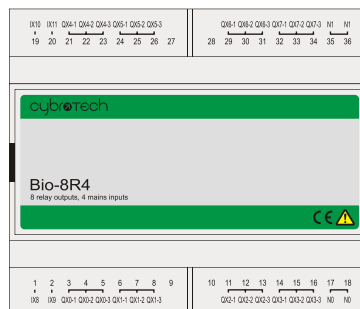


Bio-8R4

Fallback Functionality

TN-015



cybrotech

© 2008 Cybrotech Ltd

Cybrotech Ltd
14 Brinell Way, Harfreys Industrial Estate
Great Yarmouth, Norfolk, Nr31 OLU - UK
tel: +44 (0)1493 650 222
www.cybrotech.co.uk
info@cybrotech.co.uk

Copyright © 2008 by Cybrotech, Ltd.
Data subject to change without notice
Publication code: PBL-TN015-EN-10-2008

The logo for Cybrotech, featuring the word "cybrotech" in a lowercase, sans-serif font. The "y" is green, and the "o" is a green circle with a white dot inside, resembling a stylized eye or a target. The rest of the letters are black.

1 Introduction

Bio-8R4 modules (software version 4.0 and later) now support a new fallback feature. This feature enables limited control of Bio-8R4 relay outputs in case communication failure between the device and the CyBro-2 controller.

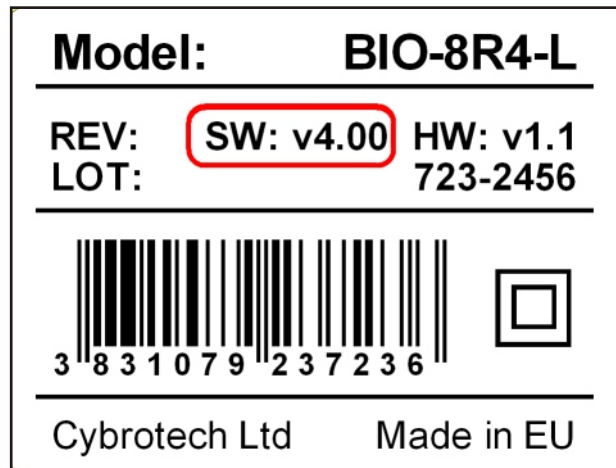


Figure 1: Label can be found on the side of the device

2 Normal operation

Fallback functionality and settings do not influence the operation of Bio-8R4 while no communication failure is detected. Module is controlled as programmed in the application in CyBro-2 controller.

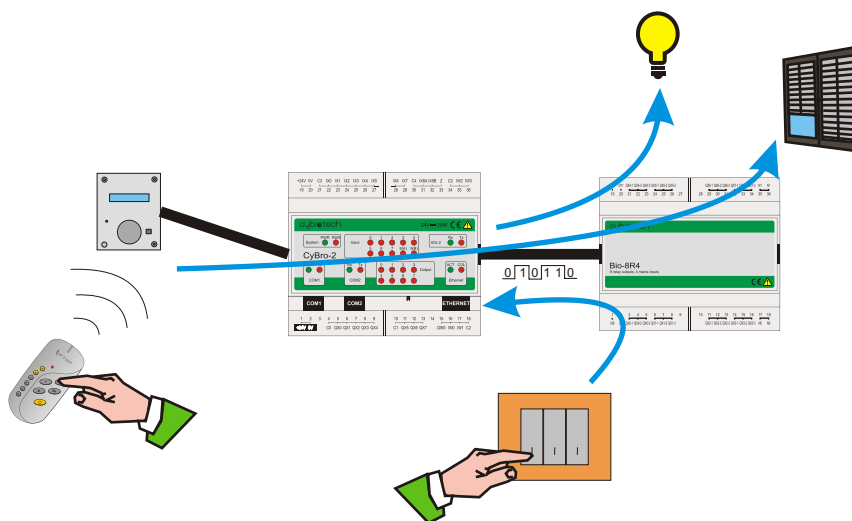


Figure 2: System in normal operation

3 Fallback operation

Should the communication between Bio-8R4 module and CyBro-2 controller be lost, the module will automatically change to fallback mode. While in this mode, module's outputs are controlled directly by its inputs. When communication to the controller is restored Bio-8R4 returns to normal operation.

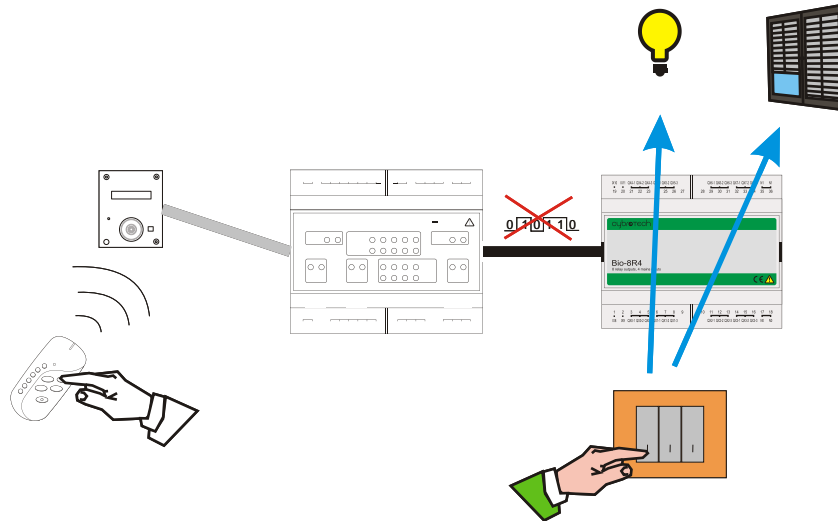


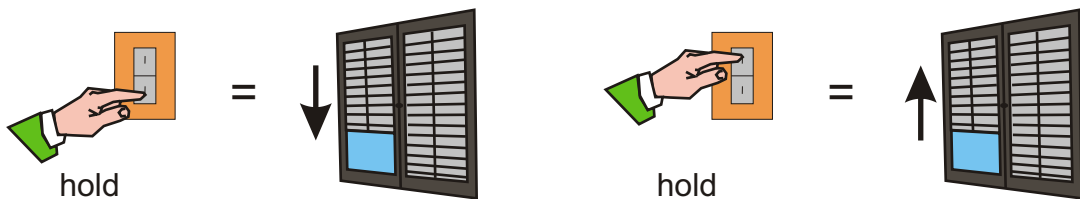
Figure 3: System in fallback operation

3.1 Configuring fallback operation

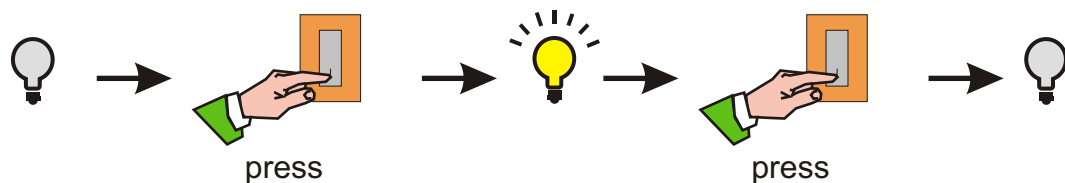
Bio-8R4 has four voltage sense inputs to which keys can be connected. In fallback mode, these inputs can be configured to control module's output relay.

Two ways of control can be set:

A) Direct



B) Toggle



At the time of hardware detection, fallback functionality can be configured by double-clicking on the hardware slot that contains Bio-8R4 module.

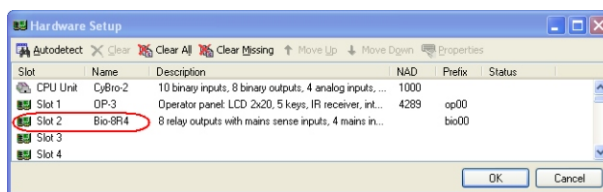


Figure 4: Hardware setup

A new window will open with where the configuring values are inserted.

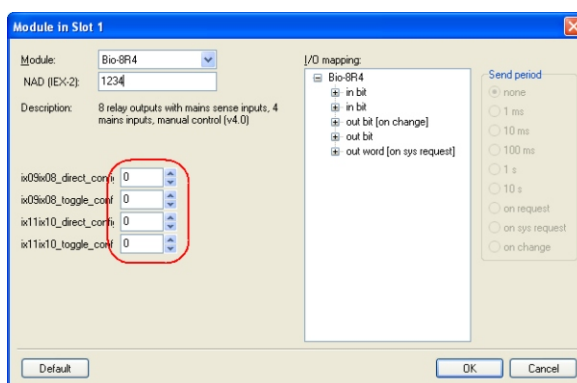


Figure 5: Module configuration

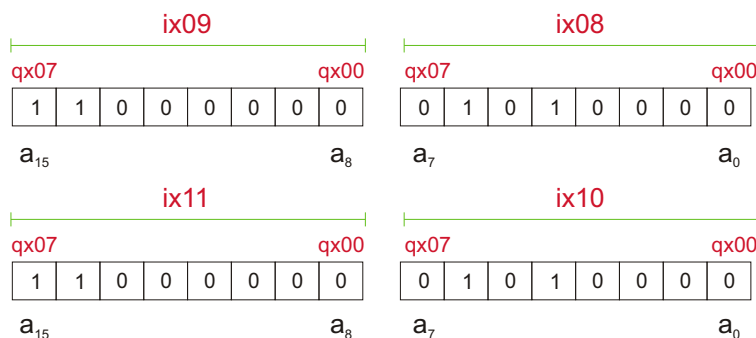
```
bio00_ix09ix08_direct_config: INT;
bio00_ix09ix08_toggle_config: INT;
bio00_ix11ix10_direct_config: INT;
bio00_ix11ix10_toggle_config: INT;
bio00_sys_req: BIT;
```

These settings are sent automatically to the module on controller power-up and STOP-to-RUN transition or by setting system request flag.

3.1.1 Calculating config values

Config values must be input in an signed integer format. High bytes correspond to the ix09 and ix11 inputs, low bytes are for ix08 and ix10.

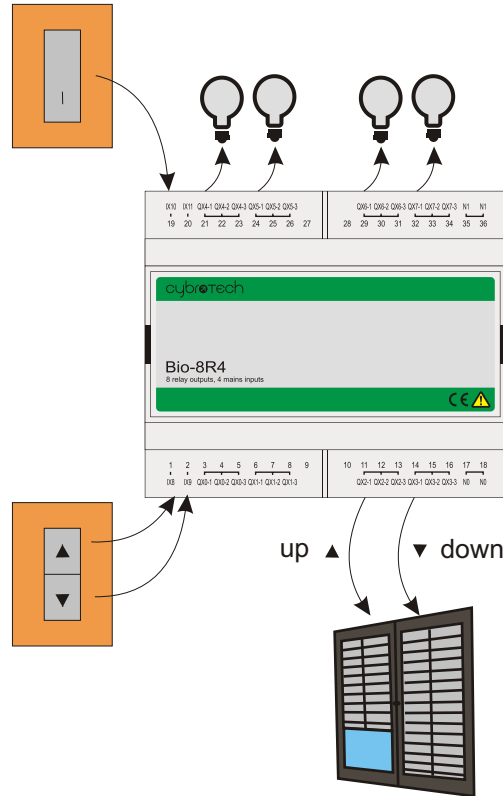
Linking an input to an output is done by setting the corresponding bit value to 1.



$$\text{config value} = -a_{15} * 2^{15} + a_{14} * 2^{14} + a_{13} * 2^{13} + \dots + a_2 * 2^2 + a_1 * 2^1 + a_0 * 2^0$$

4 Example

Four lights, a blind, a key and a two-way key is connected to our Bio-8R4. The goal is to configure the module to enable a limited control in case of communication failure.



- The two-way key must be able to control the movement of the blinds. While UP key is held the blinds must open and while DOWN key is held the blinds must close.
- The single key must toggle the state of both lights each time it is pressed.

Since keys UP and DOWN activate outputs only while they are held, the **direct_config** value must be used for inputs **ix08** and **ix09**. Lights must be toggled by the key press and therefore the **toggle_config** value must be set for input **ix10**.

	ix09								ix08								
ix09ix08_direct_config	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
ix09ix08_toggle_config	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ix11								ix10								
ix11ix10_direct_config	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ix11ix10_toggle_config	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0

ix09ix08_direct_config = 0000 1000 0000 0100b = 0804h = 2052
 ix09ix08_toggle_config = 0
 ix11ix10_direct_config = 0
 ix11ix10_toggle_config = 0000 0000 1111 0000b = 00F0h = 240