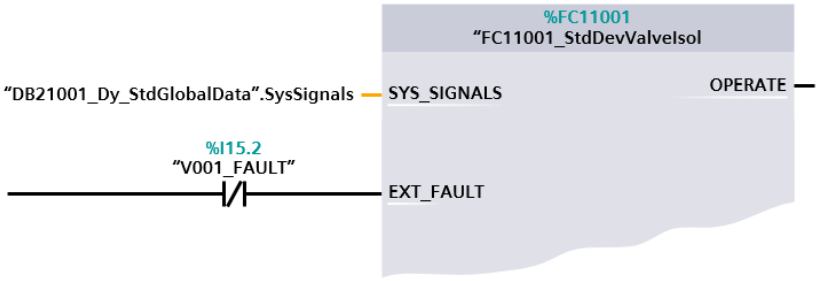


Figure 2.3 Inverting a failsafe external fault signal

(5) If the EXT_FAULT signal is not required, the input should be set to _False (similar to the safety and interlock signals that are not used, see § 2.4).

(6) The EXT_FAULT signal is non-latching; it does not require a mode_Reset signal to clear the fault. The external fault will automatically reset whenever the EXT_FAULT signal is set to 0.

(7)

2.3 Fault handling

(1) Safety and interlock signals are optional signals, passed as individual parameters to the module (in the same manner as IO signals), all such signals are optional, a valve can have all, none or any combination of the signals.

(2) Safety and interlock signals act to override the normal operation of the valve (whether in automatic or manual mode) and force the valve to a particular state.

(3) There is one safety signal and three interlock signals, each having its own formal module parameter:

PARAMETER	SAFETY/ INTERLOCK	FUNCTION	TYPE	DETAILS
E_STOP	Safety	Safety or emergency stop signals are hard-wired signals that remove power from a device forcing it to a safe state.	Bool	1 = E-stop active 0 = healthy
TRIP	Interlock	A latching interlock, it activates whenever a set of events are true and forces the devices to a given state. It will not deactivate until the triggering conditions are removed and a reset signal has been given.	Bool	1 = trip active 0 = healthy
INTERLOCK	Interlock	A simple interlock that is active whenever a set of conditions is true, it will force the devices to a given state.	Bool	1 = interlock active 0 = healthy
PERMISSIVE	Interlock	Takes no action if a device is in a non-permitted state, but once the device is in a that state will prevent a transition back to the non-permitted state.	Bool	1 = permissive active 0 = healthy

Table 2.21 Safety and interlock parameters

(4) Each safety and interlock signal has its own configuration signal that must be used to formally disable the parameter if the signal is not required, these are all within the STATIC_DATA parameter UDT (*UT11001_St_DevValveIsol*) as follows:

DATA STRUCTURE	*UT11001_St_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
CONFIG_ESTOP_DIS	Disables the safety signal. If set to 1, any signal on the E_STOP parameter will be ignored	Bool	1 = E_STOP disabled 0 = E_STOP enabled
CONFIG_TRIP_DIS	Disables the trip interlock signal. If set to 1, any signal on the TRIP parameter will be ignored	Bool	1 = TRIP disabled 0 = TRIP enabled
CONFIG_ILOCK_DIS	Disables the interlock signal. If set to 1, any signal on the INTERLOCK parameter will be ignored	Bool	1 = INTERLOCK disabled 0 = INTERLOCK enabled
CONFIG_PERM_DIS	Disables the permissive interlock signal. If set to 1, any signal on the PERMISSIVE parameter will be ignored	Bool	1 = PERMISSIVE disabled 0 = PERMISSIVE enabled

Table 2.22 Safety and interlock disable signals

(5) The disable signals are used to formally configure the valve, they allow any supervisory system to know the correct configuration of the valve and display information accordingly.

2.4.1 Safety signal operation

(1) Safety or emergency stop signals are hardwired signals that remove power from a device forcing it to a safe state. Such signals act independently of the software (the pressing of an emergency stop button will physically remove power from the affected devices).

(2) The emergency stop signal is passed to the module as the parameter E_STOP; it informs the module that the safety signal has been activated and allows the module to adjust the state of the valve to follow the emergency stop conditions.

(3) When the emergency stop signal is removed, the valve will resume the requested state (with the appropriate operation timer).

(4) Under emergency stop conditions the valve is always deenergised (both physically by the hardware and within the software via the actions of this module). I.e. if an emergency stop condition exists, the OPERATE output is set to zero (OPERATE = 0). For a normally closed valve (CONFIG_NO_NC = 0) this means the valve will close, for a normally open valve (CONFIG_NO_NC = 1), the valve will open.

(5) The emergency stop signal is absolute and will override any other demand made of the valve. There is no mechanism for bypassing this form of safety signal.

Typical usage of the E_STOP parameter

(6) Any safety signal connected to the E_STOP parameter must be active high (i.e. the signal must be set to 1 to activate the safety signal response). It is often the case, where hardwired emergency stop signals are used, that the emergency stop condition is activated by the absence of a signal (a *failsafe* response) rather than the application of a signal. The module can accommodate this sense of signal by inverting the E_STOP input:

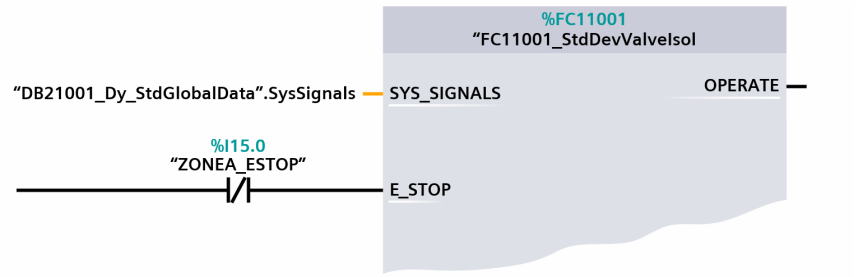


Figure 2.4 Inverting a failsafe emergency stop signal

(7) Non failsafe safety signals are directly applied to the parameter:

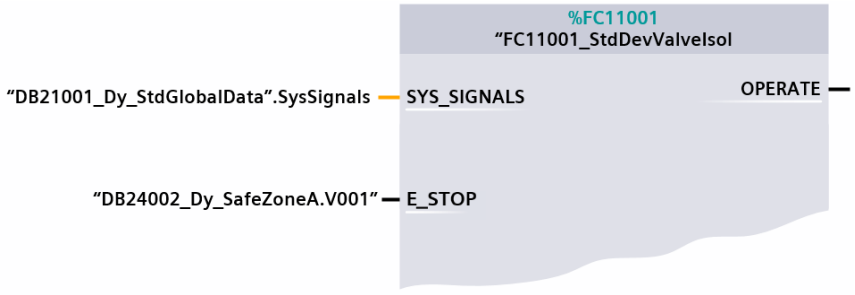


Figure 2.5 Non failsafe emergency stop signal

(8) Finally, if no safety signal is used, the parameter is set to the `_False` tag (note, also that the configuration signal CONFIG_ESTOP_DIS should also be set to 1).

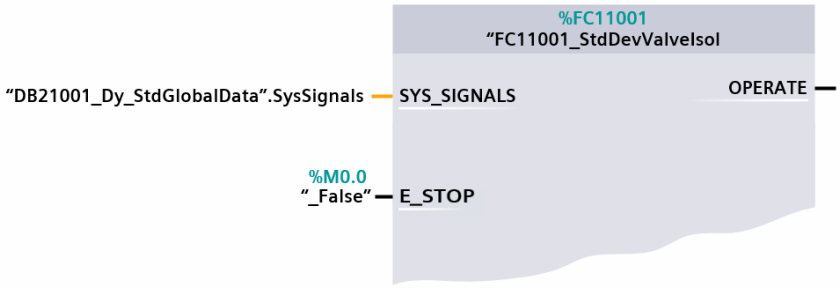


Figure 2.6 No emergency stop signal

2.4.2 Interlock signals

(1) Interlock signals are a more flexible mechanism for overriding the requested position of the valve. Interlock signals can be both hardware driven (similar to a safety signal) or can be purely software derived (unlike a safety signal).

(2) Software derived interlock signals can accommodate more complex and more flexible arrangements, it would for example, be possible to disable a software derived interlock if the process were undergoing some particular function, such as cleaning &c.

(3) There are three different types of interlock arrangements:

① **Interlock**: a simple interlock that is active whenever a set of conditions is true; if active, it will force the devices to a specified state.

② **Permissive**: designates a specified "permitted" state for the device. A permissive interlock takes no action if the device is in the non-permitted state, but once the device changes to the permitted state, it will be prevented from changing back to the non-permitted state (e.g. if the permitted state was closed and the valve was currently open when the permissive signal activated, it would not force the valve to close, but once it was closed, it would prevent it from re-opening).

③ **Trip**: a latching interlock, it activates whenever a set of events are true (like an interlock), but will not deactivate until the triggering conditions are removed and a reset signal has been given (effectively a latching interlock); if active, it will force the device to a specified state. The resetting of the trip is handled by the trip logic itself (and not within this module).

(4) For any type of interlock, the specified interlock state (or permitted state) can be either open or closed irrespective of the type of valve (normally open or normally closed); unlike the safety signal, it does not have to be the deenergised state of the valve.

(5) That said, the most common interlock state is the deenergised state of the valve; i.e. the failsafe position of the valve.

Typical usage of interlock signals

Typical usage of interlock signals

(6) Each type of interlock is passed as its own parameter: TRIP, INTERLOCK or PERMISSIVE. Like the safety signal (E_STOP), each parameter is active high (i.e. the signal must be set to 1 to activate the associated interlock signal response).

(7) Again, it is often the case, where hardwired interlock signals are used, that the condition is activated by the absence of a signal (i.e. a failsafe response) rather than the application of a signal. The module can accommodate this sense of signal by inverting the relevant input:

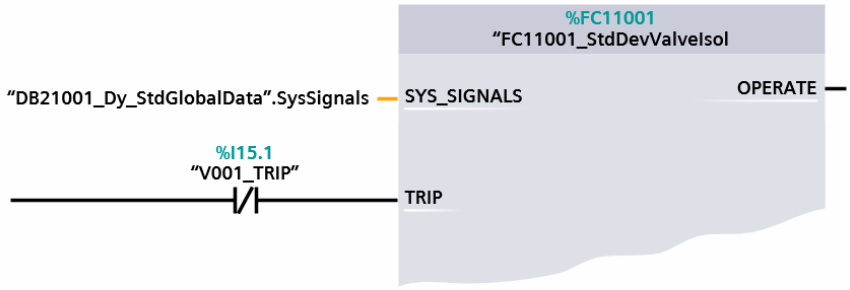


Figure 2.7 Inverting a failsafe interlock signal

(8) Non failsafe signals are directly applied to the parameter:

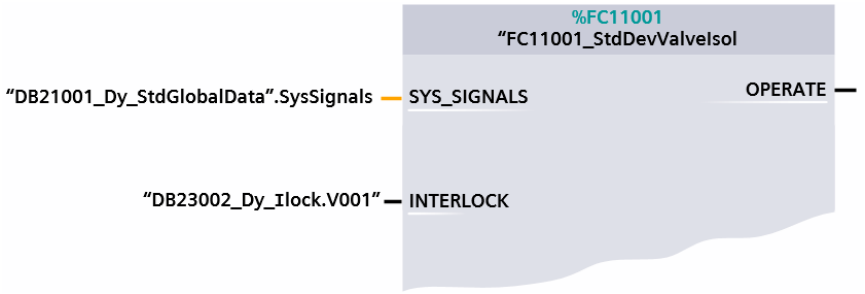


Figure 2.8 Non failsafe interlock signal

(9) Finally, if no interlock signal is used, the parameter is set to the `_False` tag (note also: the appropriate configuration signal should also be set to `1`: CONFIG_TRIP_DIS, CONFIG_ILOCK_DIS or CONFIG_PERM_DIS).

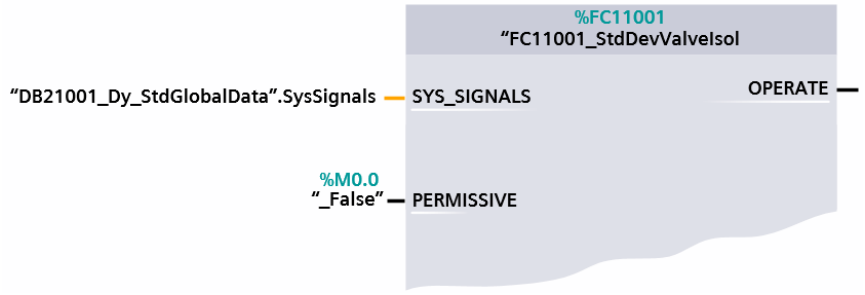


Figure 2.9 No interlock signal

Configuring the interlock signals

Configuring the interlock signals

(10) The state adopted by the valve when a particular interlock is active (TRIP or INTERLOCK) or the permitted state of the valve (PERMISSIVE) are set individually for each type of interlock signal. These configuration signals are stored within the STATIC_DATA parameter UDT as follows:

DATA STRUCTURE	*UT11001_St_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
CONFIG_TRIP_OP_CL	The required state of a valve if the TRIP signal is active	Bool	1 = valve will open, 0 = valve will close
CONFIG_ILOCK_OP_CL	The required state of a valve if the INTERLOCK signal is active	Bool	1 = valve will open, 0 = valve will close
CONFIG_PERM_OP_CL	The permitted state of a valve if the PERMISSIVE signal is active	Bool	1 = valve may open, 0 = valve may close

Table 2.23 Safety and interlock disable signals

(11) For example, if CONFIG_ILOCK_OP_CL is set to 1, the valve will open whenever the INTERLOCK signal is active (set to 1), this will override any automatic or manual requirement for the valve (but will not override a safety signal, see below). If CONFIG_ILOCK_OP_CL were set to 0, the valve would close whenever the INTERLOCK signal was active.

(12) The TRIP signal behaves in exactly the same way as the INTERLOCK signal, but uses CONFIG_TRIP_OP_CL to set the tripped state of the valve.

(13) Although the TRIP signal requires a reset signal to disengage the interlock conditions, this is handled by the trip logic itself (usually handled by the trip application modules: FC23200) and not via this module; simply deactivating the TRIP signal removes the trip state from the valve and allows it to operate normally.

(14) The PERMISSIVE signal operates in a different way, if CONFIG_PERM_OP_CL were set to 1 (valve is permitted to open), and the valve was currently closed under automatic control when the PERMISSIVE signal became active, nothing would happen to the valve (it would remain closed), if the valve then subsequently opened under automatic or manual control, the PERMISSIVE signal would allow the valve to do so (the valve permitted state is open). However, any attempt to re-close the valve would not be allowed by the PERMISSIVE signal.

(15) Conversely, if CONFIG_PERM_OP_CL were set to 0 (valve is permitted to close), and the valve was currently open under automatic control when the PERMISSIVE signal became active, nothing would happen to the valve (it would remain open), if the valve then subsequently closed under automatic or manual control, the PERMISSIVE signal would allow the valve to do so (the valve permitted state is close). However, any attempt to re-open the valve would not be allowed by the PERMISSIVE signal.

Note: *Where multiple interlock signals are used, it is possible for the different interlocks to apply different states to the valve, for example an active TRIP could close the valve and an active INTERLOCK open it. Such conflicts are managed by the different priorities applied to the interlock signals (see § 2.5).*

Bypassing interlocks

Bypassing interlocks

(16) Unlike safety signals, all interlock signals can be overridden (or bypassed). The bypass signal is a single signal that, if active, overrides and deactivates any TRIP, INTERLOCK or PERMISSIVE signal that may have been activated.

(17) The bypass signal is specified within the DYNAMIC_DATA parameter UDT as follows:

DATA STRUCTURE	*UT31001_Dy_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
Mode_BypassOn	Activates bypass mode, if active, any TRIP, INTERLOCK or PERMISSIVE signal will be ignored	Bool	1 = bypass mode on, 0 = bypass mode off

Table 2.24 Bypass mode selection

(18) Bypass mode is a manual operation, generally activated by the operator via a faceplate available at the supervisory system (SCADA/HMI), or via some hardwired signal, such as a key-switch.

(19) It is possible to configure the module such that bypass mode is permanently disabled, this uses the following STATIC_DATA parameter:

DATA STRUCTURE	*UT11001_St_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
CONFIG_BYPASS_DIS	Prevents bypass mode being activated under all circumstances	Bool	1 = no bypass mode, 0 = bypass permitted

Table 2.25 Bypass mode configuration

(20) If bypass mode is disabled (configuration signal CONFIG_BYPASS_DIS is set to 1), then the bypass mode cannot be turned on under any circumstances, the option will be greyed out on the supervisory system faceplate and the module will constantly reset the mode_BypassOn signal.

Note: *The safety (E_STOP) signals cannot be bypassed.*

Similarly, if an interlock signal has a hardwired component (that will override any software actions), it also must not be bypassed.

2.5 Priority of operation

(1) The valve is normally controlled by either the automatic system or by manual (operator) control. However, this operation can be overridden by the various safety and interlock operations. Each of these operations is assigned a priority and if more than one is active at the same time, the one with the highest priority will take precedence. The priorities for the various operations are as follows:

SIGNAL TYPE	PRIORITY	CAN BE BYPASSED
Emergency Stop	40	High
Trip	30	↑
Interlock	20	↑
Permissive	10	↑
Automatic or manual control	0	Low

Table 2.26 Operational signal priority — High priority signal take precedence over lower priority signals

(2) Thus, if both the TRIP and INTERLOCK signals were active at the same time, the module will respond only to the TRIP signal (with the higher priority).

2.6 Operating modes

(1) This module FC11001_StdDevValveIsol supports various operating modes:

- Automatic/manual mode
- Interlock bypass mode
- Simulation mode
- Remote/local mode
- Faceplate disabled mode

(2) Examining these in turn:

2.6.1 Automatic and manual modes

(1) Under normal operation, the valve is in automatic mode and is controlled directly by the Controller software (usually, sequential or continuous logic); however, it is possible for the operator to take direct control of the valve by switching it to manual mode.

(2) If manual mode is active, the valve will open and close directly at the command of the operator and will ignore any automatic mode open and close requests.

(3) If the valve is operating under automatic mode, the manual open/close request follows that of the automatic mode; this ensures that the valve remains in the same state when initially switched from automatic mode to manual mode, referred to as bumpless transfer. This approach minimises any disturbance to the process should the valve be inadvertently switched to manual mode.

(4) The following data points are associated with automatic/manual mode:

DATA STRUCTURE SIGNAL	*UT11001_St_DevValveIsol* FUNCTION	TYPE	DETAILS
CONFIG_MAN_DIS	Prevents manual mode being activated under all circumstances, device is permanently in auto, mode_AutMan is set permanently to 1	Bool	1 = no manual mode, 0 = manual permitted

DATA STRUCTURE SIGNAL	*UT31001_Dy_DevValveIsol* FUNCTION	TYPE	DETAILS
mode_AutMan	Identifies the operating mode of the valve. If the manual mode is disabled, the mode_AutMan is set permanently to 1 (automatic mode)	Bool	1 = automatic mode, 0 = manual mode
ctrl_Aut_OpenClose	Causes the valve to open and close if automatic mode is active (mode_AutMan = 1) If automatic mode is not active, the signal is ignored	Bool	*Automatic mode* 1 = open the valve, 0 = close the valve
ctrl_Man_OpenClose	Causes the valve to open and close if manual mode is active (mode_AutMan = 0) If manual mode is not active, the signal is set to match ctrl_Aut_OpenClose.	Bool	*Manual mode* 1 = open the valve, 0 = close the valve

Table 2.27 Auto/manual mode signals

(5) Manual mode can only be activated by the operator via a faceplate available at the supervisory system (SCADA/HMI).

(6) If manual mode is disabled (CONFIG_MAN_DIS = 1), then manual mode cannot be turned on under any circumstances, the option will be greyed out on the supervisory system faceplate and the valve will be permanently set to automatic mode (mode_AutMan = 1).

2.6.2 Interlock bypass mode

(1) All interlock signals (TRIP, INTERLOCK or PERMISSIVE) can be overridden (bypassed). The bypass signal is a single signal that, if active, overrides and deactivates any TRIP, INTERLOCK or PERMISSIVE signal that may have been activated.

(2) It is possible to permanently disable the bypass mode for any given valve (preventing the operator from activating it).

(3) The following data points are associated with the bypass mode:

DATA STRUCTURE SIGNAL	*UT11001_St_DevValveIsol* FUNCTION	TYPE	DETAILS
CONFIG_BYPASS_DIS	Prevents bypass mode being activated under all circumstances	Bool	1 = no bypass mode, 0 = bypass permitted

DATA STRUCTURE SIGNAL	*UT31001_Dy_DevValveIsol* FUNCTION	TYPE	DETAILS
mode_BypassOn	Activates bypass mode, if active, any TRIP, INTERLOCK or PERMISSIVE signal will be ignored	Bool	1 = bypass mode on, 0 = bypass mode off
status_BypassOn	Status indication, shows if bypass mode is active	Bool	1 = Bypass mode active, 0 = Bypass mode inactive

Table 2.28 Bypass mode signals

(4) Bypass mode can only be activated by the operator usually via a faceplate available at the supervisory system (SCADA/HMI), alternatively it may be via some form of hard-wired signal such as a key-switch &c.

(5) If bypass mode is disabled (configuration signal CONFIG_BYPASS_DIS is set to 1), then the bypass mode cannot be turned on under any circumstances, the option will be greyed out on the supervisory system faceplate and the module will constantly reset the mode_BypassOn signal.

2.6.3 Simulation mode

(1) The module can be switched to a simulation mode, in which the position feedback for the valve is replaced by simulated signals specified by the operator.

(2) Simulation mode is generally used during testing, but can also be applied in a process environment if a fault condition is detected; this scenario allows the plant to continue operating (albeit under some degree of manual control) until the valve is repaired or replaced.

(3) The following data points are associated with the simulation mode

DATA STRUCTURE	*UT11001_St_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
CONFIG_SIM_DIS	Prevents simulation mode being activated under all circumstances	Bool	1 = no simulation, 0 = simulation permitted

DATA STRUCTURE	*UT31001_Dy_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
mode_SimOn	Activates simulation mode, if active, the position feedback will be specified by the mode_SimValue	Bool	1 = simulation mode on, 0 = simulation mode off
mode_SimValue	Specifies the type of simulated position feedback in use	int	0 = no position feedback 1 = permanently open 2 = permanently closed 3 = follow demand
status_SimOn	Status indication, shows if simulation mode is active	Bool	1 = sim mode active, 0 = sim mode inactive

Table 2.29 Simulation mode signals

(4) Simulation mode can only be activated by the operator via a faceplate available at the supervisory system (SCADA/HMI).

(5) If simulation mode is activated, the position feedback for the valve is forced to one of four states specified by the mode_SimValue variable (this is an enumerated variable):

		SIMULATED VALUES	
mode_SimValue	DESCRIPTION	FBK_OPEN	FBK_CLOSED
0	Permanently disables all position feedback (irrespective of any valve position)	0	0
1	Permanently sets the valve position feedback to open (irrespective of any valve position)	1	0
2	Permanently sets the valve position feedback to closed (irrespective of any valve position)	0	1
3	Position feedback follows the requested state of the valve	0 if valve is closed 1 if valve is open	1 if valve is closed 0 if valve is open

Table 2.30 Enumerated mode_SimValue meaning

(6) The mode_SimValue is restricted to the range 0-3.

(7) If simulation mode is disabled (configuration signal CONFIG_SIM_DIS is set to 1), then the simulation mode cannot be turned on under any circumstances, the option will be greyed out on the supervisory system faceplate and the module will constantly reset the mode_SimOn signal.

(8) If simulation mode is permitted (CONFIG_SIM_DIS = 0), then setting mode_SimOn to 1 will activate simulation mode and the any position feedback signals (connected to FBK_OPEN and FBK_CLOSED) will be overridden according to the value stored in mode_SimValue (see Table 2.30).

(9) The operator can set and adjust the simulation value (mode_SimValue) from the instrument faceplate on the supervisory system. The simulation value can be adjusted at any time by the operator (even if simulation mode has not been activated), but will not be applied until simulation mode is turned on (mode_SimOn = 1).

(10) If simulation mode is completely disabled (CONFIG_SIM_DIS is set to 1) then the operator will not be able to adjust the simulated value (mode_SimValue will be greyed out on the faceplate).

2.6.4 Remote/local mode

(1) Remote and local operating modes refer to the supervisory system that has control of the device (i.e. which system can write to the device and change the operating mode of the device).

(2) This type of mode is usual present where a plant has a remote central control location (a control room) that normally controls the device (remote control), but also has a field panel with a local HMI that an operator in the field can select to take over control of the device (local control) for maintenance purposes.

(3) The modes are as follows (and are mutually exclusive, only one will be active):

Remote	Only the remote system in the control room can control the device
Local	A local system has taken control of the device and the remote system can no longer issue commands to it
Remote/local disabled (ALL mode)	The system does not use remote/local modes and any supervisory system can issue commands to the device

Table 2.31 Remote/local modes

(4) Remote/local control is generally handled by the supervisory systems themselves; however, the supervisory systems need a storage area per device to record which mode that device is in, local control is often activated by a panel key-switch, that changes the state of several devices from remote to local (usually all the devices controlled by the local panel).

(5) The module simply identifies the remote/local mode from the mode signals and configuration signals provided to the module.

(6) For systems that do not use remote/local modes (any supervisory system can control the device, or control is determined by operator privileges), the remote/local modes can be disabled (CONFIG_RL_EN is set to zero) and the device is effectively in ALL mode (any system can control the device).

(7) If remote/local operation is in use, the supervisory system that **does not** have control will still display all the information from the device, but will not be able to control the device (it could not for example, activate simulation mode) and all control function (faceplate functions) will be greyed out.

(8) The following signals are associated with remote/local/all modes:

DATA STRUCTURE SIGNAL	*UT11001_St_DevValveIsol* FUNCTION	TYPE	DETAILS
CONFIG_RL_EN	Allows Remote and Local modes to be selected. If set to zero, the device does not have remote and local operation and both mode_RemoteOn and mode_LocalOn are set to zero (the device is in ALL mode).	Bool	1 = Remote/local can be selected 0 = All mode is active

DATA STRUCTURE SIGNAL	*UT31001_Dy_DevValveIsol* FUNCTION	TYPE	DETAILS
mode_RemoteOn	Activates remote mode and resets local mode (CONFIG_RL_EN must be set to 1)	Bool	1 = remote mode on, 0 = remote mode off
mode_LocalOn	Activates local mode and resets remote mode (CONFIG_RL_EN must be set to 1)	Bool	1 = local mode on, 0 = local mode off
status_RemoteOn	Status indication, active if device is in remote mode (mode_RemoteOn = 1)	Bool	1 = remote mode on, 0 = remote mode off
status_LocalOn	Status indication, active if device is in local mode (mode_LocalOn = 1)	Bool	1 = local mode on, 0 = local mode off
status_RLOff	Status indication, ALL mode is active (mode_RemoteOn = 0 and mode_LocalOn = 0)	Bool	1 = ALL mode on, 0 = ALL mode off

Table 2.32 Remote/local signals

(9) By default, remote/local is disabled — ALL mode is selected.

2.6.5 Faceplate disable mode

(1) It is possible to disable the supervisory system faceplate from within the module, the configuration signal:

DATA STRUCTURE	*UT11001_St_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
CONFIG_FP_DIS	Disables the supervisory system faceplate. If active, clicking the block icon or device symbol will not activate the device faceplate pop-up. I.e. the operator will never be able to issue instructions to the device	Bool	1 = Faceplate disabled 0 = normal

Table 2.33 Remote/local signals

(2) If the faceplate is disabled (CONFIG_FP_DIS = 1), the supervisory system must not allow the device faceplate to be opened (normally achieved by clicking the block icon or device symbol), this in turn prevents the operator from affecting the device in anyway, it would not, for example, be possible to put the device into simulation mode.

(3) Similarly, if the faceplate is disabled, the operator will not have access to the mode_Reset signal, meaning that alternative reset mechanisms must be in place (such as a hardwired pushbutton for example).

3 Supervisory system interface

(1) The module is designed to interface with a supervisory system (a SCADA or HMI), it does this through the use of a symbol, a block icon and faceplates.

(2) The symbol shows the true state of the device (open, closed, fault &c.) and provides some configuration information (normally open, normally closed &c.).

(3) The block icon provides status information for the valve, it reports tag information, operating modes, interlock conditions and output states.

(4) The faceplates are pop-up windows that provide additional information about the device and allow the operator to select and activate the various options and modes available to it.

3.1 Symbols and associated signals

(1) Supervisory systems use the status signals of Table 3.1(below) to display an animated representation of the valve, the following are various typical symbols that satisfy the display requirements of the module:

Standard NC valve	Standard NO valve	Motorised NC valve	Motorised NC valve	SIGNAL	DESCRIPTION
				status_Closed	Closed
				status_Opening	Opening
				status_Opened	Opened
				status_Closing	Closing
				status_Fault	Fault (valve body shows state)
				N/A	Loss of communications

Table 3.1 Typical valve symbols NC — Normally closed NO — Normally open

(2) The symbols show the various states of the valve, and also identify the type of valve (normally open or normally closed). The type of valve is indicated by the presence of an upwards pointing triangle in the actuator area of the valve:

Standard valve	Motorised valve	SIGNAL	DESCRIPTION
		CONFIG_NO_NC = 1	Normally closed valve
		CONFIG_ NO_NC = 1	Normally open valve

Table 3.2 Typical valve symbols NC — Normally closed NO — Normally open

(3) The standard arrangement is for symbols to show discrete opening and closing symbols (Table 3.1); there are some instances (usually on less sophisticated HMIs) where it is necessary to show only that the valve is in the process of changing state.

(4) This is achieved through the use of the status_Operating signal (this is active whenever either status_Opening or status_Closing is active), status_Operating indicates that the valve is either opening or closing, but does not distinguish which.

(5) Typical symbols for the operating state are:

Standard NC valve	Standard NO valve	Motorised NC valve	Motorised NC valve	SIGNAL	DESCRIPTION
				status_Operating	Opening or closing

Table 3.3 Valve operating symbols NC — Normally closed NO — Normally open

3.1.1 Symbol status signals

(1) The module constantly reports the status of the valve (i.e. open, closed, opening &c.) via various status signals. These signals can be used within the Controller software to determine specific device conditions (e.g. the valve is confirmed open), the signals may also be used by supervisory systems to display (in an animated fashion) the state of the valve.

(2) The module reports the following valve statuses:

DATA STRUCTURE	*UT31001_Dy_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
status_Closed	Indicates the valve is in the closed state	Bool	1 = valve is closed, 0 = valve is not closed
status_Open	Indicates the valve is in the open state	Bool	1 = valve is open, 0 = valve is not open
status_Closing	Indicates the valve is in the process of closing (i.e. closing operation timer is running)	Bool	1 = valve is closing, 0 = valve is not closing
status_Opening	Indicates the valve is in the process of opening (i.e. opening operation timer is running)	Bool	1 = valve is opening, 0 = valve is not opening
status_Operating	Indicates the valve is opening or closing (i.e. either status_Closing or status_Opening is active – set to 1)	Bool	1 = valve is changing state, 0 = valve is static
status_Fault	The valve is in a fault condition (at least one of the five alarm conditions is active)	Bool	1 = valve is in fault, 0 = valve is healthy

Table 3.4 Symbolic status information

3.2 Block icons and associated signals

(1) This module has a partner block icon that represents the valve and its various modes and status in a graphical format (block icon) that is displayed on a supervisory system (SCADA or HMI). The basic block icon has the following appearance:

Figure 3.1 FC11001 block icon (typical)

(2) The five grey boxes at the bottom are used to display various status information for the instrument, Figure 3.2 show all the permutations and the signals that drive the various states:

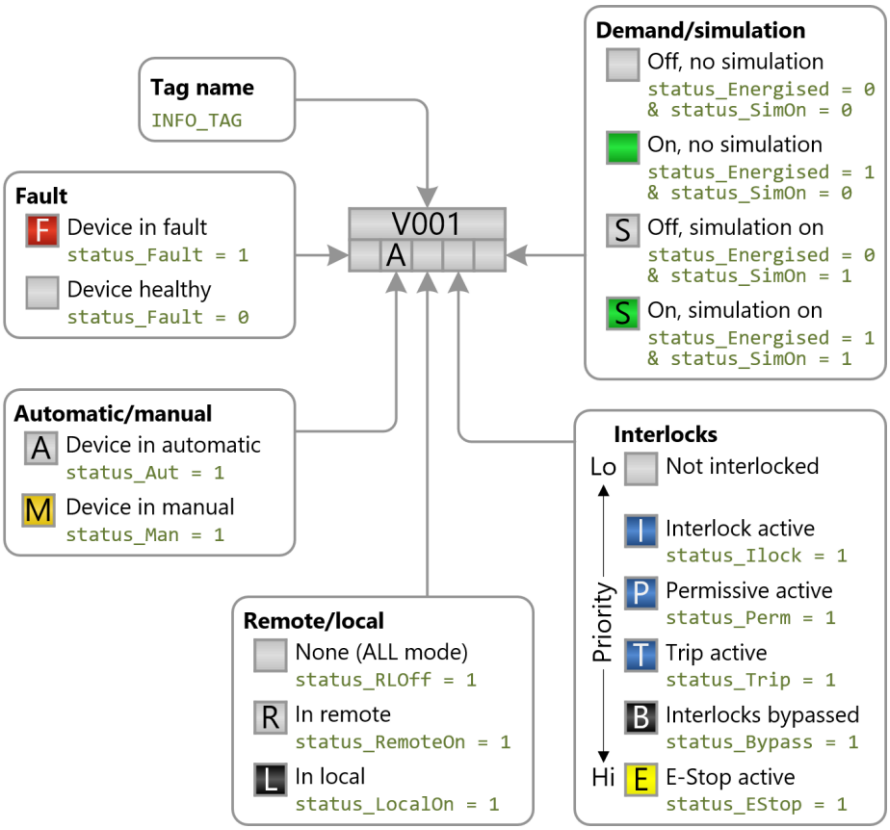


Figure 3.2 FC11001 block icon breakout

3.2.1 Block icon signals

(1) The block icon uses the following signals:

DATA STRUCTURE	*UT11001_St_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
INFO_TAG	The tag name for the device (e.g. V0001)	String[20]	An alpha numeric string

DATA STRUCTURE	*UT31001_Dy_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
status_Fault	The valve is in a fault condition (at least one of the five alarm conditions is active)	Bool	1 = valve is in fault, 0 = valve is healthy
status_Aut	The valve is in automatic mode (mode_AutMan = 1)	Bool	1 = in automatic mode 0 = not in automatic mode
status_Man	The valve is in manual mode (mode_AutMan = 0)	Bool	1 = in manual mode 0 = not in manual mode
status_Energised	Energised/deenergised state of the valve (status_Energised = OPERATE)	Bool	1 = valve energised 0 = valve deenergised
status_SimOn	Status indication, shows if simulation mode is active	Bool	1 = sim mode active, 0 = sim mode inactive
status_RemoteOn	Status indication, active if device is in remote mode (mode_RemoteOn = 1)	Bool	1 = remote mode on, 0 = remote mode off
status_LocalOn	Status indication, active if device is in local mode (mode_LocalOn = 1)	Bool	1 = local mode on, 0 = local mode off
status_RLOff	Status indication, ALL mode is active (mode_RemoteOn = 0 and mode_LocalOn = 0)	Bool	1 = ALL mode on, 0 = ALL mode off
status_EStop	Safety signal is active (E_STOP = 1)	Bool	1 = safety signal active, 0 = safety signal inactive
status_Trip	Trip signal is active (TRIP = 1)	Bool	1 = trip signal active, 0 = trip signal inactive
status_Ilock	Interlock signal is active (INTERLOCK = 1)	Bool	1 = interlock signal active, 0 = interlock signal inactive
status_Perm	Permissive signal is active (PERMISSIVE = 1)	Bool	1 = permissive active, 0 = permissive inactive
status_BypassOn	Interlock bypass mode is active (mode_BypassOn = 1)	Bool	1 = bypass active, 0 = Bypass inactive

Table 3.5 Block icon status information

(2)

3.3 Faceplates and associated signals

(1) This module is designed to be used with a supervisory system faceplate, this is a pop-up window that appears on the supervisory system, overlaying the plant mimic.

(2) The faceplate has several, selectable tabs that allow various options and modes to be selected; the faceplate also allows specific configuration information for the valve to be viewed.

(3) The faceplate for FC11001 has the following functional areas:

① **Status**
Shows the status of the valve and clearly identifies the selected operating modes

② **Mode**
Displays the operating modes available to the valve and allows the operator to activate or deactivate any such modes

③ **Interlocks**
Shows the interlock states and allows the interlocks to be bypassed (if permitted)

④ **Simulation**
Allows the valve to be switched to simulation mode and lets the operator select the various simulation modes

⑤ **Configuration**
Displays the primary configuration information for the valve (operating times &c.)

⑥ **Messages**
Displays any alarms that may be active for the valve

(4) The following shows a typical set of faceplates for the module:

Figure 3.3 FC11001 faceplate (typical)

(5) The following figures examine each of these in detail and show the associated data points that are used to control (or be controlled by) each faceplate option.

3.3.1 Faceplate construction and signals

(1) The following section shows each faceplate tab and the signals it uses:

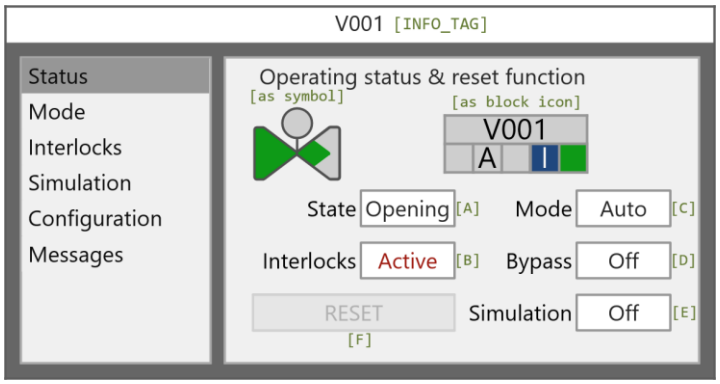


Figure 3.4 FC11001 faceplate — Status tab

(2) This has the following data points:

ELEMENT ID	[A]		
ELEMENT NAME	State	OBJECT TYPE	Text display box
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
FAULT	status_Fault = 1		None
Closed	status_Fault = 0 & status_Closed = 1		None
Opening	status_Fault = 0 & status_Opening = 1		None
Open	status_Fault = 0 & status_Open = 1		None
Closing	status_Fault = 0 & status_Closed = 1		None
ELEMENT ID	[B]		
ELEMENT NAME	Interlocks	OBJECT TYPE	Text display box
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Active	status_Estop = 1 + status_Trip = 1 + status_ILock = 1 + status_Perm = 1		None
Healthy	status_Estop = 0 & status_Trip = 0 & status_ILock = 0 & status_Perm = 0		None

ELEMENT ID	[C]		
ELEMENT NAME	Mode	OBJECT TYPE	Text display box
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Auto	status_Aut = 1		None
Manual	status_Man = 1		None
ELEMENT ID	[D]		
ELEMENT NAME	Bypass	OBJECT TYPE	Text display box
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Off	status_BypassOn = 0		None
ON	status_BypassOn = 1		None
ELEMENT ID	[E]		
ELEMENT NAME	Simulation	OBJECT TYPE	Text display box
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Off	status_SimOn = 0		None
ON	status_SimOn = 1		None
ELEMENT ID	[F]		
ELEMENT NAME	Reset	OBJECT TYPE	Text display box
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
RESET	msg_FailToOpen = 0 & msg_FailToClose = 0 & msg_FailWhileOpen = 0 & msg_FailWhileClosed = 0		None
RESET	msg_FailToOpen = 1 + msg_FailToClose = 1 + msg_FailWhileOpen = 1 + msg_FailWhileClosed = 1		Set mode_Reset = 1

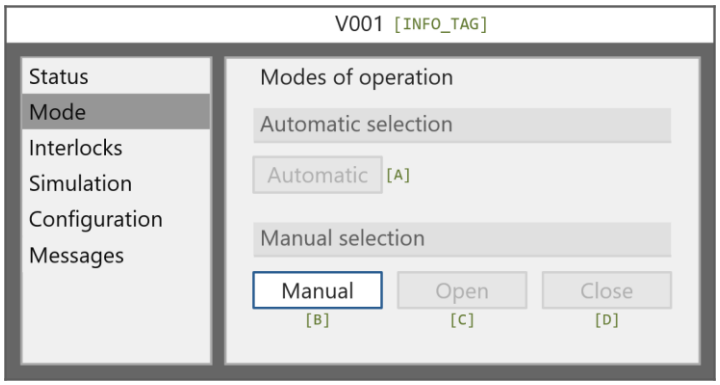


Figure 3.5 FC11001 faceplate — Mode tab

(3) This has the following data points:

ELEMENT ID	[A]		
ELEMENT NAME	Automatic selection	OBJECT TYPE	Clickable text display button
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Automatic	status_Aut = 0		Set mode_AutMan = 1
Automatic	status_Aut = 1		None
ELEMENT ID	[B]		
ELEMENT NAME	Manual selection	OBJECT TYPE	Clickable text display button
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Manual	status_Man = 0		Set mode_AutMan = 0
Manual	status_Man = 1		None
ELEMENT ID	[C]		
ELEMENT NAME	Manual open	OBJECT TYPE	Clickable text display button
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Open	status_Man = 0 + ctrl_Man_OpenClose = 1		None
Open	status_Man = 1 & ctrl_Man_OpenClose = 0		Set ctrl_Man_OpenClose = 1
ELEMENT ID	[D]		
ELEMENT NAME	Manual close	OBJECT TYPE	Clickable text display button
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Close	status_Man = 0 + ctrl_Man_OpenClose = 0		None
Close	status_Man = 1 & ctrl_Man_OpenClose = 1		Set ctrl_Man_OpenClose = 0

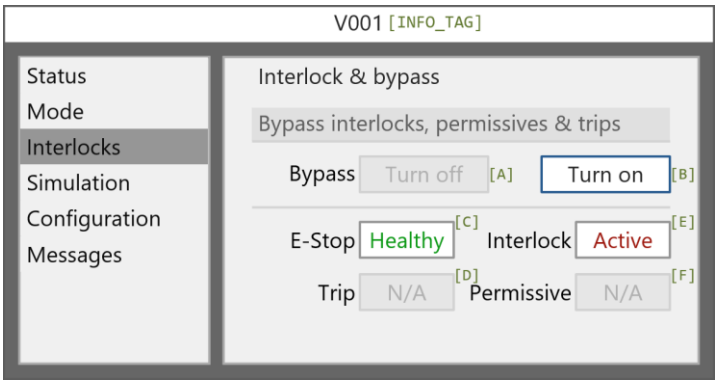


Figure 3.6 FC11001 faceplate — Interlock tab

(4) This has the following data points:

ELEMENT ID	[A]		
ELEMENT NAME	Bypass off selection	OBJECT TYPE	Clickable text display button
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Turn off	status_BypassOn = 0		None
Turn off	status_BypassOn = 1		Set mode_BypassOn = 0
ELEMENT ID	[B]		
ELEMENT NAME	Bypass on selection	OBJECT TYPE	Clickable text display button
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Turn off	status_BypassOn = 1		None
Turn off	status_BypassOn = 0		Set mode_BypassOn = 1
ELEMENT ID	[C]		
ELEMENT NAME	E-Stop status	OBJECT TYPE	Text display box
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
N/A	CONFIG_ESTOP_DIS = 1		None
Healthy	CONFIG_ESTOP_DIS = 0 & status_Estop = 0		None
Active	CONFIG_ESTOP_DIS = 0 & status_Estop = 1		None

ELEMENT ID	[D]		
ELEMENT NAME	Trip status	OBJECT TYPE	Text display box
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
N/A	CONFIG_TRIP_DIS = 1		None
Healthy	CONFIG_TRIP_DIS = 0 & status_BypassOn = 0 & status_Trip = 0		None
Active	CONFIG_TRIP_DIS = 0 & status_BypassOn = 0 & status_Trip = 1		None
Bypassed	CONFIG_TRIP _DIS = 0 & status_BypassOn = 0		None
ELEMENT ID	[E]		
ELEMENT NAME	Interlock status	OBJECT TYPE	Text display box
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
N/A	CONFIG_ILOCK_DIS = 1		None
Healthy	CONFIG_ILOCK_DIS = 0 & status_BypassOn = 0 & status_ Ilock = 0		None
Active	CONFIG_ILOCK_DIS = 0 & status_BypassOn = 0 & status_Ilock = 1		None
Bypassed	CONFIG_ILOCK_DIS = 0 & status_BypassOn = 0		None
ELEMENT ID	[F]		
ELEMENT NAME	Permissive status	OBJECT TYPE	Text display box
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
N/A	CONFIG_PERM_DIS = 1		None
Healthy	CONFIG_PERM_DIS = 0 & status_BypassOn = 0 & status_Perm = 0		None
Active	CONFIG_PERM_DIS = 0 & status_BypassOn = 0 & status_Perm = 1		None
Bypassed	CONFIG_PERM_DIS = 0 & status_BypassOn = 0		None

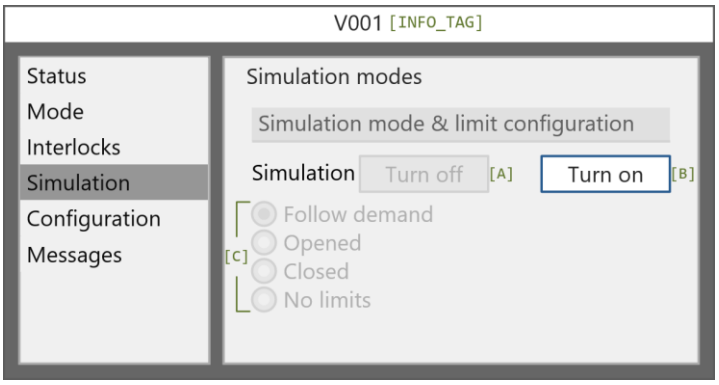


Figure 3.7 FC11001 faceplate — Simulation tab

(5) This has the following data points:

ELEMENT ID	[A]		
ELEMENT NAME	Sim off selection	OBJECT TYPE	Clickable text display button
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Turn off	status_Simon = 0		None
Turn off	status_Simon = 1		Set mode_SimOn = 0
ELEMENT ID	[B]		
ELEMENT NAME	Sim on selection	OBJECT TYPE	Clickable text display button
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Turn on	status_Simon = 1		None
Turn on	status_Simon = 0		Set mode_SimOn = 1
ELEMENT ID	[C]		
ELEMENT NAME	Sim type	OBJECT TYPE	Clickable text display button
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Follow demand Opened Closed No limits	CONFIG_SIM_DIS = 1		None
(●) Follow demand ( ) Opened ( ) Closed ( ) No limits	status_SimOn = 1 & mode_SimValue = 3		Set mode_SimValue = 3
( ) Follow demand (●) Opened ( ) Closed ( ) No limits	status_SimOn = 1 & mode_SimValue = 2		Set mode_SimValue = 2
( ) Follow demand ( ) Opened (●) Closed ( ) No limits	status_SimOn = 1 & mode_SimValue = 1		Set mode_SimValue = 1
( ) Follow demand ( ) Opened ( ) Closed (●) No limits	status_SimOn = 1 & mode_SimValue = 0		Set mode_SimValue = 0

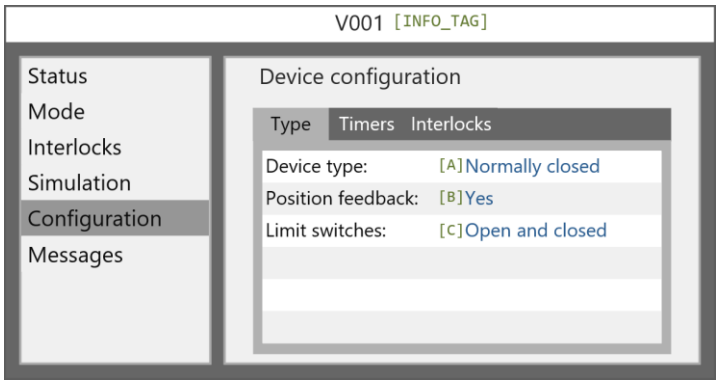


Figure 3.8 FC11001 faceplate — Configuration (Type) tab

(6) This has the following data points:

ELEMENT ID	[A]		
ELEMENT NAME	Device type	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Normally open	CONFIG_NO_NC = 1		None
Normally closed	CONFIG_NO_NC = 0		None
ELEMENT ID	[B]		
ELEMENT NAME	Position feedback	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
No	CONFIG_FBK_OPEN_EN = 0 & CONFIG_FBK_CLOSED_EN = 0		None
Yes	CONFIG_FBK_OPEN_EN = 1 + CONFIG_FBK_CLOSED_EN = 1		None
ELEMENT ID	[C]		
ELEMENT NAME	Limit switches	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
None	CONFIG_FBK_OPEN_EN = 0 & CONFIG_FBK_CLOSED_EN = 0		None
Open only	CONFIG_FBK_OPEN_EN = 1 & CONFIG_FBK_CLOSED_EN = 0		None
Closed only	CONFIG_FBK_OPEN_EN = 0 & CONFIG_FBK_CLOSED_EN = 1		None
Open and closed	CONFIG_FBK_OPEN_EN = 1 & CONFIG_FBK_CLOSED_EN = 1		None

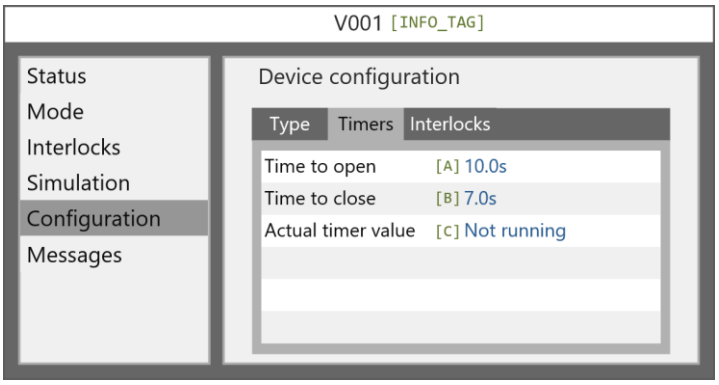


Figure 3.9 FC11001 faceplate — Configuration (Timer) tab

(7) This has the following data points:

ELEMENT ID	[A]		
ELEMENT NAME	Time to open	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
[TIME_OPENING_MAX]	TIME_OPENING_MAX		None
ELEMENT ID	[B]		
ELEMENT NAME	Time to close	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
[TIME_CLOSING_MAX]	TIME_CLOSING_MAX		None
ELEMENT ID	[C]		
ELEMENT NAME	Actual timer value	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Not running	status_Operating = 0		None
[actual_Operating_Timer]	status_Operating = 1		None

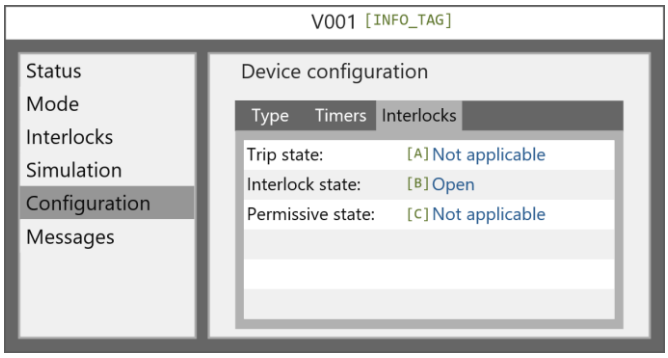


Figure 3.10 FC11001 faceplate — Configuration (Interlocks) tab

(8) This has the following data points:

ELEMENT ID	[A]		
ELEMENT NAME	Trip state	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Not applicable	CONFIG_TRIP_DIS = 1		None
Open	CONFIG_TRIP_DIS = 0 & CONFIG_TRIP_OP_CL = 1		None
Closed	CONFIG_TRIP_DIS = 0 & CONFIG_TRIP_OP_CL = 0		None
ELEMENT ID	[B]		
ELEMENT NAME	Interlock state	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Not applicable	CONFIG_ILOCK_DIS = 1		None
Open	CONFIG_ILOCK_DIS = 0 & CONFIG_ILOCK_OP_CL = 1		None
Closed	CONFIG_ILOCK_DIS = 0 & CONFIG_ILOCK_OP_CL = 0		None
ELEMENT ID	[B]		
ELEMENT NAME	Permissive state	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Not applicable	CONFIG_PERM_DIS = 1		None
Open	CONFIG_PERM_DIS = 0 & CONFIG_PERM_OP_CL = 1		None
Closed	CONFIG_PERM_DIS = 0 & CONFIG_PERM_OP_CL = 0		None

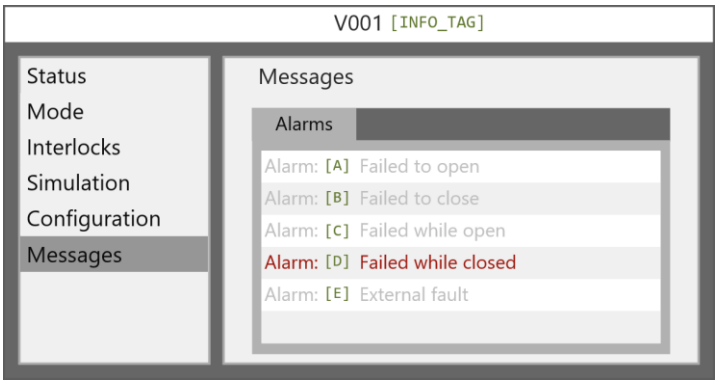


Figure 3.11 FC11001 faceplate — Message tab

(9) This has the following data points:

ELEMENT ID	[A]		
ELEMENT NAME	Fail to open	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Alarm: Failed to open	msg_FailToOpen = 0		None
Alarm: Failed to open	msg_FailToOpen = 1		None
ELEMENT ID	[B]		
ELEMENT NAME	Fail to close	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Alarm: Failed to close	msg_FailToClose = 0		None
Alarm: Failed to close	msg_FailToClose = 1		None
ELEMENT ID	[C]		
ELEMENT NAME	Fail while open	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Alarm: Failed while open	msg_FailWhileOpen = 0		None
Alarm: Failed while open	msg_ FailWhileOpen = 1		None
ELEMENT ID	[D]		
ELEMENT NAME	Fail while close	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Alarm: Failed while close	msg_FailWhileClose = 0		None
Alarm: Failed while close	msg_FailWhileClose = 1		None
ELEMENT ID	[E]		
ELEMENT NAME	External fault	OBJECT TYPE	Text field
APPEARANCE	APPEARANCE DRIVER LOGIC		CLICK ACTION LOGIC
Alarm: External fault	msg_Flt_External = 0		None
Alarm: External fault	msg_Flt_External = 1		None

3.3.2 String constants for the tag name

(1) The STATIC_DATA structure holds the tag name of the valve:

DATA STRUCTURE	*UT11001_St_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
INFO_TAG	Tag name of the valve	String[20]	Device tag

Table 3.6 String constants for the valve

(2) This allows the supervisory system to populate the valve tag (V001 in the symbol, block icon and faceplates above) directly from the data stored in the Controller. It means symbols, block icons and faceplates can be used directly within the supervisory system graphic without customisation; all the data will be populated directly at runtime.

4 Parameters

The following parameters are associated with this module:

PARAMETER	FUNCTION	TYPE	IN/OUT
SYS_SIGNALS	Common system signals logic and timing signals for parametric access	UT21000	In
FBK_OPEN	Optional open state positive feedback (1 = open, 0 = not open) If not required, set permanently to _False	Bool	In
FBK_CLOSED	Optional closed state positive feedback (1 = closed, 0 = not closed) If not required, set permanently to _False	Bool	In
EXT_FAULT	Optional external fault signal for the device (1 = fault, 0 = healthy) If not required, set permanently to _False	Bool	In
E_STOP	Optional emergency stop (safety) signal (1 = emergency stop active, 0 = healthy). If not required, set permanently to _False	Bool	In
TRIP	Optional trip signal (1 = trip condition active, 0 = healthy) If not required, set permanently to _False	Bool	In
INTERLOCK	Optional interlock signal (1 = interlock condition active, 0 = healthy) If not required, set permanently to _False	Bool	In
PERMISSIVE	Optional permissive signal (1 = permissive condition active, 0 = healthy) If not required, set permanently to _False	Bool	In
STATIC_DATA	Data structure that holds the configuration data for the device (normally open/closed, limit switch and interlock configuration &c.)	UT11001	In
DYNAMIC_DATA	Data structure that holds the live data for the device (its mode, status and messages and any other dynamic information required by the module)	UT31001	InOut
OPERATE	Output to drive the valve to a particular state (1 = energised state, 0 = deenergised state)	Bool	Out

Table 4.1 Module parameters

5 Data structures and usage

The module has the following associated data structures:

DATA STRUCTURE	FUNCTION	
UT01000_St_SysRevision	Revision information for this block	See FC01001 for details
UT01001_St_SysLicence	Licence information for this block	See FC01001 for details
UT21001_Dy_SysSignals	System signals for logic and timing	See FC01001 for details
UT11001_St_DevValveIsol	Static data structure for a bistable valve.	
UT31001_Dy_DevValveIsol	Dynamic data structure for a bistable valve. Holds the live data for the device (its mode, status and messages and any other dynamic information required by the module)	

Table 5.1 Data structure list

5.1 UT11001_St_DevValveIsol

DATA STRUCTURE	*UT11001_St_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
CONFIG_NO_NC	Identifies the type of the valve as normally open or normally closed.	Bool	1 = Normally open 0 = Normally closed
CONFIG_FBK_OPEN_EN	Indicates that open positive feedback is present	Bool	1 = open feedback 0 = no open feedback
CONFIG_FBK_CLOSED_EN	Indicates that closed positive is present	Bool	1 = closed feedback 0 = no closed feedback
CONFIG_ESTOP_DIS	Disables the safety signal. If set to 1, any signal on the E_STOP parameter will be ignored	Bool	1 = E_STOP disabled' 0 = E_STOP enabled
CONFIG_TRIP_DIS	Disables the trip interlock signal. If set to 1, any signal on the TRIP parameter will be ignored	Bool	1 = TRIP disabled 0 = TRIP enabled
CONFIG_ILOCK_DIS	Disables the interlock signal. If set to 1, any signal on the INTERLOCK parameter will be ignored	Bool	1 = INTERLOCK disabled 0 = INTERLOCK enabled
CONFIG_PERM_DIS	Disables the permissive interlock signal. If set to 1, any signal on the PERMISSIVE parameter will be ignored	Bool	1 = PERMISSIVE disabled 0 = PERMISSIVE enabled
CONFIG_TRIP_OP_CL	The required state of a valve if the TRIP signal is active	Bool	1 = valve will open 0 = valve will close
CONFIG_ILOCK_OP_CL	The required state of a valve if the INTERLOCK signal is active	Bool	1 = valve will open 0 = valve will close
CONFIG_PERM_OP_CL	The permitted state of a valve if the PERMISSIVE signal is active	Bool	1 = valve can open 0 = valve can close
CONFIG_FP_DIS	Disables any faceplate; if active, clicking the block icon or device symbol will not activate the device faceplate.	Bool	1 = Faceplate disabled 0 = normal
CONFIG_MAN_DIS	Prevents manual mode being activated under all circumstances, device is permanently in auto	Bool	1 = no manual mode, 0 = manual permitted
CONFIG_BYPASS_DIS	Prevents bypass mode being activated under all circumstances	Bool	1 = no bypass mode 0 = bypass permitted
CONFIG_SIM_DIS	Prevents simulation mode being activated under all circumstances	Bool	1 = no simulation 0 = simulation permitted
CONFIG_RL_EN	Allows Remote and Local modes to be selected.	Bool	1 = Remote/local avail 0 = All mode is active
INFO_TAG	Tag name of the valve	String[20]	E.g. V001
TIME_OPENING_MAX	The maximum time required for the valve to travel from a closed state to an open state	Real	Seconds
TIME_CLOSING_MAX	The maximum time required for the valve to travel from an open state to a closed state	Real	Seconds
TIME_DER_OPEN	The percentage of TIME_OPENING_MAX that must expire before the derived FBK_OPEN signal activates.	Real	Percentage
TIME_DER_CLOSED	The percentage of TIME_CLOSING_MAX that must expire before the derived FBK_CLOSED signal activates.	Real	Percentage

Table 5.2 Static data structure *UT11001_St_DevValveIsol*

5.2 UT31001_Dy_DevValveIsol

DATA STRUCTURE	*UT31001_Dy_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
status_Closed	Indicates the valve is correctly in the closed state	Bool	1 = valve is closed 0 = valve is not closed
status_Open	Indicates the valve is correctly in the open state	Bool	1 = valve is open 0 = valve is not open
status_Closing	Indicates the valve is in the process of closing (i.e. closing operation timer is running)	Bool	1 = valve is closing 0 = valve is not closing
status_Opening	Indicates the valve is in the process of opening (i.e. opening operation timer is running)	Bool	1 = valve is opening 0 = valve is not opening
status_Operating	Indicates the valve is opening or closing (i.e. either status_Closing or status_Opening is active)	Bool	1 = valve is changing state 0 = valve is static
status_Fault	The valve is in a fault condition (at least one of the five alarm conditions is active)	Bool	1 = valve is in fault 0 = valve is healthy
status_Energised	Energised/deenergised state of the valve (status_Energised = OPERATE)	Bool	1 = valve energised 0 = valve deenergised
status_Aut	The valve is in automatic mode (mode_AutMan = 1)	Bool	1 = automatic mode active 0 = automatic mode inactive
status_Man	The valve is in manual mode (mode_AutMan = 0)	Bool	1 = manual mode active 0 = manual mode inactive
status_Estop	Safety signal is active (E_STOP = 1)	Bool	1 = safety signal active 0 = safety signal inactive
status_Trip	Trip signal is active (TRIP = 1)	Bool	1 = trip signal active 0 = trip signal inactive
status_Ilock	Interlock signal is active (INTERLOCK = 1)	Bool	1 = interlock signal active 0 = interlock signal inactive
status_Perm	Permissive signal is active (PERMISSIVE = 1)	Bool	1 = permissive active 0 = permissive inactive
status_BypassOn	Status indication, shows if bypass mode is active	Bool	1 = bypass mode active 0 = bypass mode inactive
status_SimOn	Status indication, shows if simulation mode is active	Bool	1 = sim mode active 0 = sim mode inactive
status_RemoteOn	Status indication, active if device is in remote mode (mode_RemoteOn = 1)	Bool	1 = remote mode on, 0 = remote mode off
status_LocalOn	Status indication, active if device is in local mode (mode_LocalOn = 1)	Bool	1 = local mode on, 0 = local mode off
status_RLOff	Status indication, ALL mode is active (mode_RemoteOn = 0 and mode_LocalOn = 0)	Bool	1 = ALL mode on, 0 = ALL mode off

DATA STRUCTURE	*UT31001_Dy_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
msg_FailToOpen	Valve failed to open in the specified opening time	Bool	1 = alarm active 0 = alarm inactive
msg_FailToClose	Valve failed to close in the specified closing time	Bool	1 = alarm active 0 = alarm inactive
msg_FailWhileOpen	Valve suffered position failure while in a static open state	Bool	1 = alarm active 0 = alarm inactive
msg_FailWhileClosed	Valve suffered position failure while in a static closed state	Bool	1 = alarm active 0 = alarm inactive
msg_Flt_External	A reset pulse (active for one Controller cycle)	Bool	1 = alarm active 0 = alarm inactive
mode_AutMan	Determines the operating mode of the valve. mode_AutMan = 1 (automatic mode) if faceplate disabled	Bool	1 = automatic mode 0 = manual mode
mode_BypassOn	Activates bypass mode, if active, any TRIP, INTERLOCK or PERMISSIVE signal will be ignored	Bool	1 = bypass mode on 0 = bypass mode off
mode_SimOn	Activates simulation mode, if active, the position feedback will be specified by the mode_SimValue	Bool	1 = simulation mode on, 0 = simulation mode off
mode_SimValue	Specifies the type of simulated position feedback in use	int	0 = no position feedback 1 = permanently open 2 = permanently closed 3 = follow demand
mode_RemoteOn	Activates remote mode and resets local mode (CONFIG_RL_EN must be set to 1)	Bool	1 = remote mode on 0 = remote mode off
mode_LocalOn	Activates local mode and resets remote mode (CONFIG_RL_EN must be set to 1)	Bool	1 = local mode on, 0 = local mode off
mode_Reset	A reset pulse (active for one Controller cycle)	Bool	1 = reset latching faults 0 = no action
ctrl_Aut_OpenClose	Causes the valve to open and close if automatic mode is active (mode_AutMan = 1), ignored if not in auto mode	Bool	1 = open the valve 0 = close the valve
ctrl_Man_OpenClose	Causes the valve to open and close if manual mode is active (mode_AutMan = 0), ignored if not in man mode	Bool	1 = open the valve 0 = close the valve
batch_ID	Batch (booking) ID (used for batch operations)	UInt	Batch ID number
actual_Operating_Timer	The maximum time required for the valve to travel from a closed state to an open state	Real	Seconds
actual_Last_Op_Time	The actual time for the valve to achieve its correct position (either open or close)	Real	Seconds

DATA STRUCTURE SIGNAL	*UT31001_Dy_DevValveIsol* FUNCTION	TYPE	DETAILS
$derived_fbk_open	Derived feedback open signal (if CONFIG_FBK_OPEN_EN = 0)	Bool	0 = open 1 = not open
$derived_fbk_closed	Derived feedback closed signal (if CONFIG_FBK_CLOSED_EN = 0)	Bool	1 = closed 0 = not closed
$sim_fbk_open	Simulated feedback open signal (if mode_SimOn = 1)	Bool	0 = open 1 = not open
$sim_fbk_closed	Simulated feedback closed signal (if mode_SimOn = 1)	Bool	1 = closed 0 = not closed
$ctrl_OpenClose	actual open or close request, incorporates all priorities	Bool	1 = open the valve 0 = close the valve
$pret_Open	Edge retention (+ve) rising edge of ctrl_OpenClose (change of state to opening)	Bool	Retains last state of signal
$nret_Close	Edge retention (-ve) falling edge of ctrl_OpenClose (change of state to closing)	Bool	Retains last state of signal

Table 5.3 Dynamic data structure *UT31001_Dy_DevValveIsol*

(1) With the exception of the flowing, all these signals are discussed in the previous sections. The outstanding items are firstly:

batch_ID	Batch (booking) ID (used for batch operations)	UInt	Batch ID number

(2) The batch_ID is not required for normal use; it is only used where a "booking" system is used (common in batch and routing applications). Such booking applications can acquire the device if it is free and reserve it for a particular phase of the batch. Batch applications are a topic in their own right and are discussed separately in the User Guide *[Ref. 014]*.

(3) There are internal working signals (all signals the start with a $ are internal working signals used by the module). In this instance they are used as follows:

(4) The $derived_fbk_open and $derived_fbk_closed are the internally derived open and closed feedback signal used in place of FBK_OPEN and FBK_CLOSED if either CONFIG_FBK_OPEN_EN or CONFIG_FBK_CLOSED_EN is set to zero.

(5) The $sim_fbk_open and $sim_fbk_closed are the simulated position feedback signal used in place of FBK_OPEN and FBK_CLOSED if simulation mode is active.

(6) $ctrl_OpenClose is the required state of the valve, it takes into account the auto/manual requested state and modifies this according to any interlocks, safety signals or faults that may be present.

(7) The $pret_Open and $nret_Close are edge retention signals used to detect either a rising (a change to opening) or a falling (a change to closing) edge on the $ctrl_OpenClose signal.

6 Constants & Temporary data

6.1 Constants

CONSTANT	VALUE	FUNCTION	TYPE
K_TIME_OP_MIN	0.21s	Minimum value for TIME_OPENING_MAX and TIME_CLOSING_MAX	Real

Table 6.1 Constants

6.2 Temporary (local) data

SIGNAL	FUNCTION	TYPE
revInfo	Revision information for this block	UT01000
licInfo	Licensing information for this block	UT01001
calClosedElapsedTime	Calculation - Derived closed signal elapsed time limit	Real
calOpenElapsedTime	Calculation - Derived open signal elapsed time limit	Real
wrkOpen	Working - open position feedback signal (actual or derived)	Bool
wrkClosed	Working - closed position feedback signal (actual or derived)	Bool

SIGNAL	FUNCTION	TYPE
wrkCtrlPriority40	Working - priority 40 condition controls the valve	Bool
wrkCtrlPriority30	Working - priority 30 condition controls the valve	Bool
wrkCtrlPriority20	Working - priority 20 condition controls the valve	Bool
wrkCtrlPriority10	Working - priority 10 condition controls the valve	Bool
wrkCtrlPriority00	Working - priority 00 condition controls the valve	Bool
wrkCtrlAutManOpCl	Working - consolidated auto/manual open or close request	Bool
wrkTimeFbkOpenSim	Working - value of opening timer before derived open feedback	Real
wrkTimeFbkClosedSim	Working - value of closing timer before derived closed feedback	Real
wrkInt	Working - storage area (integer)	Int
wrkReal	Working - storage (Real)	Real
wrkDInt	Working - storage area (double integer)	DInt

Table 6.2 Temporary (local) data

7 Block calls and associations

This section lists any blocks which may be called from within this software module (sub-routine functions for example) and any partner blocks with which it may be associated (for example a receive module that is partner with a transmit module &c.). It lists any system functions which may be called (e.g. reading the real time clock) and any system data types that may be used.

Finally, it lists any special calling requirements for the block (for example, must be called from within a cyclic interrupt organisation block) and if the block is using "optimised access"[1] (this is the default arrangement).

7.1 Block calls from within this module

There are no PAL block calls from within this block.

---

[1] Optimised access dynamically optimises the data storage within a block, it means however, that absolute addressing cannot be used to access the data (all access is symbolic).

By default block optimisation should always be used. The exceptions are where an older system (HMI for example) can only access data using absolute addressing, under these circumstances, it is permissible to disable the optimised accessing of associated data blocks.

7.2 Blocks associated with this module

7.2 Blocks associated with this module

This is a stand-alone block and is not associated with any other blocks.

7.3 System block calls and system data types

7.3 System block calls and system data types

There are no system block calls.

7.4 Special properties and requirements

7.4 Special properties and requirements

Block optimisation, IEC Compatibility and library conformance

Block optimisation, IEC Compatibility and library conformance

- Block optimisation is ACTIVE for this block
- The block has been checked for IEC compatibility and is compliant
- The block is compatible with all IEC library-conformance module constraints

Calling requirements

Calling requirements

There are no special calling requirements for this block.

8 Example usage

The following shows a typical deployment of FC11001:

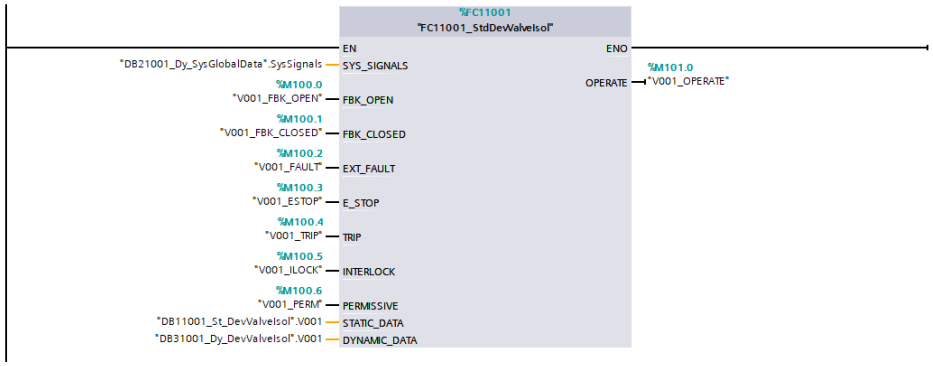


Figure 8.1 FC11001 example usage

The associated static data block is:

DB11001_St_DevValveIsol

	Name	Data type	Start value	Comment
1	Static			
2	_DB_Header	Array[0..79] of Bool		Static DB [DEVICE DRIVER] — Isolating Valve
3	_____0000_0	Bool	false	
4	_____0000_1	Bool	false	DB CURRENT REVISION
5	revInfo	"UT01000_St_SysRevision"		Revision information
6	_____0010_0	Bool	false	
7	_____0010_1	Bool	false	ISOLATING VALVES
8	V001	"UT11001_St_DevValveIsol"		Filter 1 backwash inlet
9	revInfo	"UT01000_St_SysRevision"		Revision information
10	_____0010_0	Int	0	
11	_____0010_1	Int	0	INSTRUMENT CONFIGURATION
12	CONFIG_NO_NC	Bool	false	CONFIG — Normally open or normally closed valve (1 = normally open, 0 = normally closed)
13	CONFIG_FBK_OPEN_EN	Bool	true	CONFIG — Open feedback is enabled (1 = open feedback present, 0 = no open feedback)
14	CONFIG_FBK_CLOSED_EN	Bool	true	CONFIG — Closed feedback is enabled (1 = closed feedback present, 0 = no closed feedback)
15	CONFIG_ESTOP_DIS	Bool	false	CONFIG — E_STOP signal disabled (1 = E_STOP not used, 0 = E_STOP in use)
16	CONFIG_TRIP_DIS	Bool	false	CONFIG — TRIP signal disabled (1 = TRIP not used, 0 = TRIP in use)
17	CONFIG_ILOCK_DIS	Bool	false	CONFIG — INTERLOCK signal disabled (1 = INTERLOCK not used, 0 = INTERLOCK in use)
18	CONFIG_PERM_DIS	Bool	false	CONFIG — PERMISSIVE signal disabled (1 = PERMISSIVE not used, 0 = PERMISSIVE in use)
19	CONFIG_TRIP_OP_CL	Bool	false	CONFIG — TRIP state (1 = open if tripped, 0 = close if tripped)
20	CONFIG_ILOCK_OP_CL	Bool	false	CONFIG — INTERLOCK state (1 = open if interlocked, 0 = close if interlocked)
21	CONFIG_PERM_OP_CL	Bool	false	CONFIG — PERMISSIVE permitted state (1 = permitted state open, 0 = permitted state closed)
22	CONFIG_FP_DIS	Bool	false	CONFIG — Faceplate is disabled (1 = no Faceplate, 0 = normal)
23	CONFIG_MAN_DIS	Bool	false	CONFIG — Manual mode is disabled (1 = no manual mode – auto only, 0 = normal)
24	CONFIG_BYPASS_DIS	Bool	false	CONFIG — Bypass mode is disabled (1 = no Bypass, 0 = normal)
25	CONFIG_SIM_DIS	Bool	false	CONFIG — Simulation mode is disabled (1 = no Simulation, 0 = normal)
26	CONFIG_RL_EN	Bool	false	CONFIG — Remote/local mode enabled (1 = remote/local permitted, 0 = remote/local N/A)
27	_____0020_0	Int	0	
28	_____0020_1	Int	0	INSTRUMENT INFORMATION (TAG & UNITS)
29	INFO_TAG	String[20]	''	INFORMATION — Instrument ID tag
30	_____0030_0	Int	0	
31	_____0030_1	Int	0	INSTRUMENT TIMER DEFAULT VALUES
32	TIME_OPENING_MAX	Real	20.0	TIMER — Initial value of opening operation timer [seconds], counts down from this value
33	TIME_CLOSING_MAX	Real	30.0	TIMER — Initial value of closing operation timer [seconds], counts down from this value
34	TIME_DER_OPEN	Real	80.0	TIMER — Percentage of opening timer to expired before derived open feedback [percentage]
35	TIME_DER_CLOSED	Real	90.0	TIMER — Percentage of closing timer to expired before derived closed feedback [percentage]

Figure 8.2 FC11001 example static DB configuration

The associated dynamic data block is:

DB31001_Dy_DevValveIsol

	Name	Data type	Start value	Comment
1	Static			
2	_DB_Header	Array[0..79] of Bool		Dynamic DB [DEVICE DRIVER] — Isolating Valve
3	_____0000_0	Bool	false	
4	_____0000_1	Bool	false	DB CURRENT REVISION
5	revInfo	"UT01000_St_SysRevision"		Revision information
6	_____0010_0	Bool	false	
7	_____0010_1	Bool	false	ISOLATING VALVES
8	V001	"UT31001_Dy_DevValveIsol"		Filter 1 backwash inlet
9	revInfo	"UT01000_St_SysRevision"		Revision information
10	_____0010_0	Int	0	
11	_____0010_1	Int	0	STATUS (FOR BLOCK ICON AND SYMBOL)
12	status_Closed	Bool	false	STATUS — Valve is confirmed closed (1 = valve is closed, 0 = valve is not closed)
13	status_Open	Bool	false	STATUS — Valve is confirmed open (1 = valve is open, 0 = valve is not open)
14	status_Closing	Bool	false	STATUS — Valve is closing (1 = valve is closing, 0 = valve is not closing)
15	status_Opening	Bool	false	STATUS — Valve is opening (1 = valve is opening, 0 = valve is not opening)
16	status_Operating	Bool	false	STATUS — Valve is changing state (1 = valve is opening or closing, 0 = valve is not moving)
17	status_Fault	Bool	false	STATUS — Valve is in fault (1 = fault present, 0 = healthy)
18	status_Energised	Bool	false	STATUS — Valve output is on (1 = valve energised, 0 = valve deenergised)
19	status_Aut	Bool	false	STATUS — Automatic mode active (1 = in automatic mode, 0 = not in automatic mode)
20	status_Man	Bool	false	STATUS — Manual mode active (1 = in manual mode, 0 = not in manual mode)
21	status_EStop	Bool	false	STATUS — E_STOP safety signal is active (1 = E_STOP active, 0 = E_STOP not active)
22	status_Trip	Bool	false	STATUS — TRIP signal is active (1 = TRIP active, 0 = TRIP not active)
23	status_Ilock	Bool	false	STATUS — INTERLOCK signal is active (1 = INTERLOCK active, 0 = INTERLOCK not active)
24	status_Perm	Bool	false	STATUS — PERMISSIVE signal is active (1 = PERMISSIVE active, 0 = PERMISSIVE not active)
25	status_BypassOn	Bool	false	STATUS — Valve is in bypass mode (1 = bypass mode on, 0 = normal)
26	status_SimOn	Bool	false	STATUS — Valve is in simulation mode (1 = simulation mode on, 0 = normal)
27	status_RemoteOn	Bool	false	STATUS — Valve is in remote mode (1 = remote mode, 0 = remote mode off)
28	status_LocalOn	Bool	false	STATUS — Valve is in local mode (1 = local mode, 0 = local mode off)
29	status_RLOff	Bool	false	STATUS — Remote/local mode disabled (1 = ALL mode on, 0 = RL mode selected)
30	_____0020_0	Int	0	
31	_____0020_1	Int	0	MESSAGES (ALARMS, WARNINGS, FAULTS AND EVENTS)
32	msg_FailToOpen	Bool	false	MESSAGE — Alarm – "Fail to open" is active (1 = alarm active, 0 = alarm not active)
33	msg_FailToClose	Bool	false	MESSAGE — Alarm – "Fail to close" is active (1 = alarm active, 0 = alarm not active)
34	msg_FailWhileOpen	Bool	false	MESSAGE — Alarm – "Fail while open" is active (1 = alarm active, 0 = alarm not active)
35	msg_FailWhileClosed	Bool	false	MESSAGE — Alarm – "Fail while closed" is active (1 = alarm active, 0 = alarm not active)
36	msg_Flt_External	Bool	false	MESSAGE — Alarm – "External fault" is active (1 = alarm active, 0 = alarm not active)
37	_____0030_0	Int	0	
38	_____0030_1	Int	0	OPERATING MODE SELECTION (FROM FACEPLATE OR PANEL)
39	mode_AutMan	Bool	true	MODE — Automatic or manual mode selection (1 = automatic mode, 0 = manual mode)
40	mode_BypassOn	Bool	false	MODE — Bypass mode (1 = bypass mode active, 0 = normal)
41	mode_SimOn	Bool	false	MODE — Simulation mode (1 = simulation mode active, 0 = normal)
42	mode_SimValue	Int	0	MODE — Simulation value (0 no limits, 1 always open, 2 always closed, 3 follow demand)
43	mode_RemoteOn	Bool	false	MODE — Local HMI control enabled (1 = control active, 0 = control disabled or N/A)
44	mode_LocalOn	Bool	false	MODE — Remote SCADA control enabled (1 = control active, 0 = control disabled or N/A)
45	mode_Reset	Bool	false	MODE — Fault reset (1 = Reset any faults – 1 cycle only, 0 = does nothing)
46	_____0040_0	Int	0	
47	_____0040_1	Int	0	CONTROL SIGNALS (TO OPERATE THE DEVICE)
48	ctrl_Aut_OpenClose	Bool	false	CONTROL — Auto mode open or close valve (1 = open valve in auto, 0 = close valve in auto)
49	ctrl_Man_OpenClose	Bool	false	CONTROL — Manual mode open or close valve (1 = open valve in man, 0 = close valve in man)
50	_____0050_0	Int	0	
51	_____0050_1	Int	0	BATCH AND BOOKING DATA
52	batch_ID	UInt	0	BATCH — Booking ID (optional for batch operations)
53	_____0060_0	Int	0	
54	_____0060_1	Int	0	LIVE DATA (SCALED READING & TIMER VALUES)
55	actual_Operating_Timer	Real	0.0	ACTUAL — Timer value — operating opening/closing timer actual value [seconds]
56	actual_Last_Op_Time	Real	0.0	ACTUAL — Timer value — last operation time before feedback [seconds]
57	_____0090_0	Int	0	
58	_____0090_1	Int	0	BLOCK INTERNAL WORKING AND STORAGE AREA
59	$derived_fbk_open	Bool	false	INTERNAL — Derived feedback open signal (if CONFIG_FBK_OPEN_EN = 0)
60	$derived_fbk_closed	Bool	false	INTERNAL — Derived feedback closed signal (if CONFIG_FBK_CLOSED_EN = 0)
61	$sim_fbk_open	Bool	false	INTERNAL — Simulated feedback open signal (if mode_SimOn = 1)
62	$sim_fbk_closed	Bool	false	INTERNAL — Simulated feedback closed signal (if mode_SimOn = 1)
63	$ctrl_OpenClose	Bool	false	INTERNAL — actual open or close control signal passed to the valve (1 = open, 0 = close)
64	$pret_Open	Bool	false	INTERNAL — Edge retention (+ve) rising edge of ctrl_OpenClose (change of state to open)
65	$nret_Close	Bool	false	INTERNAL — Edge retention (-ve) falling edge of ctrl_OpenClose (change of state to close)

Figure 8.3 FC11001 example dynamic DB configuration

9 Test and verification path

The following diagram shows the full test and verification path for the formal release of each version of the software module: FC11001. See the Test Plan *[Ref. 003]* for full details, available here: http://psop.uk/pal-tp-p.

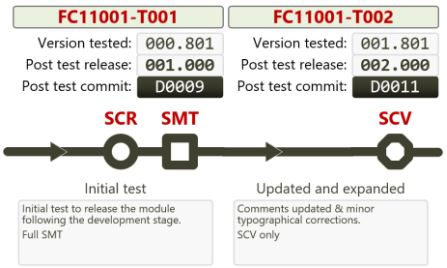


Figure 9.1 FC11001 test and verification path

Where:

ABB.	MEANING	DESCRIPTION
SCR	Source Code Review	A visual inspection of the software to ensure it has been written to the correct standards, uses the correct structures within the software and is generally suitable for deployment.
SMT	Software Module Test	A full and detailed test of an individual software module in isolation; such testing requires that all branches of the software are tested. It tests all the interfaces to the module, any data recorded or stored by the module, all error and exception handling functions and tests all timed and interrupt driven operations.
SMTp	SMT partial retest	A partial repeat of some aspect of the full SMT to retest a small change made to some aspect of the software. The change must be small, specific and localised.
SCV	Software Compatibility Verification	A mechanism for verifying that no executable code software changes have been made to a module following changes to comment fields. Allows typographical errors &c. to be corrected without forcing a full SMT on a module that has not functionally changed.

Note: An SCV can only be performed on a module that has at some point previously, successfully completed both an SCR and SMT. Where an SCV is implemented, it replaces the requirement for both the SCR and the SMT for the test iteration in question.

10 Modification history

The following table lists the modification history for each formal release of the module:

MODULE	*FC11001_StdDevValveIsol*		
REVISION	DATE	COMMIT/AUTH	DESCRIPTION
001.101	2024-01-11	D0000x-000.000	Comments updated and standardised
001.000	2022-03-04	D0009	First release

Table 10.1 FC11001 modification history

11 References

The following is a list of the main Project documents:

REF.	DOC NO	AUTHOR	TITLE
001	PS2001-5-0101-001	PSP	Quality Plan (QP)
002	PS2001-5-0121-002	PSP	Validation Plan (VP)
003	PS2001-5-0121-002	PSP	Test Plan (TP)
004	PS2001-5-1101-001	PSP	User requirements specification (URS)
005	PS2001-5-1111-001	PSP	Requirement Traceability Matrix (RTM)
006	PS2001-5-2101-001	PSP	Functional Specification (FS)
007	PS2001-5-2211-001	PSP	Hardware Design Specification (HDS)
008	PS2001-5-2311-001	PSP	Software Design Specification (SDS)
009	PS2001-5-2313-011	PSP	Style Guide (SG)
010	PS2001-5-234101-001	PSP	ES/WDP Configuration Manual
011	PS2001-5-2312-fcNo	PSP	Software Module Design Specifications (SMDSs)
012	PS2001-5-2301-001	PSP	Software Module Register (SMR)
013	PS2001-5-2302-011	PSP	Software Control Mechanism (SCM)
014	PS2001-5-7111-001	PSP	User Guide (UG)

Table 11.1 Table for references

A full list of all project documentation is available here:

https://practicalseries.com/2001-pal/21-project/01-00-docs.html

12 Typicals

12 Typicals

12.1 SecHeading

12.1 SecHeading

12.1.1 SubHeading

12.1.1 SubHeading

InlineHeading

InlineHeading

12.2 Figures

Unscaled full-page images have the following sizes:

ORIENTATION	WIDTH IN PIXELS	HEIGHT IN PIXELS	ASPECT RATIO
Horizontal	529	729	1:1.38
Vertical	506	745	1:1.47

Table 12.1 Full-page image unscaled resolutions

Extended full-page images have the following sizes:

ORIENTATION	WIDTH IN PIXELS	HEIGHT IN PIXELS	ASPECT RATIO
Extended Horizontal	624	850	1:1.36
Extended Vertical	602	864	1:1.44

Table 12.2 Extended full-page image unscaled resolutions

The most common figure is a partial page (full width) image:

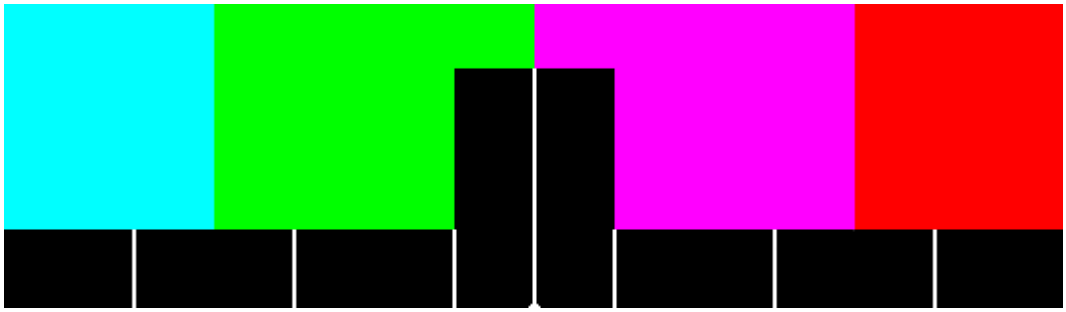


Figure 12.1 Partial page image (529 pixels wide)

The following pages show all other variations:

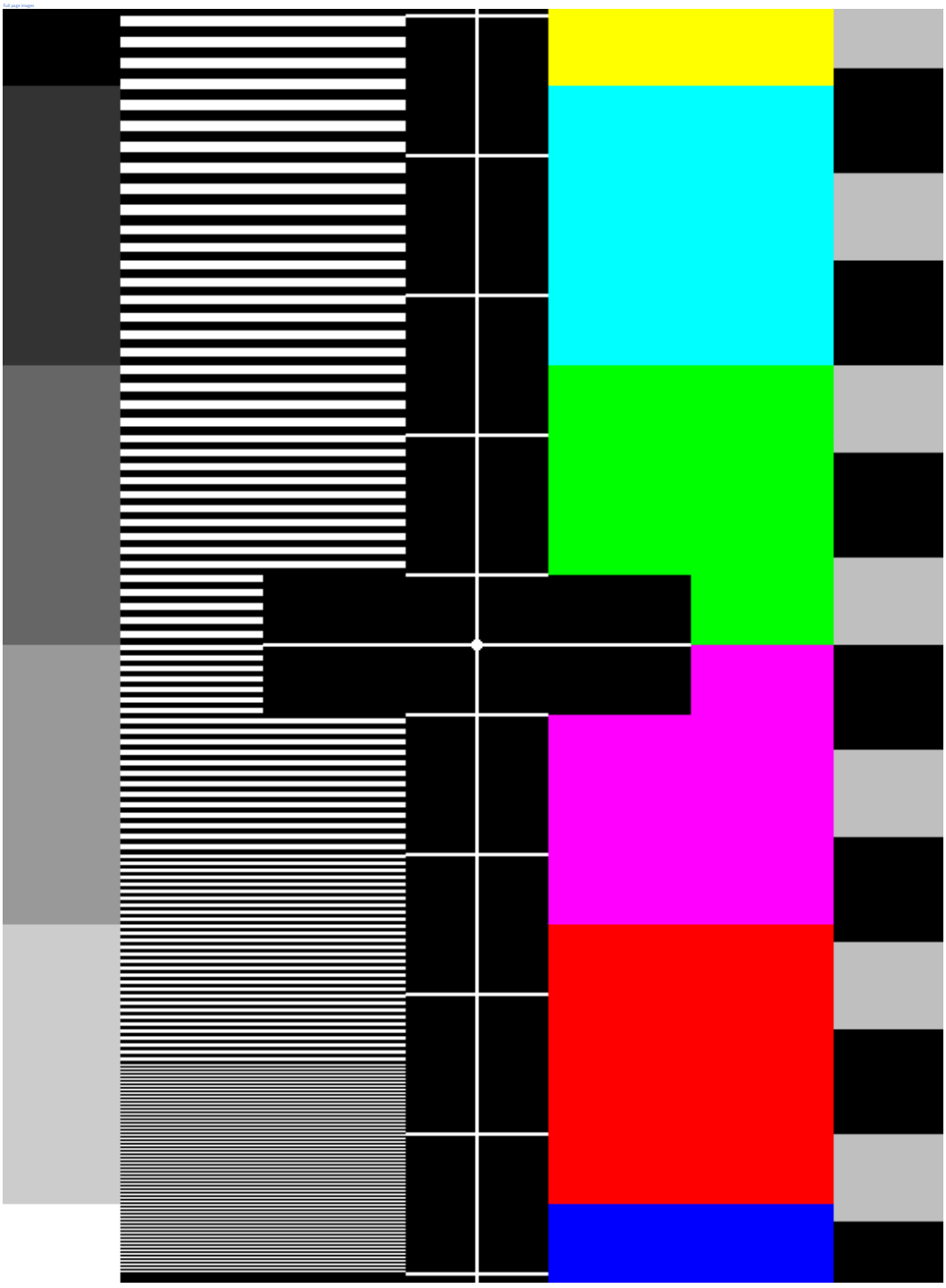


Figure 12.2 Full-page image horizontal arrangement (529 × 729 pixels)

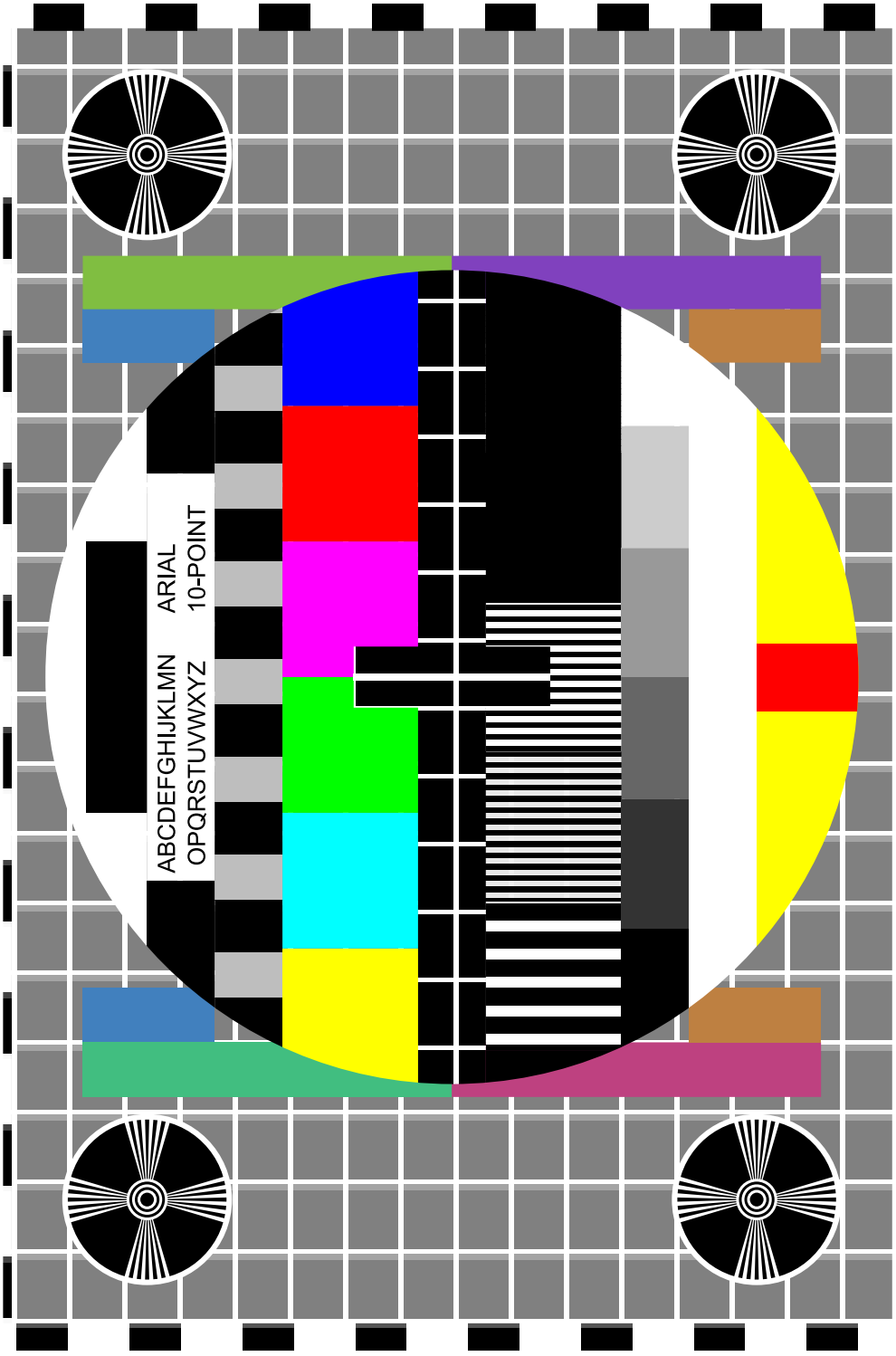


Figure 12.3 Full odd page image vertical arrangement (506 × 745 pixels)

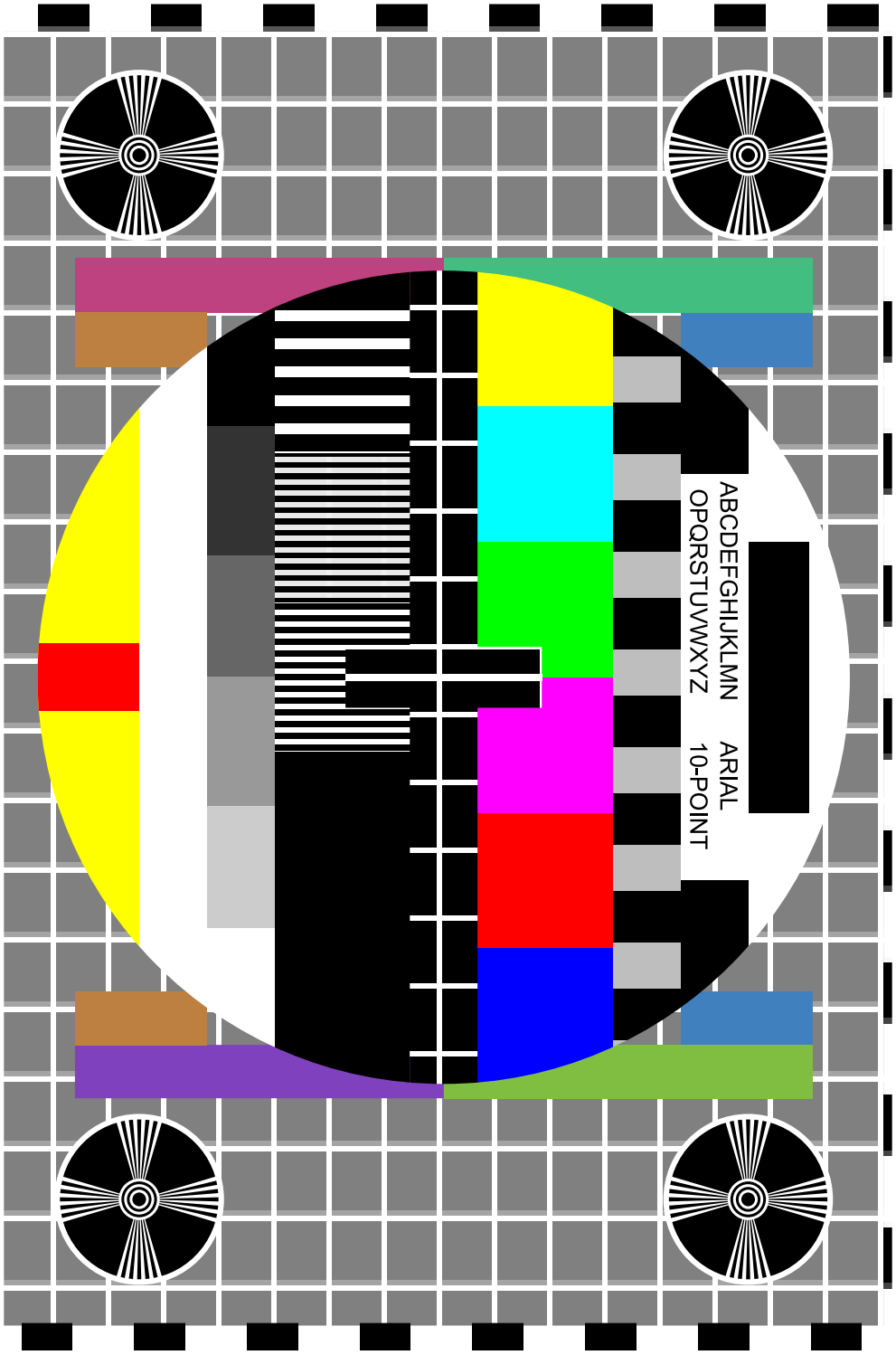


Figure 12.4 Full even page image vertical arrangement (506 × 745 pixels)

Half page image

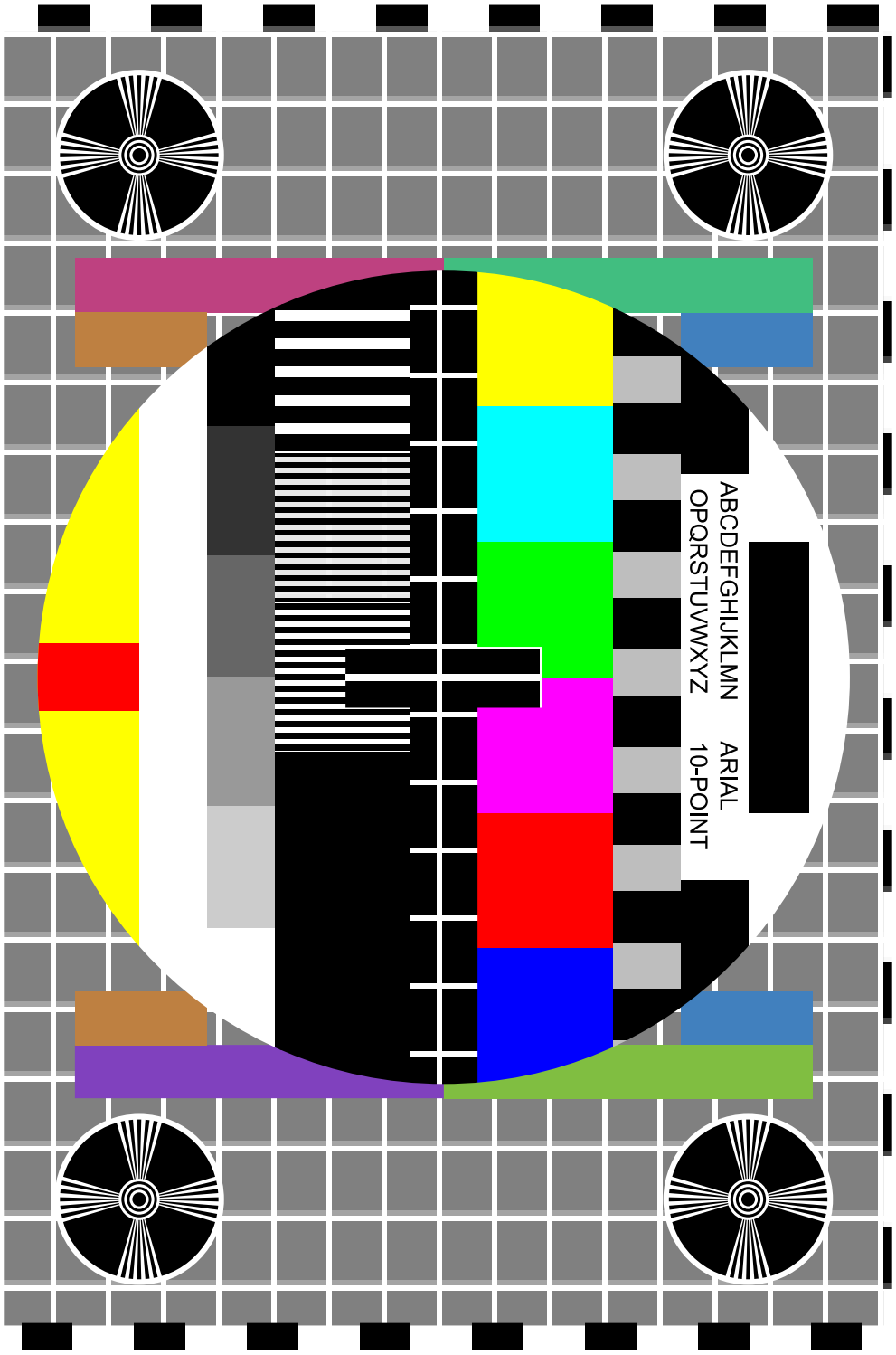


Figure 12.4 Full even page image vertical arrangement (506 × 745 pixels)

Half page image

Half page image with native resolution of 539 pixels wide:

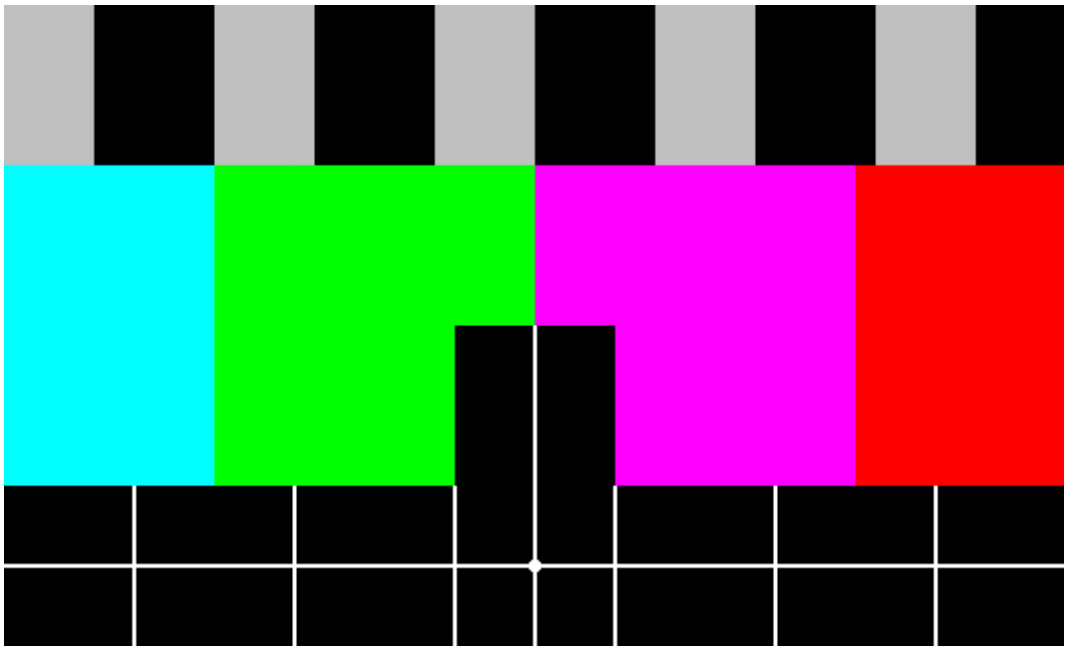


Figure 12.5 Partial page image (529 pixels wide)

Two side-by-side images

Two side-by-side images

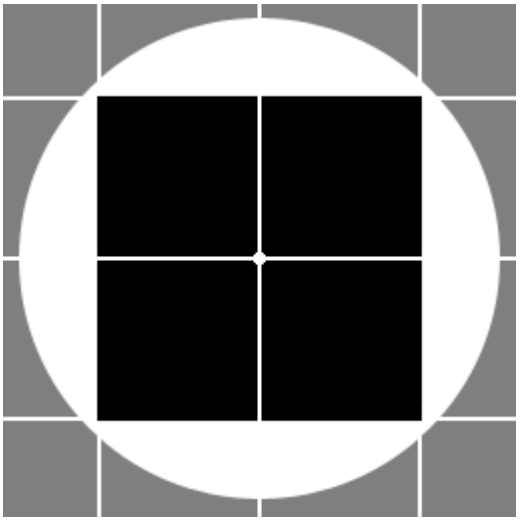


Figure 12.6 Side by side image A (256 pixels wide)

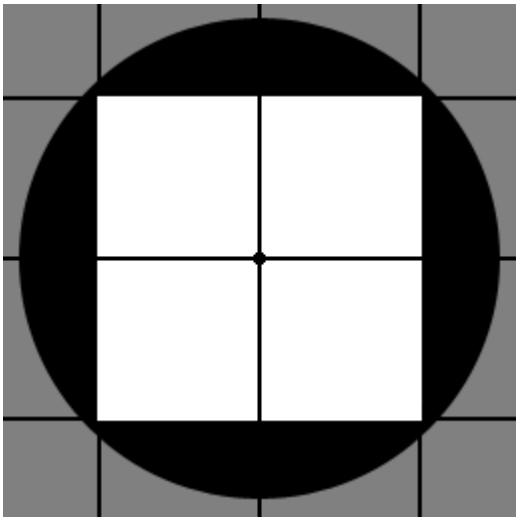


Figure 12.7 Side by side image B (256 pixels wide)

Three side-by-side images

Three side-by-side images

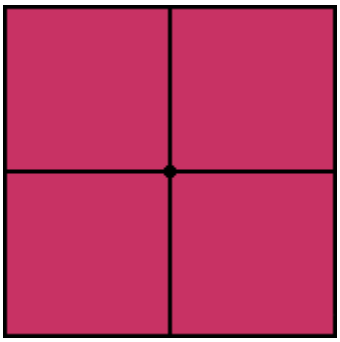


Figure 12.8 Image A (166 pixels wide)

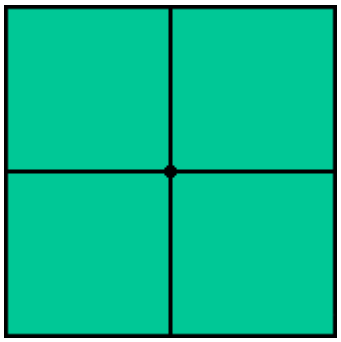


Figure 12.9 Image B (166 pixels wide)

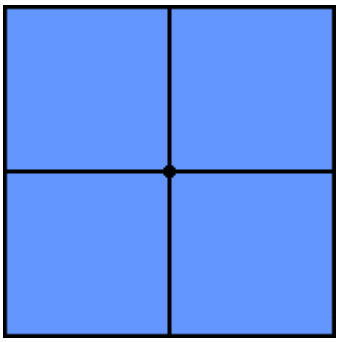


Figure 12.10 Image C (166 pixels wide)

Inline images

Inline images

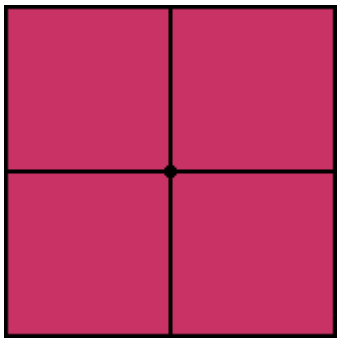


Figure 12.11 Inline image left

I remember him as if it were yesterday, as he came plodding to the inn door, his sea-chest following behind him in a hand-barrow — a tall, strong, heavy, nut-brown man, his tarry pigtail falling over the shoulder of his soiled blue coat, his hands ragged and scarred, with black, broken nails, and the sabre cut across one cheek, a dirty, livid white. I remember him looking round the cover and whistling to himself as he did so, and then breaking out in that old sea-song that he sang so often afterwards:

in the high, old tottering voice that seemed to have been tuned and broken at the capstan bars. Then he rapped on the door with a bit of stick like a handspike that he carried, and when my father appeared, called roughly for a glass of rum. This, when it was brought to him, he drank slowly, like a connoisseur, lingering on the taste and still looking about him at the cliffs and up at our signboard.

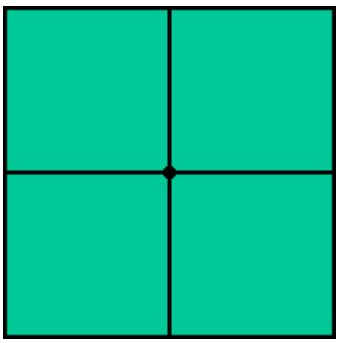


Figure 12.12 Inline image right

12.3 Tables and equations

12.3 Tables and equations

Simple table

Simple table

ITEM	FUNCTION
01	Functional Description
02	Functional Description
03	Functional Description

Table 12.3 Simple table style

Simple table indented

Simple table indented

ITEM	FUNCTION
01	Functional Description
02	Functional Description
03	Functional Description

Table 12.4 Simple table indented

Reference table

Reference table

REF.	DOC NO	AUTHOR	TITLE
001	PS2001-5-0101-001	PSP	Quality Plan (QP)
002	PS2001-5-0121-002	PSP	Validation Plan (VP)
003	PS2001-5-0121-002	PSP	Test Plan (TP)
004	PS2001-5-1101-001	PSP	User requirements specification (URS)
005	PS2001-5-1111-001	PSP	Requirement Traceability Matrix (RTM)
006	PS2001-5-2101-001	PSP	Functional Specification (FS)

Table 12.5 Table for references

Glossary table (complete)

Glossary table (complete)

APPREVIATION.	DESCRIPTION
AC	Alternating Current
AI	Analogue Input
AQ	Analogue Output
ASCII	American Standard Code for Information Interchange
BS	British Standard
BS EN	British standards (BS) adoption of a European Standard (EN)
CAD	Computer Aided Design
CFR	Code of Federal Regulations
CPU	Central Processing Unit
CSS	Cascading Style Sheet
DC	Direct Current
DB	Data Block
DI	Digital Input
DNS	Domain Name System
DOL	Direct Online
DQ	Digital Output
DS	Design Specification (general reference to any design document)
DTL	Date Time Long
EEMUA	Engineering Equipment and Materials Users' Association
EoC	End of Cycle
EN	European Standards
ERP	Enterprise Resource Planning
ES	Engineering Station
EudraLex	European Union Drug Regulation Authority Legislation
EU	European Union
FAT	Factory Acceptance Test
FB	Function Block
FC	Function
FMS	Fieldbus Message Specification
FS	Functional Specification

APPREVIATION.	DESCRIPTION
GAMP	Good Automated Manufacturing Practice
GMP	Good Manufacturing Practice
GRAFCET	GRAPHe de Commande Etape-Transition (sequence documentation)
GxP	Collective abbreviation for GMP and GXP
HDS	Hardware Design Specification
HMI	Human Machine Interface
HTML	Hypertext Mark-up Language
ID	Instance data block or Identifier
iDB	Instance Data Block
IEC	International Electro-technical Commission
IEC 61131-3	IEC standard for the syntax and semantics for PLC programming languages
IET	Institution of Engineering and Technology
IM	Interface Module
IO	Input/Output
IP	Internet Protocol
IQ	Installation Qualification
ISPE	International Society for Pharmaceutical Engineering
ISO	International Standards Organisation
IT	Information Technology
JavaScript	A web-based scripting language
jQuery	A library of JavaScript objects, commonly used in web development
LAD	Ladder Logic (PLC programming language)
Ladder	Ladder Logic (PLC programming language)
LTSB	Long-Term Service Branch
MDF	Medium-density Fibreboard
MIT	Massachusetts Institute of Technology (Licence)
MRPII	Management Resource Planning 2
NC	Normally Closed (type of valve)
NO	Normally Open (type of valve)
OB	Organisation Block
OQ	Operational qualification
OSL	Operating State Logic

APPREVIATION.	DESCRIPTION
PAL	Practical Series Automation Library
P&ID	Piping and Instrumentation Diagram
PC	Personal Computer
PDF	Portable Document Format
PDT	PLC Data Type
PG	Programmer (or programming device, see ES)
PI	Process Image
PID	Proportional, Integral, Derivative — a common type of control loop
PII	Process Image of Inputs
PIP	Process Image Partition
PIPI	Process Image Partition of Inputs
PIPQ	Process Image Partition of Outputs
PIQ	Process Image of Outputs
PLC	Programmable Logic Controller (another name for a Siemens Controller)
PN/IE	Profinet/Industrial Ethernet
ProfiBus	Process Field Buss
Profinet	Process Field Net
PSP	Practical Series of Publications
QHD	Quad High Definition
QMS	Quality Management System
QP	Quality Plan
RAL	Colour standards (Reichs-Ausschuß für Lieferbedingungen und Gütesicherung)
RAM	Random Access Memory
RoC	Rate of Change
RTD	Resistance Temperature Device
RT	Run Time
RTM	Requirements Traceability Matrix
SCADA	Supervisory Control and Data Acquisition
SCM	Software Control Mechanism
SDS	Software Design Specification
SDT	System Data Type
SG	Style Guide

APPREVIATION.	DESCRIPTION
SIT	Software Integration Test document
SMDS	Software Module Design Specification
SMT	Software Module Test document
SoC	Start of Cycle
SQL	Structural Query Language
SSD	Solid State Drive
STL	Statement List (PLC programming language)
TIA	Totally Integrated Solutions (TIA Portal, a Siemens programming tool)
TC	Thermocouple (when referring to IO cards)
TCP/IP	Transmission Control Protocol/Internet Protocol
UDT	User Data Type
UG	User Guide
UI or U/I	Voltage and current (when referring to IO cards)
UK	United Kingdom
URS	User requirements specification
US	United States of America
USB	Universal Serial Bus
UT	User Data Type (alternative abbreviation)
VAC	Voltage (alternating current)
VDC	Voltage (direct current)
VP	Validation Plan
VSD	Variable Speed Drive
WinCC	A Siemens Simatic SCADA system

Table 12.6 Glossary table (complete)

Parameter table

Parameter table

PARAMETER	FUNCTION	TYPE	IN/OUT
SYS_SIGNALS	Common system signals logic and timing signals for parametric access	UT21000	In
FBK_OPEN	logic and timing signals for parametric access (1 = open, 0 = not open) If not required, set permanently to _False	Bool	In
FBK_CLOSED	Optional open state positive feedback (1 = closed, 0 = not closed) If not required, set permanently to _False	Bool	In
STATIC_DATA	Data structure that holds the configuration data for the device (limit switch and interlock configuration &c.)	UT11101	In
DYNAMIC_DATA	Data structure that holds the live data for the device (mode, status, messages and other dynamic data)	UT31101	In/Out
OPEN_CMD	Output to drive the valve to an open state (1 = Open valve, 0 = no action)	Bool	Out
CLOSE_CMD	Output to drive the valve to a closed state (1 = Close valve, 0 = no action)	Bool	Out

Table 12.7 Module parameters

Data structure table

Data structure table

DATA STRUCTURE	FUNCTION	
UT01000_St_SysRevision	Revision information for this block	See FC01001 for details
UT01001_St_SysLicence	Licence information for this block	See FC01001 for details
UT21001_Dy_SysSignals	System signals for logic and timing	See FC01001 for details
UT11001_St_DevValveIsol	Static data structure for a bistable valve.	
UT31001_Dy_DevValveIsol	Dynamic data structure for a bistable valve. Holds the live data for the device (its mode, status and messages and any other dynamic information required by the module)	

Table 12.8 Data structure list

Static data table

Static data table

DATA STRUCTURE	*UT11001_St_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
CONFIG_MAINT_OP	Maintains or deenergises the outputs (OPEN _CMD or CLOSE_CMD) when the valve reaches the required state	Bool	1 = maintain outputs 0 = deenergise outputs
CONFIG_FBK_OPEN_EN	Indicates that open positive feedback is present	Bool	1 = open feedback 0 = no open feedback
CONFIG_FBK_CLOSED_EN	Indicates that closed positive is present	Bool	1 = closed feedback 0 = no closed feedback

Table 12.9 Static data structure *UT11001_St_DevValveIsol*

Dynamic data table

Dynamic data table

DATA STRUCTURE	*UT31001_Dy_DevValveIsol*		
SIGNAL	FUNCTION	TYPE	DETAILS
status_Closed	Indicates the valve is correctly in the closed state	Bool	1 = valve is closed 0 = valve is not closed
status_Fault	The valve is in a fault condition (at least one of the five alarm conditions is active)	Bool	1 = valve is in fault 0 = valve is healthy
msg_FailToOpen	Valve failed to open in the specified opening time	Bool	1 = alarm active 0 = alarm inactive
msg_FailToClose	Valve failed to close in the specified closing time	Bool	1 = alarm active 0 = alarm inactive
mode_AutMan	Determines the operating mode of the valve. mode_AutMan = 1 (automatic mode) if faceplate disabled	Bool	1 = automatic mode 0 = manual mode
mode_BypassOn	Activates bypass mode, if active, any TRIP, INTERLOCK or PERMISSIVE signal will be ignored	Bool	1 = bypass mode on 0 = bypass mode off
ctrl_Aut_OpenClose	Causes the valve to open and close if automatic mode is active (mode_AutMan = 1), ignored if not in auto mode	Bool	1 = open the valve 0 = close the valve
ctrl_Man_OpenClose	Causes the valve to open and close if manual mode is active (mode_AutMan = 0), ignored if not in man mode	Bool	1 = open the valve 0 = close the valve
actual_Operating_Timer	Actual (elapsed) time of the operation timer	Real	Seconds
actual_Runon_Timer	Actual (elapsed) value of the run-on timer	Real	Seconds

Table 12.10 Dynamic data structure *UT11001_St_DevValveIsol*

The details column may be replaced with values or examples as required.

Static/dynamic data linkage

Static/dynamic data linkage

PARAM	ASSOCIATED UDT	ASSOCIATED DATA BLOCK
STATIC_DATA	*UT11001_St_DevValveIsol*	*DB11001_St_DevValveIsol*
DYNAMIC_DATA	*UT31001_St_DevValveIsol*	*DB31001_St_DevValveIsol*

Table 12.11 Associated UDTs and data blocks

Constants table
Static/dynamic data linkage

Static/dynamic data linkage

PARAM	ASSOCIATED UDT	ASSOCIATED DATA BLOCK
STATIC_DATA	*UT11001_St_DevValveIsol*	*DB11001_St_DevValveIsol*
DYNAMIC_DATA	*UT31001_St_DevValveIsol*	*DB31001_St_DevValveIsol*

Table 12.11 Associated UDTs and data blocks

Constants table

CONSTANT	VALUE	FUNCTION	TYPE
K_TIME_OP_MIN	0.21s	Minimum value for TIME_OPENING_MAX and TIME_CLOSING_MAX	Real
K_TIME_SIM_MIN	0.21s	Elapsed operating timer value for follow demand sim signals	Real

Table 12.12 Constants

Local data table

Local data table

SIGNAL	FUNCTION	TYPE
revInfo	Revision information for this block	UT01000
licInfo	Licensing information for this block	UT01001
calClosedElapsedTime	Calculation - Derived closed signal elapsed time limit	Real
calOpenElapsedTime	Calculation - Derived open signal elapsed time limit	Real

Table 12.13 Temporary (local) data

Black cell table

Block call table

BLOCK	FUNCTION	USAGE
RD_SINFO	Read OB start information	Identifies the first cycle after a CPU start
RD_LOC_T	Read the local system (RTC) time	Reads the CPU real time clock
RT_INFO	Read runtime statistics	Returns the previous, min and max cycle times

Table 12.14 System block calls

System data structure table

System data structure table

DATA STRUCTURE	*SI_ProgramCycle*	
SIGNAL	FUNCTION	TYPE
SI_Format	Start information format, indicates if the current OB is using block optimisation: 16#FF = No information 16#FE = Block optimised	USInt
OB_Class	Class of the calling OB, for this data structure the result is always 1 (indicating cyclic OB)	USInt
OB_Nr	Number of the calling OB	UInt
Initial_Call	Indicates the first call of the block 1 = First call (in the case of OB1 this is the CPU first cycle)	Bool
Remanence	Indicates the block is using retentive storage 1 = Retentive storage has been allocated to the block	Bool

Table 12.15 System data structure

Test and verification table

Test and verification table

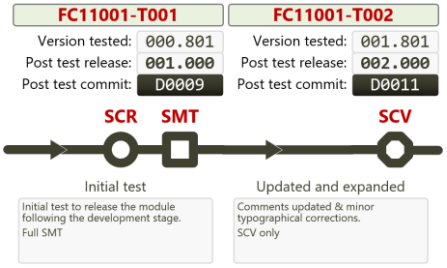


Figure 12.13 FC11001 test and verification path

Where:

Abb.	Meaning	Description
SCR	Source Code Review	A visual inspection of the software to ensure it has been written to the correct standards, uses the correct structures within the software and is generally suitable for deployment.
SMT	Software Module Test	A full and detailed test of an individual software module in isolation; such testing requires that all branches of the software are tested. It tests all the interfaces to the module, any data recorded or stored by the module, all error and exception handling functions and tests all timed and interrupt driven operations.
SMTp	SMT partial retest	A partial repeat of some aspect of the full SMT to retest a small change made to some aspect of the software. The change must be small, specific and localised.
SCV	Software Compatibility Verification	A mechanism for verifying that no executable code software changes have been made to a module following changes to comment fields. Allows typographical errors &c. to be corrected without forcing a full SMT on a module that has not experienced a software change.
Note:	*An SCV can only be performed on a module that has at some point previously, successfully completed both an SCR and SMT. Where an SCV is implemented, it replaces the requirement for both the SCR and the SMT for the test iteration in question.*	

Module verification table

Module verification table

Module	*FC11001_StdDevValveIsol*		
Revision	Date	Commit/Auth	Description
001.101	2024-01-11	D0000x-000.000	Comments updated and standardised
001.000	2022-03-04	D0009	First release

Table 12.16 FC11001 modification history

By-the-Way table

By-the-Way table

By the Way (Heading) — #2861 GH BTW-Head (San BTW Bo)
By the Way body text area — #2163 GB BTW-Body (San BTW Asp)
By the early 1960s, creating an engine to withstand the temperatures and the pressures of the F-1 was, thanks to new metallurgical and engineering techniques, not a formidable problem.

Abstract table

Abstract table

ABSTRACT

FC11001_StdDevValveIsol controls the operation of either a normally closed or normally open isolating valve configured with either open, closed, both open and closed or no position feedback.

The module generates fault logic for the valve that will trigger specific alarms depending on the fault in question. The alarms within this module are: The valve module supports all forms of interlock, permissive and trip signals, and emergency stop signals..

FC11001	Software module design specification (SMDS)
	The PDF version of the SMDS for this module is available online here: https://psop.uk/pal-smds-fc11001-p It is also available as part of the TIA Portal project *User Defined Documentation*; it can be accessed by selecting *FC11001* in the Project Tree (on the left) and pressing SHIFT+F1

Code fragment table

Code fragment table

Index.html

```
<div class="rg-row">                          <!-- Start of cover row -->
    <div class="rg-col rg-span1-5"></div>     <!-- Left column NOT USED -->
    <div class="rg-col rg-span3-5">           <!-- Start of Cover area column -->
```

12.4 Equations

12.4 Equations

Simple equation

Simple equation

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos\frac{n\pi x}{L} + b_n \sin\frac{n\pi x}{L} \right) \tag{12.1}$$

Wide equation

Equation (12.1)

$$actual\_Value = \frac{(RAW\_ANALOG \ - actual\_RawMin) \times (CONFIG\_SCALE\_MAX - CONFIG\_SCALE\_MIN)}{(actual\_RawMax - actual\_RawMin)} + CONFIG\_SCALE\_MIN$$

Equation (12.2)

Extendedequation

Extended equation

The quadratic equation:

$$ax^2 + bx + c = 0$$

Has the roots:

$$\alpha_1 = -\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1}$$

$$\alpha_2 = -\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1}$$

Let:

$$\beta_1 = -\alpha_1 = +\zeta\omega_n + \omega_n\sqrt{\zeta^2 - 1} \tag{12.3}$$

$$\beta_2 = -\alpha_2 = +\zeta\omega_n - \omega_n\sqrt{\zeta^2 - 1}$$

So:

$$\beta_1\beta_2 = \omega_n^2 \tag{12.4}$$

12.5 Sidebars

Odd page sidebar

Sidebars are text boxes inserted in the outer margin area; the text box is exactly 30 mm wide by 198 mm high. The sidebars are precisely positioned to match the height of the body text area of the page. The positioning is different for odd and even numbered pages:

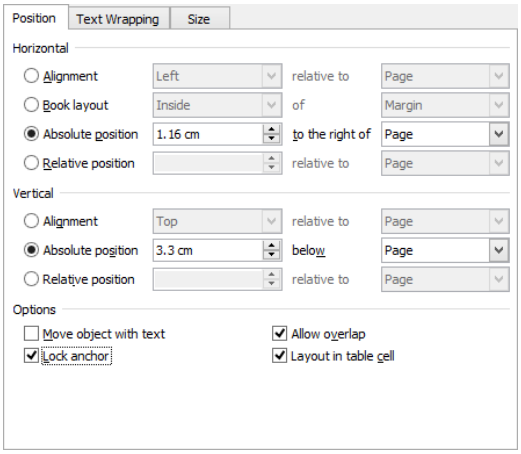


Figure 12.14 Even page sidebar positioning

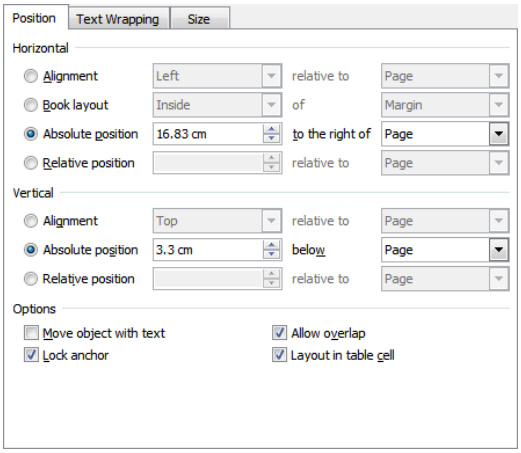


Figure 12.15 Odd page sidebar positioning

Sidebar

The sidebar for odd numbered pages is 11mm from the text box to the page edge (1/2 the inside margin).

Odd page sidebars are left justified

Sidebars contain a table of the same width with left and right margins set to zero. By changing the row heights within the table, it is possible to line sidebar text with the main body text

Sidebars are text boxes inserted in the outer margin area; the text box is exactly 30 mm wide by 198 mm high

The bottom line of the first sidebar text line should align with the bottom of the corresponding line in the main body.

Even page sidebar
Odd page sidebar

Odd page sidebar

Sidebars are text boxes inserted in the outer margin area; the text box is exactly 30 mm wide by 198 mm high. The sidebars are precisely positioned to match the height of the body text area of the page. The positioning is different for odd and even numbered pages:

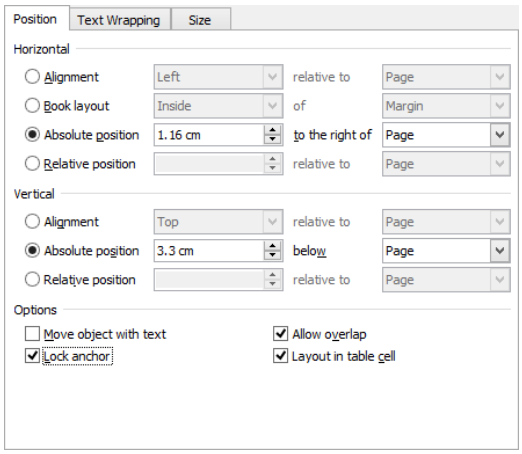


Figure 12.14 Even page sidebar positioning

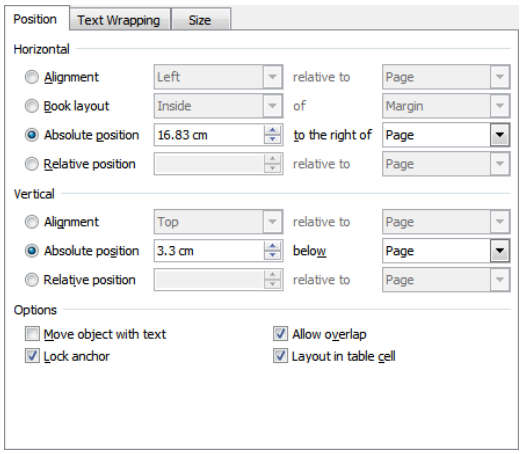


Figure 12.15 Odd page sidebar positioning

Sidebar

The sidebar for odd numbered pages is 11mm from the text box to the page edge (1/2 the inside margin).

Even page sidebar

Sidebars are text boxes inserted in the outer margin area; the text box is exactly 30 mm wide by 198 mm high. The sidebars are precisely positioned to match the height of the body text area of the page. The positioning is different for odd and even numbered pages:

Odd page sidebars are right justified

Sidebars contain a table of the same width with left and right margins set to zero. By changing the row heights within the table, it is possible to line sidebar text with the main body text

Sidebars are text boxes inserted in the outer margin area; the text box is exactly 30 mm wide by 198 mm high

The bottom line of the first sidebar text line should align with the bottom of the corresponding line in the main body.

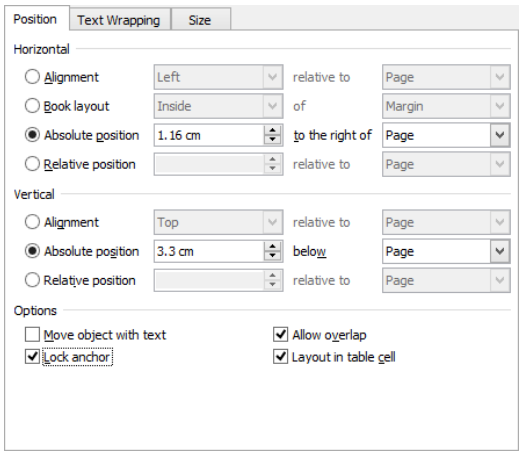


Figure 12.16 Even page sidebar positioning

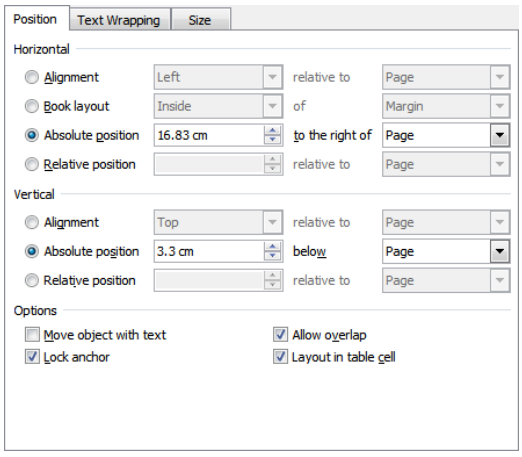


Figure 12.17 Odd page sidebar positioning

12.6 Alt codes

The following table lists the most common alt codes along with the HTML (&) symbol.

Unicode decimal references are given in HTML as &#nnnn.

hexadecimal HTML as &#xhhhh and in CSS as \hhhh (hhhh being the hexadecimal equivalent of the decimal ALT nnnn number given here).

	MATHEMATICAL	HTML	ALT
×	Multiplication sign	×	alt + 0215
÷	Division sign	÷	alt + 0247
−	Minus sign	&minus	alt + 8722
±	Plus/minus sign	±	alt + 0177
⁄	Fraction slash	⁄	alt + 8260
ⁿ	Superscript lowercase n	-	alt + 8319
℮	Estimated symbol	-	alt + 8494
∂	Partial differential	-	alt + 8706
Δ	Increment	-	alt + 8710
∏	N-array product	∏	alt + 8719
∑	N-array summation	∑	alt + 8721
∕	Division slash	-	alt + 8725
∙	Bullet operator	-	alt + 8729
√	Square root	√	alt + 8730
∞	Infinity	∞	alt + 8734
∟	Right angle	-	alt + 8735
∩	Intersection	∩	alt + 8745
∫	Integral	∫	alt + 8747
⌠	Top half integral	-	alt + 8992
⌡	Bottom half integral	-	alt + 8993

	MATHEMATICAL CONT.	HTML	ALT
≈	Almost equal to	≈	alt + 8776
≠	Not equal to	≠	alt + 8800
≡	Identical to	≡	alt + 8801
<	Less than	<	alt + 0060
>	Greater than	>	alt + 0062
≤	Less than or equal to	≤	alt + 8804
≥	Greater than or equal to	≥	alt + 8805
	Miscellaneous		
¦	Broken vertical bar	¦	alt + 0166
°	Degree sign	°	alt + 0176
·	Middle dot	·	alt + 0183
•	Bullet	•	alt + 8226
←	Leftwards arrow	←	alt + 8592
↑	Upwards arrow	↑	alt + 8593
→	Rightwards arrow	→	alt + 8594
↓	Downwards arrow	↓	alt + 8595
↔	Left right arrow	↔	alt + 8596
↕	Up down arrow	-	alt + 8597
&	Ampersand	&	alt+0026
	Non breaking space		alt+0160

	NUMBERS	HTML	ALT
¹	Superscript one	¹	alt + 0185
²	Superscript two	²	alt + 0178
³	Superscript three	³	alt + 0179
½	Fraction one half	½	alt + 0189
⅓	Fraction one third	-	alt + 8531
⅔	Fraction two thirds	-	alt + 8532
¼	Fraction one quarter	&frac14	alt + 0188
¾	Fraction three quarters	&frac34	alt + 0190
⅛	Fraction one eighth	-	alt + 8539
⅜	Fraction three eighths	-	alt + 8540
⅝	Fraction five eighths	-	alt + 8541
⅞	Fraction seven eighths	-	alt + 8542

	CURRENCY		
¢	Cent sign	¢	alt + 0162
£	Pound sign	£	alt + 0163
¤	General currency sign	¤	alt + 0164
₣	French franc sign	-	alt + 8355
₤	Lira sign	-	alt + 8356
₧	Peseta sign	-	alt + 0158
₪	New sheqel sign	-	alt + 8362
₫	Dong sign	-	alt + 8363
€	Euro sign	€	alt + 0128

	LEGAL & TECHNICAL		
§	Section sign	§	alt + 0167
¶	Paragraph sign	¶	alt + 0182
©	Copyright sign	©	alt + 0169
®	Registered trademark sign	®	alt + 0174
™	Trademark sign	™	alt + 8482
ª	Feminine ordinal indicator	ª	alt + 0170
¬	Not sign	¬	alt + 0172
⌐	Reversed not sign	-	alt + 8976
µ	Micro sign	µ	alt + 0181
‰	Per mille sign	‰	alt + 8240
′	Prime (straight quote)	′	alt + 8242
″	Double prime (straight quote)	″	alt + 8243
℅	Care of	-	alt + 8453
№	Numero sign	-	alt + 8470
Ω	Ohm sign	-	alt + 8486
⌂	House	-	alt + 8962
†	Dagger	†	alt+8224
‡	Double dagger	‡	alt+8225

	PUNCTUATION	HTML	ALT
‼	Double exclamation mark	-	alt + 8252
¡	Inverted exclamation mark	¡	alt + 0161
¿	Inverted question mark	¿	alt + 0191
“	Left double quote	“	alt + 8220
”	Right double quote	”	alt + 8221
„	Double low-9 quote	„	alt + 8222
‘	Left single quote	‘	alt + 8216
’	Right single quote	& rsquo;	alt + 8217
,	Single low-9 quote	‚	alt + 8218
‛	Single high-reversed-9 quote	-	alt + 8219
«	Left double angle quote	«	alt + 0171
»	Right double angle quote	»	alt + 0187
‹	Single left angle quote	‹	alt + 8249
›	Single right-angle quote	›	alt + 8250
…	Horizontal ellipsis	…	alt + 8230
–	En dash	–	alt + 8211
—	Em dash	—	alt + 8212
―	Horizontal bar	-	alt + 8213
‗	Double low line	-	alt + 8215
‾	Overline	‾	alt + 8254

	SHAPES		
▀	Upper half block	-	alt + 9600
▄	Lower half block	-	alt + 9604
█	Full block	-	alt + 9608
▌	Left half block	-	alt + 9612
▐	Right half block	-	alt + 9616
░	Light shade	-	alt + 9617
▒	Medium shade	-	alt + 9618
▓	Dark shade	-	alt + 9619
■	Black square	-	alt + 9632
□	White square	-	alt + 9633
▪	Black small square	-	alt + 9642
▫	White small square	-	alt + 9643
▬	Black rectangle	-	alt + 9644
▲	Black up triangle	-	alt + 9650
▶	Black right pointer	-	alt + 9658
▼	Black down triangle	-	alt + 9660
◀	Black left pointer	-	alt + 9668
◊	Lozenge	◊	alt + 9674
○	White circle	-	alt + 9675
●	Black circle	-	alt + 9679
◘	Inverse bullet	-	alt + 9688
◙	Inverse white circle	-	alt + 9689
◦	White bullet	-	alt + 9702

	GREEK LETTERS	HTML	ALT		GREEK LETTERS	HTML	ALT
α	Greek small Alpha	α	alt + 0945	Α	Greek capital Alpha	Α	alt + 0913
β	Greek small Beta	β	alt + 0946	Β	Greek capital Beta	Β	alt + 0914
γ	Greek small Gamma	γ	alt + 0947	Γ	Greek capital Gamma	Γ	alt + 0915
δ	Greek small Delta	δ	alt + 0948	Δ	Greek capital Delta	Δ	alt + 0916
ε	Greek small Epsilon	ε	alt + 0949	Ε	Greek capital Epsilon	Ε	alt + 0917
ζ	Greek small Zeta	ζ	alt + 0950	Ζ	Greek capital Zeta	Ζ	alt + 918
η	Greek small Eta	η	alt + 0951	Η	Greek capital Eta	Η	alt + 0919
θ	Greek small Theta	θ	alt + 0952	Θ	Greek capital Theta	Θ	alt + 0920
ι	Greek small Iota	ι	alt + 0953	Ι	Greek capital Iota	Ι	alt + 0921
κ	Greek small Kappa	κ	alt + 0954	Κ	Greek capital Kappa	Κ	alt + 0922
λ	Greek small Lambda	λ	alt + 0955	Λ	Greek capital Lambda	Λ	alt + 0923
μ	Greek small Mu	μ	alt + 0956	Μ	Greek capital Mu	Μ	alt + 0924
ν	Greek small Nu	ν	alt + 0957	Ν	Greek capital Nu	Ν	alt + 0925
ξ	Greek small Xi	ξ	alt + 0958	Ξ	Greek capital Xi	Ξ	alt + 0926
ο	Greek small Omicron	ο	alt + 0959	Ο	Greek capital Omicron	Ο	alt + 0927
π	Greek small Pi	π	alt + 0960	Π	Greek capital Pi	Π	alt + 0928
ρ	Greek small Rho	ρ	alt + 0961	Ρ	Greek capital Rho	Ρ	alt + 0929
ς	Greek small Sigma 1	ς	alt + 0962	Σ	Greek capital Sigma	Σ	alt + 0931
σ	Greek small Sigma 2	σ	alt + 0963				
τ	Greek small Tau	τ	alt + 0964	Τ	Greek capital Tau	Τ	alt + 0932
υ	Greek small Upsilon	υ	alt + 0965	Υ	Greek capital Upsilon	Υ	alt + 0933
φ	Greek small Phi	φ	alt + 0966	Φ	Greek capital Phi	Φ	alt + 0934
χ	Greek small Chi	χ	alt + 0967	Χ	Greek capital Chi	Χ	alt + 0935
ψ	Greek small Psi	ψ	alt + 0968	Ψ	Greek capital Psi	Ψ	alt + 0936
Ω	Greek small Omega	ω	alt + 0969	Ω	Greek capital Omega	Ω	alt + 0937

Table 12.17 Table of alt codes, Unicode characters and HTML equivalent characters

12.7 Colour pallets

Error! Reference source not found.

THEME COLOURS									
FFFFFF	000000	E7E6E6	44546A	4472C4	ED7D31	A5A5A5	FFC000	5B9BD5	70AD47
F2F2F2	7F7F7F	D0CECE	D6DCE4	D9E2F3	FBE5D5	EDEDED	FFF2CC	DEEBF6	E2EFD9
D8D8D8	595959	AEABAB	ADB9CA	B4C6E7	F7CBAC	DBDBDB	FEE599	BDD7EE	C5E0B3
BFBFBF	3F3F3F	757070	8496B0	8EAADB	F4B183	C9C9C9	FFD965	9CC3E5	A8D08D
A5A5A5	262626	3A3838	323F4F	2F5496	C55A11	7B7B7B	BF9000	2E75B5	538135
7F7F7F	0C0C0C	171616	222A35	1F3864	833C0B	525252	7F6000	1E4E79	375623
STANDARD COLOURS									
C00000	FF0000	FFC000	FFFF00	92D050	00B050	00B0F0	0070C0	002060	7030A0

Table 12.18 Office 2021 standard colour pallet

Error! Reference source not found.

THEME COLOURS									
FFFFFF	000000	E7E6E6	1F497D	4F81BD	C0504D	9BBB59	8064A2	4BACC6	F79646
F2F2F2	7F7F7F	DDD9C3	C6D9F0	DBE5F1	F2DCDB	EBF1DD	E5E0EC	DBEEF3	FDEADA
D8D8D8	595959	C4BD97	8DB3E2	B8CCE4	E5B9B7	D7E3BC	CCC1D9	B7DDE8	FBD5B5
BFBFBF	3F3F3F	938953	548DD4	95B3D7	D99694	C3D69B	B2A1C7	92CDDC	FAC08F
A5A5A5	262626	494429	17365D	366092	953734	76923C	5F497A	31859B	E36C09
7F7F7F	0C0C0C	171616	0F243E	244061	632423	4F6128	3F3151	205867	974806
STANDARD COLOURS									
C00000	FF0000	FFC000	FFFF00	92D050	00B050	00B0F0	0070C0	002060	7030A0

Table 12.19 Office 2007-2010 standard colour pallet

Badge colours

Badge colours

COLOUR	DESCRIPTION	MEANING
BF504D	Red	Development
E60D18	Bright red	Work in progress (proof of concept)
F79545	Orange	*Not allocated*
FF6C09	Bright orange	Highlight colour (sometime used in images)
9DBB61	Green	*Not allocated*
00B050	Bright green	Released
4BABC5	Turquoise	*Not allocated*
4F81BC	Blue	Published
AB99C0	Violet	Under test
3F3F30	Grey	Development (alternative)

Table 12.20 Badge colours

12.8 Helpful notes

12.8 Helpful notes

Highlight colours

Highlight colours

On item selection in images — the orange highlight colour is RGB 255,108,009 (#FF6C09) for both the outline and the fill, the fill uses the secondary colour of Paint.Net and this has a transparency of 100.(#FF6C0964)

The border for rectangles is 2 pixels with anti-aliasing active. Font size in paint.net for highlights is generally 14-point.

Removing text background colours

Removing text background colours

Sometimes copying code fragments from text editors leaves a light grey background colour applied to the text (not the cell); I don't know why, but here's how you get rid of it.

Select the affected cells right click and select BORDERS AND SHADING. Select the SHADING tab and change the fill box to NO COLOUR; in the APPLY TO box select TEXT (not cell). This removes the colour from the text, but not the background from the cell.

Configuring footnotes

Configuring footnotes

First switch to draft view, on the ribbon select: VIEW tab, VIEWS group, DRAFT. Now display footnotes, again on the ribbon select REFERENCES tab, FOOTNOTES group, SHOW NOTES.

To change the footnote separator, click the dropdown where it says ALL FOOTNOTES: and select FOOTNOTE SEPARATOR, this will now show the footnote separator.

GxP blank page

GxP blank page

The following page is a GxP compliant blank page, with the text "intentionally left blank". It does not need a section break (it has the same footer arrangements as normal).

99 Typicals and examples

The first section on a new web page must contain enough text to accommodate the left side bar table of contents.

This can require multiple sections, subsections and inline sections to be incorporated into the first <section> element of the page.

These are referred to as supplementary sections, subsections and inline sections. An example of each is included below.

This is a complete section with left side bar

PAGE INTENTIONALLY LEFT BLANK

Appendices

APPENDICES

PAGE INTENTIONALLY LEFT BLANK

A An empty Appendix

This appendix contains various typical components and pages that can be used in different types of documents (particularly engineering documents).

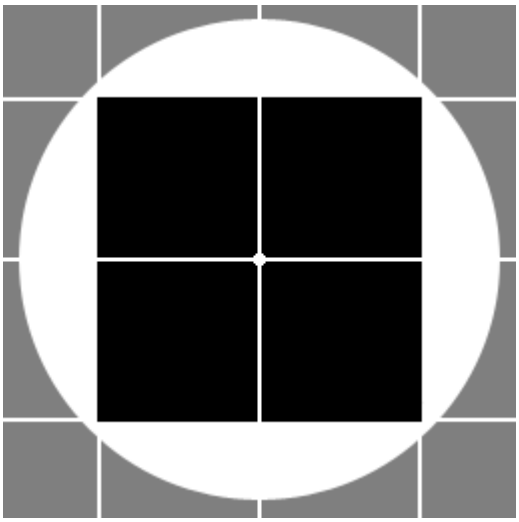


Figure A.1 Figure Caption

This is Figure A.1.

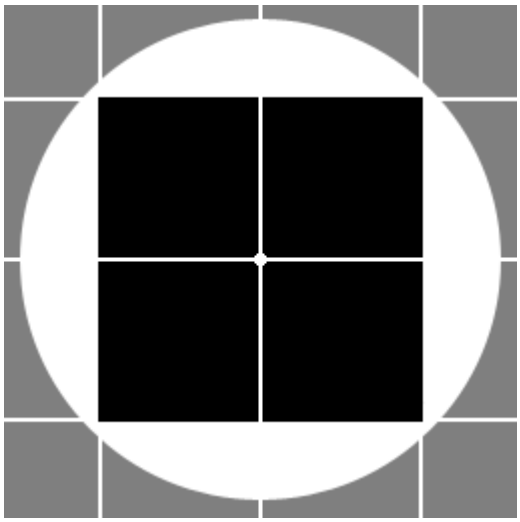


Figure A.2 Figure Caption

This is Figure A.2.

HEADING	HEADING	HEADING

Table A.1 Table Caption

This is Table A.1

HEADING	HEADING	HEADING

Table A.2 Table Caption

This is Table A.2

Equation area (A.1)

This is Equation (A.1).

A.1 Appendix headings

A Appendix chapter

A Appendix chapter

A.1 Appendix section

A.1 Appendix section

A.1.1 Appendix subsection

A.1.1 Appendix subsection (inline)