MICROSENS

Application Note

Leveraging General Purpose I/O Ports with microScript

MICROSENS GmbH & Co. KG
Küferstr. 16
59067 Hamm/Germany

Tel. +49 2381 9452-0
FAX +49 2381 9452-100
E-Mail info@microsens.de
Web www.microsens.de

Summary

The feature set of MICROSENS G6 Industrial Switches (PL+, PLM) includes two general purpose I/O ports. Signal states at the input ports can be used to trigger predefined actions on the output ports.

The MICROSENS script language microScript allows users to expand the functionality of their MICROSENS switches through individual scripts. This application note offers an easy entry into creation and usage of microScript based scripts. This objective is achieved by explaining some documented script examples of the handling of state changes at the I/O ports.

Table of Contents

SUMMARY 2
TABLE OF CONTENTS 2
1 INTRODUCTION 3
1.1 Understanding the general purpose I/O ports 3
1.1.1 Alarm Input Ports (Signal In) 4
1.1.2 Alarm Output Ports (Signal Out) 5
1.2 MICROSENS microScript 6
2 WORKING WITH MICROSENS MICROSCRIPT 7
2.1 Create a microScript file 7
2.1.1 Example 1 8
2.1.2 Example 2 9
2.2 Upload the script file to the switch 10
2.3 Assign the script to the respective event 11
2.4 Hints for advanced CLI users 12
DISCLAIMER 13

1 Introduction

1.1 Understanding the general purpose I/O ports

The input and output ports *Signal In* and *Signal Out* (a. k. a. *Alarm Inputs*, *Alarm Outputs*, Figure 1) are electrically designed for maximum compatibility with varying installation scenarios. Input and output contacts are electrically isolated, allowing connection to external circuits within different reference voltage ranges (Figure 2).

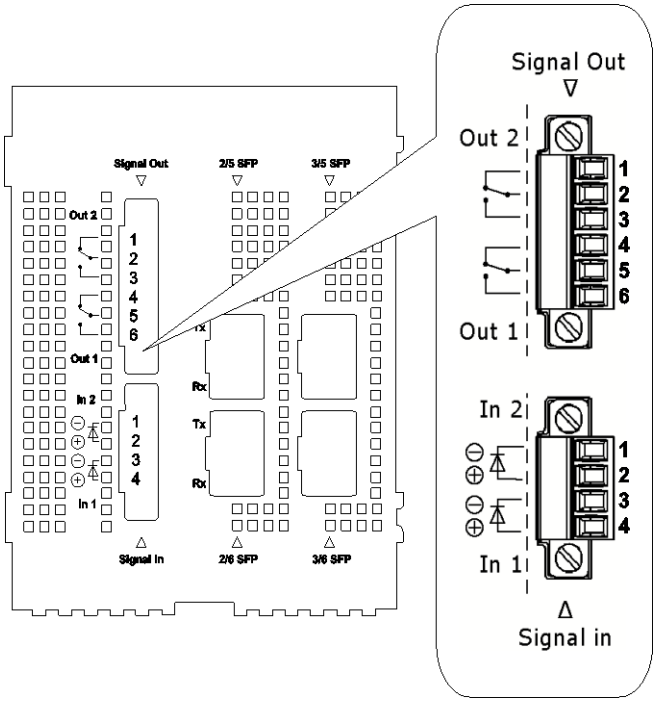


Figure 1: Arrangement of the I/O contacts at the bottom of the switch case

The **Input Ports** are specified as:

- 2x 2-pole monitoring ports, screw connection
- potential-free input, unshielded
- for connecting signal sources via opto-coupler (cable length: max. 3 m)

The **Output Ports** are specified as:

- 2x 3-pole relay connector
- potential-free output, screw connection, unshielded
- for connecting external monitoring devices (cable length: max. 3 m)

Note:
For a complete description of I/O port positions and LED signal codes on the front panel please refer to the Quick Installation Guide delivered with the respective MICROSENS G6 Industrial Switch.

1.1.1 Alarm Input Ports (Signal In)

The contacts at the Alarm Input Ports require active signaling. A simple external circuit may be required as shown in the schematic.

Physically, the Alarm Input ports consist of two opto-couplers which are used to decouple the external circuit from the internally used operating power.

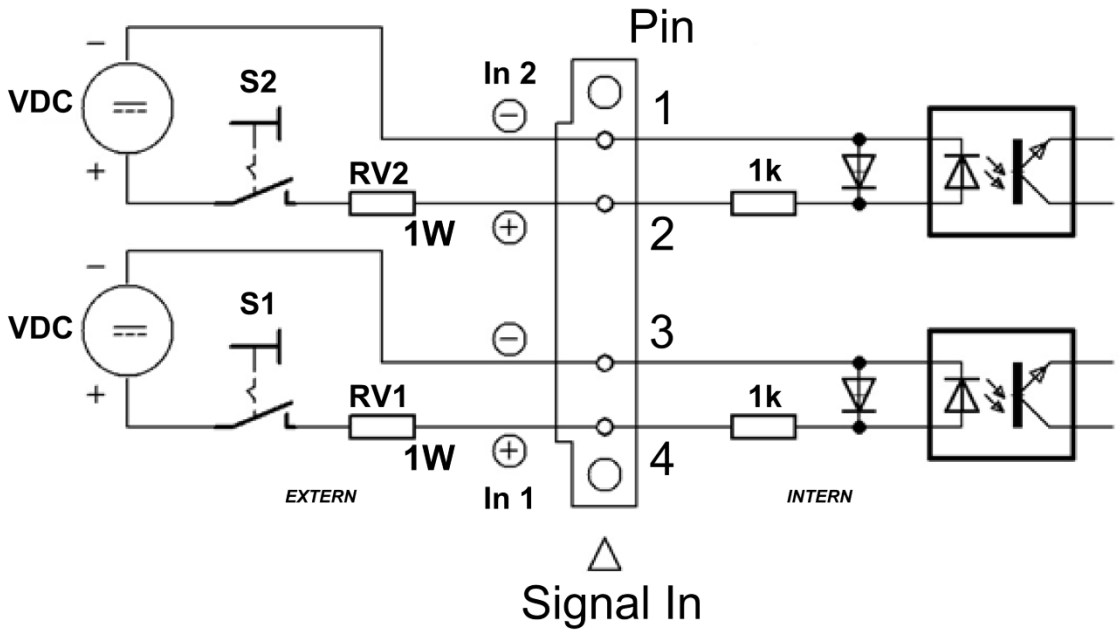


Figure 2: Proper wiring of the Alarm Input Ports

An opto-coupler contains an LED whose light emission is used to control a transistor. An LED must be operated with DC power. To limit the electrical current flow through the LED, each LED requires a series resistor **RV***n*. The resistors value depends on the voltage level applied (Figure 2).

Select the value of the resistors RV1 and RV2 according to the following table:

VDC Range	**3-12V**	**12-15V**	**15-24V**	**24-36V**	**36-48V**	**48-59V**
RV1, RV2	0 Ohm	300 Ohm	1200 Ohm	2400 Ohm	3600 Ohm	4700 Ohm
	Resistor power rating: ≥ 1W					

Table 1: Relationship between range of DC input voltage and resistor values

ATTENTION: Max. rating = **57VDC / 25mA**

Note:
For each input port it is possible to (pre-)define whether the alarm condition is a physically low or high voltage (see also section "2.4 Hints for advanced CLI users"). Due to the respective configuration either a 'High' or a 'Low' input signal can cause an alarm. Therefore, an input signal 'High' does not necessarily mean that the logical status at the Input Port is an 'active alarm state' ('Active Alarm State' = True).

Upon change of an input signal, an internal event is generated. This event may be used to trigger a microScript as explained in the following section.

Thus it is possible to start any user defined action via the input contacts. This may include special trap handling, shutdown of certain functions to safeguard against manipulation (door contact) or perhaps to switch to lower port speed.

For *Signal In*, there are 2 LEDs on the front panel, each is associated with one input.

Color	Indication
Off	Caption of Input Signals is disabled
Green	Input is logically in normal condition.
Red (blinking)	Input is logically in alarm condition Blinking is used when the signal_mode is configured to any of the two blink modes.

Table 2: Interpretation of Front Panel LED colours related to Signal In

1.1.2 Alarm Output Ports (Signal Out)

Physically, the Alarm Output Ports consist of two relays under software control. For safety reason the relays are designed to only be operated under low voltage.

There is a *normally closed* and a *normally open* contact and a *common contact* for each relay (Figure 1). The following tables define the logical and the electrical behaviour of both output ports:

Logical State	Signal Out 1	Signal Out 2
DISABLED, OFF	4-5 (short) 5-6 (open)	1-2 (short) 2-3 (open)
ON	4-5 (open) 5-6 (short)	1-2 (open) 2-3 (short)

Table 3: Relationship between logical state and electrical behavior of Signal Out

ATTENTION: Max. rating = **57 AC or DC / 1A**.

Just like the input pins the alarm relays operation can be configured by script.

The relays of the Alarm Output Ports can be configured to switch under the following conditions (see also section "2.4 Hints for advanced CLI users"):

- If operating power is present.
- If a redundant power supply fails.
- If the units temperature behaviour impends to exceed a limit.
- Manually *On* or *Off*.
- Under script/app control (user programmable).

The relays may be set to blink mode. In this case the relays will, when an error condition needs to be indicated, blink with approx 1Hz.

There are 2 LEDs on the front panel, each associated with one relay.

Color	Indication
Off	Output is disabled
Green	Output is in normal condition. The configured alarm condition is not present. **Note:** When relay is configured to "ON_WHILE_RUNNING" the LED will indicate green, even though the relay is actually activated, since this is not an error condition.
Red (blinking)	Output is in alarm condition. Blinking is used when the "signal_mode" is configured to any of the two blink modes.

Table 4: Interpretation of Front Panel LEDs colours related to Signal Out

1.2 MICROSENS microScript

The script language MICROSENS microScript enables the user to expand the switch functionality according to his demands without changing the systems firmware.

Basically a simple script can store a value for every parameter of the switch directly. Thus it is possible to use a script file to back up a switches configuration or to configure several identical switches with the same parameter settings easily.

With using variables and conditional expressions it is possible to realize comprehensive and flexible configurations, i.e., depending on the I/O ports input states.

Note:
Users with no or little experience in using the command line interface are strongly recommended to use the switches web interface for script management. The next chapter shows how to use scripts with the Web Manager primarily and only gives short references to the respective CLI parameters.

2 Working with MICROSENS microScript

Utilization of microScript files follows these steps:

1. Create a microScript file
2. Upload the script file to the switch
3. Assign the script to the respective event
4. Configure the events

2.1 Create a microScript file

MICROSENS microScript files are simple text files which can be created and edited with a standard text editor (e.g. Notepad).

Note:
Script files use UNIX-style line termination (LF). When editing and downloading scripts from a Windows environment, please use an editor program or tool to convert Windows-style line termination (CR+LF) to UNIX compatible line termination.

Save these script files locally either as a microScript file with the extension `.ms` or as a simple text file with the extension `.txt`. MICROSENS switches are able to cope with both file types.

2.1.1 Example 1

The following script activates the PoE function on port 2/1 when input 1 reports itself "active". Line numbers only serve as information and are not to be used in the script.

```
1    Device.Hardware.io_signal_status.input_1_alarm_active %
2    :if V0 = True
3    :   print "Input 1 is active -> turning on output 1"
4    Device.POE.config[2/1].mode = AUTOMATIC
5    :else
6    :   print "Input 1 is off -> turning off output 1"
7    Device.POE.config[2/1].mode = DISABLED
8    :endif
```

- **`1  Device.Hardware.io_signal_status.input_1_alarm_active %`**
  The script reads the state of input 1. This value either contains the state "True" for an active port or "False" for an inactive port.

- **`2  :if V0 = True`**
  If input 1 is active due to a prior signal change it contains "True" and the script continues on line 3.
  If input 1 is not active due to a prior signal change it contains "False" and the script continues on line 5.

- **`3  :  print "Input 1 is active -> turning on output 1"`**
  The script displays the message "Input 1 is active -> turning on output 1" on the screen.

- **`4  Device.POE.config[2/1].mode = AUTOMATIC`**
  The script sets the PoE parameter on port 2/1 to "AUTOMATIC". Now this port supplies connected devices over PoE. The script continues on line 8.

- **`5  :else`**
  With an inactive input 1 (state "False") the script skips lines 3 and 4 and continues on line 6.

- **`6  :  print "Input 1 is off -> turning off output 1"`**
  The script displays the message "Input 1 is off -> turning off output 1" on the screen.

- **`7  Device.POE.config[2/1].mode = DISABLED`**
  The script sets the PoE parameter on port 2/1 to "DISABLED". Now this port does not supply connected devices over PoE anymore.

- **`8  :endif`**
  End of script.

Note:
Please keep in mind that due to the respective configuration either a high or a low input signal can cause an alarm. Therefore an input signal "high" does not mean necessarily the input is in an active alarm state (state = True).

2.1.2 Example 2

The following script configures the IP address of the switch depending on the signal levels of input 1 and 2. Line numbers only serve as information and are not to be used in the script.

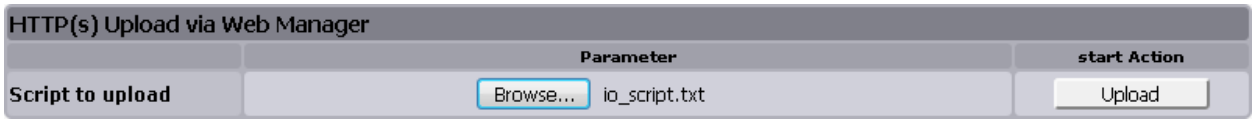
```
1    :var $S1
2    :var $S2
3    Device.Hardware.io_signal_status.input_1_alarm_active %
4    :set $S1=V0
5    Device.Hardware.io_signal_status.input_2_alarm_active %
6    :set $S2=V0
7    :if $S1 = True
8    :   if $S2 = False
9          Device.IP.v4_config.dhcp_mode = DISABLED
10         Device.IP.v4 config.static device ip = 192.168.40.244
11   :   endif
12   :   if $S2 = True
13         Device.IP.v4_config.dhcp_mode = DISABLED
14         Device.IP.v4_config.static_device_ip = 192.168.50.244
15   :   endif
16   :endif
17   :if $S1 = False
18   :   if $S2 = False
19         Device.IP.v4_config.dhcp_mode = DISABLED
20         Device.IP.v4_config.static_device_ip = 192.168.20.244
21   :   endif
22   :   if $S2 = True
23         Device.IP.v4_config.dhcp_mode = DISABLED
24         Device.IP.v4_config.static_device_ip = 192.168.30.244
25   :   endif
26   :endif
```

- **1 :var $S1**
  2 :var $S2
  The script defines two variables "$S1" and "$S2".
- **3 Device.Hardware.io_signal_status.input_1_alarm_active %**
  4 :set $S1=V0
  5 Device.Hardware.io_signal_status.input_2_alarm_active %
  6 :set $S2=V0
  It reads the states of the input ports and assigns the specific states ("True" or "False") to the respective variable "$S1" and "$S2".
- **7 :if $S1 = True**
  8 : if $S2 = False
  9 Device.IP.v4_config.dhcp_mode = DISABLED
  10 Device.IP.v4_config.static_device_ip = 192.168.40.244
  11 : endif
  12 : if $S2 = True
  13 Device.IP.v4_config.dhcp_mode = DISABLED
  14 Device.IP.v4_config.static_device_ip = 192.168.50.244
  15 : endif
  16 :endif

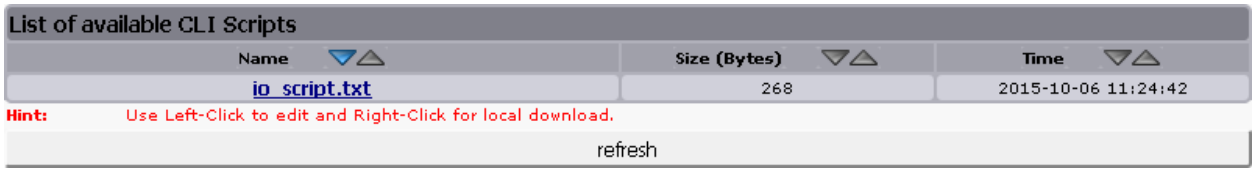
In nested conditional expressions the script assigns a specific IP address to the switch depending on the respective input signals. The DHCP mode is disabled on all possible input states. Lines 17 to 26 correspond to lines 7 to 16.

2.2 Upload the script file to the switch

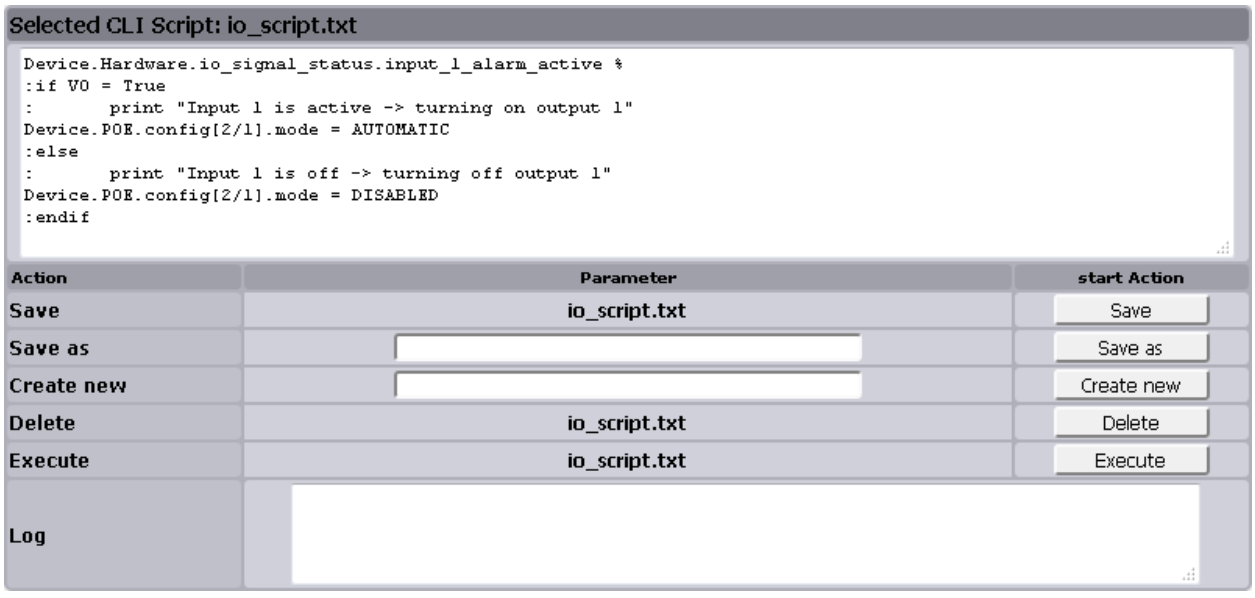
1. Start the Web Manager and open the menu "Scripting" and change to the tab "CLI scripts".
2. In the section "HTTP(S) Upload via Web Manager" click on "Browse" and choose the respective microScript file ("<script_name>.ms" or "<script_name>.txt"). Confirm the dialog with "Open". The file name appears in the parameter column.



3. Click on "Upload" to upload the script file into the switch. The scripts file name appears in section "List of available CLI Scripts".



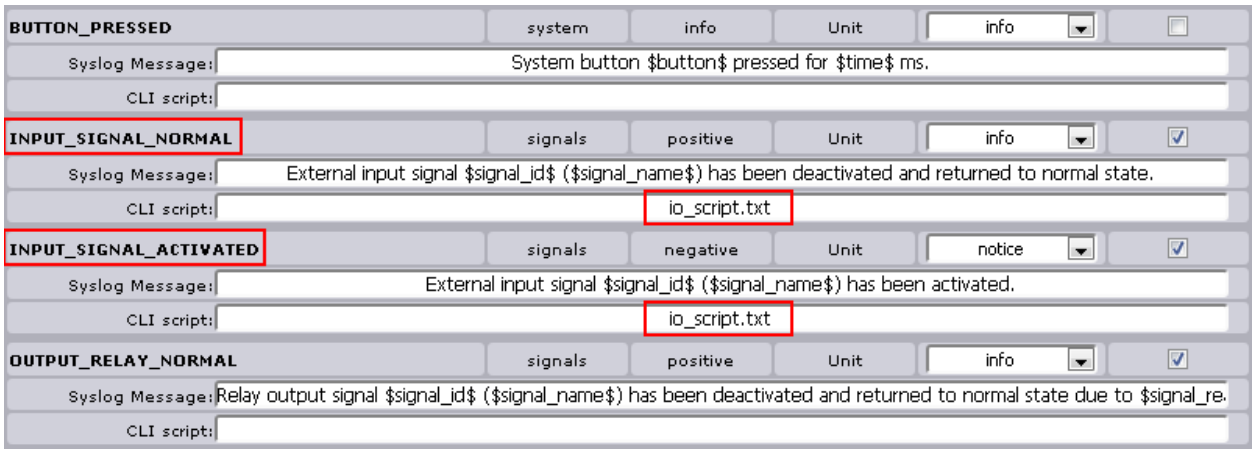
Note:
To edit the script with Web Manager click on it with the left mouse button for loading it into the editor section "Selected CLI Script: <file name>".



2.3 Assign the script to the respective event

To make specific events trigger the script, you have to assign these events to the script.

1. Open the menu "Events" and scroll down to the entries "INPUT_SIGNAL_NORMAL" and "INPUT_SIGNAL_ACTIVATED"
2. In the field "CLI script" enter the file name of the script file you have uploaded before.



The appropriate CLI commands are:

- **`Management.Event.event_list[INPUT_SIGNAL_NORMAL].cli_script io_script.txt`**
- **`Management.Event.event_list[INPUT_SIGNAL_ACTIVATED].cli_script io_script.txt`**

3. Click "apply to running configuration" to finish the appliance of a MICROSENS microScript. A signal change on an input port now triggers the respective script.

2.4 Hints for advanced CLI users

The value of the **input signal** can be monitored via the management system.

- **`device.hardware.io_signal_status.`**

For each input it is possible to define whether the alarm condition is physically a low or a high voltage. Upon change of an input signal, an internal event is generated. This event can be forwarded via SNMP or Syslog to any desired manager. Additionally, each signal change may be used to trigger a script/app.Thus it is possible to start any user defined action through the input contacts. This may include special trap handling, shutdown of certain functions to saveguard against manipulation (door contact) or perhaps to switch to lower port speed.

The relays of the **Alarm Output Port** can be configured to switch under various conditions For details please refer to section "1.1.2 Alarm Output Ports (Signal Out)" and to the parameter description following

- **`device.hardware.io_signal_config.`**

Disclaimer

Document ID: AN-15002_2015-11-05