



INTEGRITI PROGRAMMING EXAMPLES



**INNER RANGE recommends that all Inner Range systems
be installed & maintained by FACTORY CERTIFIED
TECHNICIANS.**

**For a list of Accredited Dealers in your area refer to the
Inner Range Website.**

<http://www.innerrange.com>

Examples

This document contains programming examples to help get a better understanding of how to fully utilise the Integriti controller.

Example 1

Flashing auxiliaries during a time period

This example demonstrates the use of macros in conjunction with a few entities. This example will toggle an auxiliary once every 5 seconds. Every fifth toggle, another auxiliary will turn on for 10 seconds. This macro will only run while a time period is valid.

Entities used:

- 1x General variable
- 1x Macro
- 1x Time period
- 2x Auxiliaries

Entity configuration:

In this example we have created a time period called 'Working Hours'. This time period has been configured for 09:00 to 17:30 Monday to Friday.

The general variable used has been called 'Example Counter'. The test value for the general variable has been set to 4.

The two auxiliaries (C01:X01 and C01:X02) have been labelled 'Flasher 1' and 'Flasher 2'.

Macro statements:

A total of eight statements are required to achieve this goal:
Macro: **Example 1**

#	Statement	Configuration	
1	Wait for Condition...	Expression:	TP1
2	Do an Action	Action:	Control Aux
		Auxiliary:	C01:X01
		When Asserted:	Toggle
3	Pause for Time...	Expression:	50

4	Set Entity To Expression...	Expression:	GV1+1
		Entity to Set:	GV1
5	Goto <label> if...	Expression:	!GV1
		Label:	SkipOn
6	Do an Action	Action:	Control Aux
		Auxiliary:	C01:X02
		On Time:	00:00:10
7	Set Entity To Expression...	Expression:	0
		Entity to Set:	GV1
8	Define a Label	Label:	SkipOn

Statement summary for macro: **Example 1**

- Prevent further execution of the macro until the time period (TP1) is valid.
 - TP1 (Work Hours) has been configured for 09:00 to 17:30 hours Monday to Friday. If the panel time and day fall within these values then the test of time period TP1 will return true and remainder of the macro will be executed.
- This statement toggles the first auxiliary (Flasher 1).
 - C01:X01 (Flasher 1) has been to toggle on assert. Every time this statement is executed the state of the auxiliary will be inverted.
- Pause for 50 x 100ms (5 seconds).
 - The macro will pause for a period of 5 seconds.
- Add 1 to the general variable (GV1).
 - The general variable GV1 is set to a value that is itself plus one.
- If the test of general variable (GV1) does not return true, go to the label 'SkipOn'.
 - When the general variable (GV1) was configured, its test value was set to four. If the general variable is greater than this value, it will return true.
 - The expression '!GV1' is a test to see if GV1 is not returning true (If the value is equal to four or less).
 - If the expression returns true, the following two statements will be skipped.
- Turn on the second auxiliary (Flasher 2) for 10 seconds.
 - If an on time other than zero is specified for an auxiliary action, the target auxiliary will turn off after the time expires.
- Reset the general variable (GV1) back to 0.
 - The general variable GV1 is set to zero.
- This is the label (SkipOn) that the macro moves execution to if the general variable (GV1) does not return true (GV1 is greater than 4).
- Return to step 1.

Example 2

Random bag inspections

During work hours random bag inspections are carried out. Users going through door x will be denied access at random. An input or users with special access can be used to reset the random bag inspection and allow the user to pass through.

Entities used:

- 1x Door
- 1x General variable
- 1x Input
- 1x Named Action
- 1x Reader module
- 1x Time period
- 2x Macros
- 2x Permission Groups
- 2x Users

Entity configuration:

In this example we have configured a door (Door x) as a normal entry door associated with a reader module (Reader x).

We have two users - User 'Employee' is subject to random bag inspections. The 'Supervisor' user is not. We will name the two permission groups 'Employees' and 'Supervisors' respectively. Each user has its own credential.

We have called the time period 'Working Hours'. This time period has been configured for 09:00 to 17:30 Monday to Friday.

The general variable 'Random inspection' is used to determine user access. It is assigned a random number using the first macro 'Random bag' and reset to zero by the other macro 'Reset random bag'.

The named action 'Start random bag' is triggered by 'Door x' but only when working hours are valid. 'Start random bag' will start the macro 'Random bag' every time 'Door x' is opened.

The Input 'Reset bag' is used by the supervisor on duty. Pressing the button attached to the input 'Reset bag' will start the macro 'Reset random bag'.

Macro statements:

Both macros are small (only containing two statements each):

Macro: **Random bag**

#	Statement	Configuration
1	Set Entity To Expression...	Expression: 8388607 Entity to Set: GV2
2	End Current Macro	

Macro: **Reset random bag**

#	Statement	Configuration
1	Set Entity To Expression...	Expression: 0 Entity to Set: GV2
2	End Current Macro	

Statement summary for macro: **Random bag**

1. The entity GV2 'Random inspection' is assigned a value of 8388607.
 - The number 8388607 is a magic number. Entities assigned this number are actually given a random number between 1 and 8388067. For more information, please refer to the appendices.
2. Stop the macro.
 - This statement will cause the macro to stop.

Statement summary for macro: **Reset random bag**

1. The entity GV2 'Random inspection' is assigned a value of 0.
2. Stop the macro.
 - This statement will cause the macro to stop.

