

MOD-IO2 extension board **USER'S MANUAL**

Document revision G, July 2024
Board designed by OLIMEX Ltd



All boards produced by Olimex LTD are ROHS compliant

DISCLAIMER

© 2024 Olimex Ltd. Olimex®, logo and combinations thereof, are registered trademarks of Olimex Ltd. Other product names may be trademarks of others and the rights belong to their respective owners.

The information in this document is provided in connection with Olimex products. No license, express or implied or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Olimex products.

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://www.creativecommons.org/licenses/by-sa/3.0/>.



This hardware design by Olimex LTD is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.

The software is released under GPL.

It is possible that the pictures in this manual differ from the latest revision of the board.

The product described in this document is subject to continuous development and improvements. All particulars of the product and its use contained in this document are given by OLIMEX in good faith. However all warranties implied or expressed including but not limited to implied warranties of merchantability or fitness for purpose are excluded. This document is intended only to assist the reader in the use of the product. OLIMEX Ltd. shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.

This evaluation board/kit is intended for use for engineering development, demonstration, or evaluation purposes only and is not considered by OLIMEX to be a finished end-product fit for general consumer use. Persons handling the product must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards.

Olimex currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive. Olimex assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

THERE IS NO WARRANTY FOR THE DESIGN MATERIALS AND THE COMPONENTS USED TO CREATE MOD-IO2. THEY ARE CONSIDERED SUITABLE ONLY FOR MOD-IO2.

Table of Contents

| | |
|---|------------------|
| <u>DISCLAIMER.....</u> | <u>2</u> |
| <u>CHAPTER 1: OVERVIEW.....</u> | <u>5</u> |
| <u>1. Introduction to the chapter.....</u> | <u>5</u> |
| <u>1.1 Features.....</u> | <u>5</u> |
| <u>1.2 MOD-IO vs MOD-IO2.....</u> | <u>5</u> |
| <u>1.3 Board layout (top view).....</u> | <u>6</u> |
| <u>1.4 Target market and purpose of the board.....</u> | <u>6</u> |
| <u>1.5 Organization.....</u> | <u>7</u> |
| <u>CHAPTER 2 SETTING UP THE MOD-IO2 BOARD.....</u> | <u>8</u> |
| <u>2. Introduction to the chapter.....</u> | <u>8</u> |
| <u>2.1 Electrostatic warning.....</u> | <u>8</u> |
| <u>2.2 Typical setup.....</u> | <u>8</u> |
| <u>2.3 Requirements.....</u> | <u>8</u> |
| <u>2.4 Powering the board.....</u> | <u>8</u> |
| <u>2.5 Firmware description and basic usage under Linux.....</u> | <u>9</u> |
| <u>2.5.1 I2C-tools for controlling MOD-IO2 under Linux.....</u> | <u>10</u> |
| <u>2.5.2 Custom software tool for controlling MOD-IO2 under Linux.....</u> | <u>13</u> |
| <u>2.5.3 Arduino IDE demos.....</u> | <u>15</u> |
| <u>CHAPTER 3: THE CH32V003 MICROCONTROLLER.....</u> | <u>16</u> |
| <u>3. Introduction to the chapter.....</u> | <u>16</u> |
| <u>3.1 The CH32V003F4P6 features.....</u> | <u>16</u> |
| <u>CHAPTER 4 CONNECTORS AND PINOUT.....</u> | <u>17</u> |
| <u>4. Introduction to the chapter.....</u> | <u>17</u> |
| <u>4.1 CON4.....</u> | <u>17</u> |
| <u>4.2 UEXT connectors.....</u> | <u>18</u> |
| <u>4.2.1 Female connector.....</u> | <u>18</u> |
| <u>4.2.2 Male connector.....</u> | <u>18</u> |
| <u>4.3 Relay connectors.....</u> | <u>19</u> |
| <u>4.4 CON1, CON2, CON3 connectors.....</u> | <u>20</u> |
| <u>4.5 PWR Jack.....</u> | <u>20</u> |
| <u>4.6 Jumper description.....</u> | <u>21</u> |
| <u>4.6.1 PGM1.....</u> | <u>21</u> |
| <u>4.6.2 SDA_E1/SCL_E1.....</u> | <u>21</u> |
| <u>4.6.3 UEXT_FPWR_E1.....</u> | <u>21</u> |
| <u>4.6.4 UEXT_MPWR_E1.....</u> | <u>21</u> |
| <u>4.7 Additional hardware components.....</u> | <u>21</u> |
| <u>CHAPTER 5 SCHEMATICS.....</u> | <u>22</u> |
| <u>5. Introduction to the chapter.....</u> | <u>22</u> |
| <u>5.1 KiCAD schematic.....</u> | <u>22</u> |
| <u>5.2 Physical dimensions.....</u> | <u>24</u> |

CHAPTER 6 REVISION HISTORY AND SUPPORT.....25
6. Introduction to the chapter.....25
6.1 Document revision.....25
6.2 Board's revision.....26
6.3 Useful web links and purchase codes.....27
6.4 Product support.....28

CHAPTER 1: OVERVIEW

1. Introduction to the chapter

Thank you for choosing the MOD-IO2 expansion board from Olimex! This document provides a user's guide for the Olimex MOD-IO2 board.

The MOD-IO2 board allows expanding your existing hardware with GPIOs, relays, PWMs, analog inputs and more. The design uses the CH32V003 RISC-V microcontroller. The hardware and software design are considered open source.

Important! This document applies to MOD-IO2 boards from hardware revision C or newer. If you are owner of MOD-IO2 boards from hardware revisions A or B, please use the older user's manual, revision F, available here:

https://www.olimex.com/Products/Modules/IO/MOD-IO2/resources/MOD-IO_manual_F.pdf

Other outdated resources suitable for previous hardware revisions of the board can be found here:

<https://www.olimex.com/wiki/MOD-IO2>

As an overview, this chapter gives the scope of this document and lists the board's features. The differences between the members of the MOD-IO2 and MOD-IO boards are mentioned. The document's organization is then detailed.

1.1 Features

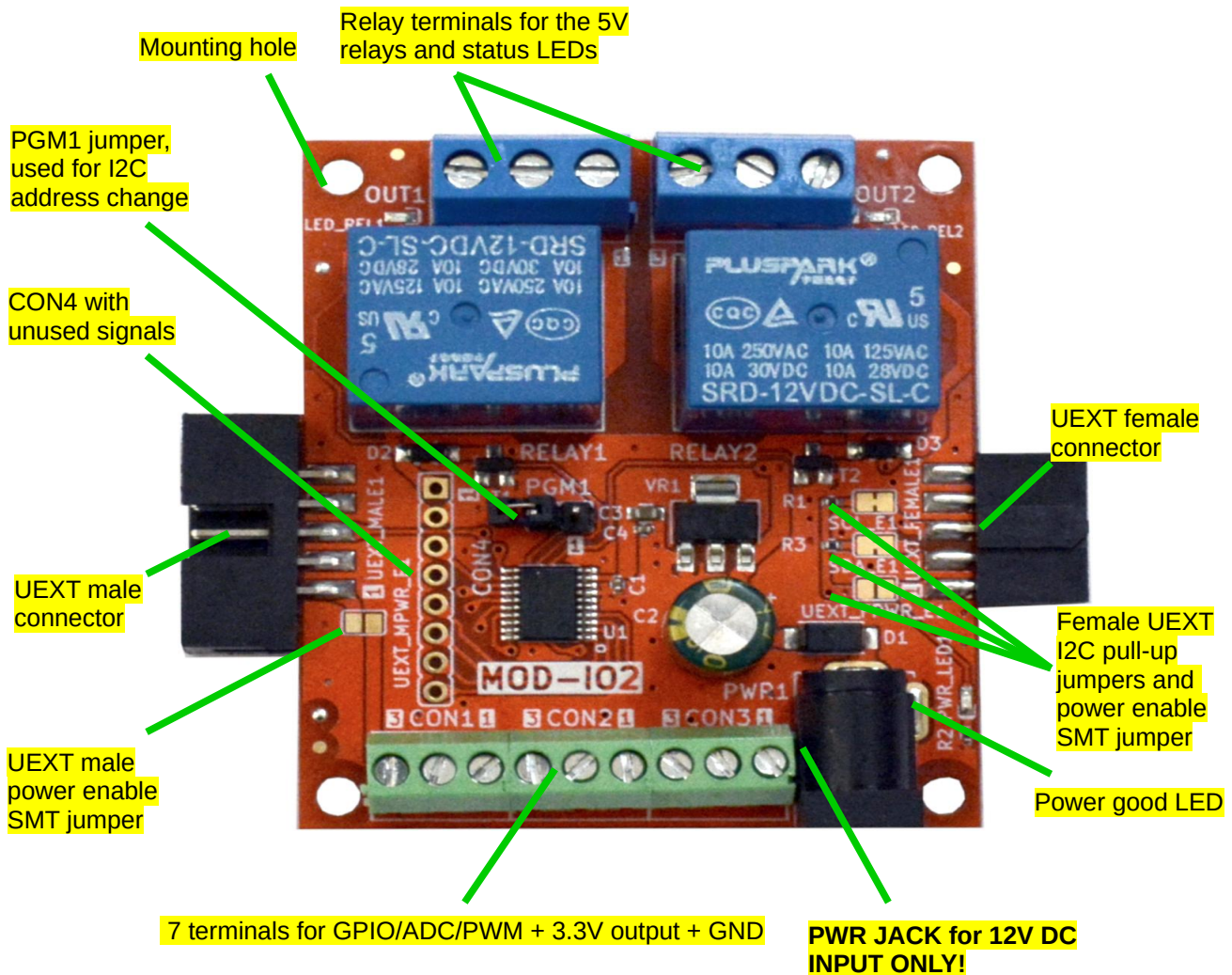
MOD-IO2 has the following features:

- ◆ CH32V003 RISC-V microcontroller pre-loaded open-source firmware for easier interfacing
- ◆ Board uses I2C, allows I2C address change
- ◆ Stackable, [UEXT male and female connectors](#)
- ◆ 9-pin terminal screw connector for 7 GPIOs, 3.3V and GND
- ◆ Firmware supports: 7 x GPIO, 2 x RELAY, 3 x ANALOG IN, 2 x PWM (GPIO, ANALOG IN, PWM multiplexed)
- ◆ 2 relay outputs with 15A/250VAC contacts with screw terminals
- ◆ RELAY output status LEDs
- ◆ [PWR jack for 12V DC](#)
- ◆ Four mounting holes 3.3mm ~ (0.13)"
- ◆ UEXT female-female cable included
- ◆ FR-4, 1.5mm ~ (0.062)", red soldermask, white silkscreen component print
- ◆ Dimensions: (61 x 52)mm ~ (2.40 x 2.05)"

1.2 MOD-IO vs MOD-IO2

MOD-IO2 is a smaller-sized input output extension module compared to MOD-IO both in terms of size and in terms of functionality, however in a lot of situations MOD-IO2 might provide to be the better choice. Designs which need optocouplers should consider MOD-IO. Additionally MOD-IO has a better power supply with the option to be provide voltage in the 8-30VDC range.

1.3 Board layout (top view)



1.4 Target market and purpose of the board

MOD-IO2 is an extension development board which can interface other Olimex boards via UEXT connector it adds add RELAYs and GPIOs (or ADC, or PWM). Multiple MOD-IO2s are stackable and individually addressable. The firmware allows you to interact with the board using simple commands and yet if you wish you can modify the firmware for your needs.

If you work with any of our development boards with UEXT connector and you need more GPIOs and RELAY outputs you can add these by attaching MOD-IO2 to your development board. This board allows easy interfacing to 2 relays and 7 GPIOs. MOD-IO2 is stackable and addressable – these boards can be plugged together and you can add as many inputs and outputs as you want! 2-4-6-8 etc! MOD-IO2 has CH32V003 RISC-V microcontroller and the firmware is open source and available for modification.

The board is a very good addition to most of the Olimex boards if you need GPIOs or analog inputs or relays.

1.5 Organization

Each section in this document covers a separate topic, organized as follow:

- Chapter 1 is an overview of the board usage and features
- Chapter 2 provides a guide for quickly setting up the board
- Chapter 3 describes the component that is the heart of the board: the CH32V003
- Chapter 4 covers the connector pinout, peripherals, and jumper description
- Chapter 5 provides the schematics
- Chapter 6 contains the revision history, useful links and support information

CHAPTER 2 SETTING UP THE MOD-IO2 BOARD

2. Introduction to the chapter

This section helps you set up the MOD-IO2 development board for the first time. Please consider first the electrostatic warning to avoid damaging the board, then discover the hardware and software required to operate the board.

The procedure to power up the board is given, and a description of the default board behavior is detailed.

2.1 Electrostatic warning

MOD-IO2 is shipped in a protective anti-static package. The board must not be exposed to high electrostatic potentials. A grounding strap or similar protective device should be worn when handling the board. Avoid touching the component pins or any other metallic element.

2.2 Typical setup

The most typical setup for MOD-IO2 is to attach it via the female UEXT connector to another board with female UEXT connector. This establishes I2C interface between these two boards. Then run code on the main board and communicate via I2C with MOD-IO2.

2.3 Requirements

In order to set up the MOD-IO2 optimally, the following items are required:

- A board with a free I2C or any OLIMEX board that has an UEXT connector;
- 12V source of power for the relay operation; it should fit the on-board power jack.

If you wish to reprogram the board or modify the firmware you will also need:

- Third-party board capable to act as programmer for CH32V003 or dedicated programmer – here we use ESP32-S2-DevKit-Lipo-USB

Some of the suggested items can be purchased by Olimex, for instance:

SY0612E – power supply adapter 12V/0.5A for European customers, comes with power jack that fits the connector of MOD-IO2

ESP32-S2-DevKit-Lipo-USB – ESP32 development board that can be flashed with [firmware](#) to act as a CH32V003 programmer.

2.4 Powering the board

The board is powered by the power jack. You should provide 12V DC. More info about the power jack our boards use can be found here:

<https://www.olimex.com/wiki/PWRJACK>

For the European customers we sell an affordable power supply adapter 12V/0.5A – **SY0612E**. If you power the board correctly, the on-board red LED named PWR_LED1 would turn on.

2.5 Firmware description and basic usage under Linux

There is firmware loaded on the CH32V003 of the board that allows easier use of MOD-IO2 via I2C protocol. The firmware of MOD-IO2 has gone through several iterations. The latest firmware is revision 5.0.

You can download the latest firmware from this archive:

https://www.olimex.com/Products/Modules/IO/MOD-IO2/resources/MOD-IO2_firmware_v5.zip

It contains binary, sources, and explanation in PDF format.

IMPORATANT! Due to main chip change, the latest firmware – version 5.0 and newer – can NOT be programmed to boards older than hardware version C (those with PIC16 chip). For boards with PIC16 chip (and ICSP connector), please use firmware version 4.3.

Using the firmware with not-Linux enabled host boards please refer to the READMEIO2.pdf in the archive that contains the firmware sources.

Firmware revisions 1, 2, and 3 are NOT compatible. These firmware revisions define different MOD-IO2 board addresses and different command set. Firmware revisions 3.0, 3.1, and 3.02 (3.xx), and 4.3 are compatible. Firmware revision 5.0 and newer can only be used with boards that have CH32V003 chip (firmware 5.0 and newer is incompatible with boards with PIC16).

Please note that the custom firmware might NOT support all of the hardware capabilities of MOD-IO2. In some cases you might need to adapt the firmware to use the hardware of MOD-IO2 to its full potential!

2.5.1 I2C-tools for controlling MOD-IO2 under Linux

Instead of the custom program mentioned in 2.4.1 you might use the popular Linux tool “i2c-tools”. Download it with apt with:

```
apt install i2c-tools
```

MOD-IO2 is compatible with i2c-tools since the release of its firmware 3.

In that case the commands are the most popular ones from the i2c-tools – i2cdetect, i2cdump, i2cget, i2cset, i2ctransfer.

Use the above commands and the information about the firmware to send (i2cset) and receive (i2cget) different data. The information about the firmware is located in a README.pdf file in the archive of the firmware; the archive containing the latest firmware (version 5) might be found here:

https://www.olimex.com/Products/Modules/IO/MOD-IO2/resources/MOD-IO2_firmware_v5.zip

There is a table with most available commands:

| # | Command code | (Set/Get) Description | Data range |
|----|--------------|----------------------------------|--------------|
| 1 | 0x01 | Set GPIO direction | 0x00 - 0x7f |
| 2 | 0x02 | Set GPIO output level | 0x00 - 0x7f |
| 3 | 0x03 | Get GPIO input level | 0x00 - 0x7f |
| 4 | 0x04 | Set GPIO pull-up | 0x00 - 0x1f |
| 5 | 0x10 | Get analog input GPIO0 | 0xHHLL |
| 6 | 0x15 | Get analog input GPIO5 | 0xHHLL |
| 7 | 0x16 | Get analog input GPIO6 | 0xHHLL |
| 8 | 0x20 | Get board ID | 0x23 |
| 9 | 0x21 | Get firmware version | 0x50 |
| 10 | 0x40 | Set relay(s) on/off | 0x00 - 0x03 |
| 11 | 0x41 | Set relay(s) on | 0x01 - 0x03 |
| 12 | 0x42 | Set relay(s) off | 0x01 - 0x03 |
| 13 | 0x43 | Get current relay state | 0x00 - 0x03 |
| 14 | 0x50 | Set PWM off | 0x01 or 0x02 |
| 15 | 0x51 | Set PWM1 and duty cycle | 0x00 - 0xff |
| 16 | 0x52 | Set PWM2 and duty cycle | 0x00 - 0xff |
| 17 | 0xF0 | Change I2C address (PGM1 closed) | 0x01 - 0x7f |

Some examples for setting/reading MOD-IO2's peripherals in Linux using i2c-tools:

- Turning on the relays:

```
i2cset -y 2 0x21 0x40 0x03
```

,where

i2cset – command for sending data;

-y – to skip the y/n confirmation prompt;

2 – board's hardware I2C number (typically 0 or 1 or 2);

0x21 – board address (0x21 should be used for writing);

0x40 – Turn on or off relay operation (as seen in the firmware README.pdf);

0x03 – should be interpreted as binary 011 – turns on both relays (0x02 would turn only second relay, 0x01 only the first, 0x00 would turn both off – 0x03 again would turn them off also);

Expected result: a specific sound would occur and relay LED would turn on.

- Reading the status of the relays (available since MOD-IO2's firmware revision 3.02):

```
i2cset -y 2 0x21 0x43
```

and then the read command

```
i2cget -y 2 0x21
```

,where

i2cset – command for sending data;

-y – to skip the y/n confirmation prompt;

2 – I2C number (usually 0, 1, or 2);

0x21 – board address (0x21 should be used for writing);

0x43 – read relay operations (as seen in the firmware README.pdf);

Expected results: 0x00 – meaning both relays are off; 0x03 – should be interpreted as binary 011, e.g. both relays are on; etc.

- Reading ADC:

```
i2cset -y 2 0x21 0x10
```

and then the read command

```
i2cget -y 2 0x21
```

,where

0x10 – the first analog IO (AN0 at GPIO0), the second is 0x15 and the third 0x16;

The big thing here is that to read you actually have to write (“that you would read”). Read is a combination of i2cset and i2cget!

You can instead combine them in one line using `i2ctransfer`:

```
i2ctransfer -y 2 w1@0x21 0x10 r2
```

,where

`r2` is the number of bytes to read, we need two bytes for this.

Expected results: on the terminal you would receive random and changing number or `0x00` or `0x08` or `0xFF` whether you have the GPIO floating or set to `0V` or set to `3.3V`.

- Set all GPIOs to high level

```
i2cset -y 2 0x21 0x01 0x01
```

,where

`0x01` is used to define port directions;

`0x01` is the high level (for low level use `0x00`).

- Reading state of GPIOs:

```
i2cset -y 2 0x21 0x01
```

```
i2cget -y 2 0x21
```

or as single command:

```
i2ctransfer -y 2 w1@0x21 0x01 r1
```

Detailed explanation of the preloaded software may be found in the firmware archive available for download at MOD-IO2's web-page.

- Changing I2C device address – especially useful if you use more than one MOD-IO2

```
i2cset 2 0x21 0xF0 0xHH
```

,where

`0xF0` is the command code for I2C change

`HH` is new address in hexadecimal format

Note that PROG jumper must be closed to be able to change the address. After the address change, remember to open the jumper. If you forget the number of the address you can use the `modio2tool` to find the address, the command and parameter would be "`modio2tool -l`". You can also reset the default address (`0x21`) with the command and parameter "`modio2tool -X`".

- Set PWM1 (at GPIO6 pin)

```
i2cset -y 2 0x21 0x51 0x7f
```

`7f` is half the duty of PWM1

2.5.2 Custom software tool for controlling MOD-IO2 under Linux

To make the things even simpler we have written a software tool for controlling MOD-IO2 under Linux. You might find it here:

<https://github.com/OLIMEX/OLINUXINO/tree/master/SOFTWARE/UEXT%20MODULES/MOD-IO2/linux-access-tool>

You can download the binary to the board via Internet with:

```
wget https://github.com/OLIMEX/OLINUXINO/raw/master/SOFTWARE/UEXT%20MODULES/MOD-IO2/linux-access-tool/modio2tool
```

then give it privileges with:

```
chmod 7 modio2tool
```

then execute and see help with:

```
./modio2tool -h
```

This software tool requires a Linux-enabled board with I2C support. The tool works with MOD-IO2 units loaded with firmware revision 3 or newer (including firmware 5 or newer).

Most of the commands require the hardware I2C number as defined in your Linux distribution with parameter `-B X`, where `X` is the number of the I2C interface. Note that by default the software is set for use with hardware I2C interface #2 and board ID 0x21 – if your setup has different you would need to specify every time by using `-B X` (`X` is the hardware I2C number) and `-A 0xXX` (`XX` is the I2C address of the module).

Some examples for the usage of modio2tool and MOD-IO2 in Linux:

- Bringing up the help menu:

```
./modio2tool -h
```

,where

```
./modio2tool  – executes the binary  
-h           – parameter used to request the help information
```

Expected result: the format of commands would be shown and a list of commands would be printed.

- Switching on both relays:

```
./modio2tool -B 2 -s 3
```

,where

```
-B 2  – sets the board to use its hardware I2C #0 (typically either “0”, “1”, or “2”)  
-s 3  – “s” is used to turn on the relays; “3” specifies to turn on both relays (use “1” or “2” for only the
```

first or only the second relay)

Expected result: a specific sound would occur and relay LEDs would turn on.

- Switching off both relays:

```
./modio2tool -B 0 -c 3
```

,where

-B 0 – sets the board to use its hardware I2C #0 (typically either “0”, “1”, or “2”)
-c 3 – “c” is used to switch off the state relays; “3” specifies to turn off both relays (use “1” or “2” for only the first or only the second relay)

Expected result: a specific sound would occur and the relay LEDs would turn off.

- Reading the status of the relays (available since MOD-IO2's firmware revision 3.02):

```
./modio2tool -B 2 -r
```

,where

-B 2 – sets the board to use its hardware I2C #0 (typically either “0”, “1”, or “2”)
-r – “r” is used to read the relays;

Expected result: the state of the relays would be printed. 0x03 means that both relays are on (equivalent of the binary 0x011).

- Reading analog inputs (AN0, AN5, AN6):

```
./modio2tool -B 2 -a 0
```

,where

-B 2 – sets the board to use its hardware I2C #0 (typically either “0”, “1”, or “2”)
-a 0 – “a” is used to read the analog input; “1” is the analog input that is read – you can use “0”, “5”, “6” since not all AN signals are available.

Expected result: the voltage at the AN would be printed. If there is nothing connected it can be anything like “ADC1: 2.311V”.

ADC is 10-bit.

- Changing the I2C address – if you use more than one MOD-IO2 (available since MOD-IO2's firmware revision 3.02)

```
./modio2tool -B 0 -x 15
```

,where

-B 0 – sets the board to use its hardware I2C #0 (typically either “0”, “1”, or “2”)
-x 15 – “x” is used to change the I2C address of the board; “15” is the desired number – it is different

from the default “0x21”.

Expected result: the board would have new I2C address and you would need to specify it with `-A 0xXX` if you wish to use the `modio2tools` in future.

For more information refer to the help returned by `modio2tools` or to `modio2tools`'s source code.

2.5.3 Arduino IDE demos

We have written an Arduino IDE library suitable for the board, with a number of simple examples. If you use MOD-IO2 with Arduino IDE compatible board, you can download the ZIP archive then add it to your Arduino IDE installation.

Sketch → Include Library → Add .ZIP library... then point to the zip archive (no need to extract it)

You can find the archive here:

https://www.olimex.com/Products/Modules/IO/MOD-IO2/resources/MOD_IO2.zip

Once it is added you can explore the examples. The library and examples have been updated to work with firmware 5 of the board.

We have a number of Arduino boards that work with MOD-IO2, wide range of boards like OLIMEXINO-328 and ESP32-POE and ESP32-POE2.

CHAPTER 3: THE CH32V003 MICROCONTROLLER

3. Introduction to the chapter

In this chapter is located the information about the heart of MOD-IO2 – its CH32V003F4P6 microcontroller.

3.1 The CH32V003F4P6 features

- QingKe 32-bit RISC-V2A processor, supporting 2 levels of interrupt nesting
- Maximum 48MHz system main frequency
- 2KB SRAM, 16KB Flash
- Power supply voltage: 3.3/5V
- Multiple low-power modes: Sleep, Standby
- Power on/off reset, programmable voltage detector
- 1 group of 1-channel general-purpose DMA controller
- 1 group of op-amp comparator
- 1 group of 10-bit ADC
- 1×16-bit advanced-control timer, 1×16-bit general-purpose timer
- 2 WDOG, 1×32-bit SysTick
- 1 USART interface, 1 group of I2C interface, 1 group of SPI interface
- 18 I/O ports, mapping an external interrupt
- 64-bit chip unique ID
- 1-wire serial debug interface (SDI)
- Package: TSSOP20

For comprehensive information on the microcontroller visit the WCH's web page for a datasheet. At the moment of writing the documentation for the chip can be found at the following link:

https://www.wch-ic.com/downloads/CH32V003DS0_PDF.html

CHAPTER 4 CONNECTORS AND PINOUT

4. Introduction to the chapter

In this chapter are presented the connectors that can be found on the board all together with their pinout and notes about them. Jumpers functions are described. Notes and info on specific peripherals are presented. Notes regarding the interfaces are given.

4.1 CON4

This is a connector with left over signals not used by the firmware or the peripherals. Can be used for own purposes if you wish to modify the firmware. The firmware source code is available to the customer if you decide to implement extra functionality.

| CON4 | |
|-------|-------------|
| Pin # | Signal Name |
| 1 | +12 VDC |
| 2 | GND |
| 3 | PC3 |
| 4 | PD0 |
| 5 | PD4 |
| 6 | PD5 |
| 7 | PD6 |
| 8 | PD7 |

4.2 UEXT connectors

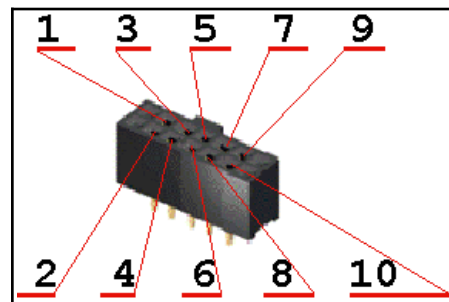
MOD-IO2 board has two UEXT connectors (male and female) and can interface Olimex's UEXT boards. For more information on UEXT please visit:

<https://www.olimex.com/Products/Modules/UEXT/>

4.2.1 Female connector

The female connector is used either to connect to a board directly (without using the female-female cable) or to connect the module to another MOD-IO2 – to create a stackable module that can be addressed via the I2C. Remember to change the I2C address of each board when using multiple boards. By default the I2C address is 0x21.

| Female UEXT | | | |
|-------------|---------------|-------|---------------|
| Pin # | Signal name | Pin # | Signal name |
| 1 | +3.3V out* | 6 | SDA |
| 2 | GND | 7 | Not connected |
| 3 | Not connected | 8 | Not connected |
| 4 | Not connected | 9 | Not connected |
| 5 | SCL | 10 | Not connected |



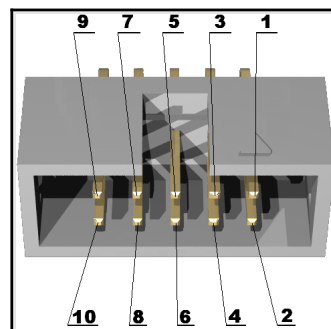
*Pin 1 is disconnected by default (can be enabled by modifying jumper UEXT_FPWR_E1). To avoid short-circuit with other modules that might also have 3.3V out.

I2C pins SCL and SDA have 4.7k pull-up resistors enabled by default. They can be disconnecting by disconnecting jumpers SCL_E1 and SDA_E1.

4.2.2 Male connector

The male connector is used with the ribbon cable in the package to connect to another male UEXT or to connected to another MOD-IO2.

| Male UEXT | | | |
|-----------|---------------|-------|---------------|
| Pin # | Signal name | Pin # | Signal name |
| 1 | +3.3V out* | 6 | SDA |
| 2 | GND | 7 | Not connected |
| 3 | Not connected | 8 | Not connected |
| 4 | Not connected | 9 | Not connected |
| 5 | SCL | 10 | Not connected |



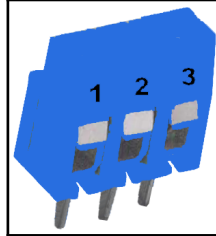
*Pin 1 is disconnected by default (can be enabled by modifying jumper UEXT_MPWR_E1). To avoid short-circuit with other modules that might also have 3.3V out.

SCL and SDA have NO pull-ups.

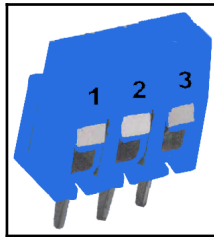
4.3 Relay connectors

There are two electromechanical relays in MOD-IO. Their output signals are the standard Normal Closed (NC), Normal Open (NO) and the Common (COM).

| REL1 - OUT1 | |
|-------------|--------------------|
| Pin # | Signal name |
| 1 | NO - normal open |
| 2 | NC - normal closed |
| 3 | COM - common |



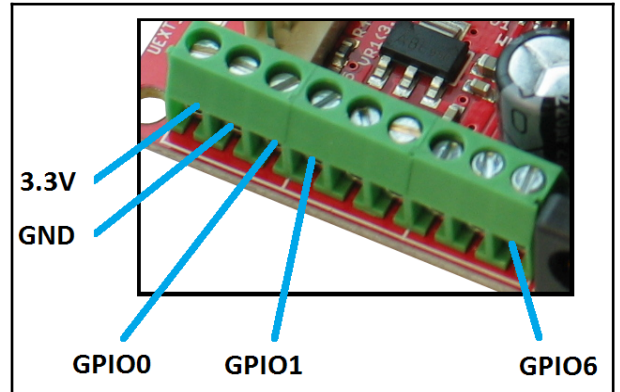
| REL2 - OUT2 | |
|-------------|--------------------|
| Pin # | Signal name |
| 1 | COM - common |
| 2 | NO - normal open |
| 3 | NC - normal closed |



4.4 CON1, CON2, CON3 connectors

The GPIO connectors are used for GPIO, ADC, PWM. There is also 3.3V output and GND. The GPIO, ADC, PWM functions are supported by the firmware. There is multiplexing between some GPIO pins and ADC and PWM. Relays are not multiplexed with CON1, CON2, CON3 signals. Note that pin names are also printed at the bottom of the board.

| Pin # | Signal name | Analog Input (firmware) | PWM (firmware) |
|-------|-------------|-------------------------|----------------|
| 1 | 3.3V | - | - |
| 2 | GND | - | - |
| 3 | GPIO0 | AN0 | - |
| 4 | GPIO1 | - | - |
| 5 | GPIO2 | - | - |
| 6 | GPIO3 | - | - |
| 7 | GPIO4 | - | - |
| 8 | GPIO5 | AN5 | PWM2 |
| 9 | GPIO6 | AN6 | PWM1 |

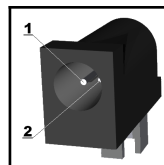


4.5 PWR Jack

The DC barrel jack has 2.0mm inner pin and 6.3mm hole. More information about the exact component might be found here: <https://www.olimex.com/wiki/PWRJACK>

For the European customers, we also stock and sell basic power supply adapters compatible with the power jack. MOD-IO2 requires 12V DC of voltage.

| Pin # | Signal name |
|-------|--------------|
| 1 | 12V DC input |
| 2 | GND |



4.6 Jumper description

Please note that almost all (except for PGM1 that is PTH) of the jumpers on the board are SMT type. If you feel insecure in your soldering/cutting technique it is better not to try adjusting SMT jumpers. Also if you feel incapable of removing the PTH jumper with hands better use tweezers.

4.6.1 PGM1

PTH jumper required to change I2C address via software means. Used to restrict changing I2C address. If you want to change I2C address you need to close it.

Default position is open.

4.6.2 SDA_E1/SCL_E1

Jumpers that connect or disconnect 4.7K pull-up resistors.

When you have more than one MOD-IO2 connected you need to keep those two jumpers closed, else the I2C line will be disconnected.

The default positions for both jumpers are closed (pull-ups enabled).

4.6.3 UEXT_FPWR_E1

If closed provides 3.3V at the female UEXT connector (but be careful since if you close that jumper and also you close the male one on the next MOD-IO2 line this might cause short-circuit and damage the board).

Default position is open.

4.6.4 UEXT_MPWR_E1

If closed provides 3.3V at the male UEXT connector (but be careful since if you close that jumper and also you close the female one on the next MOD-IO2 line this might cause short-circuit and damage the board).

The default position is open.

4.7 Additional hardware components

The components below are mounted on MOD-IO2 but are not discussed above. They are listed here for completeness:

Relay LEDs + Power LED

CHAPTER 5 SCHEMATICS

5. Introduction to the chapter

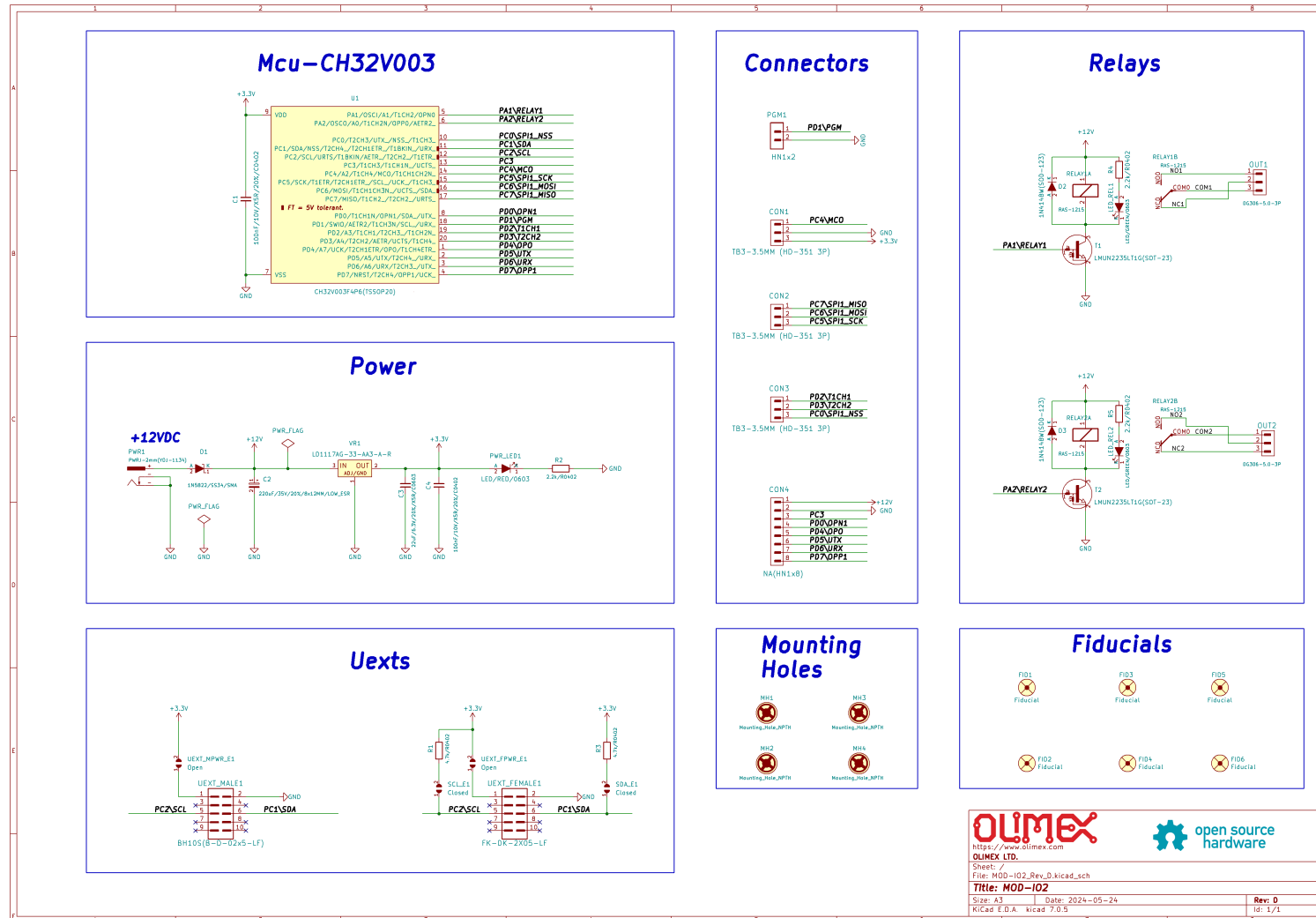
In this chapter are located the schematics describing logically and physically MOD-IO2.

5.1 KiCAD schematic

The latest MOD-IO2 schematic is situated on the next page for quicker reference. You can also find it in the HARDWARE section on the web page for MOD-IO2 at our site:

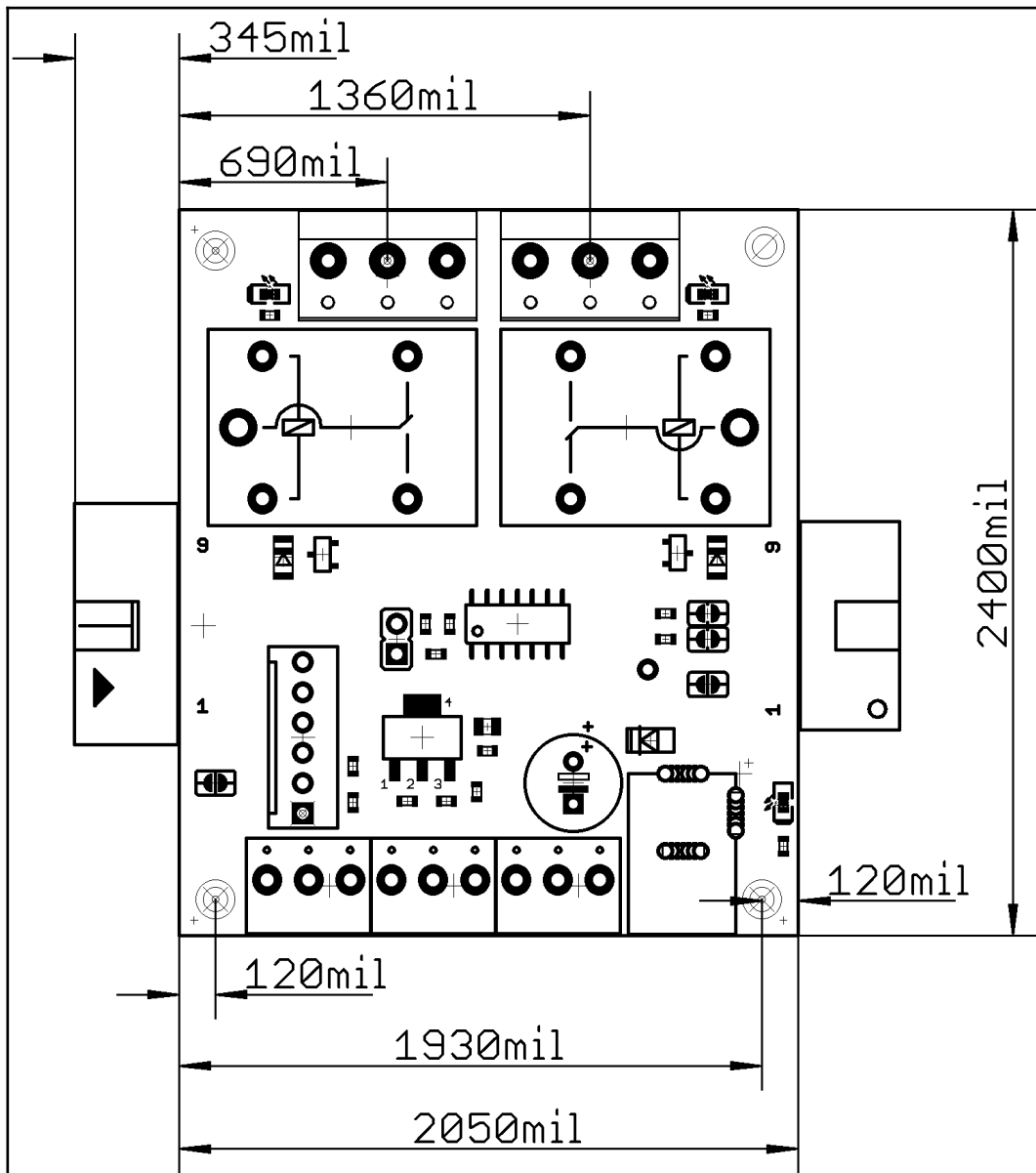
<https://www.olimex.com/Products/Modules/IO/MOD-IO2/open-source-hardware>

We also provide the hardware sources for KiCAD (which is open-source software CAD tool). MOD-IO2 is truly open source hardware project.



5.2 Physical dimensions

Note that all dimensions are in mils.



The three highest elements on the board in order from the tallest to the shortest are: relay T1 – 0.600" (15.25 mm) over the pcb; relay T2 – 0.600" (15.25 mm).

Note that the above measures does not include the PCB.

CHAPTER 6 REVISION HISTORY AND SUPPORT

6. Introduction to the chapter

In this chapter you will find the current and the previous version of the document you are reading. Also the web-page for your device is listed. Be sure to check it after a purchase for the latest available updates and examples.

6.1 Document revision

| Revision | Changes | Modified page# |
|-------------|--|---------------------|
| A, 27.08.12 | - Initial creation | All |
| B, 16.10.12 | - Fixed several leftovers from the template which were referencing wrong processors and boards - Updated links | 6, 10, 20 |
| C, 24.10.13 | - Updated Disclaimer to fit the open-source nature of the board - Added few examples and firmware version 3 explanation - Updated Product support - General formatting improvements | 2 7 23 All |
| D, 27.05.15 | - Updated the manual to reflect latest firmware revision 3.02 - Added information about the new Linux tool - modio2tools | 7, 8, 9, 10, 11 |
| E, 27.09.19 | - Updated the manual to reflect latest firmware revision 4.3 | 7, 8, 9, 10, 11 |
| F, 17.05.24 | - fixed wrong info about I2C address change command | 13, 19 |
| G, 25.07.24 | - adapted document for board revision C with WCH chip | All |

6.2 Board's revision

| Revision, release date | Revision notes |
|------------------------|---|
| B, 18.06.12 | Initial release |
| C, unreleased | <ol style="list-style-type: none">1. Board re-made in KiCAD2. PIC chip changed with CH32V003F4P63. LDO now LD1117AG-33-AA3-A-R4. ICSP connector no longer needed, replaced by CON4 with leftover GPIOs from the CH32V003 chip. |
| D, 20.07.24 | <ol style="list-style-type: none">1. Changed a number of white print labels at the bottom so that they fit the firmware2. Moved a few GND viases to make the white print readable3. PGM1 connector value was NA by mistake, now made available to HN1x2 |

6.3 Useful web links and purchase codes

The web page you can visit for more info on your device is:

<https://www.olimex.com/Products/Modules/IO/MOD-IO2/>

ORDER CODES:

MOD-IO2 – the version of the board discussed in this document

SY0612E – power supply adapter 12V/0.5A for MOD-IO2 – 220V (European compatibility)

How to order?

You can purchase directly from our online shop or from any of our distributors. Note that usually it is faster and cheaper to purchase Olimex products from our distributors. List of confirmed Olimex LTD distributors and resellers: <https://www.olimex.com/Distributors>.

Check <https://www.olimex.com/> for more info.

6.4 Product support

For product support, hardware information, and error reports mail to: support@olimex.com. All document or hardware feedback is welcome. Note that we are primarily a hardware company and our software support is limited. Please consider reading the paragraph below about the warranty of Olimex products.

All goods are checked before they are sent out. In the unlikely event that goods are faulty, they must be returned, to OLIMEX at the address listed on your order invoice.

OLIMEX will not accept goods that have clearly been used more than the amount needed to evaluate their functionality.

If the goods are found to be in working condition, and the lack of functionality is a result of lack of knowledge on the customers part, no refund will be made, but the goods will be returned to the user at their expense.

All returns must be authorized by an RMA Number. Email support@olimex.com for authorization number before shipping back any merchandise. Please include your name, phone number and order number in your email request.

Returns for any unaffected development board, programmer, tools, and cables permitted within 7 days from the date of receipt of merchandise. After such time, all sales are considered final.

Returns of incorrect ordered items are allowed subject to a 10% restocking fee. What is unaffected? If you hooked it to power, you affected it. To be clear, this includes items that have been soldered to, or have had their firmware changed. Because of the nature of the products we deal with (prototyping electronic tools) we cannot allow returns of items that have been programmed, powered up, or otherwise changed post shipment from our warehouse.

All returned merchandise must be in its original mint and clean condition. Returns on damaged, scratched, programmed, burnt, or otherwise 'played with' merchandise will not be accepted.

All returns must include all the factory accessories which come with the item. This includes any In-Circuit-Serial-Programming cables, anti-static packing, boxes, etc.

With your return, enclose your PO#. Also include a brief letter of explanation of why the merchandise is being returned and state your request for either a refund or an exchange. Include the authorization number on this letter, and on the outside of the shipping box.

Please note: It is your responsibility to ensure that returned goods reach us. Please use a reliable form of shipping. If we do not receive your package we will not be held liable.

Shipping and handling charges are not refundable. We are not responsible for any shipping charges of merchandise being returned to us or returning working items to you.

The full text might be found at <https://www.olimex.com/wiki/GTC#Warranty> for future reference.