How to Make a J-LINK by Raspberry Pi Zero

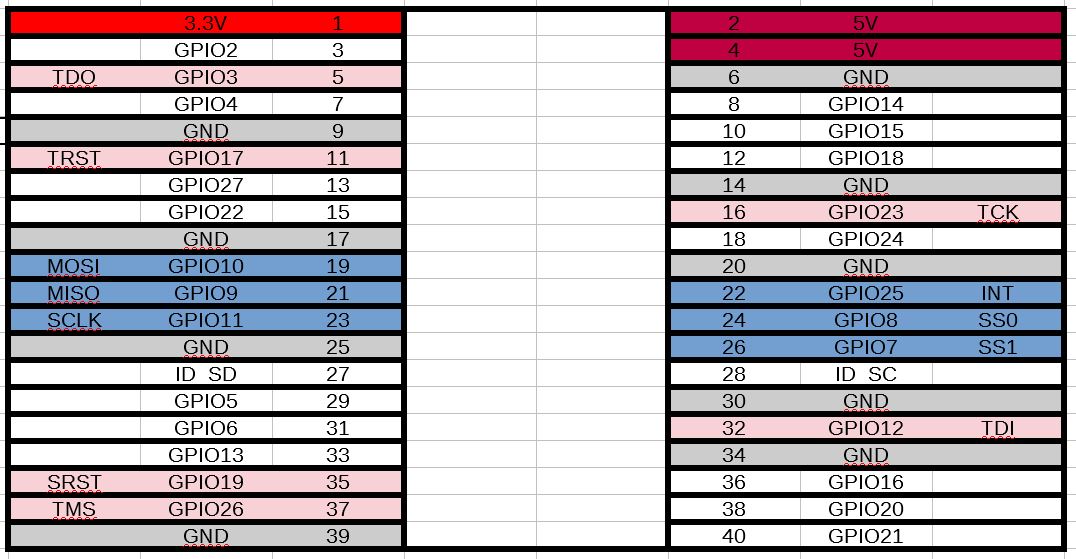
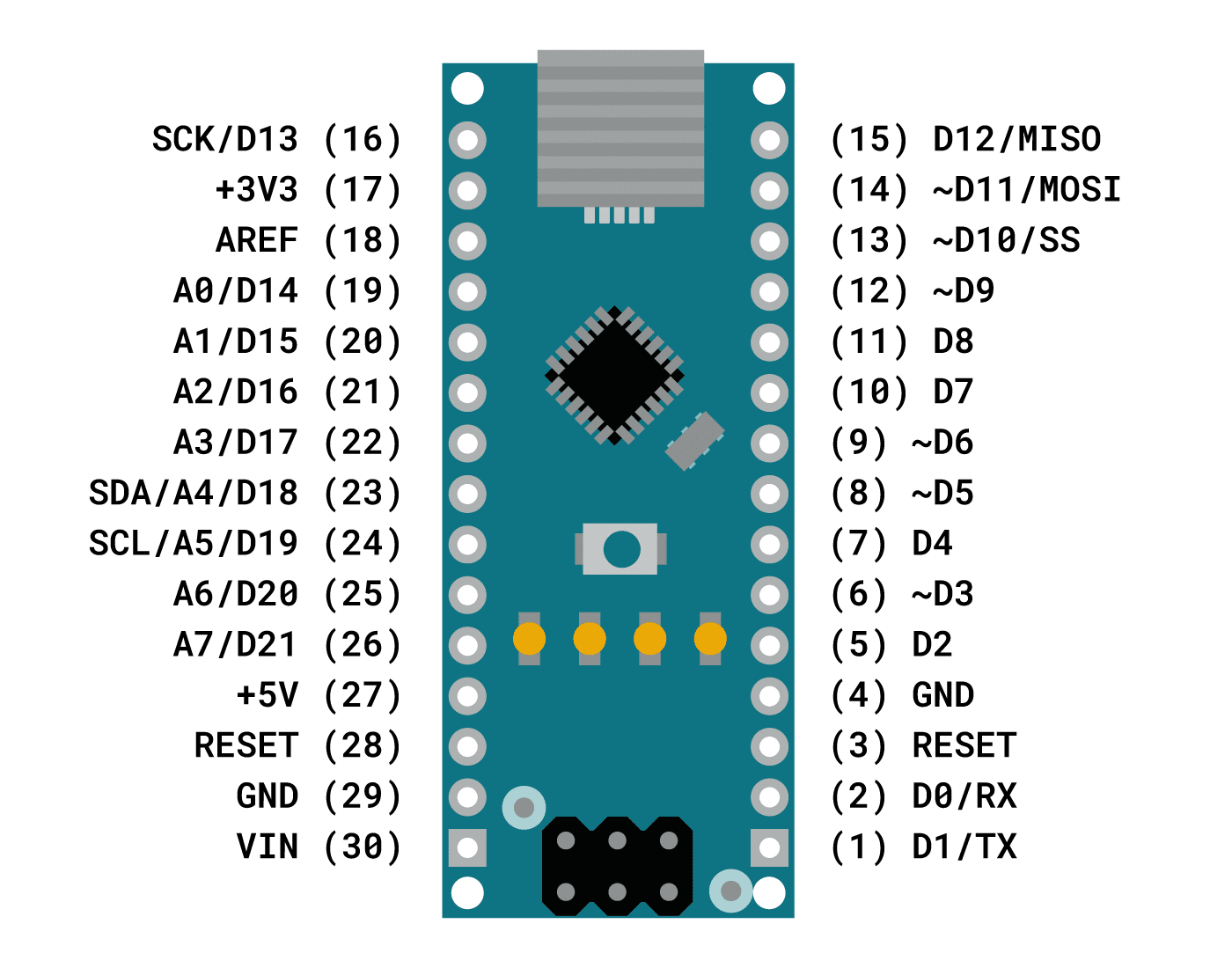
1. Set up Raspberry Pi Zero
   1. Download Raspberry Pi Imager from <https://www.raspberrypi.com/software/>
   2. Flash Debian TF card
   3. Edit /boot/config.txt and add ‘enable\_uart=1’ to the end of the file, and save.
   4. Login by user: pi, pass: raspberry
2. Set up ethernet extension board
   1. Connect a Raspberry Pi Zero to an ethernet extension board, pinout as below

Take an ethernet extension PCB from Amazon

(https://www.amazon.ca/gp/product/B087TD33KM/ref=ppx\_yo\_dt\_b\_asin\_title\_o00\_s00?ie=UTF8&psc=1)

as example:

|  |  |  |
| --- | --- | --- |
| Signal name | Raspberry Pi Zero | Ethernet Extension |
| SCK | Pin 23(GPIO14) | Pin 16 |
| MISO | Pin 21(GPIO13) | Pin 15 |
| MOSI | Pin 19(GPIO12) | Pin 14 |
| SS | Pin 24(GPIO10) | Pin 13 |
| INT | Pin 22(GPIO6) | Pin 4 of ENC28J60 on PCB |



* 1. Edit /boot/config.txt and add ‘dtoverlay=enc28j60’ to the end of the file, and save.
  2. Enable SPI as below:

#sudo raspi-config

Go to Interface options, enable SPI

* 1. Assign a static IP

sudo nano /etc/dhcpcd.conf

Add:

interface eth0

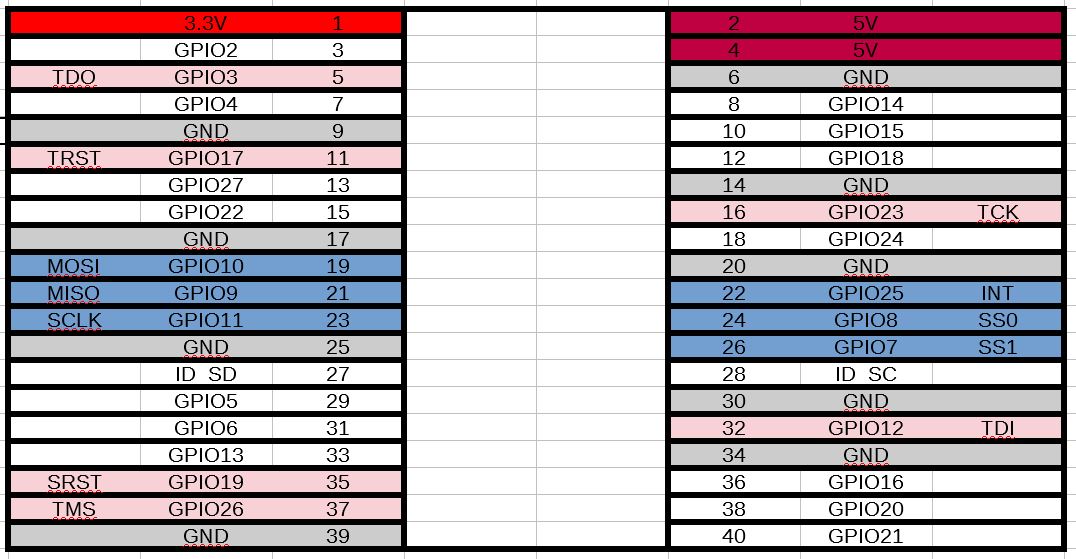
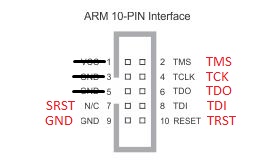
static ip\_address=192.168.1.142/24

static routers=192.168.2.1

static domain\_name\_servers=192.168.2.1

* 1. Reboot

1. JTAG Pinout



1. Install and config OpenOCD (Also see: <https://neucrack.com/p/142>)
   1. sudo apt-get update
   2. sudo apt-get install -y autoconf libtool libftdi-dev libusb-1.0-0 libusb-1.0-0-dev git
   3. sudo apt-get install gpiod
   4. sudo apt-get install libgpiod-dev
   5. sudo apt-get install git
   6. git clone --recursive git://git.code.sf.net/p/openocd/code openocd-code
   7. cd openocd-code
   8. ./bootstrap

./configure --enable-sysfsgpio\

--enable-linuxgpiod \

--enable-maintainer-mode \

--disable-werror \

--enable-ft2232\_libftdi \

--enable-ep93xx \

--enable-at91rm9200 \

--enable-usbprog \

--enable-presto\_libftdi \

--enable-jlink \

--enable-vsllink \

--enable-rlink \

--enable-arm-jtag-ew \

--enable-dummy \

--enable-buspirate \

--enable-ulink \

--enable-presto\_libftdi \

--enable-usb\_blaster\_libftdi \

--enable-ft2232\_libftdi\

--prefix=/usr

* 1. make
  2. sudo make install
  3. git clone <https://github.com/Neutree/MT7688-OpenOCD> (download the cfg file from Angelic47)
  4. Edit jlink\_gpio.cfg for JTAG I/Os

adapter driver linuxgpiod

transport select jtag

# tck tms tdi tdo

linuxgpiod gpiochip 0

linuxgpiod jtag\_nums 23 26 12 3

linuxgpiod trst\_num 17

linuxgpiod srst\_num 19

adapter srst delay 100

jtag\_ntrst\_delay 100

reset\_config trst\_and\_srst

* 1. Edit jlink\_gpio.cfg as promoted after sudo openocd -f jlink\_gpio.cfg
  2. sudo openocd -f jlink\_gpio.cfg
  3. telnet localhost 4444

1. Check the availability of GPIOs

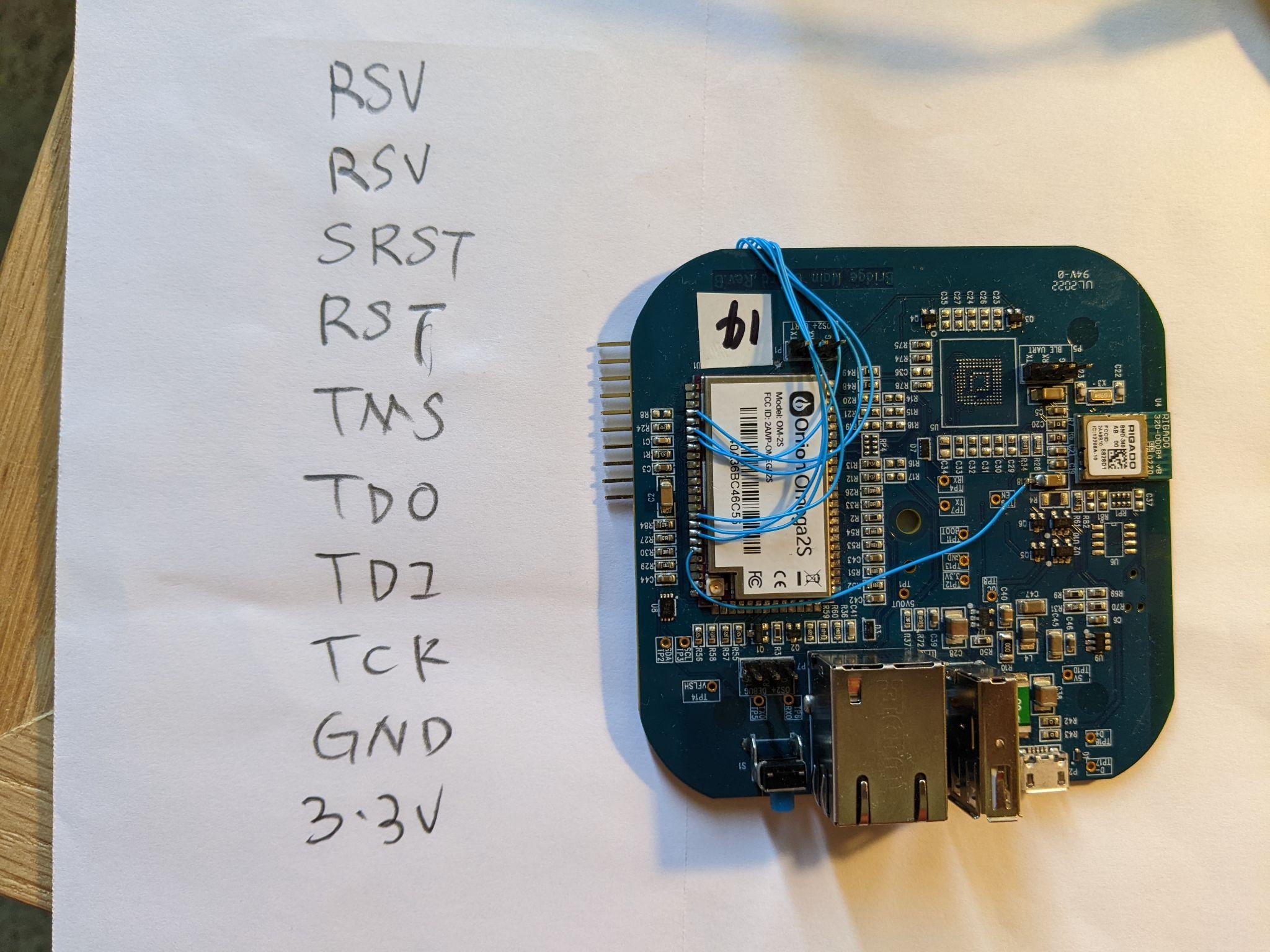
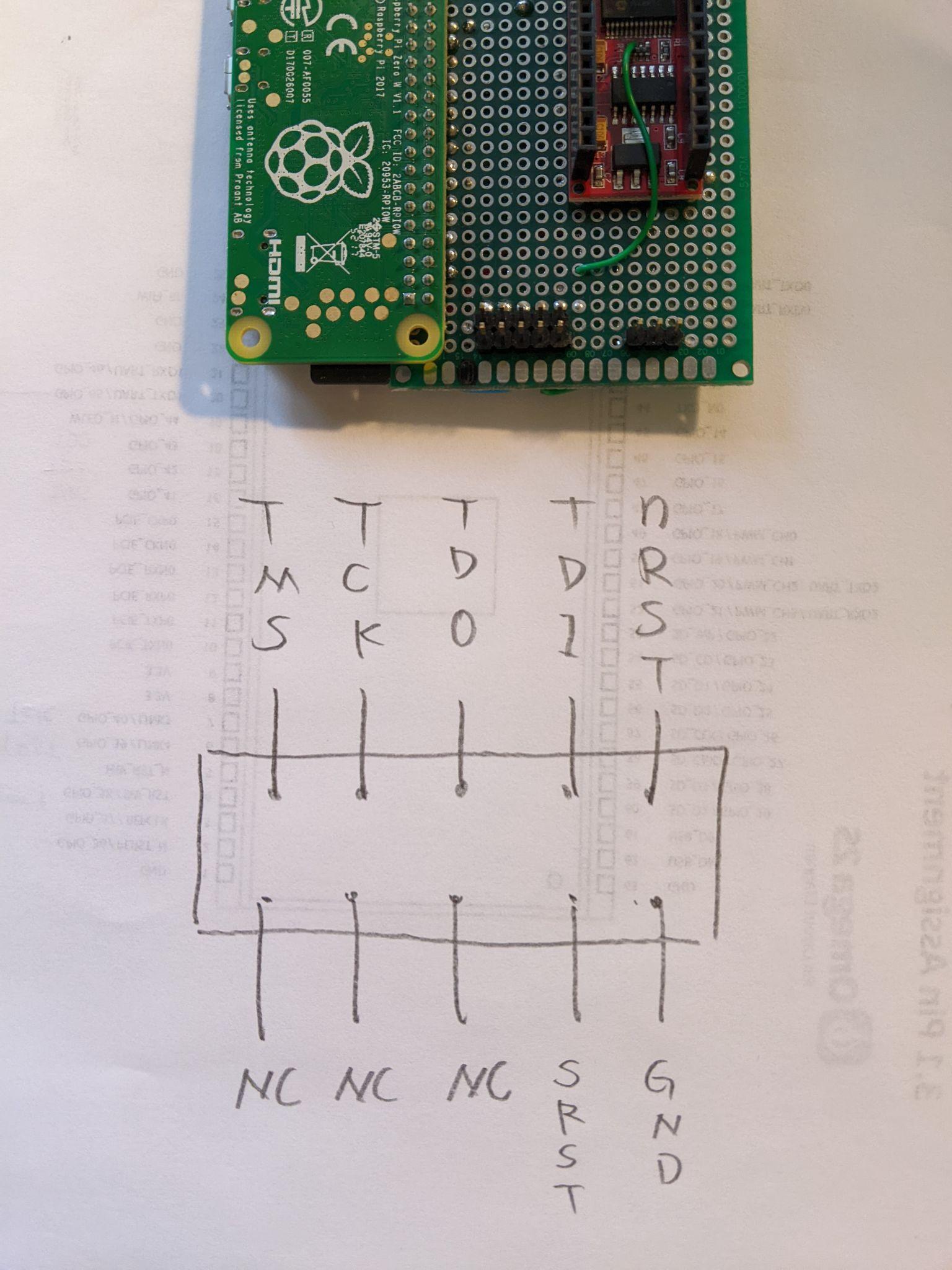
gpiodetect # detect GPIO devices

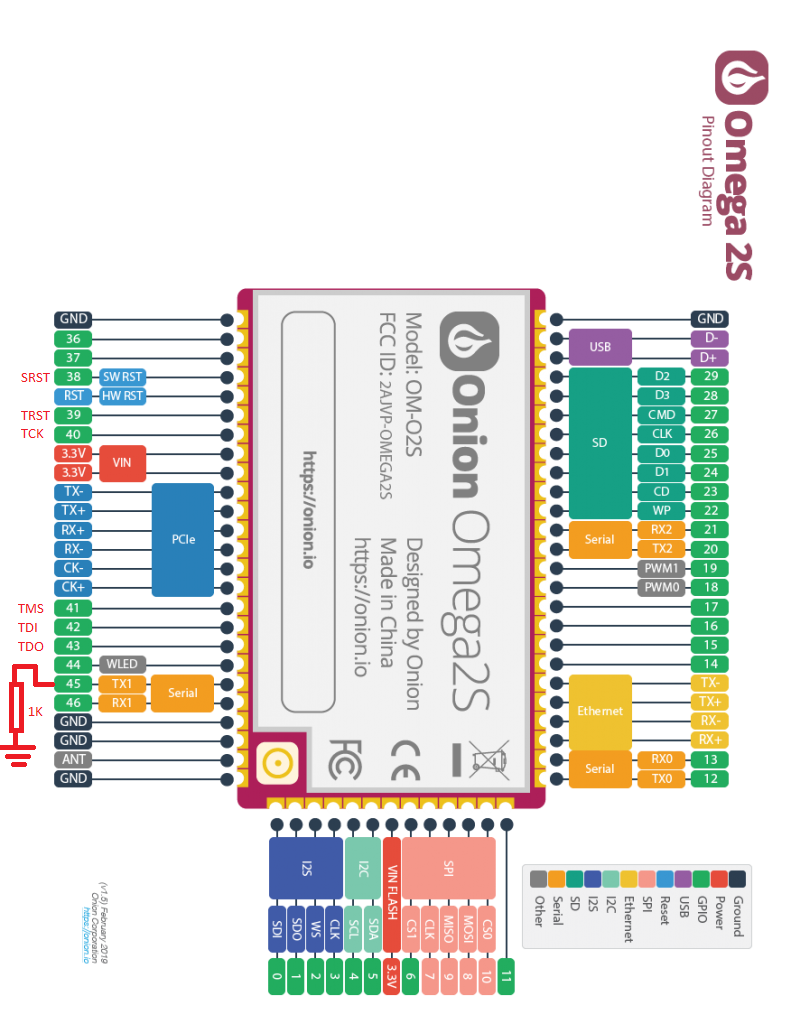
gpioinfo 0 # check the configuration of GPIO device 0

1. Install telnet

sudo apt-get install telnetd

1. Enable ssh
   1. Go into /boot
   2. sudo touch ssh
   3. Reboot
   4. sudo raspi-config
   5. Goto ‘interface options->ssh->enable it’
2. Telnet or ssh from another windows computer by PUTTY.
3. Sftp from another window computer by PSFTP
   1. lcd C:\Users\bin\Downloads
4. Rsv
5. Wiring of hardware





|  |  |  |
| --- | --- | --- |
| Signal Name | MT7688 Pin Name | O2S Pin Name |
| TDO | EPHY\_LED0, 143 |  |
| TDI | EPHY\_LED1, 142 |  |
| TMS | EPHY\_LED2, 141 |  |
| TCK | EPHY\_LED3, 140 |  |
| TRST | EPHY\_LED4, 139 |  |

1. rsv