SHARPIKEEBO 1.0

RASPBERRY PI ZERO + SHARP LS027B7DH01 Memory Display
Carrier Board + USB ATMEGA34U Keyboard

Description

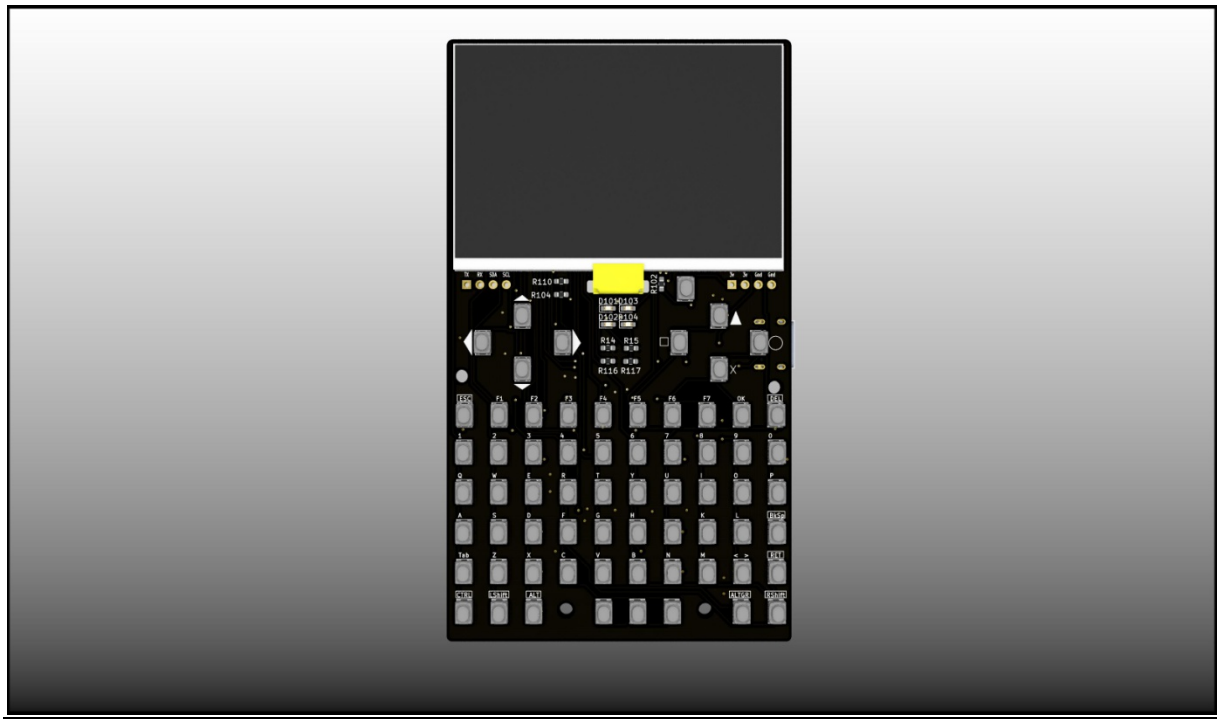
The SHARPIKEEBO carrier board has been designed to hold both a Raspberry(R) PI ZERO 2W or W + 2.7" (400x240) Sharp Memory Display. This display is capable of a fast refresh which allows the use of linux shells or display game images. A true USB Keyboard (driven by QMK firmware, so fully customized) has been added on the board.

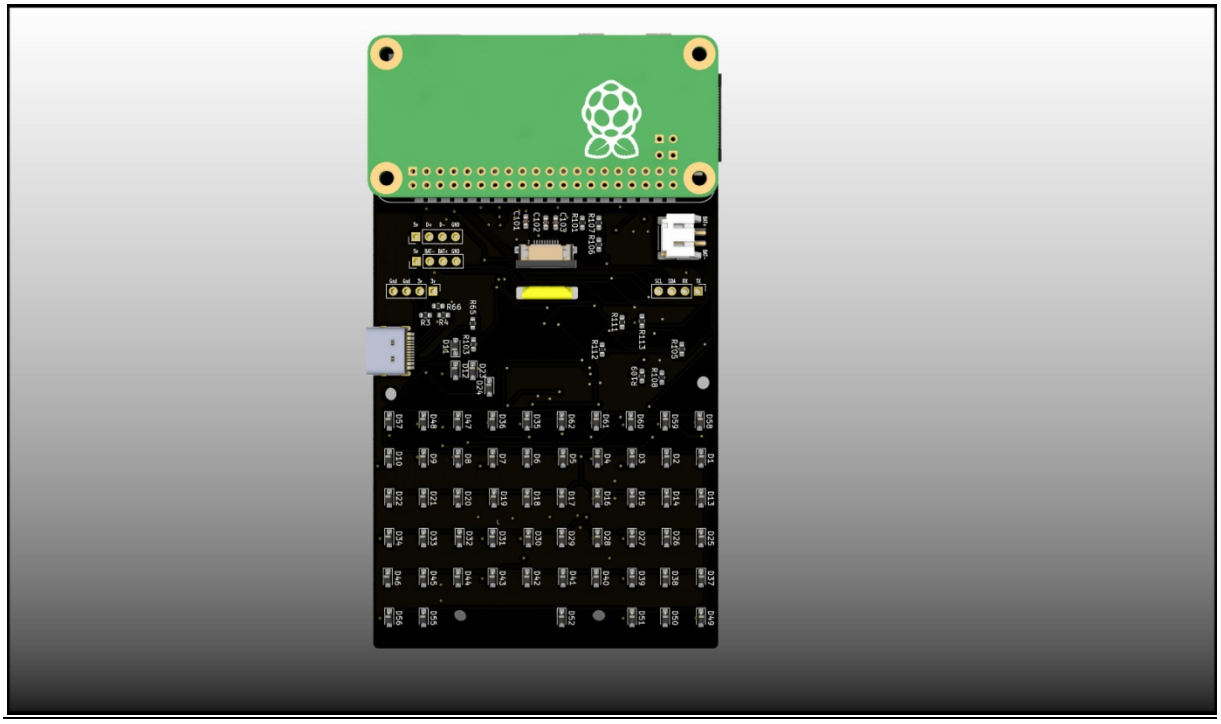
The board also has 4 independently controllable LEDs you can switch on/off with classic GPIO commands (C++ or Python)

The board also has 5 buttons you may also control via GPIO voltage detection (UP/Down voltage detection). You can use this board for gaming.

And more, you have UART GPIO and I2C GPIO easily available on board should you want to add a 3.3v or 5v module such a GPS , 4g or 5g modem etc ...

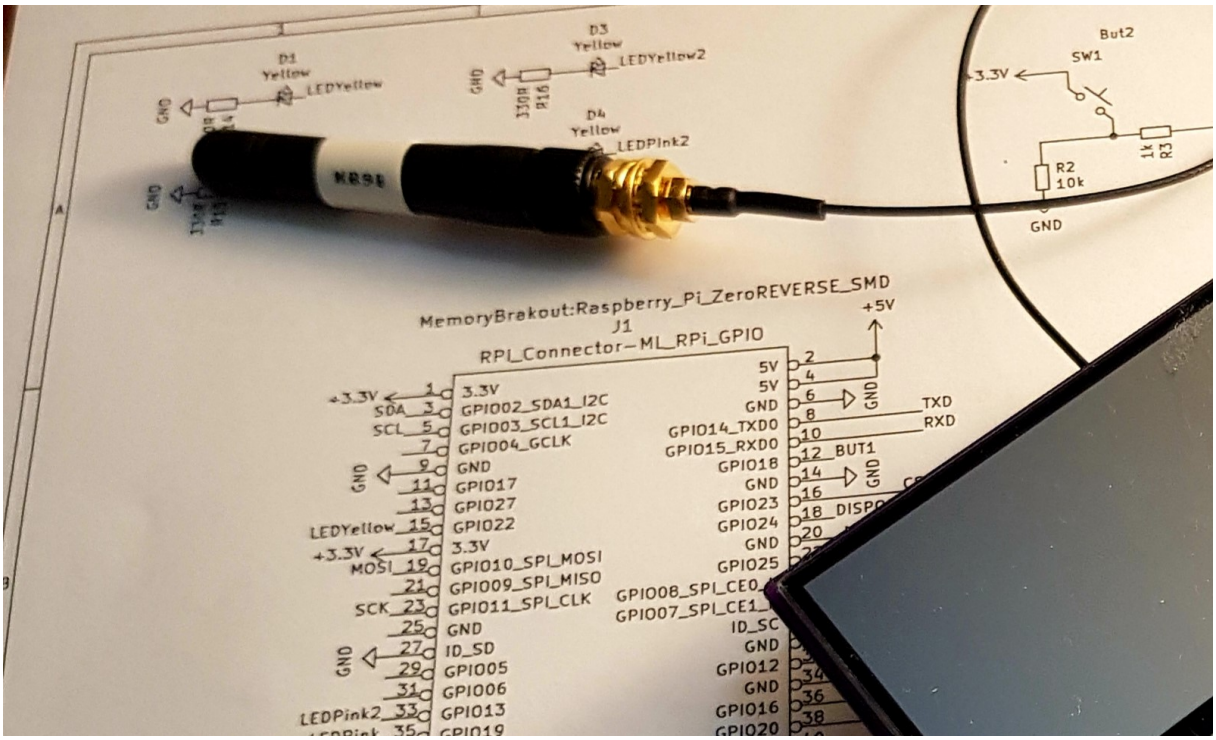
The board + RPI can be powered thought a USB C connector. The board has also been design to behave as a USB keyboard you can plug on a desktop or laptop or rack server.

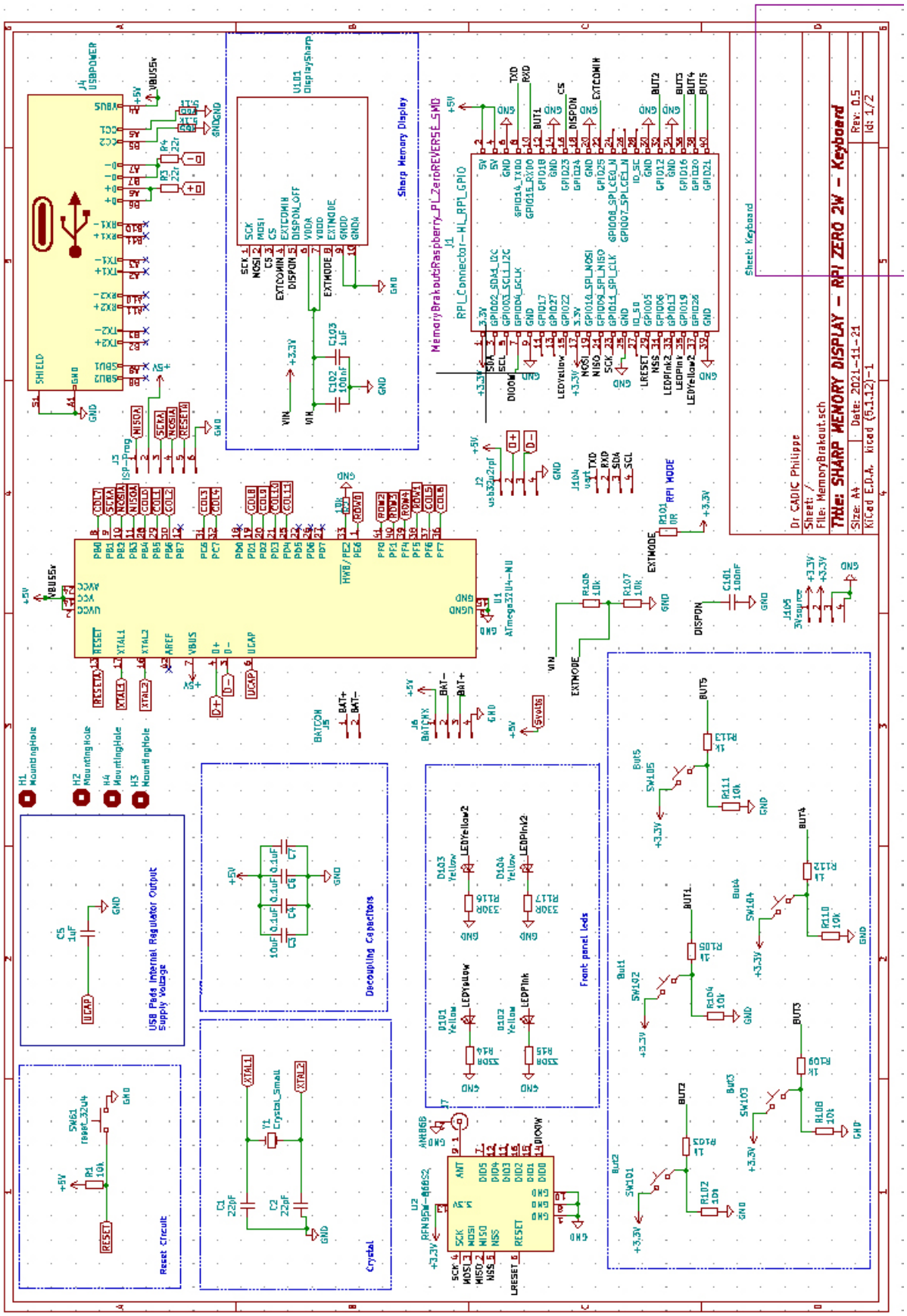


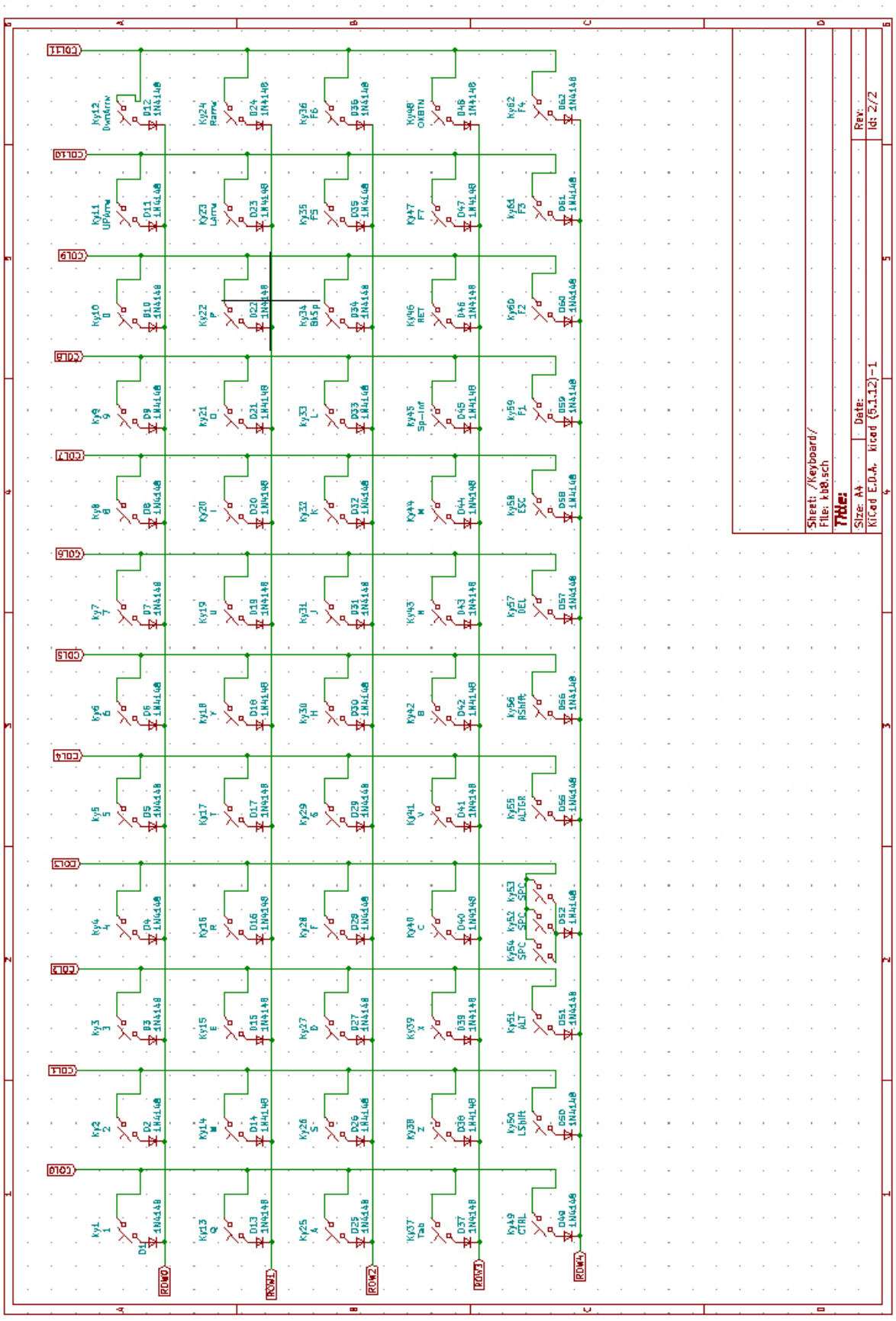


The SHARPIKEEBO is probably one of the smallest linux computers with keyboard; 400x240 pixels Sharp Memory Display + Raspberry PI Zero 2W + RFM95 long range radio transceiver. **6 cm x 11 cm x 1.5 cm ...**


pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $ lss
-bash: lss: command not fou
pi@raspberrypi:~ $ ls
1.3inch_LCD_HAT.7z Downloa
BookSheIf Music
Desktop Picture
Documents Project
pi@raspberrypi:~ $
pi@raspberrypi:~ $


D1
Yellow
LEDYellow
D3
Yellow
LEDYellow2
D4
Yellow
LEDPink2
GND
330R
R16
But2
SW1
3.3V
R2
10k
1k
R3
MemoryBreakout:Raspberry_Pi_ZeroREVERSE_SMD
J1
RPi_Connector-ML_RPi_GPIO
+5V
+3.3V
SDA
SCL
LEDYellow
MOSI
SCK
LEDPink2
3.3V
GPIO02_SDA1_I2C
GPIO03_SCL1_I2C
GPIO04_GCLK
GND
GPIO17
GPIO27
GPIO22
3.3V
GPIO10_SPI_MOSI
GPIO09_SPI_MISO
GPIO11_SPI_CLK
GND
ID_SD
GPIO05
GPIO06
GPIO13
5V
5V
GND
GPIO14_TXD0
GPIO15_RXD0
GPIO18
GND
GPIO23
GPIO24
GND
GPIO25
GPIO08_SPI_CE0
GPIO07_SPI_CE1
ID_SC
GND
GPIO12
GND
GPIO16
GPIO20
TXD
RXD
BUT1
DISP

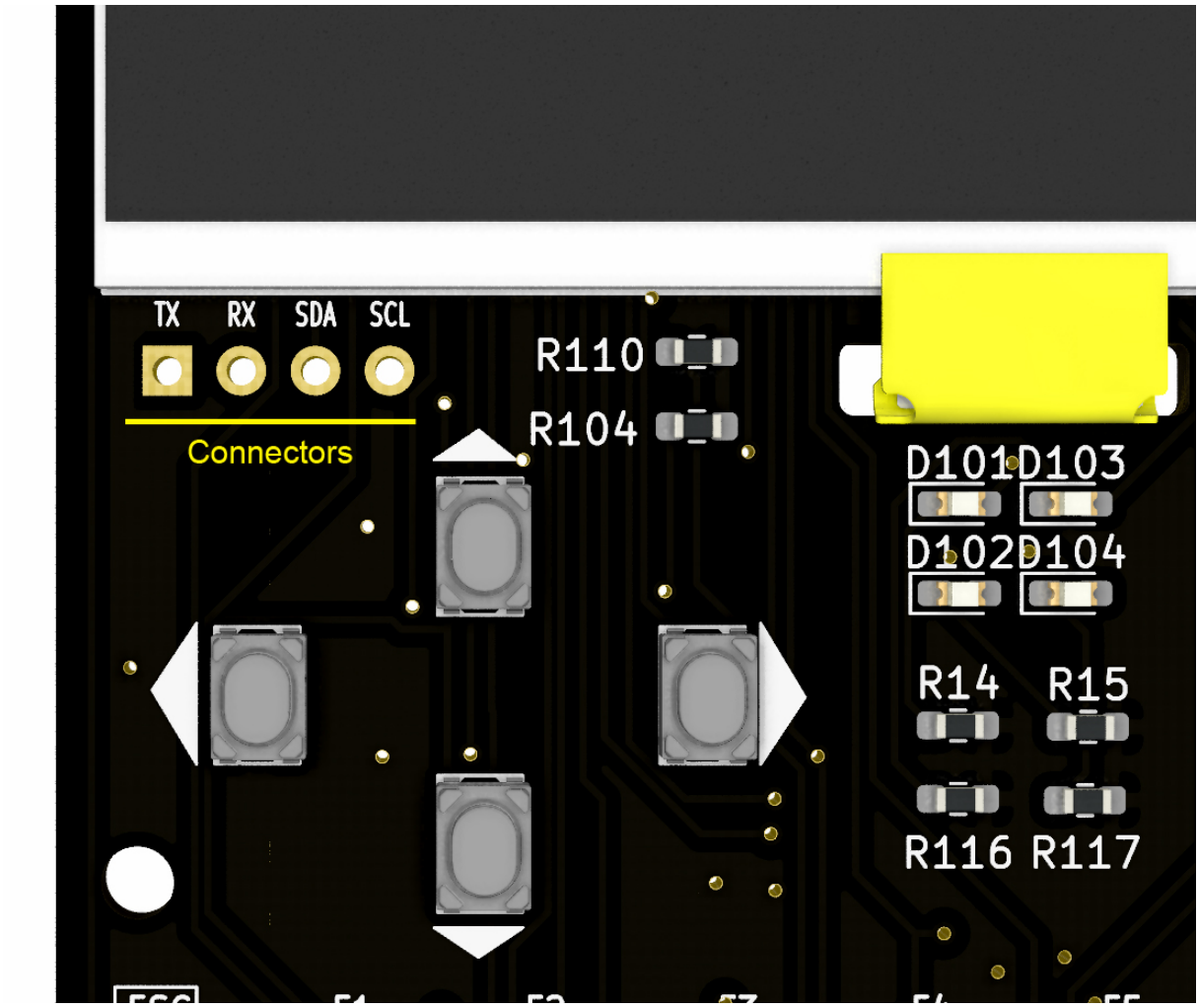

Title: SHARP MEMORY DISPLAY – RPI ZERO 2W – Keyboard
Size: A4
Date: 2021-11-21
Rev: 0.5
KiCad E.D.A. kicad (5.1.12)-1
Id: 1/2
Dr CADIC Philippe
Sheet: /
File: MemoryBrakout.sch
Sheet: Keyboard
Sharp Memory Display
Decoupling Capacitors
Front panel Leds
Crystal
Reset Circuit
USB Pads Internal Regulator Output Supply Voltage
RPI MODE


Sheet: /Keyboard/
File: kb8.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.1.12)-1
Id: 2/2

Extra GPIO available

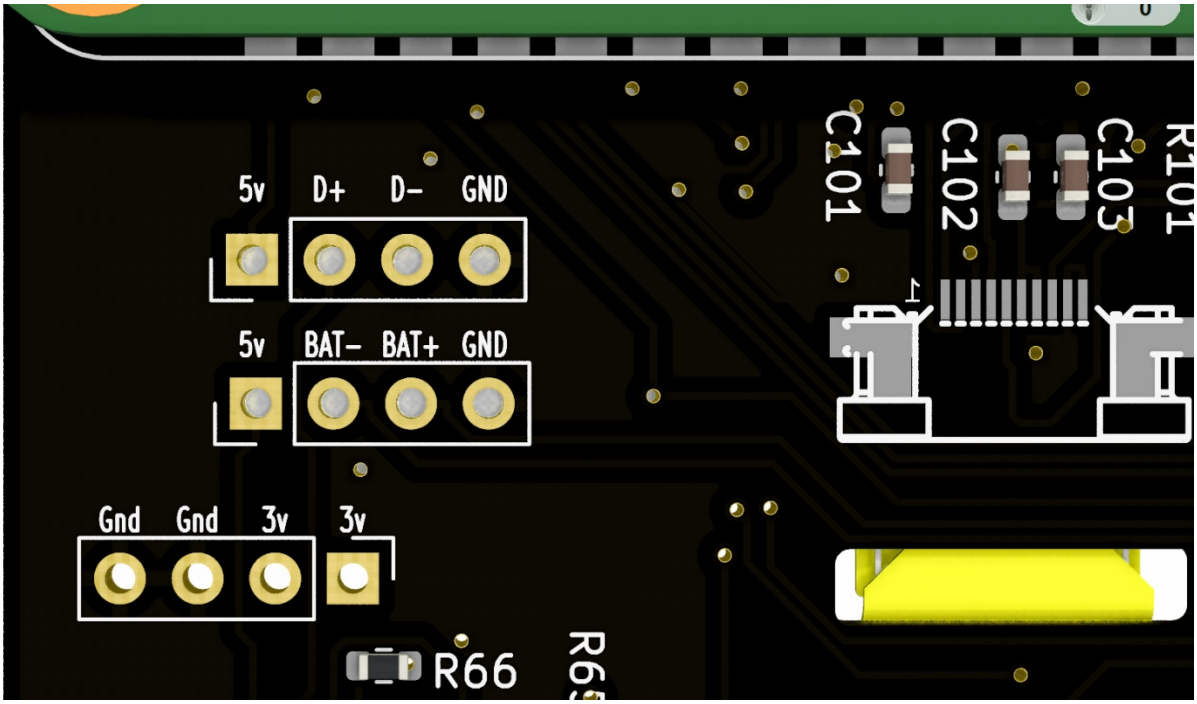
We added several extra GPIO, should you want to add an external UART extra board or a I2C board. For the maker, we also added access to the display pins with the right bottom line of 2.54 mm header pinout (SCK to GND). If you want, you can drive the LCD without the Raspberry PI ZERO , and replace it with a classic microcontroller. In the same idea, you have B1 to B5 buttons signals available on pins.

You get all the necessary connectors to add a battery charger , should your want to add a lipo and build a autonomous device or drive a 3G/4G/5G modem which require a lipo for energy reasons.

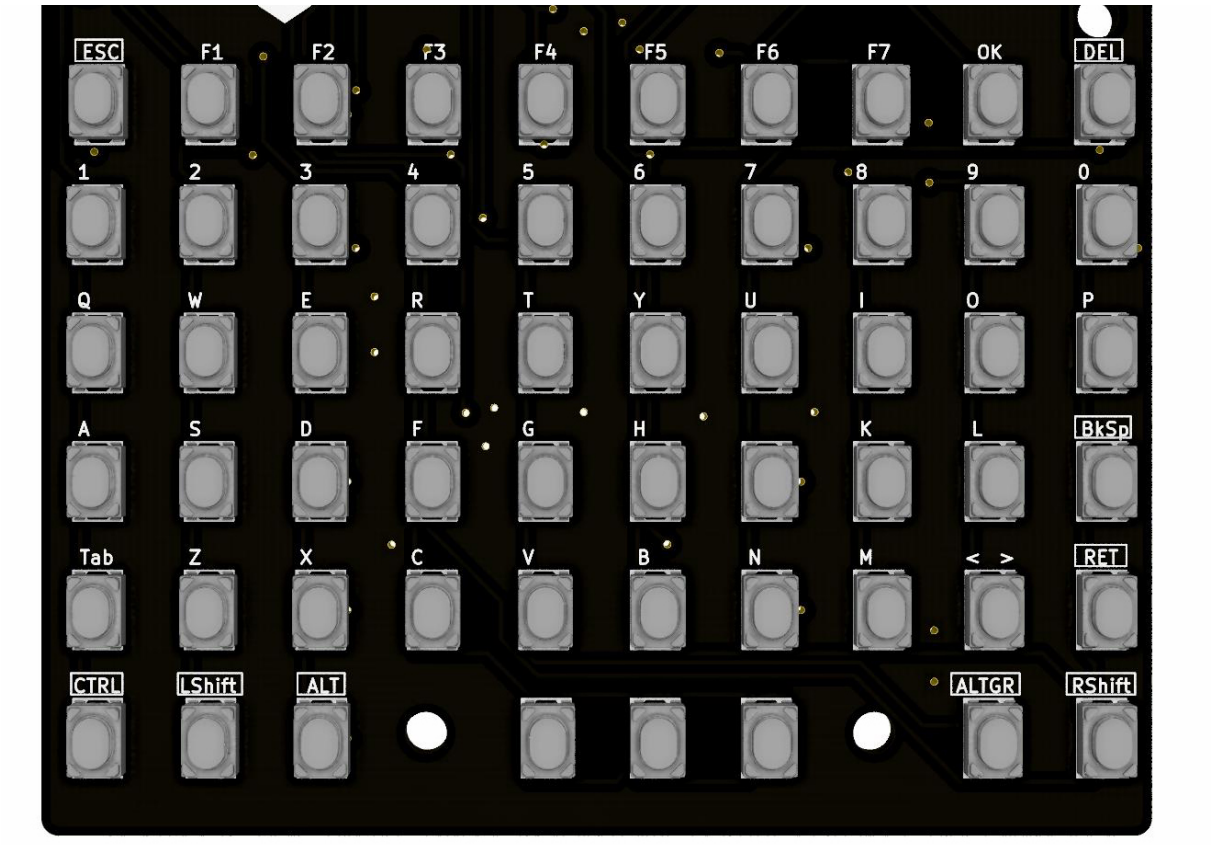


On the left you get 4 pins to connect a UART optional module or an I2C sister board. For example, you can add an optional display with the I2C interface, a GSM module such as SIM800L or newer which is often compatible with UART, an accelerometer … etc etc …

On the back side of the PC board, you get important extra connections to connect the RPI to the keyboard. You also have the necessary connectors to add the battery charging module.



The Keyboard

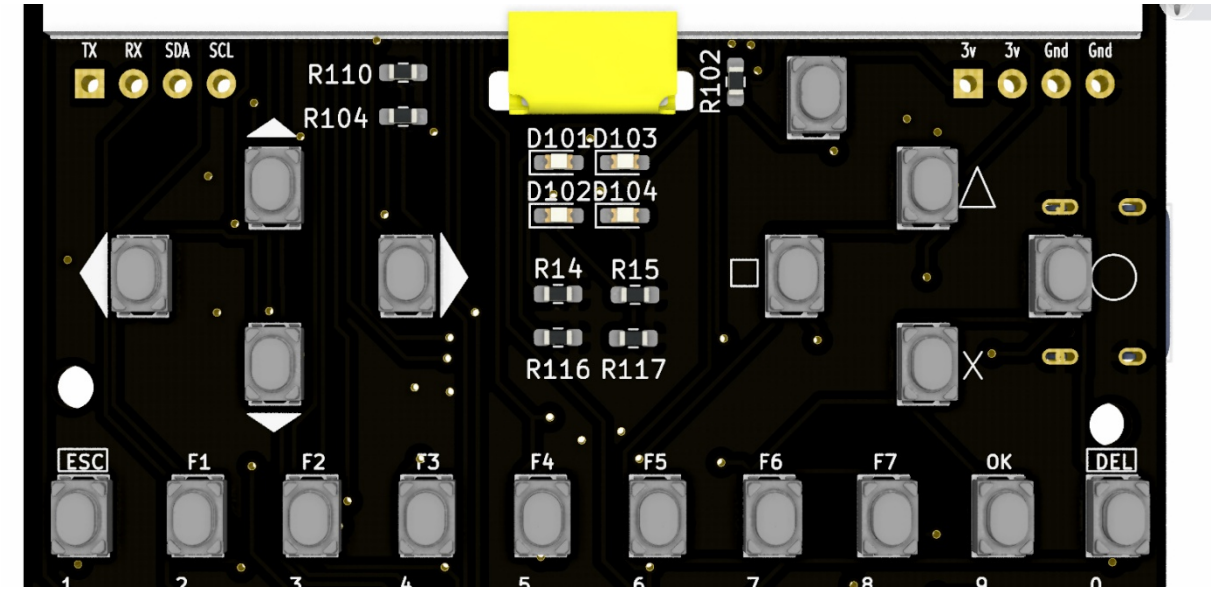


The keyboard has been designed to be a real PC keyboard. We are going to run linux in text mode so we need all kind of keys combination most keyboarded devices do not offer. To ease custom

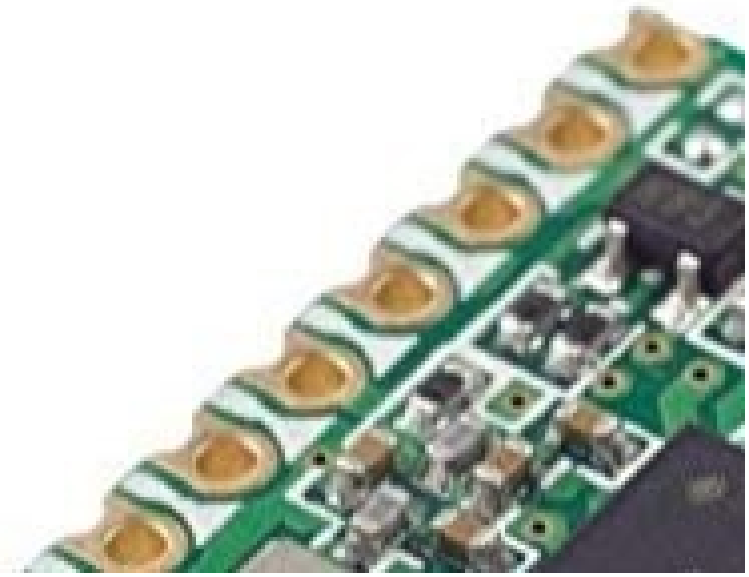
configuration , we decided to let the keyboard behave as a classical USB device you just plug into the RPI. For this, the ATMEGA32U4 chip is used. We flashed the QMK opensource firware so it can work right out of the box. Now , with QMK, you can modify the keys to any one or any script you wish.

Two joypads

Should you want to work on a port of a game system, we kept 2 joypads with 9 buttons. It will then be possible for game developers to adapt their work onto SHARPIKEEBO.



The radio module for long range communications



Installation

If the board and display come separate, please connect the display's ribbon to the white connector, ribbon contacts facing the board surface.

Software setup

The Buttons and LEDs are controllable using classic python scripts.

For the display, you have to install 2 libraries/code and compile them

1° ONE BITE DISPLAY LIB: https://github.com/bitbank2/OneBitDisplay

Install with git clone https://github.com/bitbank2/OneBitDisplay

Go to /linux/Sharp_LCD directory and type make.

2° ARMBIANIO LIB: https://github.com/bitbank2/ArmbianIO

Install the lib with **git clone https://github.com/bitbank2/ArmbianIO**

and make

Go back to Go to /linux/Sharp_LCD directory and run the compiled executable ./sharp_lcd

Have fun !

Dr CADIC Philippe

Special credits to : Larry Bank for hi precious help and support in this project.
@fast_code_r_us

Link to new LCD if an accident occurs

Link to SHARP Display: https://fr.aliexpress.com/item/4000674316500.html

QMK Firmware : https://docs.qmk.fm/#/

Link to battery module: https://fr.aliexpress.com/item/32999118705.html

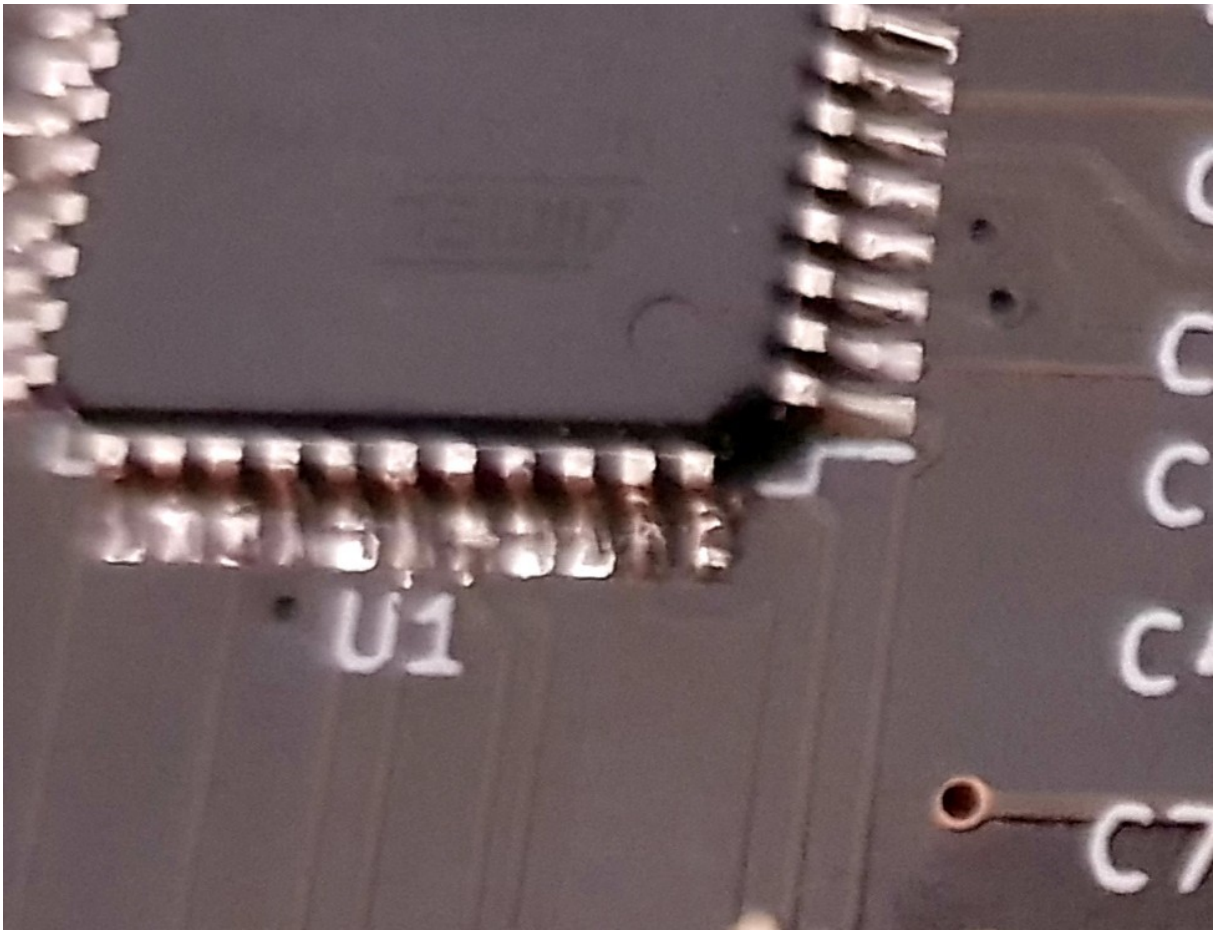
Link to RFM95 Radio module: https://www.digikey.fr/fr/products/detail/rf-solutions/RFM95W-868S2/5051755

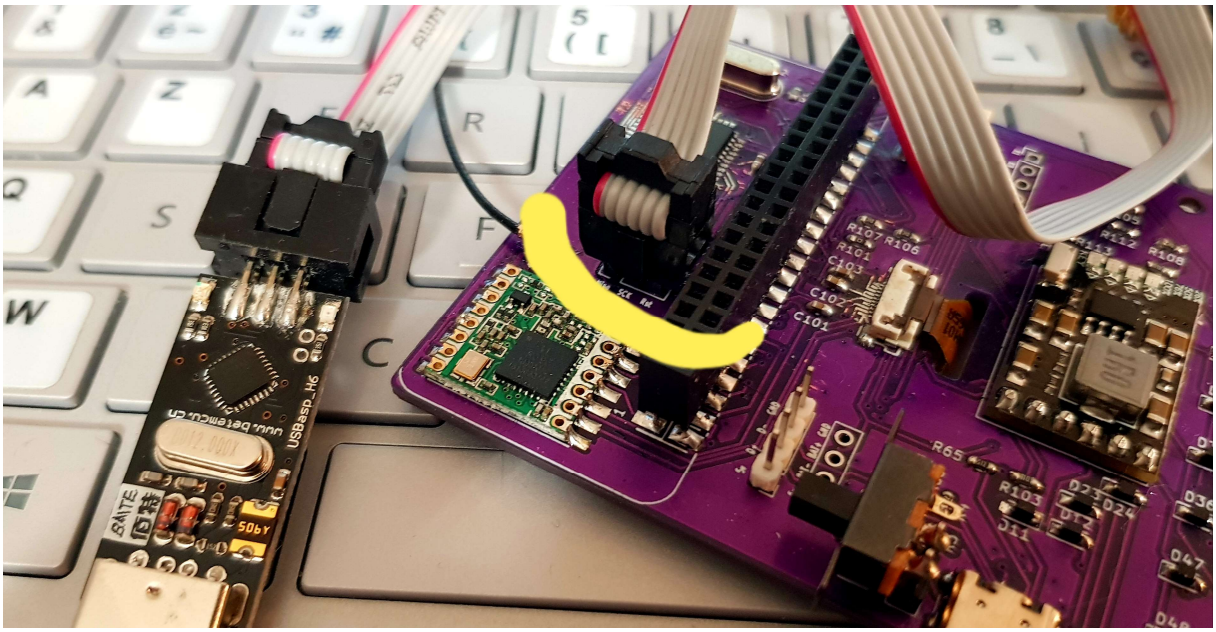
How to flash the ATMEGA32U4

File is here:
https://github.com/sparkfun/SF32u4_boards/blob/master/sparkfun/avr/bootloaders/caterina/Caterina-promicro16.hex

You will need to use a bootloader .HEX file which is compatible with the ATMEGA32U4 16 MHZ crystal version. (5v arduino pro micro equivalent). I used the Caterina-Promicro16.HEX for this.

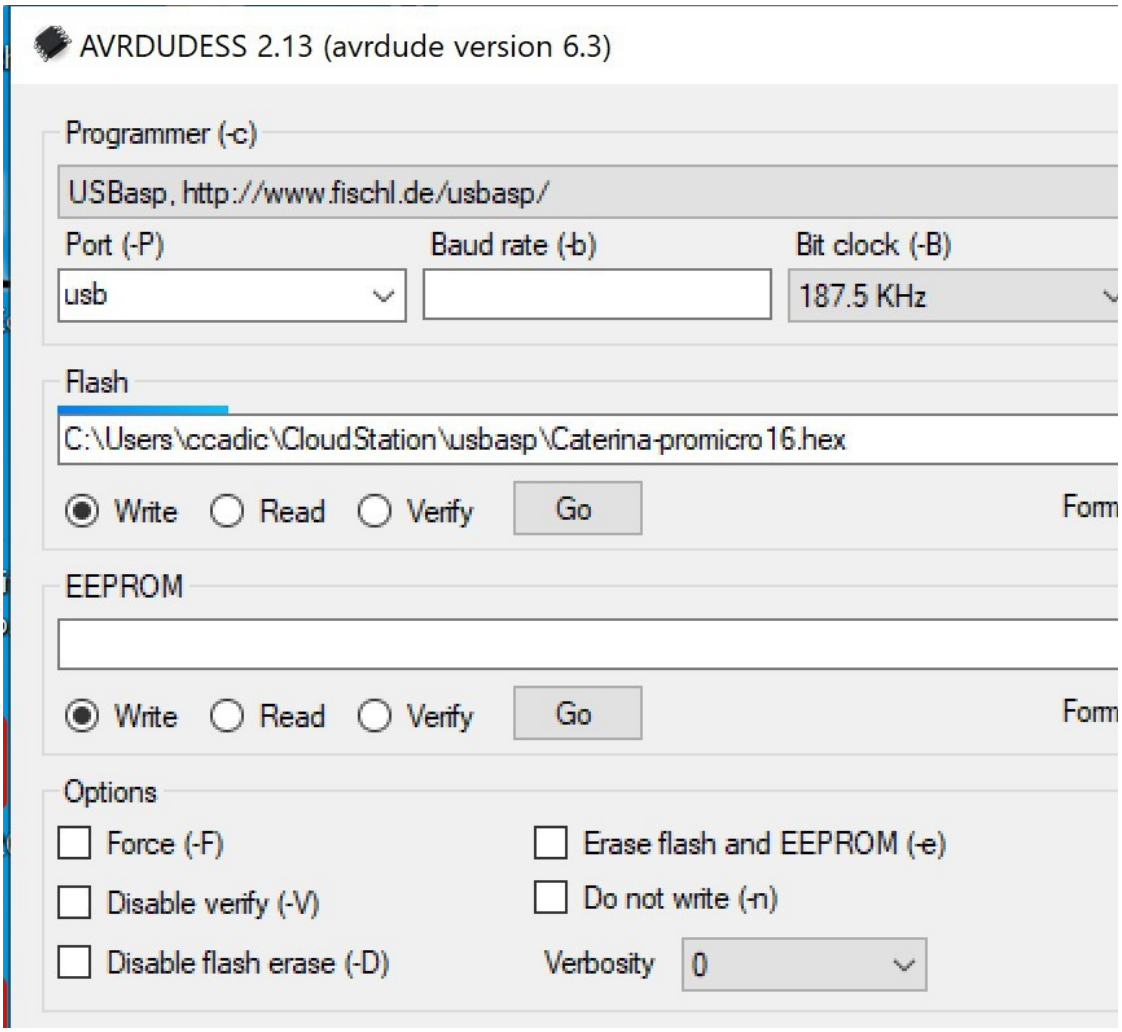
To flash the ATMEGA32U4, I used AVRDUDE with its GUI interface AVRDUDESS . You need a ISP atmel programmer or similar. Connect it on the yellow ISP programming plug. Double the plug orientation before you flash. Im using a basic USBAsp H6 low cost programmer.





Cable orientation

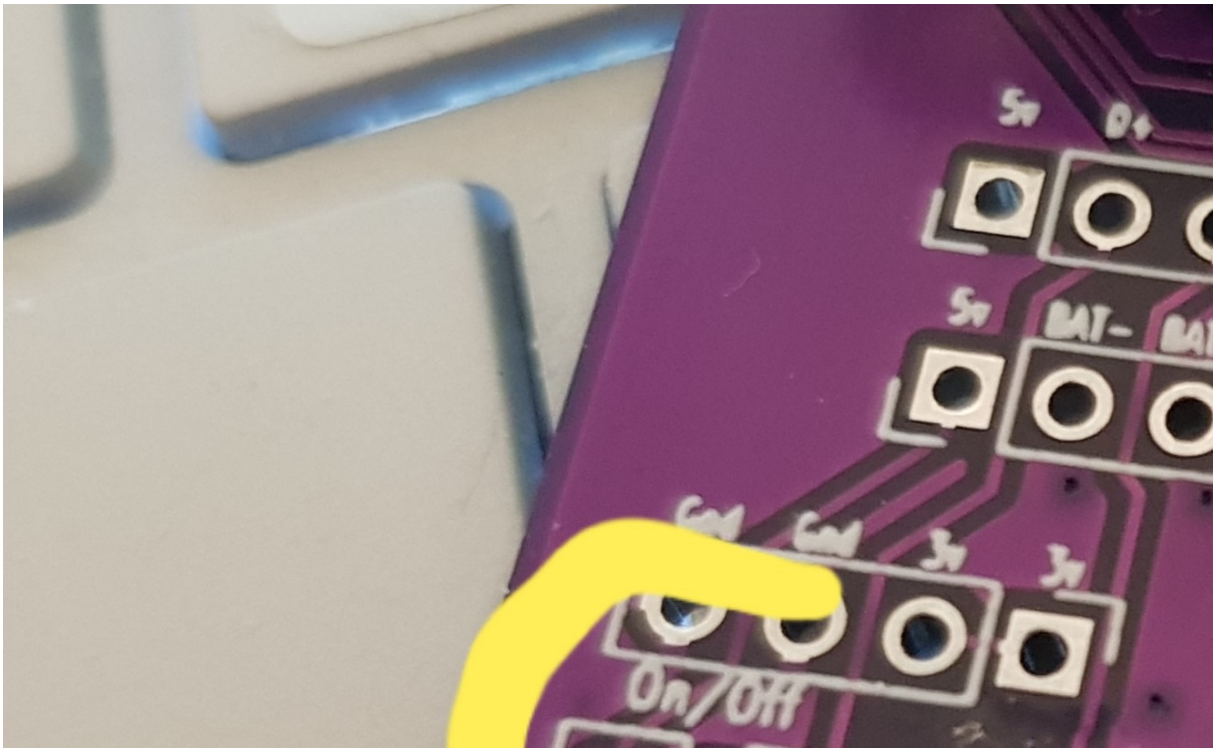
Take care when you flash the bootloader. You will need to define FUSES and BITLOCK with care. Here is the screen capture of avrdudess which will let you burn the blank ATMEGA32U4



Before you burn the HEX, you will have to click "Detect". It should display ATMEGA32U4. Then you have to press both "WRITE" buttons to write FUSES and LOCKBITS. Once done, select the Caterina-Promicro16.HEX to fill the FLASH textbox, then click PROGRAM! This should upload the ardiuno bootloader like a charm.

Caution, your ATMEGA should be recognized as an Arduino Pro Micro when you plug it onto a PC/MAC. You now have to upload the KB sketch. Caution, you do not upload the sketch though the ISP plug. You have to remove the ISP programmer. To flash the KB sketch, you will use a basic USB C cable.

Caution, to be able to upload the sketch, you MUST place an ON/OFF switch or short the 2 holes (se photo below)



You will need to add a ON/OFF switch to be able to run the system, otherwise the USB keyboard will not be recognized and the RPI will not power on.

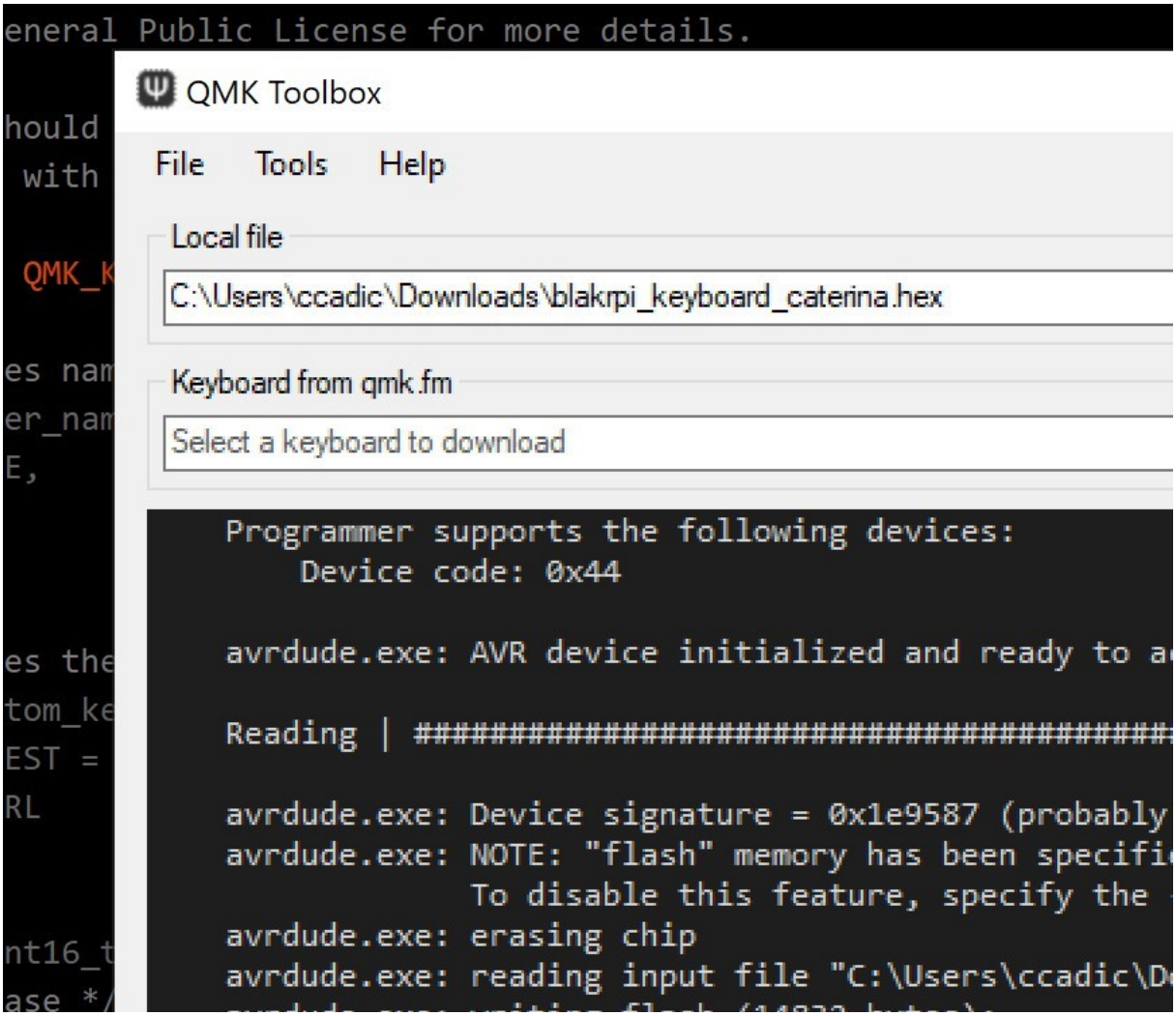
Flashing the QMK Hex file

FLASHING THE QMK HEX connect the USB C plug to the computer where QMK Toolbox is installed.

Open QMK Toolbox, select blakrpi_keyboard_caterina.hex and flash

File is here: https://cdn.hackaday.io/files/1781087620172672/blakrpi_keyboard_caterina.hex

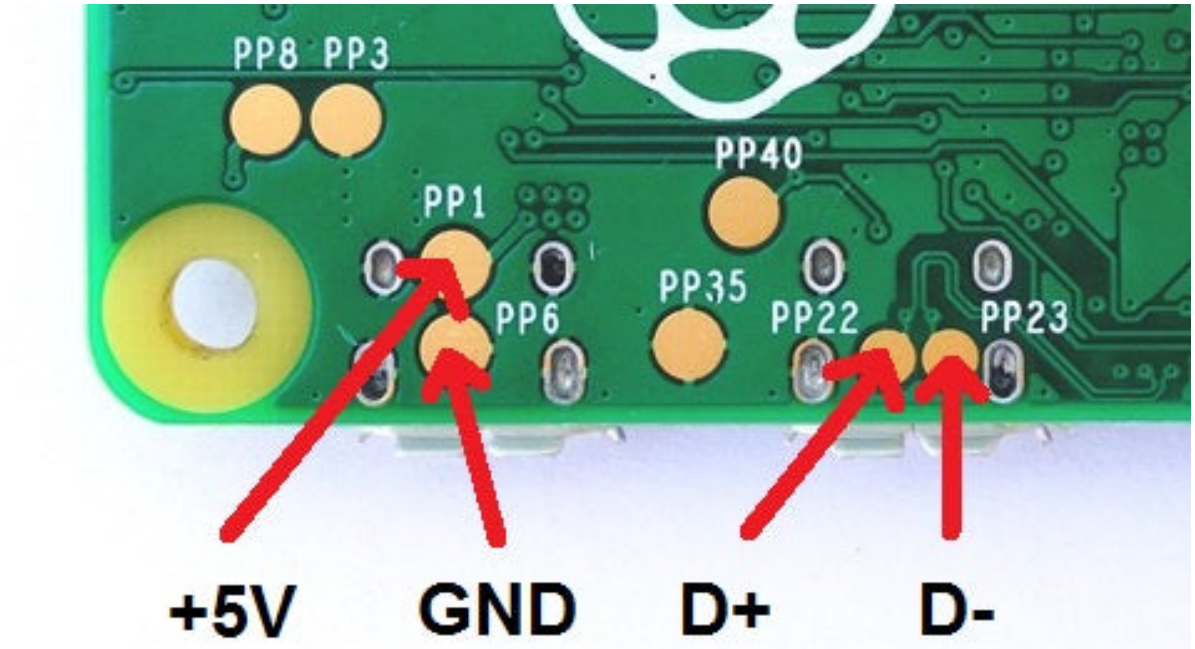
Remove the usb c plug and reconnect it. Then you can test.



Connecting the USB keyboard

You can connect the KB to the Raspberry using the the PCBboard USB C plug and the USB micro USB connector on the boarder of theRPI ZERO 2W you will need a cable to do this.

There is another solution but you need a solder iron. You can solder 4 wires 5v/D+/D-/GND from the PCboard to the back or the RPI0 2W see photos below. Doing this way, you will be able to have the KB permanently connected and you can hide the wires inside the enclosure.



Adding a UART TR/RX GSM module

We added connectors to you to add a small GSM module (SIM800L or similar). There are RT/TX pins available on the PCB. You also have voltage sources. See photos below.