`

VERSION 2.6

OCTOBER 27, 2022

IOFIREBUG ENGINE FOR UNIVERSAL ROBOTS

INSTRUCTION MANUAL

4EACH S.R.O.
WWW.4EACH.CZ

TABLE OF CONTENTS

Notes on the documentation

INTENDED AUDIENCE

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.

The following notes and explanations must be followed when installing and commissioning these components.

The responsible staff must ensure that the application or use of the products described satisfies all the requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

DISCLAIMER

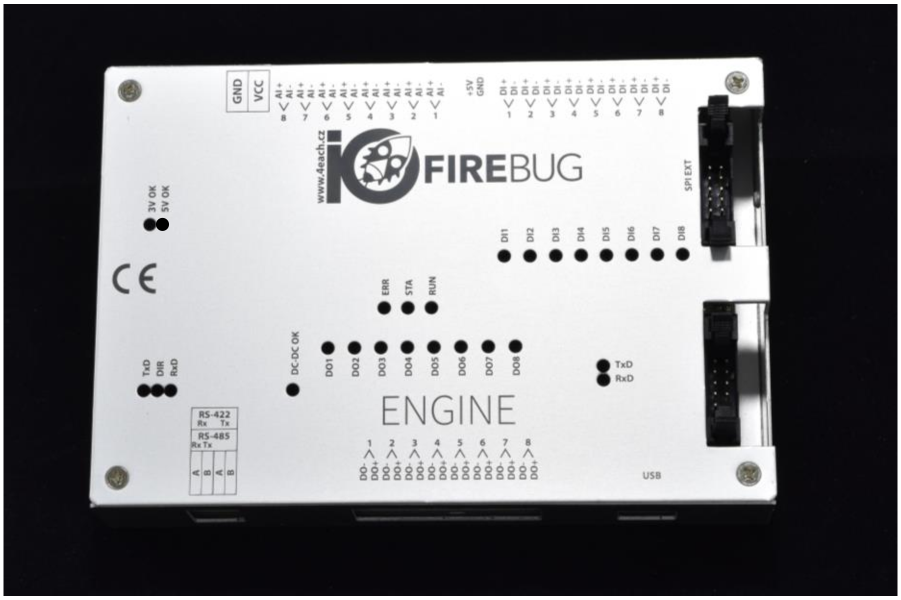
The documentation has been prepared with care. The product described is, however, constantly under development. For that reason, the documentation is not in every case checked for consistency with performance data, standards, or other characteristics. If it contains technical or editorial errors, we retain the right to make alterations at any time and without warning. No claims for the modification of products that have already been supplied may be made based on the data, diagrams, and descriptions in this documentation.

COPYRIGHT

Product overview

IOFireBug Engine block diagram is shown in the following image. The device is equipped with the following interfaces:

- 8x analog input
- 8x binary input (galvanically isolated)
- 8x switching N-FET output (galvanically isolated, all channels have common ground)
- RS485 / RS422 Interface (switchable, galvanically isolated)
- USB to Serial Interface (FTDI)
- SPI Interface for optional expanders
- I2C Intentionally unused



POWER SUPPLY

The device can be powered using a DC power supply, using 10V to 30V.

Power LED indication:

LED	**3V OK**	**5V OK**	**DC-DC OK**
Meaning	Voltage present 3,3V from the power supply	Voltage present 5,0V from the power supply	Voltage present 5,0V from the galvanically isolated power supply
Color	Red	Red	Red

If everything is ok, all three LEDs shine continuously.

USB TO COMPUTER INTERFACE

The device is equipped with a USB interface for direct communication with the robot control box unit. The status of device communication is indicated by LEDs near the USB connector.

Communication status LED:

LED	**TxD**	**RxD**
Meaning	TxD (unit is sending data)	RxD (unit is receiving data)
Color	Green	Red

RS-422 INTERFACE

IOFirebug Engine for Universal Robots supports the RS-422 communication interface. Robot CB can communicate with IOFirebug using an additional USB to RS-422 converter (ie. SB485s from papouch.com). Both interfaces (USB and RS-422) are completely interchangeable but make sure to use only one.

RS-422 LINE TERMINATION:

Using a jumper on P33, P34, P35, and P36 ports terminate the RS422 line (part of delivery). Please ensure, that these jumpers are inserted before connecting the RS-422 interface.

RS-485 COMPATIBILITY

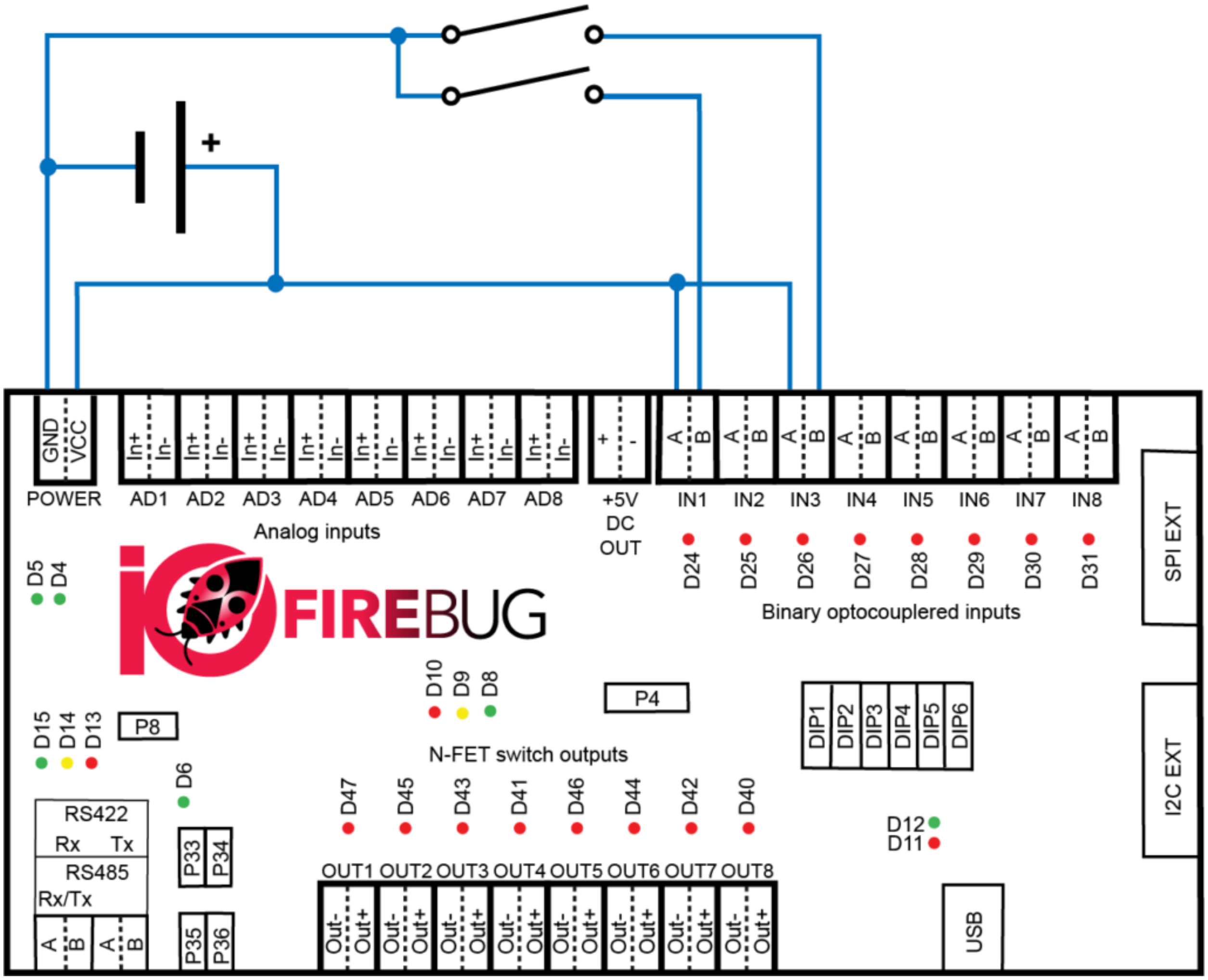
Though IOFirebug Pro supports the RS-485 interface, the IOFirebug Engine for Universal Robots currently **doesn't** support RS-485 communication.

Please do not insert the RS-485 jumper (P8) into the Engine board.

I/O connection

BINARY INPUT CONNECTION

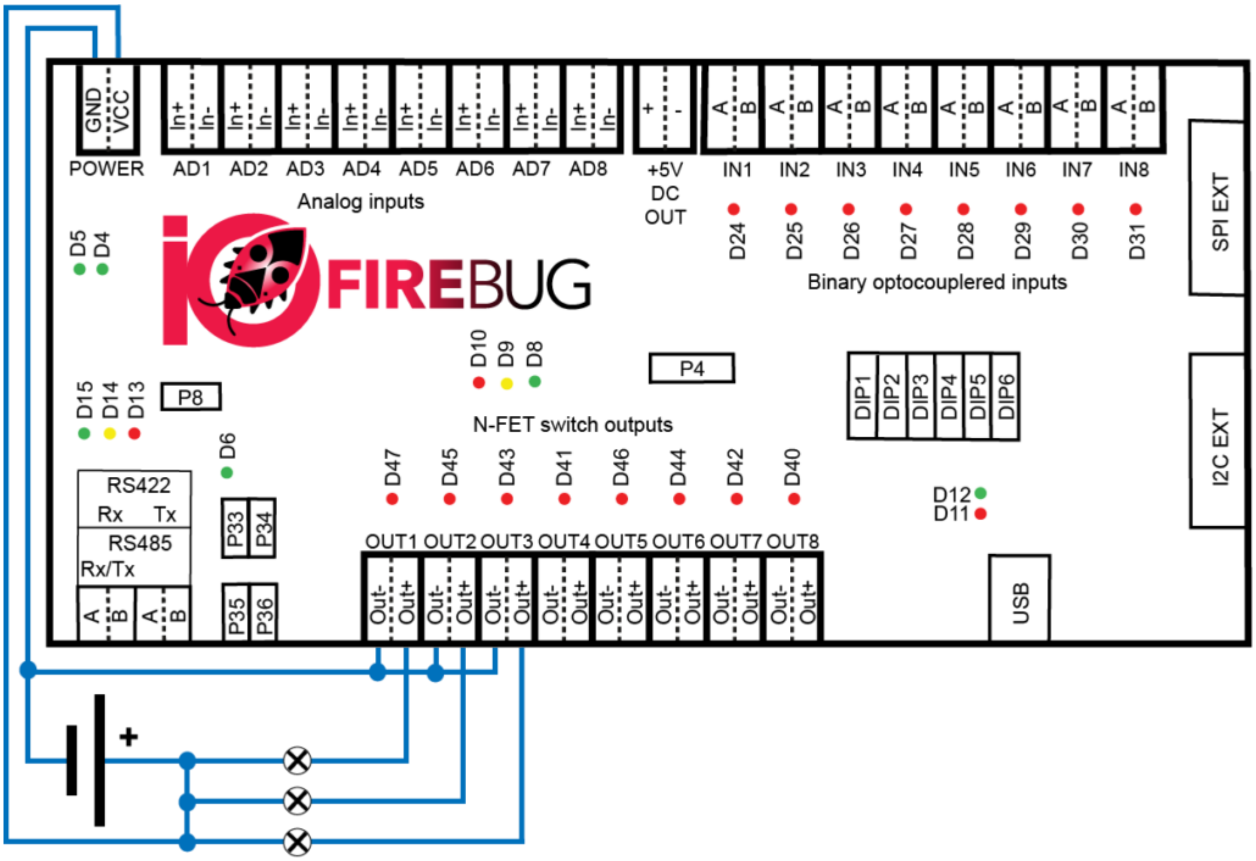
Binary inputs are galvanically isolated using optoisolators and independent of input voltage. The input voltage range is from 10V to 30V. Every input has its indication LED, which is a lid on as output is in the HIGH state.



1 BINARY INPUT WIRING EXAMPLE

BINARY OUTPUT CONNECTION

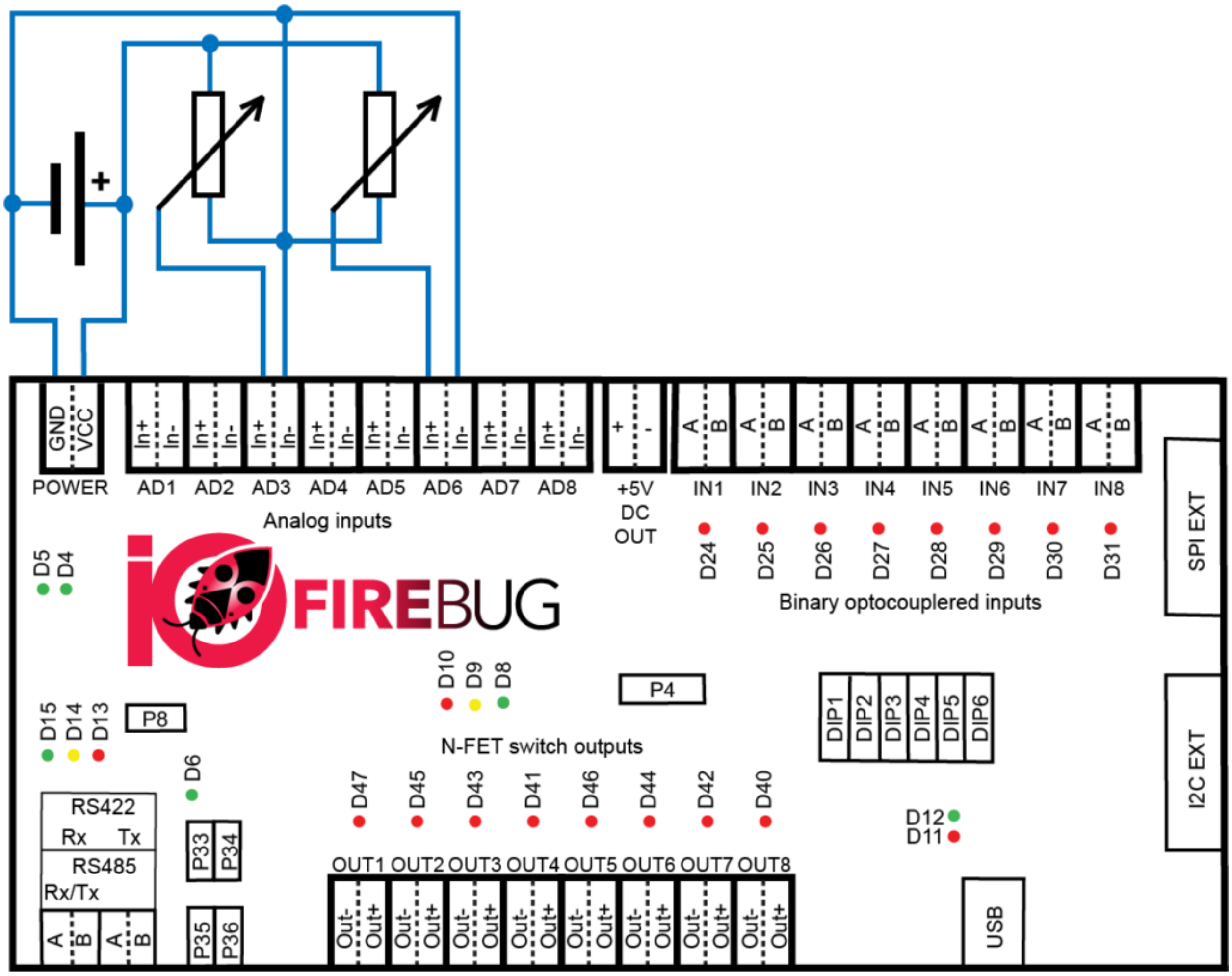
The binary outputs consist of an N-FET switching transistor. Load switching is realized by grounding (connecting negative wire). The transistor output is rated for switching 3A of continuous current, which in the short term can switch up to 4A. The maximum switching voltage is 50V. Each output has a LED indicator that shines red when the transistor (output) is closed.



2 BINARY N-FET OUTPUT WIRING EXAMPLE

ANALOG INPUT CONNECTION

Analog inputs are used to precisely determine input voltage.



3 ANALOG INPUT WIRING EXAMPLE

IOFirebug is equipped with an 8-channel 12bit A/D converter. Analog inputs are designed to measure DC voltage 0 to 10V. The minimum value is 0 for 0V, maximum safe value is 3413 for 10V (input can accept 12V for short time, in this case, the value is 4095).

- Use `iofb_get_analog_input` function for reading a value in the integer format
- Use `iofb_get_analog_inputF` function for reading a value in the float format

Wagon expansion modules

Modules are connected to the IOFireBug Engine device with 10 pin ribbon cable (part of Wagon delivery), using the SPI interface. When using extension modules, IOFireBug Engine must be properly configured by URCap. Expansion modules can be connected in the chain in any order.

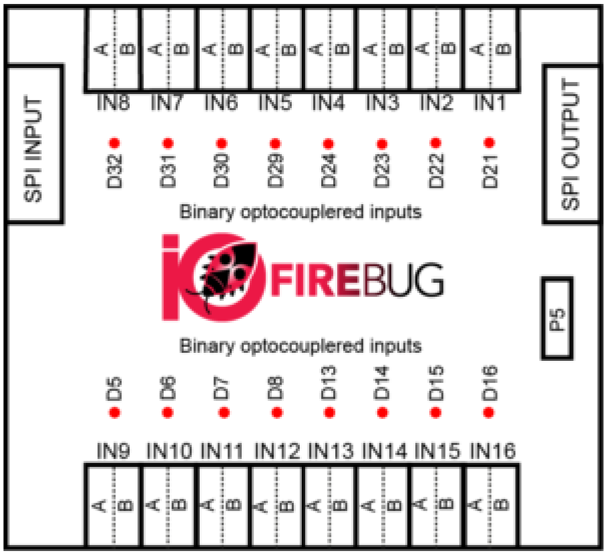
Only the last module must be terminated by the jumper.

The Engine can use a maximum of 8 expansion modules at once.

WAGON 16I EXPANSION MODULE

Wagon 16I allows you to connect additional 16 digital inputs to the IOFireBug Engine. Inputs are electrically identical to those on the Engine device. Connection circuitry is therefore analogous. If the module is last in the chain, the bus must be terminated by a P5 jumper.

The module is powered by the Engine device. Each input has an LED that shines red when the input is in the HIGH state.

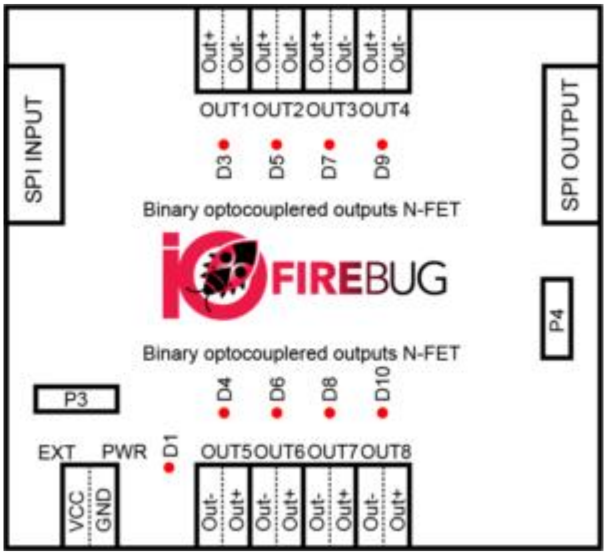

A B
IN8 IN7 IN6 IN5 IN4 IN3 IN2 IN1
D32 D31 D30 D29 D24 D23 D22 D21
SPI INPUT
SPI OUTPUT
Binary optocouplered inputs
IOFIREBUG
P5
Binary optocouplered inputs
D5 D6 D7 D8 D13 D14 D15 D16
IN9 IN10 IN11 IN12 IN13 IN14 IN15 IN16
A B

4 BINARY INPUT EXPANDER

WAGON 8O EXPANSION MODULE

The expansion Module allows you to connect additional 8 binary outputs (N-FET) to the IOFireBug Engine. Outputs are electrically identical to those on the Engine device. Connection circuitry is therefore analogous. If the module is last in the chain, the bus must be terminated by a P4 jumper.

By default, the module is powered by the Engine device, but it may also be powered by an external power supply. To select how the module is powered, use jumper P3. When the module is powered correctly, green LED D1 shines. Each output has an LED that shines red when the output is in the HIGH state.



5 BINARY N-FET OUTPUT EXPANDER

URCap plug-in

IOFirebug Engine for Universal robots is designed to be used with the URCap robot plug-in.

Please note, that other versions of IOFirebug products are not compatible with URCap.

COMPATIBILITY

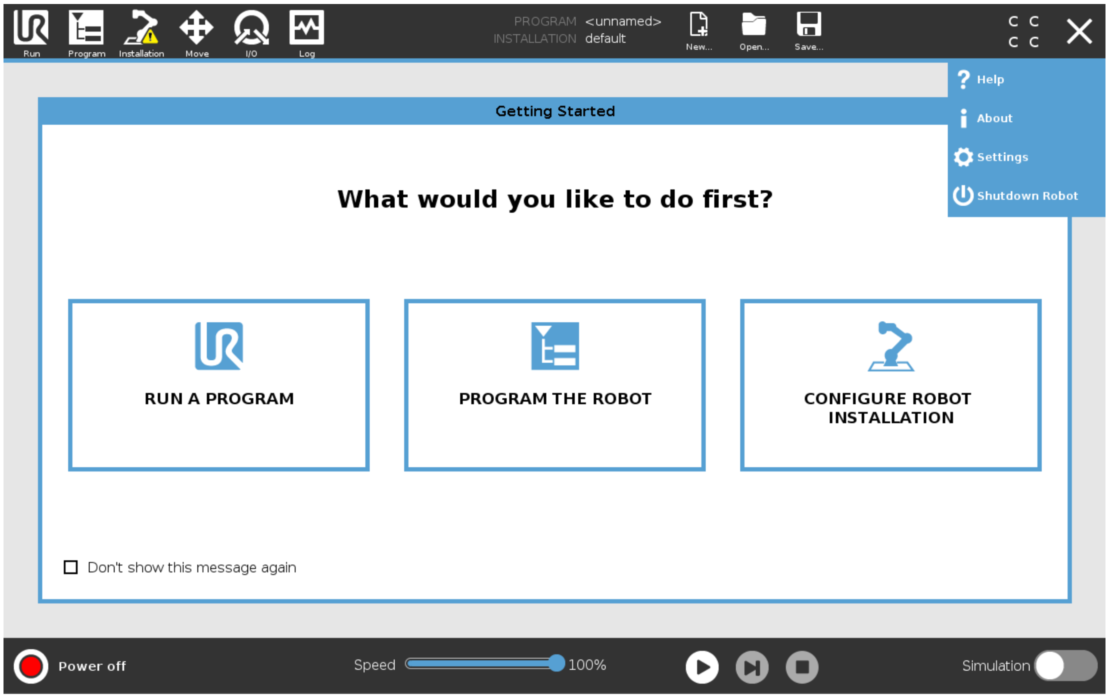
IOFirebug for Universal robots and URCap is working with CB3.1 and Polyscope 5.11 or 3.11.

Following instructions are demonstrated on the Plyscope version 5.11, but all principles are the same in new versions of Polyscope as well.

INSTALLATION

The plug-in installation procedure is following:

1) Download the URCap at: https://4each.cz/iofirebug-engine-for-universal-robots
2) Copy the .urcap file on the flash disk and insert it in the teach pedant of the robot
3) Connect the IOFirebug Engine for Universal Robots to the robot and provide the correct power supply.
4) In the upper right corner, choose “Settings”

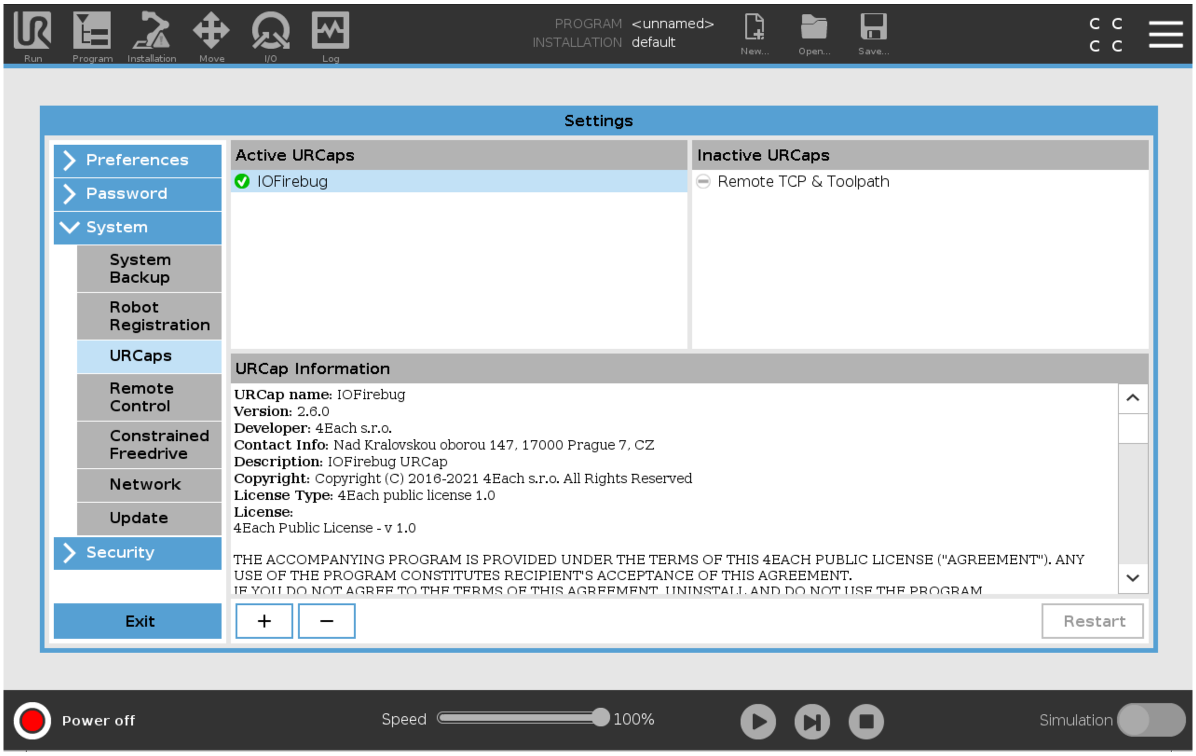


Getting Started

What would you like to do first?

□ Don't show this message again

5) In the Settings, go to “System” → “URCaps”
6) Click the “+” button for adding the plug-in and find the .urcap file on the flash disk.
7) Click on the “Restart” button and wait until the robot starts again



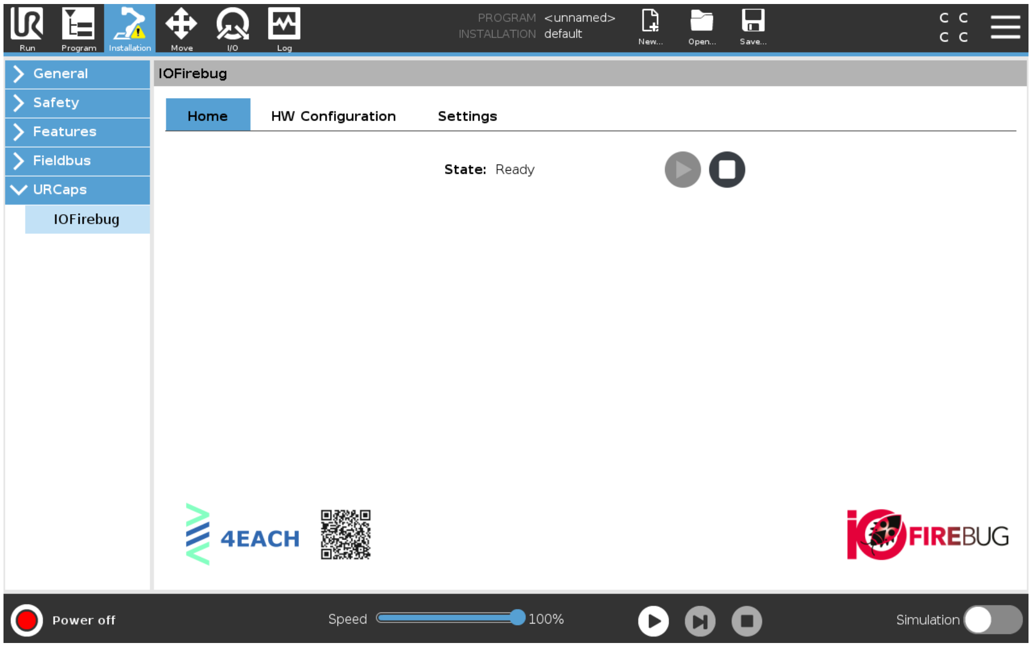
URCap name: IOFirebug
Version: 2.6.0
Developer: 4Each s.r.o.
Contact Info: Nad Kralovskou oborou 147, 17000 Prague 7, CZ
Description: IOFirebug URCap
Copyright: Copyright (C) 2016-2021 4Each s.r.o. All Rights Reserved
License Type: 4Each public license 1.0
License:
4Each Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS 4EACH PUBLIC LICENSE ("AGREEMENT"). ANY USE OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.
IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, UNINSTALL AND DO NOT USE THE PROGRAM.

If the installation was successful, a green icon appears.

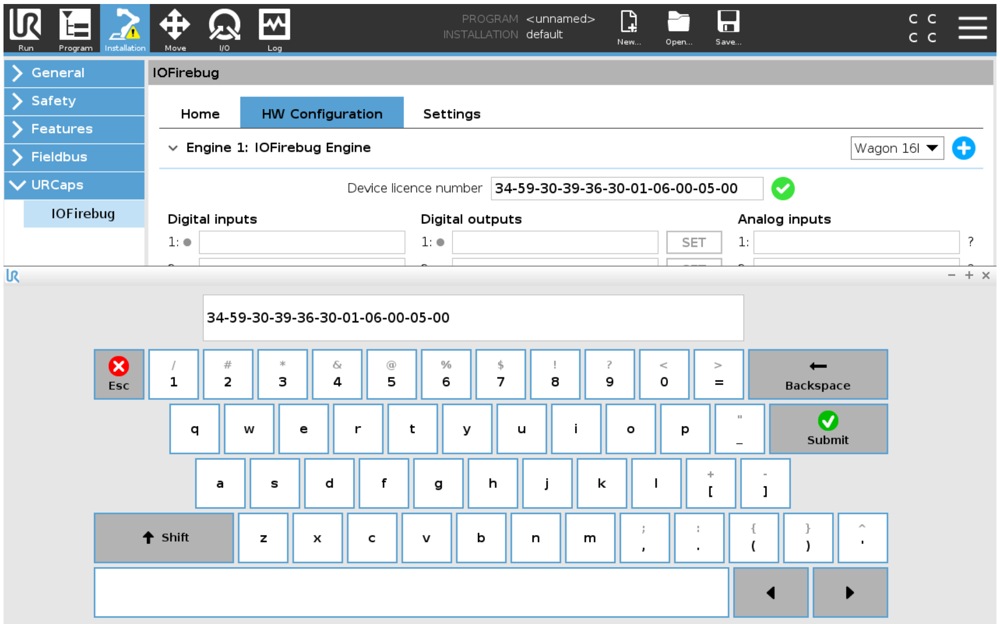
CONFIGURATION

1) Go to the Program Robot screen and choose the Installation tab
2) If the URCap installation is successful, you will find IOFirebug node. In this node there are 3 tabs — "Home", "HW Configuration" and "Settings"



State: Ready

3) On the **HW Configuration** tab, type the device license number which is part of product delivery. (If you feel that you miss your license number, do not hesitate to contact product support at 4each@4each.cz)



Engine 1: IOFirebug Engine

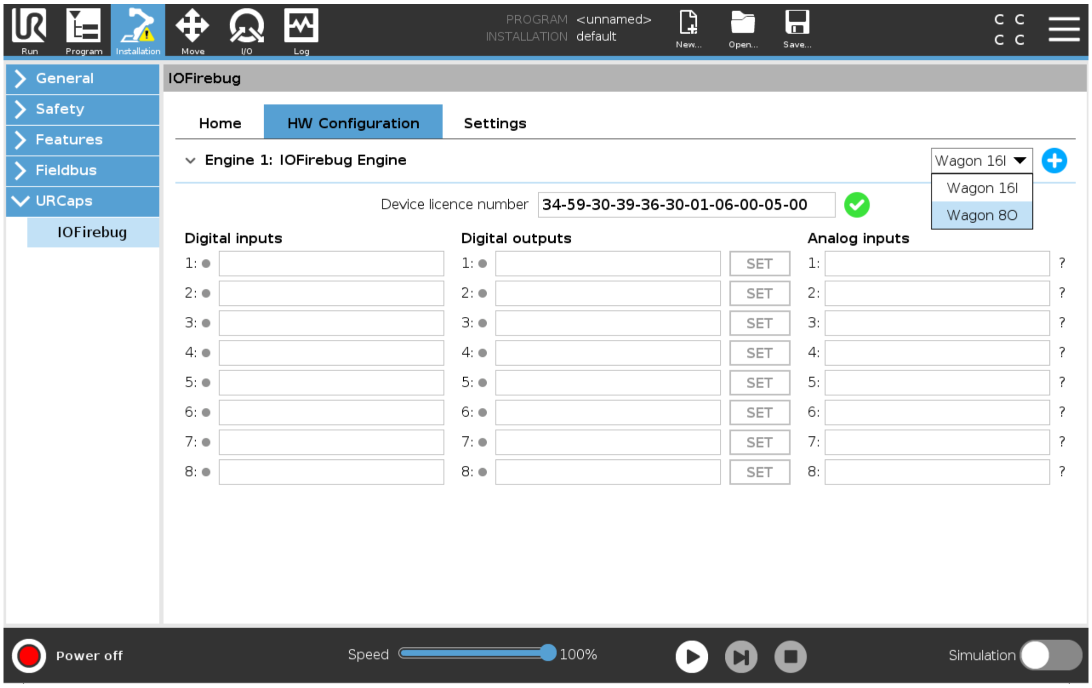
Wagon 16I

Device licence number 34-59-30-39-36-30-01-06-00-05-00

34-59-30-39-36-30-01-06-00-05-00

Note: Correct format is like 00-01-02-03-04-05-06-07-08-09-10. These values are in hexadecimal format, so they can contain characters “a” to “f” as well.

4) In this tab you can also add or modify Wagons connected to your Engine. Make sure the Wagons are added in the correct order (left to right)



PROGRAM <unnamed>
INSTALLATION default

IOFirebug

Engine 1: IOFirebug Engine

Wagon 16I

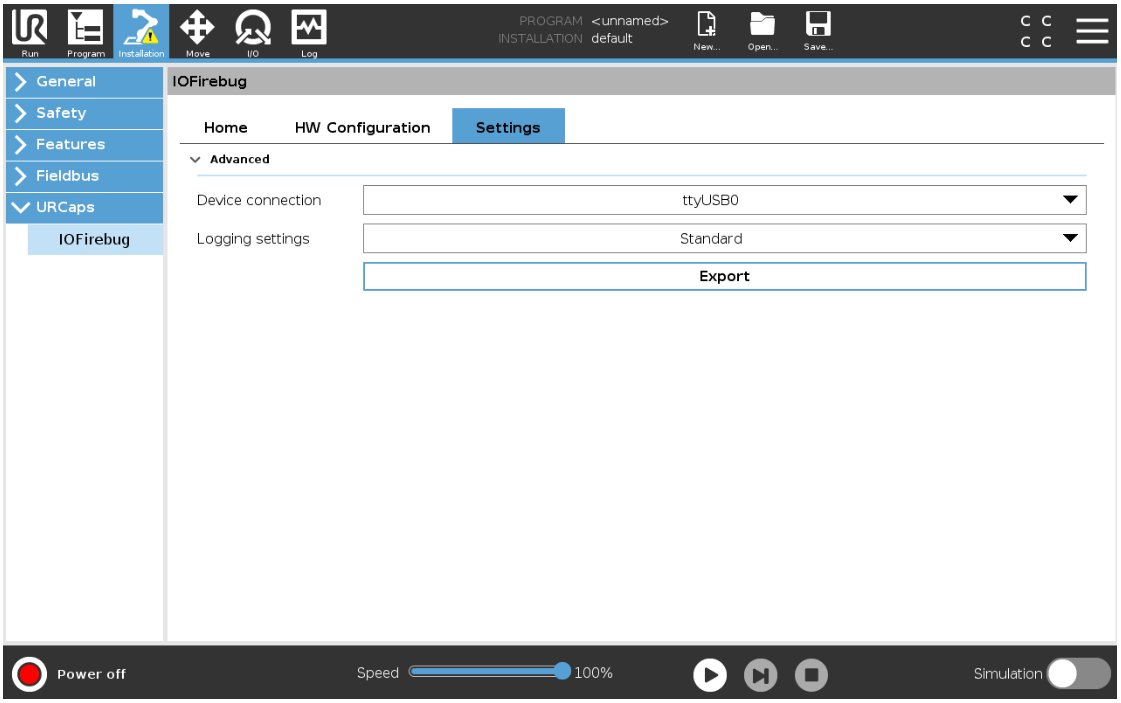
Device licence number 34-59-30-39-36-30-01-06-00-05-00

Speed 100%

5) The IOFB tab gives you a quick overview of your IOFirebug set. You can also set a name for any IO. It helps you with the orientation in the Program section.
Dot status icons turn red (negative) and green (positive) according to their state.
You can also set or reset the DO. It is very useful during the wire check. It is not necessary to create any application logic to check the wiring.

6) In the **Settings tab** define the interface for connection. If there are no other USB accessories connected to the robot it will always be “**ttyUSB0**”.

7) You can also set the logging level to “Standard” or “Debug”.



IOFirebug

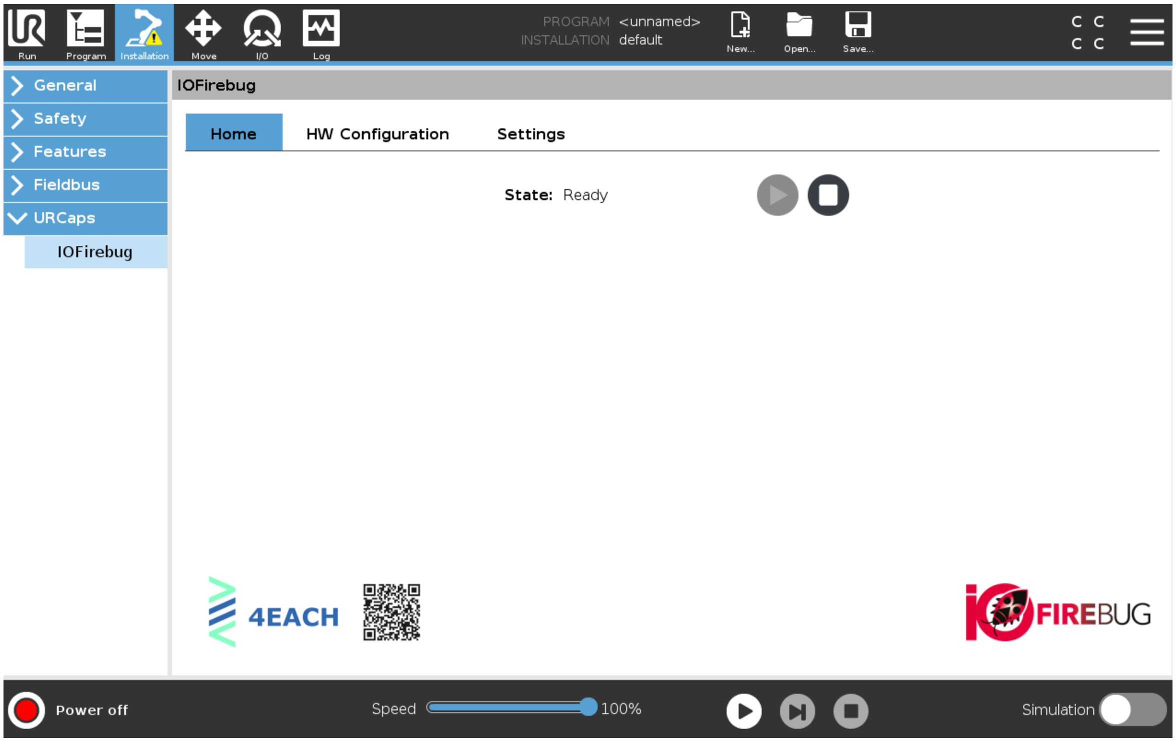
Advanced

Device connection: ttyUSB0

Logging settings: Standard

Note: The number of displayed ports does not correspond with the robot's physical interfaces. This setting is only for wider compatibility in the case of using the UR with other similar accessories from other UR+ vendors.

8) When everything is set up, click the start button on the Home tab.



IOFirebug

State: Ready

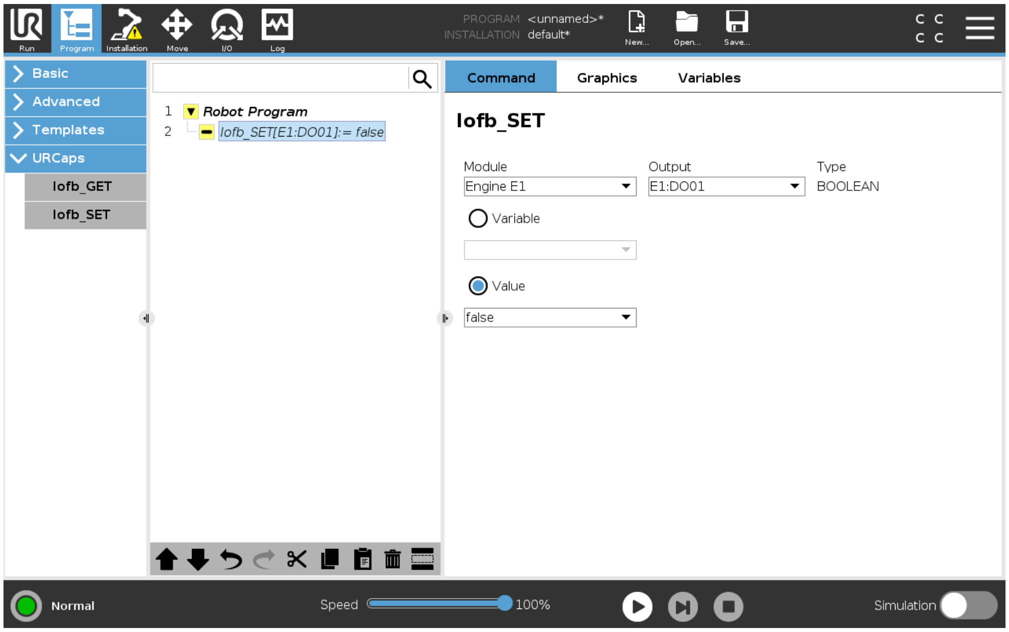
USING IOFIREBUG WITH YOUR ROBOT APPLICATION

IOFirebug products for Universal robots allow for expanding robot I/O interface depending on the count of Wagon modules connected.

In this document, we assume using one Wagon 16I and one Wagon 8O. The IOFirebug system has 24 digital inputs and 16 digital outputs configured. Engine is equipped with 8 analog inputs without any expanding capability.

IOFB_SET PROGRAM NODE

It is possible to put the Set IOFirebug output function to the UR program from the "URCap" Tab.



PROGRAM <unnamed>*
INSTALLATION default*

1 Robot Program
2 Iofb_SET[E1:DO01]:= false

Iofb_SET

Module	Output	Type
Engine E1	E1:DO01	BOOLEAN

○ Variable

● Value

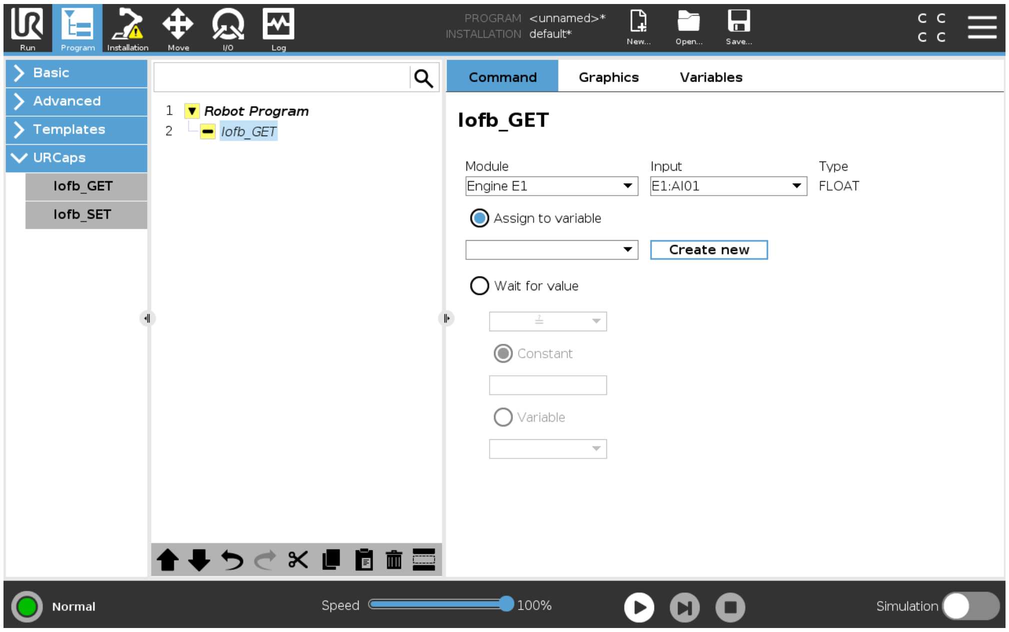
false

When this function is inserted, a program node appears in the program tree. The command tab contains a user interface for defining the module and the output index. The value can be assigned from a variable or set as a constant (true or false).

Because the IOFirebug system is connected externally, URCap supports the confirmation of commands sent to the output.

IOFB_GET PROGRAM NODE

It is possible to put the Get IOFirebug input function to the UR program from the “URCap” Tab.



1 Robot Program
2 Iofb_GET

Iofb_GET

Module: Engine E1 | Input: E1:AI01 | Type: FLOAT

● Assign to variable

Create new

○ Wait for value

● Constant

○ Variable

When this function is inserted, a program node with undefined output will appear in the program tree. The command tab contains a user interface for defining the module and the input index. The value can be assigned to a variable or “Wait for value” option can be chosen.

The “Wait for value” stops the program until the chosen input is in the demanded state.

AVAILABLE VARIABLES

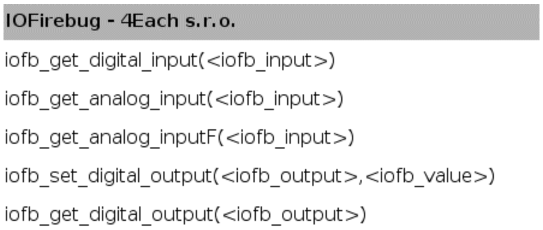
URCap defines globally available variables with the count of configured digital inputs and outputs

iofb_configured_di_cnt contains the numbed of configured digital inputs

iofb_configured_do_cnt contains the numbed of configured digital outputs

AVAILABLE FUNCTIONS

URCap adds the following functions to the expression editor:



IOFirebug - 4Each s.r.o.

iofb_get_digital_input(<iofb_input>)

iofb_get_analog_input(<iofb_input>)

iofb_get_analog_inputF(<iofb_input>)

iofb_set_digital_output(<iofb_output>,<iofb_value>)

iofb_get_digital_output(<iofb_output>)

IOFB_GET_DIGITAL_INPUT

This function allows the UR program to get the state of the selected digital input. The function expects the index of configured digital input and returns its state. If the function iofb_get_digital_input is called with a higher index of digital input than configured count, the return value will always be False. If the communication with the device is broken or the device is turned off the function returns False.

Arguments

uint iofbinput: index of configured digital input.

Return value

bool: State of digital input with *iofbinput* index

IOFB_GET_ANALOG_INPUT

This function allows the UR program to get the state of the selected analog input. The function expects the index of analog input and returns its value. If the function iofb_get_analog_input is called with index higher than 8 function returns value -1. If the communication with the device is broken or the device is turned off the function returns 0.

Arguments

uint iofbinput: index of analog input. (1…8)

Return value

uint: Value of analog input with *iofbinput* index

IOFB_GET_ANALOG_INPUTF

This function calls internally the iofb_get_analog_input method and converts the return value to decimal volts using float number.

Arguments

uint iofbinput: index of analog input. (1…8)

Return value

float: Value of analog input in volts with *iofbinput* index

IOFB_SET_DIGITAL_OUTPUT

This function allows the UR program to set the state of the selected digital output. The function expects the index of configured digital output and requested value. This function returns the confirmation status of the operation (disconnected or powered off device returns False). If the function iofb_get_digital_input is called with a higher index of digital output than configured count, the return value will always be False.

Arguments

uint iofboutput: index of configured digital output.

bool value: requested value.

Return value

bool: Confirmation status of the operation

IOFB_GET_DIGITAL_OUTPUT

This function allows the UR program to get the state of the selected digital output. The function expects the index of configured digital output. This function returns the state of the output with a small delay since the output is set by the function *iofb_set_digital_output* or *iofb_set_digital_outall*.

Arguments

uint iofboutput: index of configured digital output.

Return value

bool: status of the digital output

Troubleshooting

DEVICE INITIALISATION

During the initialization process, some issues can happen. This chapter will try to help you with what to do in case of some error messages.

Message: USB port error or device disconnected!

Recommended steps:

In case of USB connection: Be sure that the USB cable is not broken and is no longer than 1.8m and the device has the correct power supply (10 – 30 V DC).

In case of RS-422 connection using converter: Check the USB cable or try to use another converter.

Message: Communication broken or invalid address!

Recommended steps:

In case of USB connection: Be sure that all DIP switches of the device are set to OFF except DIP6, which should be ON.

In case of RS-422 connection using converter: Check the state of DIP switches states. Be sure you are using a double twisted-pair cable no longer than 100m and the RS-422 line is properly terminated (converter setting, All IOFirebug Engine jumpers P33-P36 are present).

Message: Invalid license number or device version!

Recommended steps: Stop the daemon by pressing the "Stop" button and check if the license number is correct. This message can be shown in case of misconfiguration of the device. In that case, try to use the "DEVICE DEFAULT VALUES" procedure.

Message: This is not UR+ Certified device!

Recommended steps: Buy UR+ certified product. Other IOFirebug products are not compatible with URCap.

Message: Daemon internal error!

Recommended steps: Remove installed URCap and install the version from the vendor site.

DEVICE DEFAULT VALUES

If you set all DIP switches (inside the box) to position ON and reboot the device (cut off the power supply), the unit clears the last configuration of EEPROM memory, this sets the default values. The unit indicates memory erase by flashing red LED RUN.

To return to the normal operational state, you need to switch DIP1 to DIP5 switches to an OFF state and DIP6 to an ON state. Do not forget to restart the unit one more time before use.