

University of Debrecen

Faculty of Science and Technology

INSTITUTE OF PHYSICS

Department of Electrical and Electronic Engineering

Electrical Engineering BSc

**DESIGN AND DEVELOPMENT OF A CONTROLLER
FOR CONTROLLING TOUCHSCREENS FOR THE
PURPOSE OF INFORMATION BOARD**

Tran Tien Thanh

2022

TABLE OF CONTENTS

Introduction.....	2
1. Displays.....	3
1.1 LCD – Liquid Crystal Display	3
1.2 LED – Light Emitting Diode display	4
1.3 FPT-1015.....	4
2. Controlling.....	6
2.1 PC.....	6
2.2 Micro controller.....	7
2.3 Raspberry pi 4 model B.....	9
2.3.1 Specification	10
3. Method and result.....	11
3.1 Element of the system	12
3.2 Operation.....	12
3.3 Connect to Raspberry Pi.....	14
3.3.1 Enable SSH on Raspberry Pi	15
3.3.2 Prerequisite to SSH into a Raspberry Pi	17
3.3.3 Connect to the Raspberry Pi via SSH	20
3.4 Create an alternative mouse to control Raspberry Pi	20
3.4.1 Basic function of the joystick.....	24
3.4.2 Connect joystick to Raspberry Pi.....	25
3.5 Custom PCB design for breadboard.....	37
3.5.1 Kicad.....	37
3.5.2 LPKF ProtoMat S104.....	41
3.5.3 Autodesk Fusion 360	45
3.5.4 Witbox2.....	48
3.6 Google survey	51
3.7 Website design	Error! Bookmark not defined.
4. Conclusion	Error! Bookmark not defined.

Introduction

There is a need for changes in information display around any school campus, we still use traditional ways such as sticking papers onto a board using pins. We have to manually put the flyers, calendars, ... that contain information of university, internships around many places that can reach the maximum amount of students to be efficient. So it is necessary to develop a method to gain the most attention from the student with the least effort. A solution I have come up with is placing multiple touch screens that can display information in various places on the campus and it will be monitored and controlled by a microcontroller.

As a student in the institute, I come across the bulletin board in the hallway every day and for me, this traditional type of board is old fashion and because there is a lot of information piled up in a small area, it is very difficult to see what are there on the board. An eye-catching bulletin board will build interest in every student. Educators should strive to create a bulletin board that introduces new concepts in an exciting way. Interactive bulletins are perhaps the best type of display. Students will spend more time viewing and attempting to understand interactive bulletin boards, this type of board will add some excitement to this typically visual decoration.

Showing the map of each floor of the institute will help the student to find the classroom, destination as fast as they can. There will be information about the upcoming event, internship opportunities, detail about professors with their room numbers and email address, which will help student a lot throughout their studying at the University. All the information will be integrated into an interactive touch screen, with a joystick controller to make the interaction between students and the bulletin board much easier and user-friendly.

1. Displays

A display is an output device for the presentation of information visually. The display is referred to as an electronic display when the information being presented contains an electrical signal. The cathode ray tube (CRT) (*Figure 1*), invented in 1992, was the first display ever made. It works by projecting an electron gun onto a phosphor-coated screen that forms images. The earliest CRT has monochrome and used primarily in oscilloscopes and black and white TVs.



Figure 1. Cathode ray tube found in an oscilloscope

The color CRT was introduced later in 1954 and was the most popular display technology for nearly half a century. Since then, new technology has created a wide variety of displays, the most common are:

1.1 LCD – Liquid Crystal Display

This display is a flat panel or electronically modulated optical device, using light-modulating properties of liquid crystals combined with polarizers. Its application appears everywhere, from LCD television and, computer monitors to aircraft cockpit displays. Smaller LC screens are

common in LCD projectors and portable consumers such as digital cameras, watches, digital clocks, and smartphones. In almost application, LCDs displays have taken the place of heavy, clunky cathode ray tube

1.2 LED – Light Emitting Diode display

A semiconductor light source is called a light emitting diode (*Figure 2*) when it emits light when current flow through it. Recombining electrons and electron holes in the semiconductor result in the release of energy in the form of photons. The energy needed for electrons to pass through the semiconductor's band gap determines the hue of the light, which corresponds to the energy of the photons. Using multiple semiconductors or a layer of light emitting phosphor on the semiconductor device will produce white light. LED displays are capable of providing general illumination in addition to visual displays.

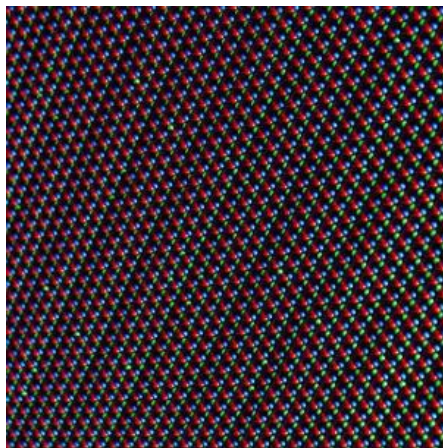


Figure 2. Detail of a LED display with a matrix of red green blue diode

1.3 FPT-1015

In this project, I am provided with a touchscreen monitor, FPT-1015 of NI (National Instruments) (*Figure 3*). This is a color thin film transistor (TFT) LCD flat panel that can be used for industrial applications and various purposes. The monitor includes VGA control signal input, the chassis is made of durable steel, and the front panel is made of Al-Mg. This monitor is a user-friendly system control interface and we use it as a remote control system.



Figure 3a. FPT 1015

1.3.1 Connection:

VGA port (DB15): We use this port to connect the display to a system via the external 15-pin DB-15 connector. Before connecting the VGA cable between the display and a PC, microcontroller, be sure that vertical frequency is within the acceptable range if we have the operating system on the PC, or microcontroller.

USB type B touch screen connector: The display includes this connector and a touch screen cable if a touchscreen is installed. We must connect it to the PC USB port.

DC 12V power in: This display needs to connect to the 12V switching power supply.



Figure 3b. Back view of the touchscreen

1.3.2 Specification:

The FPT-15 display has a heavy-duty steel chassis made from Al-Mg with NEMA4/IP65 compliance. There are physical controllers on the front side of the display, onscreen display (OSD) control pad. A mounting hole on the back of the display enables mounting the display on a wall, rack, desktop or VESA arm, and it has a resolution of **1024 x 768**.

Construction.....	Heavy duty steel chassis
Front Panel	Al-Mg, NEMA4/IP65 compliant
Control	Onscreen display (OSD) control pad on front site
Mounting.....	Panel, wallmount, rackmount, desktop or VESA arm
Weight.....	4.5kg
Vibration (operating)	5 to17 Hz, double amplitude displacement, 17 to 500 Hz, 1.0 G peak to peak
Power consumption.....	+12V $\pm$ 10% @ 2.5 A

2. Controlling

Technically, we can control the touch screen with many options, such as using microcontroller like Raspberry Pi or a PC with a proper Operating system like Windows, macOS, or Linux,... or we can even connect the display with a Laptop and control it by dual monitor.

2.1PC

A personal computer is a multi-purpose microcomputer whose size, capabilities, and price make it feasible for individual ways of using. Instead of being used by a specialist or a technician, personal computers are designed to be operated directly by the end user. A computer consists of at least one central processing unit (CPU) in the form of microprocessor, along with some type of

computer memory, typically semiconductor memory chips. A sequencing and control unit can alter the order of operation in response to data stored while the processing element performs arithmetic and logical operation.

2.2 Micro controller

So what is a microcontroller^[1]? On a small scale (**Figure 4**), microcontroller is a small computer on a single VLSI integrated circuit (Very large scale integration). One or more CPU (processor core), memory, and programmable Input/output peripherals are all included in a microcontroller.

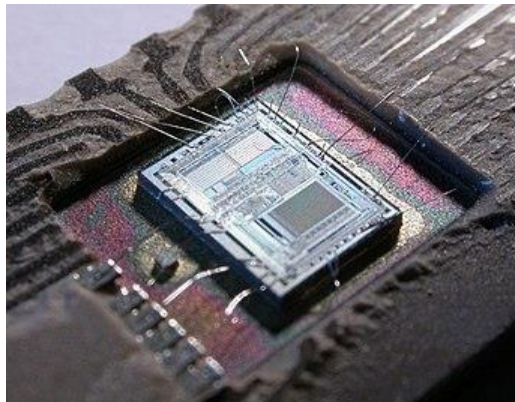


Figure 4. 8-bit microcontroller, including a CPU, RAM, and I/O port in the same chip

Automotive engine control systems, implantable medical devices, remote control devices and other automatically controlled equipment and electronics use microcontroller. There are many types of microcontrollers which have been developed, popular among all are:

PIC microcontroller

ARM microcontroller

8015 Microcontroller

AVR microcontroller

In a larger scale, a microcontroller is an Integrated Circuit (IC) (**Figure 5**), a combination of multiple small microcontrollers, that use for controlling some part of an electronics system, usually

a microprocessor unit (MPU), memory, and other peripherals. It's like a micro-computer, designed for embedded applications, making interaction with digital, analog, or even electromechanical components.

For example, an Arduino is a development platform that centers around easily programmable board with a microcontroller at its core.

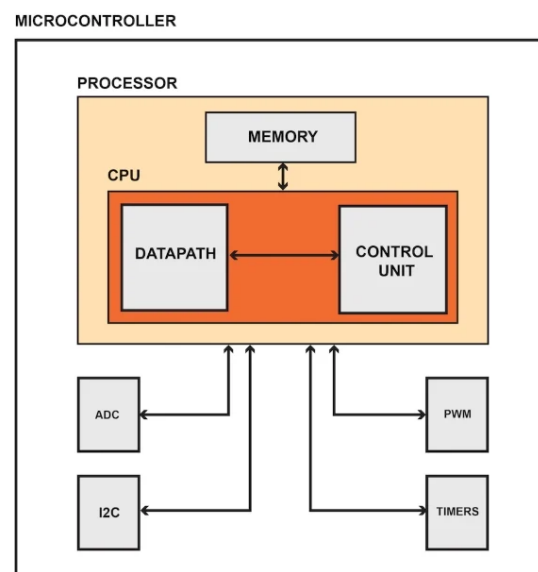


Figure 5. Microcontroller block diagram [3]

Central process Unit (CPU): It is the brain of the microcontroller, it has an Arithmetic Logic Unit (ALU) and a Control Unit (CU). It receives input, analyzes it, and then performs actions in accordance with the program stored in its memory. It collaborates with RAM and ROM to swiftly and effectively generate the desired output.

RAM (Random Access Memory): It is the data area used by the microcontroller to temporarily store constant and variable values needed during the program execution. A specific microcontroller's physical RAM storage capacity varies from one microcontroller to the next.

ROM (Read Only Memory): It is where all of the program instructions are stored. The CPU read its instruction from the ROM and carries out each one individually. In contrast to RAM, ROM must keep data even when the device is turned off. Two Partitions separate it, one is storage of

program code and other is set aside for the long-term archival of data that the microcontroller uses to run the program normally.

Input/output ports: These port can be compared to the sensory input and limbs of the controller. They are used for connecting to external devices and for communication.

ADC (Analog To Digital converter): The ADC converts the analog input signal from the external sensor to digital signal. This is necessary because the CPU needs digital input to operate. In this way, ADC is a bridge between analog input and the CPU.

DAC (Digital To Analog converter): This is the inverse of the ADC. This converter converts the digital signal to pure analog output, then is further used to control external devices.

In this project, I will use Raspberry pi 4 Model B as the controller for the FPT-1015 touch screen. Raspberry pi is a mini size computer that is affordable and runs in Linux. Little history of Raspberry Pi, this microcontroller is made by Raspberry Pi Foundation, based in UK, the goal is to advance the education of adults and children in field of computer and related subject.

2.3Raspberry pi 4 model B

Raspberry Pi ^[2] is a series of small single-board computer (SBCs) –this microcontroller is a complete computer built on a single circuit board, with microprocessor, memory, input/output, and other feature required of a functional computer.



Figure 6. Raspberry pi 4 model B

Raspberry pi 4 model B (**Figure 6**) is the latest version of the Pi family, it is the most powerful, have a major increase in processing power, enhanced peripheral connectivity, performance, memory compare to prior-generation, while retaining backward compatibility and similar power consumption.

2.3.1 Specification

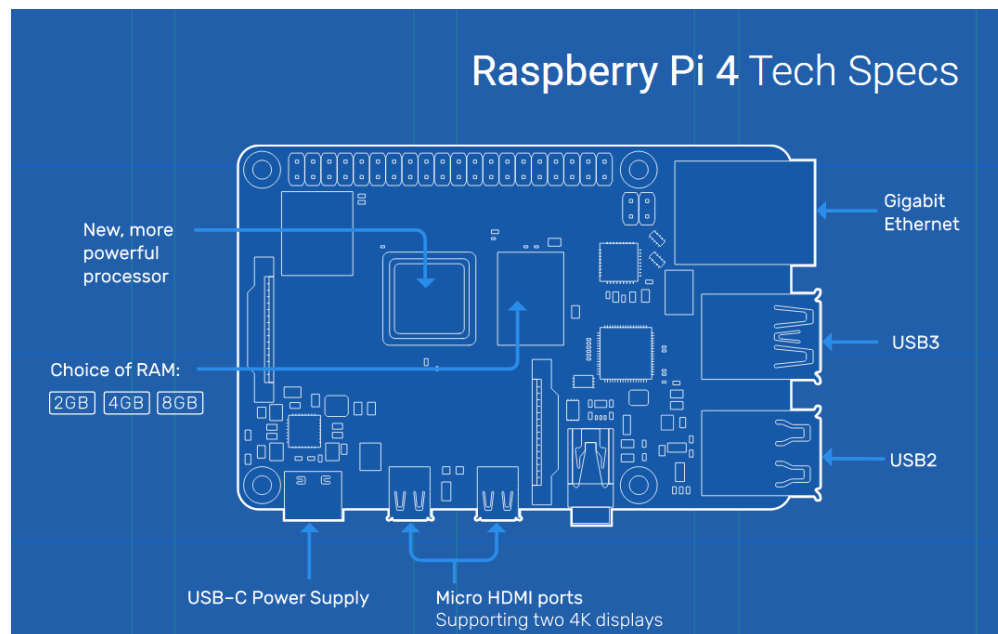


Figure 7. Component of a Raspberry Pi

[4] This Pi key feature include high performance 64-bit quad-core processor, dual display support at high resolution via 2 micro HDMI port, up to 8GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet connection, USB 3.0, and PoE capability. Dual band wireless LAN and Bluetooth have modular compliance certification, so it allows the board to be designed into end product with significantly reduced compliance testing.

This raspberry Pi is famous for its small size, just about 3.4x2.2x0.4 inches, and weight around 46 grams. As you see in the specification, there is 40-pin GPIO header. This bristling row of

input/output pin provides direct access for connecting external devices. We can connect a small size fan to cool down the CPU or a Joy-stick act like a mouse in this project.



Figure 8. Raspberry Pi size in real life

3. Method and result

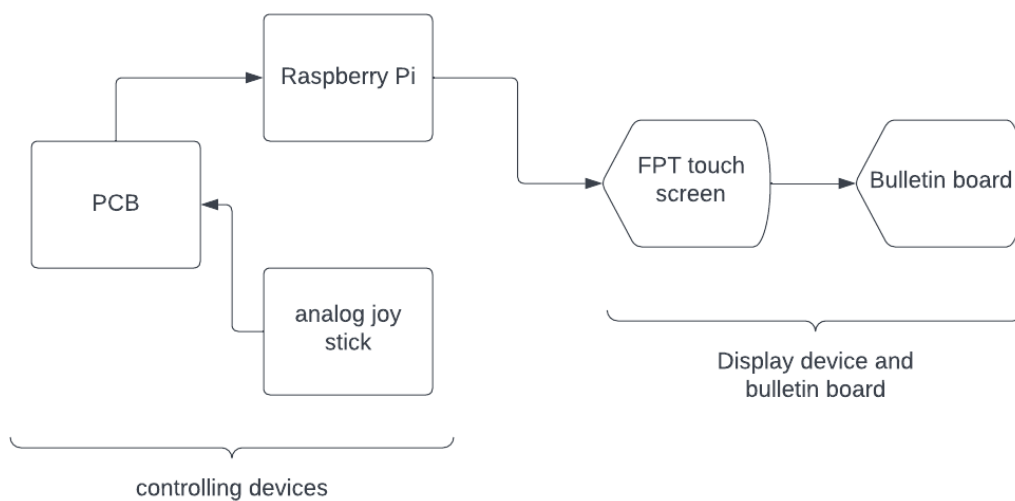


Figure . Raspberry Pi size in real life

This is the block diagram of the system (**Figure 8**), which will be implemented by assembling all of the components. The main purpose is to have a Raspberry Pi as a controller to for the FPT-1015 touch screen, the Pi is also connected to a PCB, where an ADC microcontroller MCP3008 is located, to read the signal from an analog joystick, which will be also used to control the touchscreen interaction. In the display of the touchscreen will be a website work as a bulletin board.

3.1Element of the system

- Raspberry Pi 4 model B.
- HDMI cable.
- Micro SD card.
- Power supply cable.
- Wire (female to female connection).
- Analog Joy-stick.
- FPT-1017 touchscreen.
- PCB with MCP3008 microcontroller.
- 3D printed cover box.

3.2Operation

To operate the new Raspberry Pi, we need to install an operating system to use it. It is called Raspberry Pi OS (previously call Raspbian). First, we need to have a brand new SD card to install the operating system, using the card reader attach the MicroSD card to the computer, or laptop, and install the app “SD card formatter” (**Figure 9**). The SD card will be automatically chosen. If not remove and reattach it and refresh. Choose the quick format option and then click Format. After that, the card is completely formatted and ready to be used.

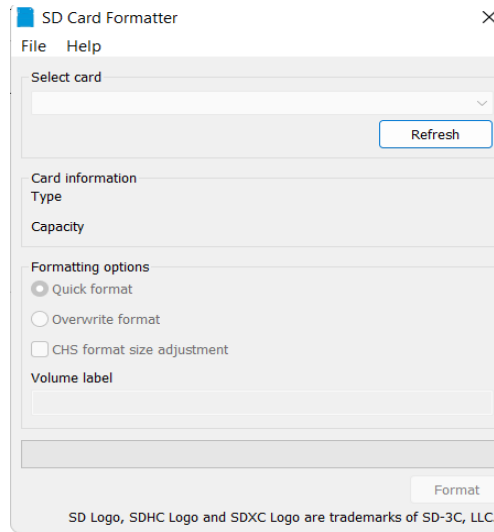


Figure 9. *SD card formatter software*

Raspberry pi Imager (**Figure 10**) is a new software, it is a quick and easy way to install Raspberry Pi OS to the SD card. Download the Imager with the new SD card that has been formatted and attached to the computer, run the application and install the operating system to the card.



Figure 10. *Raspberry Pi Imager*

Now set up the Raspberry Pi, connect Pi to the touch screen by HDMI cable with the right HDMI port on the Pi, connect mouse and keyboard to Pi by USB ports. Underneath the Raspberry

Pi, there is a socket where we can insert the SD card. Carefully insert the card into the Pi and we can start up the Pi.

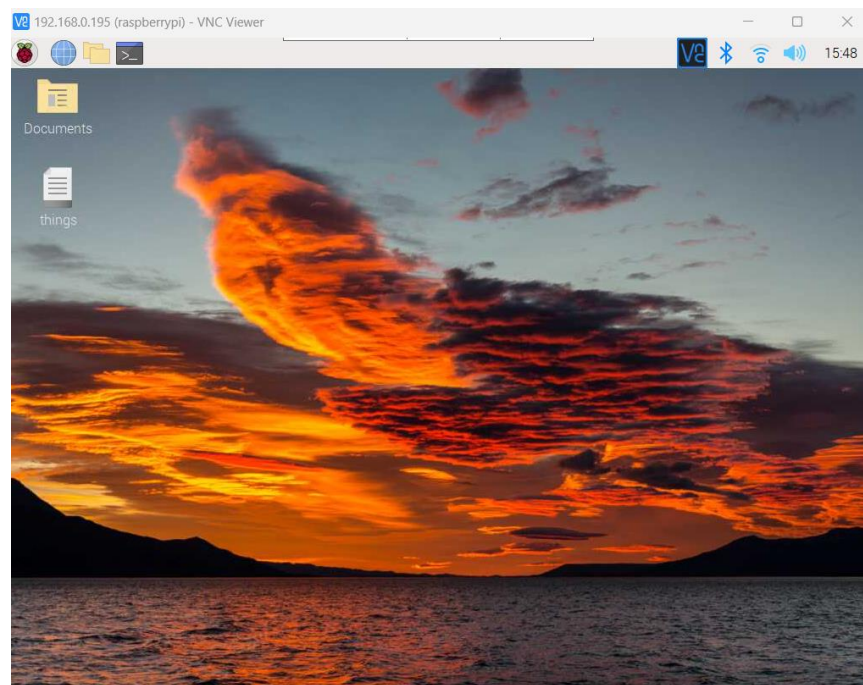


Figure 11. Raspberry Pi's screen

After plugging in the raspberry Pi with a power supply, after a few seconds the screen will turn on (*Figure 11*).

3.3 Connect to Raspberry Pi

The main way to connect and control the Pi is connecting it to FPT-1015 touch screen. On the side of the Pi, there are 2 ports of HDMI (*Figure 12*), we can directly connect it through the ports with the right cable.

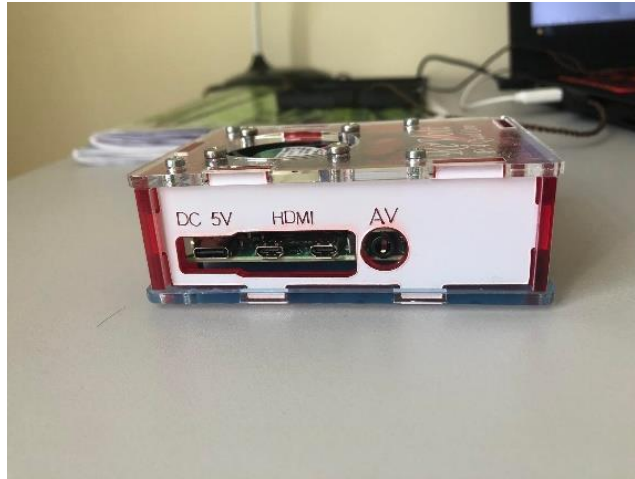


Figure 12. Ports of Raspberry Pi

The other option is to remotely control the Pi when I do not have access to any type of display, by enabling SSH. This protocol can be used to send commands to the Raspberry Pi from another computer, in this case, my study laptop, by Linux system. SSH stands for Secure Shell, it is widely used network protocol that enables secure connections between computers. Authentication is necessary, and the two computer- laptop and Pi, is encrypted.

3.3.1 Enable SSH on Raspberry Pi

On the Raspberry Pi OS, SSH is by default deactivated. So, turning it on is the first step. Depending on the setup and the installation process, I have found and successfully try 3 different ways to approach the problems:

3.3.1.1 Raspberry Pi OS with desktop display

The Raspberry Pi is connected to a display and we can operate the Pi by the display, we choose Raspberry Pi configuration tool under Preferences on top left (**Figure 13**) . After we click on “Interfaces” tab, we will choose Enabled option for SSH. Close the window and SSH is ready to use.

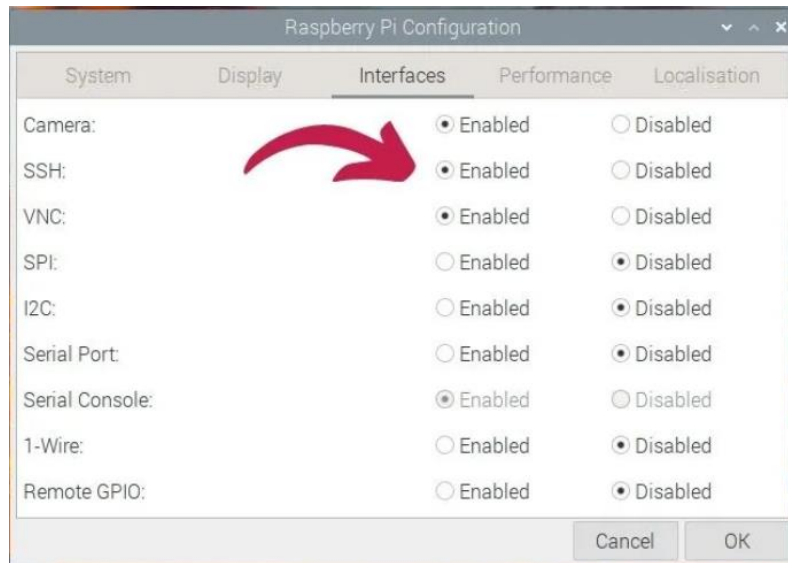


Figure 13. Pi configuration

3.3.1.2 During the SD card creation

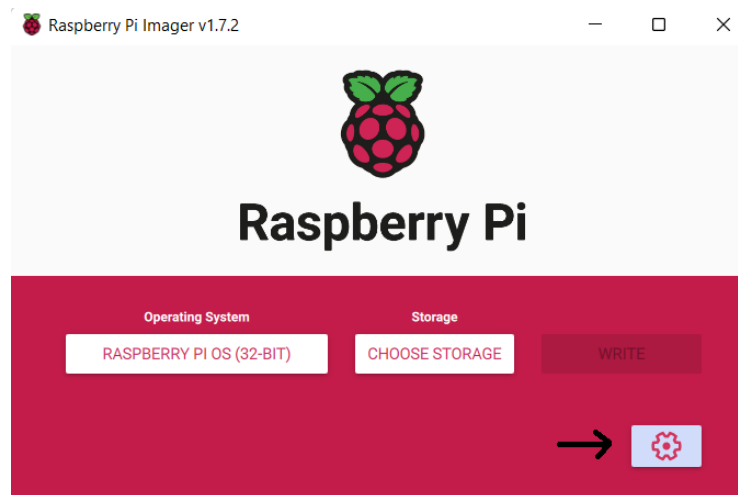


Figure 14. Raspberry Pi imager

After picking the version of Raspberry OS and the correct SD card, we can check the box “Enable SSH” and choose the user name and password of the wireless connection/ Wi-Fi (**Figure 15**) that we want the Pi connected exactly when it is powered. SSH will be enabled on the first boot.

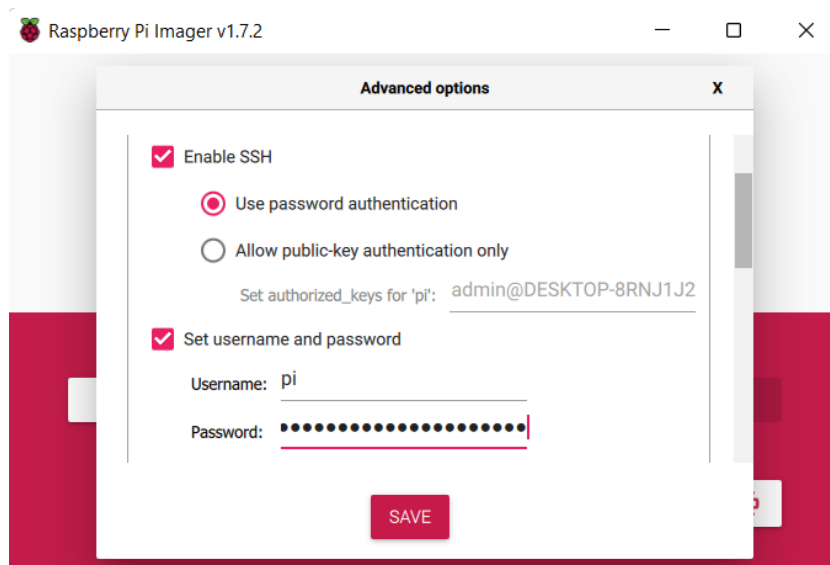


Figure 15. Raspberry Pi Imager

3.3.1.3 Manually

When I have a SD card that already has the OS, I can get the SSH enabled manually. Insert the SD card to the computer and open the file of the card, usually name “boot”, right click and create a new file named “ssh”, without any extension. Safely remove the SD card and SSH will be enabled on the next boot.

3.3.2 Prerequisite to SSH into a Raspberry Pi

When SSH is enabled, we must find the IP address of the Pi to make connection with it. IP address is a unique address that identified a device on the internet or a local network. The rules defining the format of data delivered over the internet or a local network are known as “Internet Protocol”, or IP. IP address, which carries location data and makes device reachable for communication, are essential for the exchange data between the Pi and a computer.

Concerning this problem, using Pi to control a touchscreen at the institute, I must connect it to Eduroam network of the institute to have an IP address. Raspberry OS is a Unix-like operating system based on Debian Linux, this operating system doesn’t allow you to connect to a network that use Enterprise WPA.

Wi-Fi Protected Access-Enterprise is a wireless security mechanism designed for small to large enterprise (university) wireless networks. WPA enterprise uses Remote Authentication Dial-in User Service protocol to manage user authentication so that if we want to connect to an Eduroam network, it requires a username and password that is valid ^[7].

To enable Pi to use Eduroam network, I must modify the “*wpa_supplicant*” file, this file stores all information of the network that Pi was and currently connected to. To locate this file, open the terminal and enter the codes:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

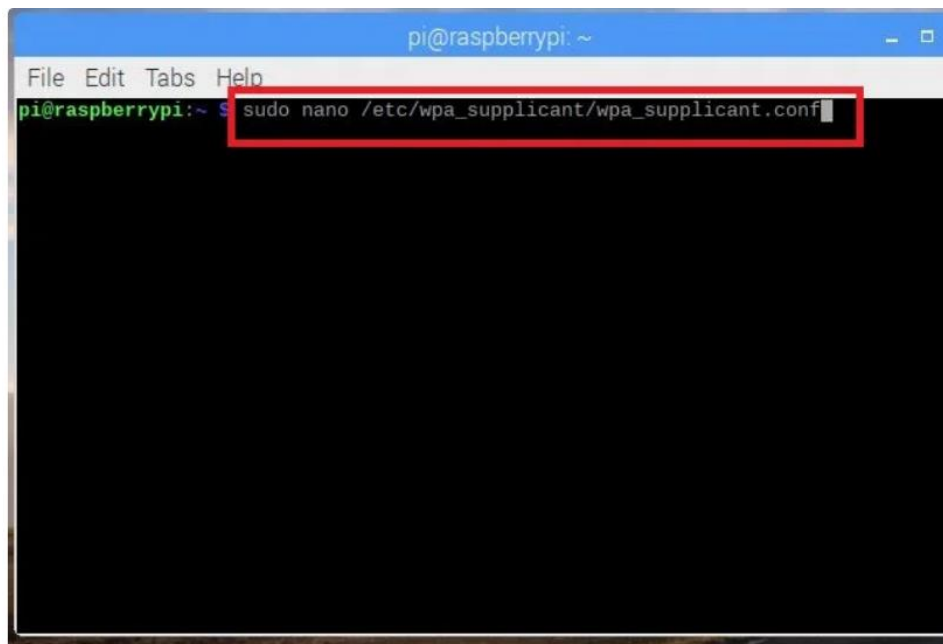


Figure 16. Terminal of Raspberry Pi

Then I have to configure this file to use Eduroam network. I have come up with a script that will add the network detail to this WPA file and after reboot, the Pi will connect automatically to the Eduroam network:

```
network={  
    ssid="eduroam"  
    scan_ssid=1  
    key_mgmt=WPA-EAP
```

```

group=CCMP TKIP
eap=PEAP
identity="trantienthanh"
password="Trantienthanh2102"
phase1="peapver=0"
phase2="MSCHAPV2"
}

```

After editing the file, hit CTRL-X, then Y, then enter to save the updated “*wpa_supplicant.conf*” file. When done, I reboot the Raspberry Pi and the network is connected to Pi. Eduroam is still grayed out but now it has a checkmark next to it.

It is very easy to find the IP of the Pi if it is directly connecting to the touchscreen or any type of display, open the terminal and use the following command:

```
ip a
```

The IP address format is 4 sections number separated by a dot (.) and the value of each section is between 1 and 254. In the early stage of the thesis, I wasn’t able to connect the Pi to any type of display, so another solution is to connect the laptop to the same wifi network that Pi is connected to. Using software called “Advanced IP scanner” (**Figure 18**), we can detect all the IPs of all the devices that currently connect to the network.

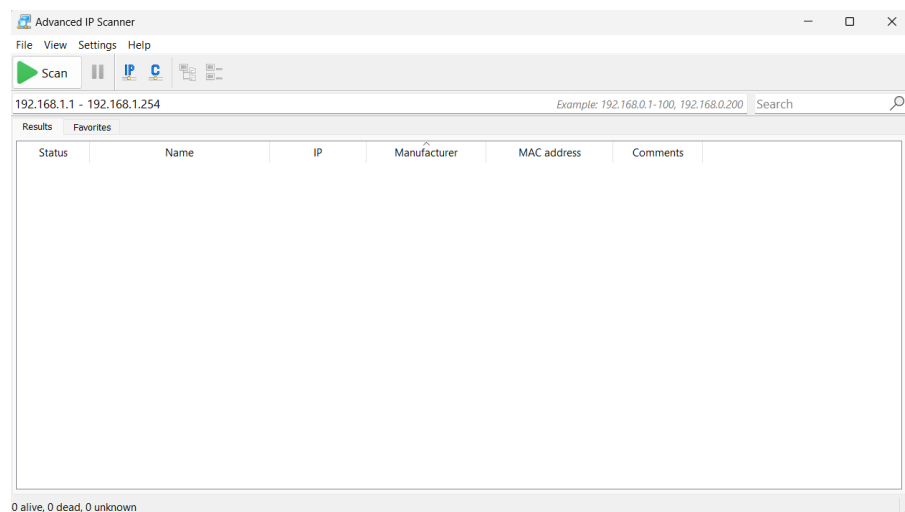


Figure 18. Advanced IP Scanner

3.3.3 Connect to the Raspberry Pi via SSH

Now the Pi is ready to be connected via SSH. There are several ways to connect to Raspberry Pi. We can use PuTTY Configuration software (**Figure 19**) to connect to Pi. PuTTY is a software application that can run from a PC, or laptop to access the Raspberry Pi command-line interface, which we can use to send commands to the Raspberry Pi and receive data from it. PuTTY is a terminal emulator that supports several network protocols, including SSH, SCP,... By inserting the IP address of Pi in Hostname, we can connect to Pi and use its terminal:

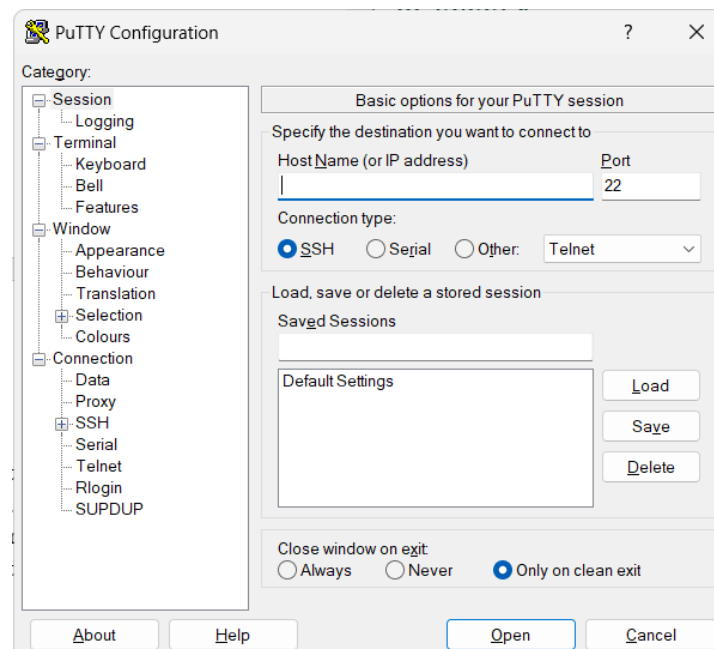


Figure 19. Putty

Then click “Open” to open up the terminal window, I am using the default user name “pi” and password “raspberrypi”, for the sake of early stage of development, I do not change or add more login user.

3.4 Create an alternative mouse to control Raspberry Pi

In addition to the interaction with the display, we can use a peripheral hardware to control the display, I use Analog 2axis Thumb Joystick (**Figure 20**) for this purpose.



Figure 20. Analog 2 axis thumb joystick with select button of Adafruit

The analog joystick is comparable to two connected potentiometers, one for horizontal movement and the other for vertical movement (Y-axis) (X-axis). A Select Switch is also including with the joystick. This pack of joystick come in 3 part (**Figure 21**) – the joystick itself, a soft-touch rubber “hat” and a nicely design breakout board.

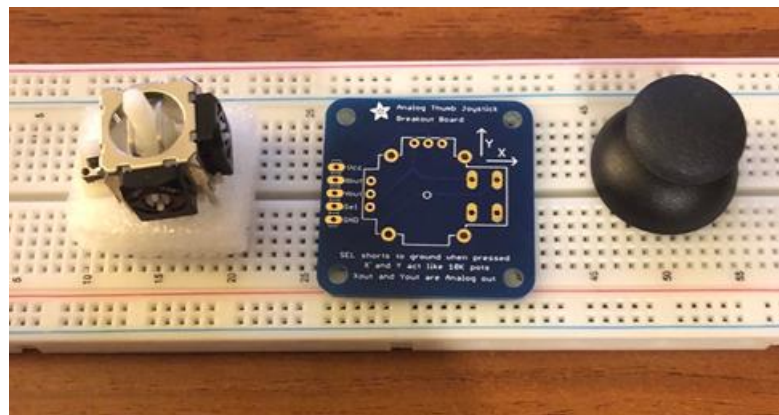


Figure 21. All the compartment of a joystick

The breakout board is designed in such way that the joystick can easily attach to the panel. A 5 pin 0.1" spaced header makes it easy to connect either in a breadboard setting or free wire. There are 5 pins on the PCB:

Pin no.	Pin name	Description
1	Gnd	Ground terminal of module
2	Vcc	Positive supply terminal of Module
3	Xout	Voltage proportional to X axis
4	Yout	Voltage proportional to Y axis
5	Sel	Switch

Xout is horizontal output voltage. Moving the joystick from left to right cause the output voltage to change from 0 to Vcc, the joystick will read approximately half of Vcc when it at default position (center), the same apply with Yout.

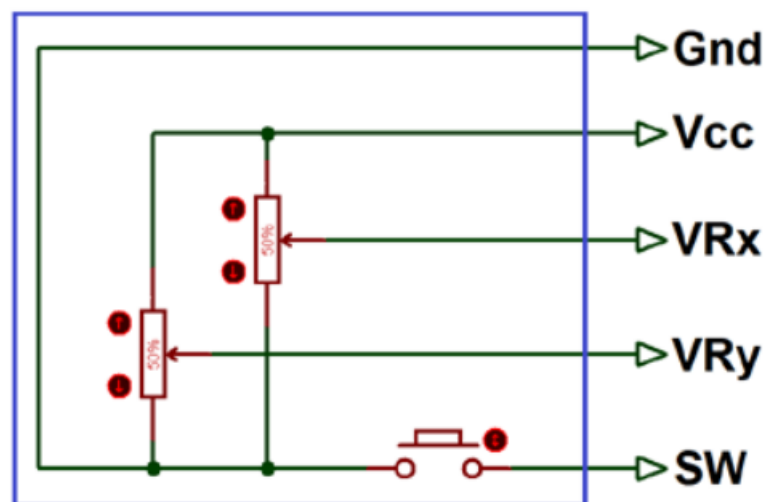


Figure 22. Schematic of Joystick module

Because it comes in 3 parts so I need to soldering the joystick into the PCB using a soldering iron and a solder wire.

A soldering iron (*Figure 23*) is a hand tool used for soldering. It supplies heat to melt solder so that it can flow into the joint between the 2 work pieces –the joystick and the PCB. A heated metal tip (the bit) plus an insulated grip make up a soldering iron. Heating is frequently achieved electrically by running an electric current through a resistive heating element (provided by an electrical cord or battery cable).

A soldering station (*Figure 23*) has an electrical power supply, a soldering iron or soldering head with a temperature sensor on the tip, control circuitry with a user-adjustable temperature and display. Normally have a stand for the hot iron when not in use and a wet sponge for cleaning the soldering iron

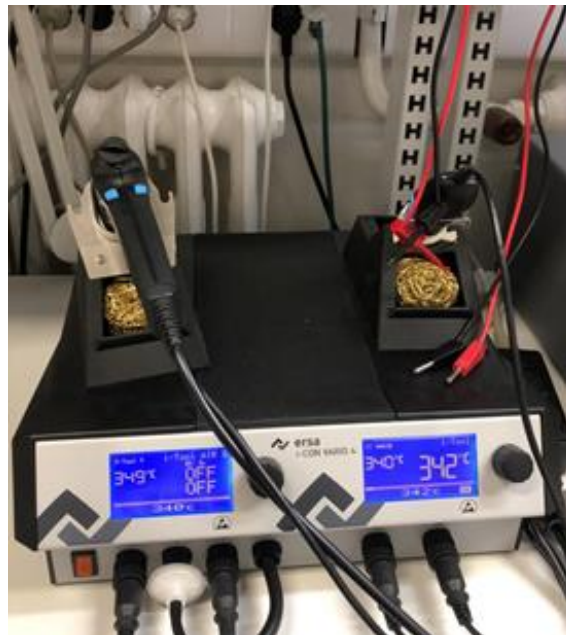


Figure 23. Solering station with temperature controller and soldering iron

In the laboratory I use Stannol S-SN68Pb40 wire. This electronic solder wire is very efficient by the high activity, which result in quick spread of solder and electrical residues. It is halide activated rosin flux.

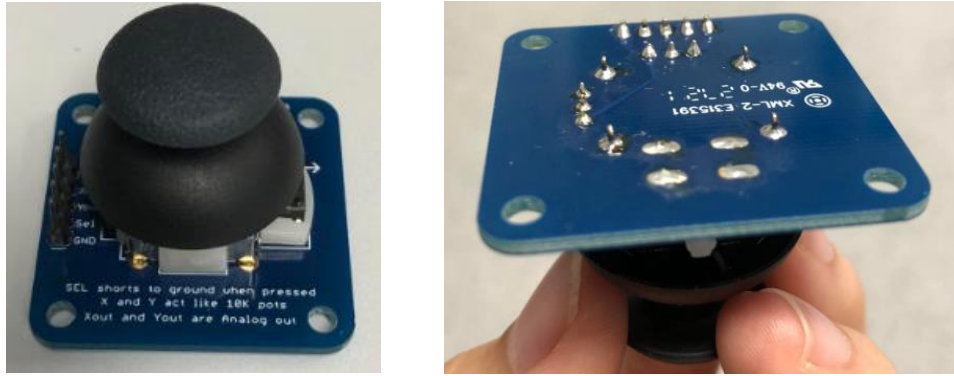


Figure 24 (a) (b) Final result of analog joystick after soldering

3.4.1 Basic function of the joystick

The basic idea is convert the stick's position on two axis (**Figure 25**) into an electrical signal that a microcontroller can process. The joystick can move between 2 directions, x and y axis. The value on each analog channel can vary from 0 to 1023 depend how I rotate the joystick.

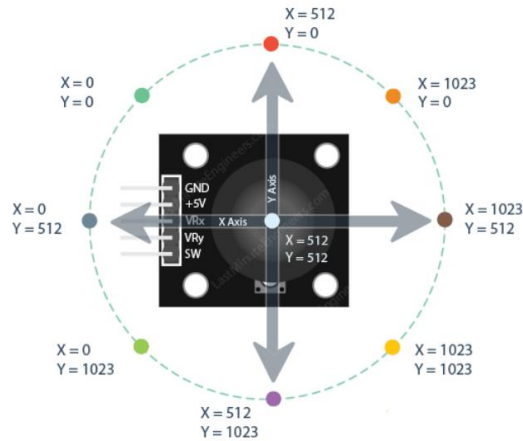


Figure 25. Demonstration of joystick motion axis

The default position for the stick (**Figure 26**) is at (x, y: 511, 511). If the stick moved on X axis from one end to the other, the X values will change from 0 to 1023 and similar thing happens when moved along the Y axis. On the same line I can read the position of the stick anywhere in upper half hemisphere from combination of these values.

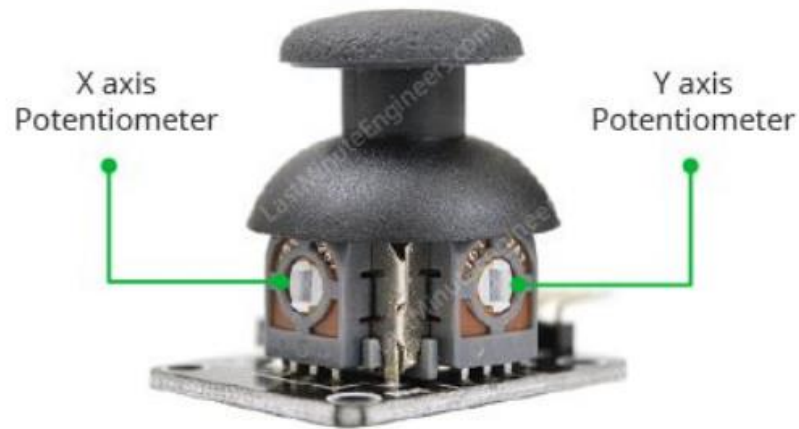


Figure 26. Potential meter

The joystick also has a momentary push button switch (**Figure 27**) that actuated when I press the joystick nob. By default, the switch output is floating, to read it, a pull up resistor is required so that when the joystick nob is pressed, the switch output become LOW, otherwise it remain HIGH. So when I want to connect the joystick to Pi, it is a must to have an internal pull up enabled, or an external pull up resistor must be connected.

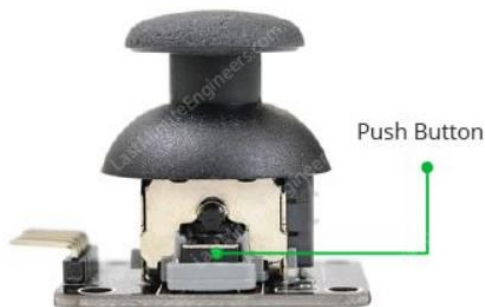


Figure 27. Push button

3.4.2 Connect joystick to Raspberry Pi

Before wire the analog joystick with Raspberry Pi, Pi doesn't have any ADC converter so that we need to have an analog to digital converter to measure the X and Y voltage. For this reason, I choose MCP3008 I/P 10 bit ADC (**Figure 28**).

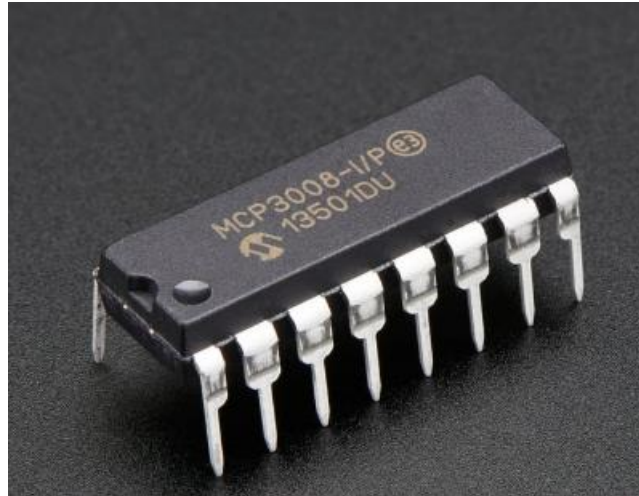


Figure 28. MCP microcontroller

This is a low-cost 8 channel 10 bit analog to digital converter. This chip is very precise and can help me to read quite a few analog signals from the joystick. It is a successive approximation converter. Successive Approximation ADC is a converter that transform a continuous analog waveform into a discrete digital representation. It does this by using a binary search over all potential quantization levels before settling on a digital output for each conversion.

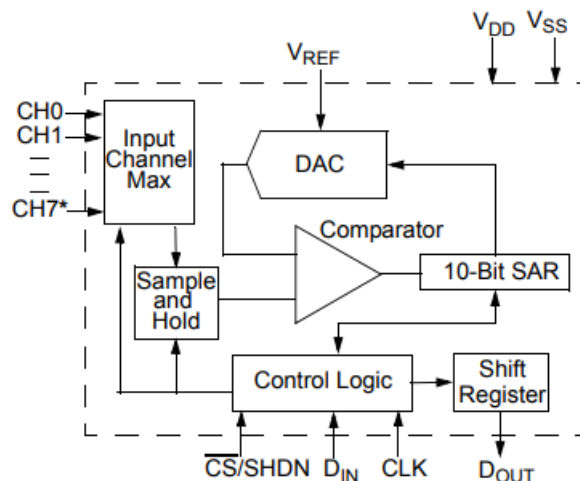


Figure 29. Functional block diagram of MCP3008

So if we want to connect MCP3008 and the analog joystick to Raspberry Pi, we must use GPIO pins of Raspberry Pi. It stands for General Purpose Input Output, it used as an input or

output, or both and controllable by user. In the right side of the Raspberry Pi, there are 2 rows of GPIO pins, connection between Pi and its peripherals.

Output pins are like switches that Pi can turn on and off a device, and also send a signal to another device. Input pins like switches that can turn on or off from outside world, or can also be a data for sensor, signal to other device. In this case we will use those pins to read signal from our MCP3008 ADC that convert signal from our analog joystick.

First step I must determine which pins needed for this connection to work between Raspberry Pi (*Figure 30*) and MCP 3008 (*Figure 31*).

3v3 Power	1	2	5v Power
GPIO 2 (I2C1 SDA)	3	4	5v Power
GPIO 3 (I2C1 SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (UART TX)
Ground	9	10	GPIO 15 (UART RX)
GPIO 17	11	12	GPIO 18 (PCM CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3v3 Power	17	18	GPIO 24
GPIO 10 (SPI0 MOSI)	19	20	Ground
GPIO 9 (SPI0 MISO)	21	22	GPIO 25
GPIO 11 (SPI0 SCLK)	23	24	GPIO 8 (SPI0 CE0)
Ground	25	26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27	28	GPIO 1 (EEPROM SCL)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM DIN)
Ground	39	40	GPIO 21 (PCM DOUT)

Figure 30. Pins of Raspberry Pi 4 from top to bottom

- **Pin 6:** Ground
- **Pin 1 (VCC):** 3v3 Power supply, had a maximum available current of only 50mA
- **Pin 24 (GPIO8):** SPI chip select 0
- **Pin 19 (GPIO10)**
- **Pin 21 (GPIO9)**
- **Pin 23 (GPIO11)**

PDIP, SOIC

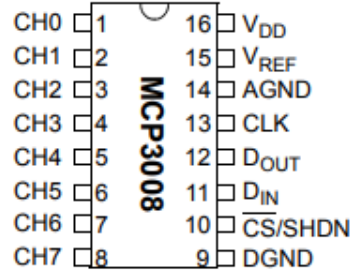


Figure 31. Pins of MCP3008

Each pin corresponds to one usage of it:

MCP3008	Symbol	Description
PDIP, SOIC		
1	CH0	Analog Input
2	CH1	Analog Input
3	CH2	Analog Input
4	CH3	Analog Input
5	CH4	Analog Input
6	CH5	Analog Input
7	CH6	Analog Input
8	CH7	Analog Input
9	DGND	Digital Ground
10	CS/SHDN	Chip Select/Shutdown Input
11	D _{IN}	Serial Data In
12	D _{OUT}	Serial Data Out
13	CLK	Serial Clock
14	AGND	Analog Ground
15	V _{REF}	Reference Voltage Input
16	V _{DD}	+2.7V to 5.5V Power Supply
–	NC	No Connection

Figure 32. Pin of MCP3008

- **Pin 9 (DGND):** Digital ground connection to internal digital circuitry.
- **Pin 10 (CS):** Chip select, use to initiate communication with the device when pulled low. When pulled high, it will end a conversion and put the device in low power standby.
- **Pin 11 (D_{IN}):** Serial Data Input, this SPI pin use to load channel configuration data into the device.

- **Pin 12 (DOUT):** Serial Data Output, this SPI pin use to shift out the result of the A/D conversion and the data always change on the falling edge of each clock as the conversion takes place.
- **Pin 13 (CLK):** Serial Clock, this SPI pin is used to initiate a conversion and clock out each bit of the conversion as it takes place.
- **Pin 14 (AGND):** Analog ground connection to internal digital circuitry.
- **Pin 15 (VREF):** Reference voltage input.
- **Pin 16 (VDD):** Power supply.

Those are the Pins that we need to connect Pi with MCP3008. The connection is the following ^[6]:

MCP3008	Pi
-----	-----
Pin 1 (CH0)	-
Pin 2 (CH1)	-
Pin 3 (CH2)	-
Pin 9 (DGND)	Pin 6 (Ground)
Pin 10 (CS)	Pin 24 (GPIO8)
Pin 11 (DIN)	Pin 19 (GPIO10)
Pin 12 (DOUT)	Pin 21 (GPIO9)
Pin 13 (CLK)	Pin 23 (GPIO11)
Pin 14 (AGND)	Pin 6 (Ground)
Pin 15 (VREF)	Pin 1 (3.3V)
Pin 16 (VDD)	Pin 1 (3.3V)

Next step is to connect the rest of MCP3008 pins to the joystick module, there are 5 pins in the joystick will correspond to 3 pin of MCP and 2 pin of Ground and VCC.

Joystick	Pi/MCP
-----	-----
GND	Pi GPIO pin 6 (Ground)
VCC	Pi GPIO pin 1 (3.3V)
Xout	MCP3008 Pin 1 (CH0)
Yout	MCP3008 Pin 2 (CH1)
SW	MCP3008 Pin 3 (CH2)

A 10k pull up resistor is needed, to pull the switch input to HIGH. When the switch is pressed, the input is connected to Ground (0V, LOW). Without the resistor, the input will be in an undefined state when the switch wasn't being pressed and read random values.

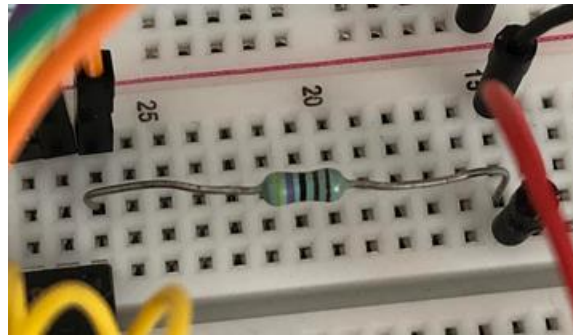


Figure 33. 10k resistor

We connect these 2 element onto a breadboard. Breadboard is a construction base use to build semi-permanent electronic circuit, allow me to create circuit without soldering. A plastic board, with a grid of tie-points (holes). Inside the board there are metal strip connecting the different tie-points in specific ways.

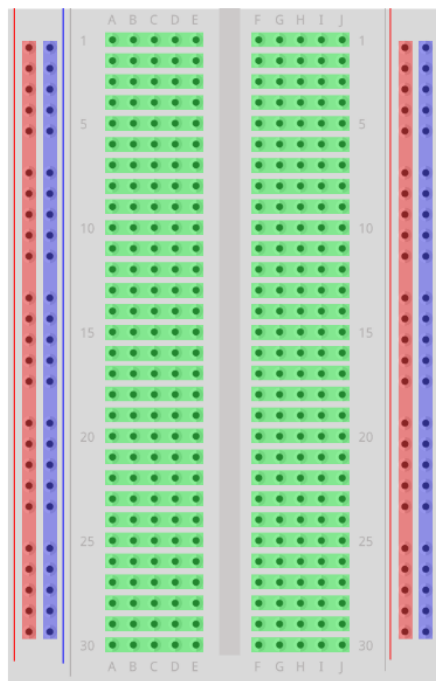


Figure 34. Breadboard

Usually the breadboard can be separated into 3 section:

- RED - Power Bus: usually connected to power supply, since the entire column is connected, I can connect power to any tie-point in the column.
- BLUE – Ground Bus: usually connected to Ground, since the entire column is connected, I can connect ground to any of the tie-point in the column.
- GREEN – connected tie-point: the tie-point of each rows are connected, 5 holes make up a row, 2 rows in the same line and separated by the Trench (in the middle).

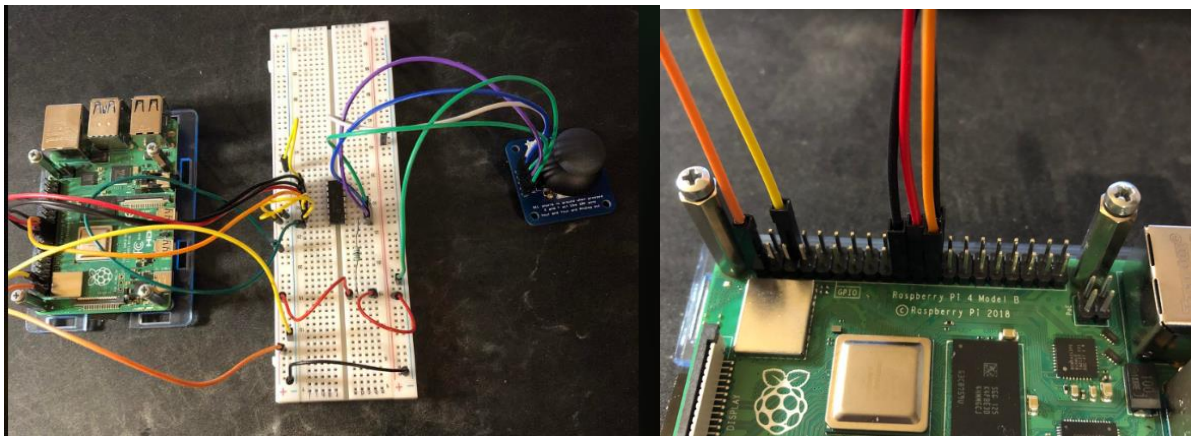


Figure 35(a)(b). Wire connection

After the connection, to make the setup work, I have to configure SPI bus (GPIO) on Raspberry Pi ^[8]. SPI stands for Serial Peripheral Interface, it's a communication protocol used to transfer data between Raspberry Pi and peripheral devices, in this case, it is MCP3008. The SPI pin of Pi is 19,21,23 and 24. By default, Raspbian image disables SPI so before using it, it must be enabled. I have found 2 methods that can be done:

a. Using Command Line

Open terminal in Raspberry pi and running this command:

```
sudo raspi-config
```

This will boot up the configuration window (*Figure 36*), select “Interface Options” and enable SPI option, reboot after this and SPI will be ready to used.

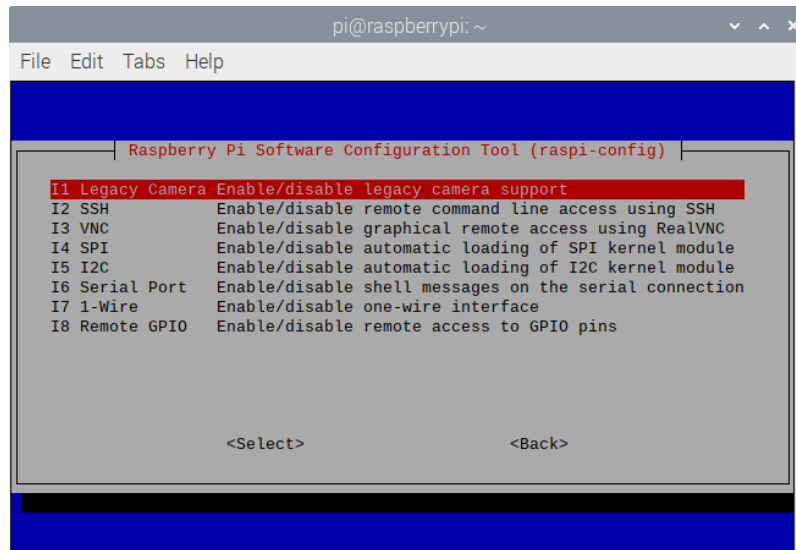


Figure 36. Raspberry Pi Configuration

b. Using Raspberry Pi configuration

The easier way is on top left of the Raspberry Pi screen, we choose Preferences and Raspberry Pi configuration (**Figure 37**), choose “Enabled” SPI and select reboot to take effect.

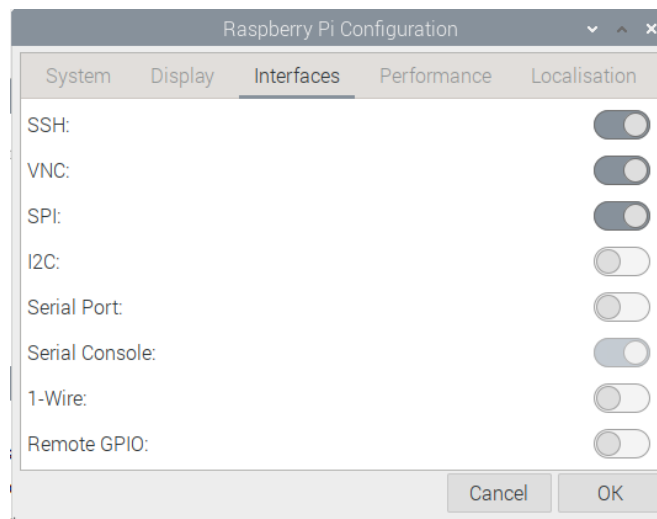


Figure 37. Raspberry Pi Configuration

Now the system is ready to use, but in order to read the signal from the SPI bus, we will use Python libraries by installing it to Raspberry Pi. Python SPI Wrapper and Python3 must be installed in the environment.

```
sudo apt-get install -y python-dev python3-dev
```

And:

```
sudo apt-get install -y python -spidev python3-spidev
```

The library is ready to use and now we can read the value from the SPI bus. I have developed a test program in Python in order to read the joystick value.

```

import spidev
import time
import os

# Open SPI bus
spi = spidev.SpiDev()
spi.open(0,0)

# Function to read SPI data from MCP3008 chip
# Channel must be an integer 0-7
def ReadChannel(channel):
    adc = spi.xfer2([1, (8+channel)<<4,0])
    data = ((adc[1]&3) << 8) + adc[2]
    return data

# Define sensor channels
# (channels 3 to 7 unused)
swt_channel = 0
vrx_channel = 1
vry_channel = 2

# Define delay between readings (s)
delay = 0.5

while True:

    # Read the joystick position data
    vrx_pos = ReadChannel(vrx_channel)
    vry_pos = ReadChannel(vry_channel)

    # Read switch state
    swt_val = ReadChannel(swt_channel)

    # Print out results
    print "-----"
    print("X : {}  Y : {}  Switch :
{}".format(vrx_pos,vry_pos,swt_val))

    # Wait before repeating loop
    time.sleep(delay)

```

After compose, Save

The basic idea is MCP3008 is a 10-bit converter so it can report a range of number from 0 to 1023 (2 to the power of 10). A reading of 0 means the input is 0V and a reading of 1023 mean the input is 3.3V. As the X and Y joystick are moved from one extreme to another, their value will fluctuate between 0 and 1023. In the center, we would expect a value of 511.5 in both X and Y, but in reality the value is varying between 505 and 514.

```

Shell
The current pointer position is (545, 465)
-----
X : 506 Y : 514 Switch : 1022
The current pointer position is (545, 485)
-----
X : 506 Y : 511 Switch : 1023
The current pointer position is (545, 485)
-----
X : 507 Y : 513 Switch : 1021
The current pointer position is (545, 485)
-----
X : 507 Y : 513 Switch : 1022
The current pointer position is (545, 485)
-----
X : 507 Y : 511 Switch : 1023
The current pointer position is (545, 485)
-----
X : 505 Y : 512 Switch : 1022
The current pointer position is (545, 485)
-----
X : 507 Y : 514 Switch : 1021

```

Figure 38. Value of X and Y

In the circuit the switch is read 1023 (3.3V) by default and if it is pressed, it will read 0 (0V)

```

Thonny - /home/pi/bbt/mouse2try.py @ 79.20
New Load Save Run Debug Over Into Out Stop Zoom Quit
mouse2try.py
1 import spidev
2 import time
3 import os
4 import pynput
5
6 from pynput.mouse import Button, Controller
7
8 mouse = Controller()
9
10 # Open SPI bus
11 spi = spidev.SpiDev()
12 spi.open(0,0)
13 spi.max_speed_hz=1000000
14
15 # Function to read SPI data from MCP3008 chip
16 # Channel must be an integer 0-7
17 def ReadChannel(channel):
18     adc = spi.xfer2([1] * (8+channel)) & 0xFF

```

```

Shell
X : 506 Y : 511 Switch : 0
The current pointer position is (456, 618)
-----
X : 506 Y : 512 Switch : 0
The current pointer position is (456, 618)
-----
X : 506 Y : 512 Switch : 0

```

Figure 39. Switch show 0 in value

```

import spidev
import time
import os
import pynput

from pynput.mouse import Button, Controller

mouse = Controller()

# Open SPI bus
spi = spidev.SpiDev()
spi.open(0,0)
spi.max_speed_hz=1000000

# Function to read SPI data from MCP3008 chip
# Channel must be an integer 0-7
def ReadChannel(channel):
    adc = spi.xfer2([1, (8+channel)<<4,0])
    data = ((adc[1]&3) << 8) + adc[2]
    return data

# Define sensor channels
# (channels 3 to 7 unused)
swt_channel = 0
vrx_channel = 1
vry_channel = 2

# Define delay between readings (s)
delay = 0.1

while True:

    # Read the joystick position data
    vrx_pos = ReadChannel(vrx_channel)
    vry_pos = ReadChannel(vry_channel)

    # Read switch state
    swt_val = ReadChannel(swt_channel)

    if vrx_pos > 520:
        mouse.move(0, -1)
    if vrx_pos < 510:
        mouse.move(0, 1)
    if vrx_pos > 900:
        mouse.move(0, -10)
    if vrx_pos < 100:
        mouse.move(0, 10)

```

After successfully connecting the element and reading the value of the joystick, now we can transform the joystick into a cursor to control the display on the touch screen. In python, we have a library dedicated to controlling and monitoring input devices, it is “pynput” library. In this package, there are sub-packages for each type of input device: Mouse and Keyboard, but we only need Mouse packages ^[9].

I have developed a program using Python and pynput library and successfully transform the joystick into a controller that can control the touch screen.

3.5 Custom PCB design for breadboard.

3.5.1 Kicad

I decided to make a PCB layout for the wire connection between Pi, MCP3008 and the joystick because the need of portable and easy to carry around, and everything fit in a 3d printed box that will be introduced later in this document.

Kicad ^[11] is an open source software suit for Electronic Design Automation. The program handles PBC layout with Gerber output, the main reason I use this software is because it is made to hide the complexity of advanced design feature so that it remains approachable by student. The goal of Kicad is to provide the best possible cross platform electronics design application for professional designer.

The first step is to create a project in Kicad (**Figure 40**), where it will store all of the element of your PCB like schematic or PCB layout.

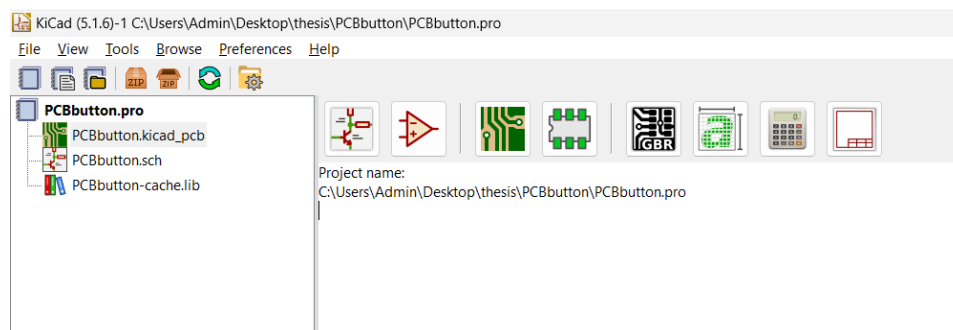


Figure 40. Create new project

In new project, we can see that there are “.sch” file, which is schematic of the circuit that I want to design, “.pcb” file which is PCB layout of the schematic and “.lib” file, which is the library of the footprint of the element of the circuit. To create a schematic of the breadboard, we have to consider all of the wire and parts that connect to it: Raspberry Pi, MCP3008, and joystick. To start, we have to choose the symbol that stand for the part, on the right side menu on Kicad, there is an option to choose.

This open up a library of symbol that we can find and select the correct symbol that we need. Now, we choose the symbol for the connector for the connection between Pi and the PCB, then we set the footprint for it, which is connector 5 pins female.

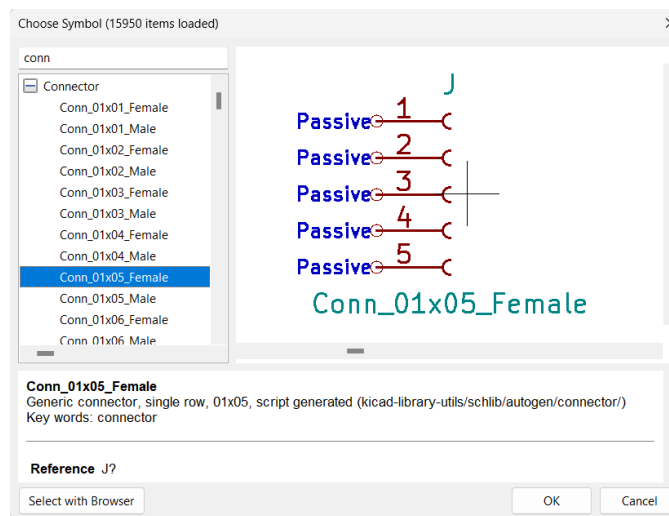


Figure 41. 5 pin connector

MCP3008:

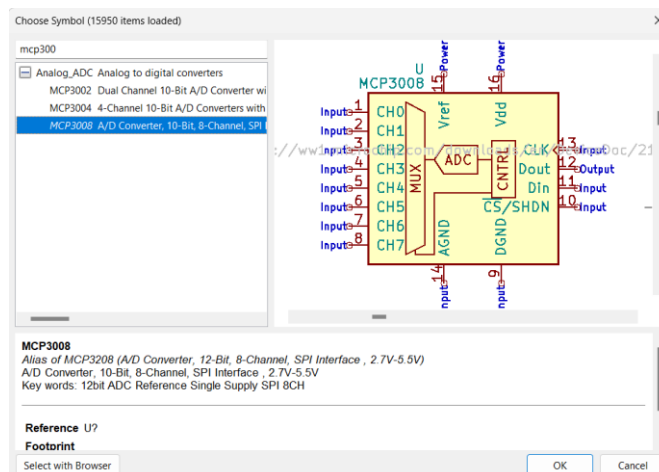


Figure 42. Microcontroller MCP3008

Resistor 10k:

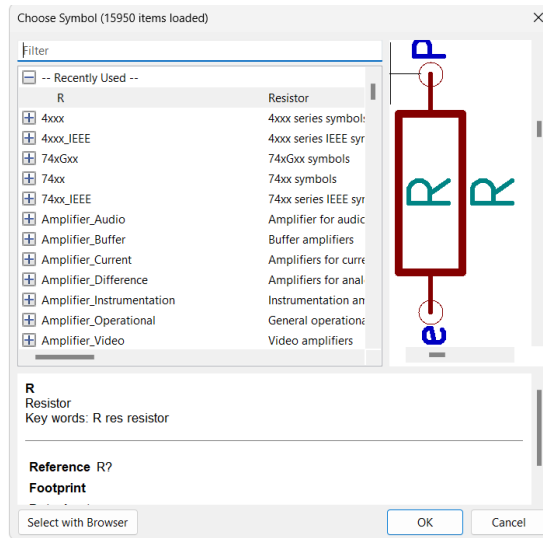


Figure 43. resistor

Connector for connection between PCB and joystick, 6 pins female:

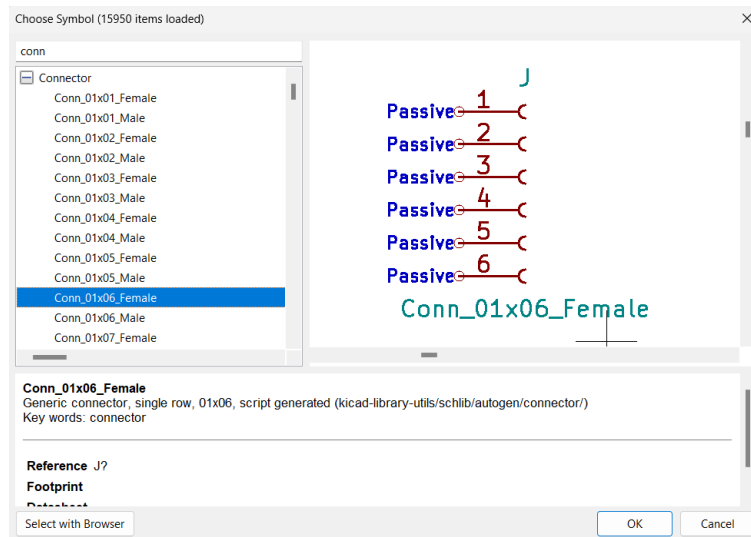


Figure 44. 6 pin connector

Based on the connection we have made earlier, we can connect all the above parts with wires, and place a Ground connection and VCC (3,3V) connection.

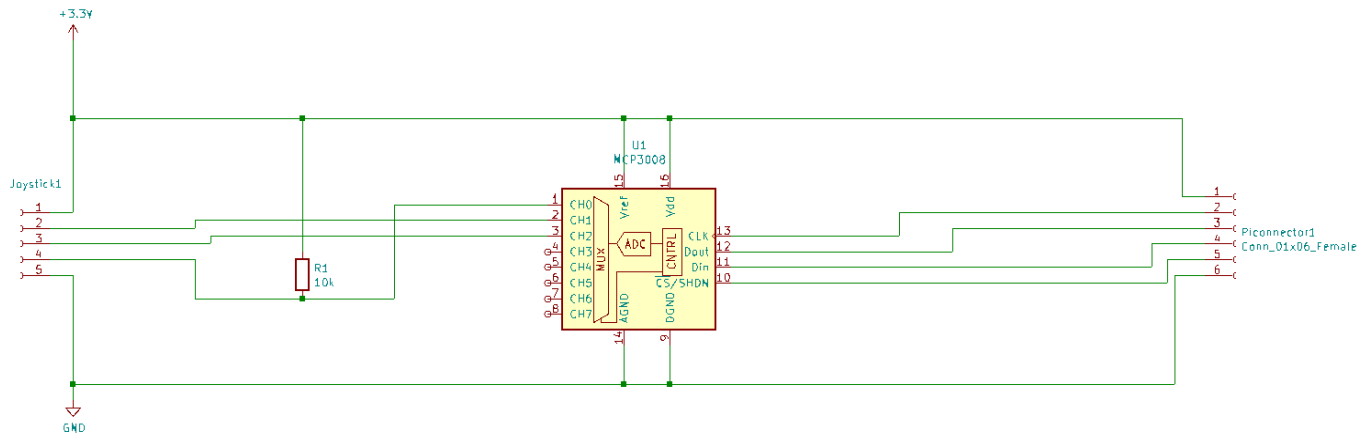


Figure 45. schematic of circuit

Then from this schematic, we can turn it into PCB layout. Choose the create PCB option on the taskbar and it will bring us to PCB window with the elements unorganized. We have to rearrange the parts and connect the wire accordingly and we can obtain the final PCB layout.

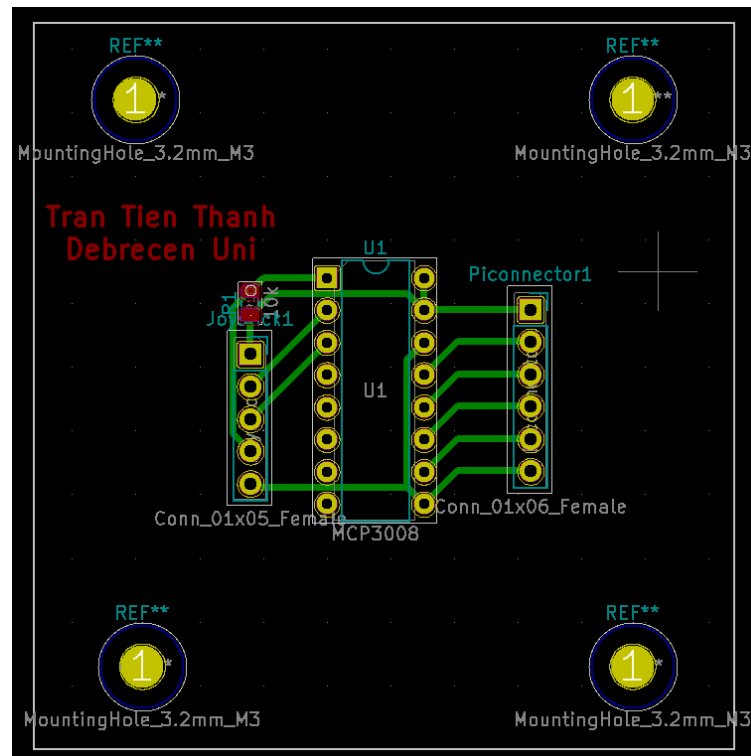


Figure 46. PCB layout

3.5.2 LPKF ProtoMat S104

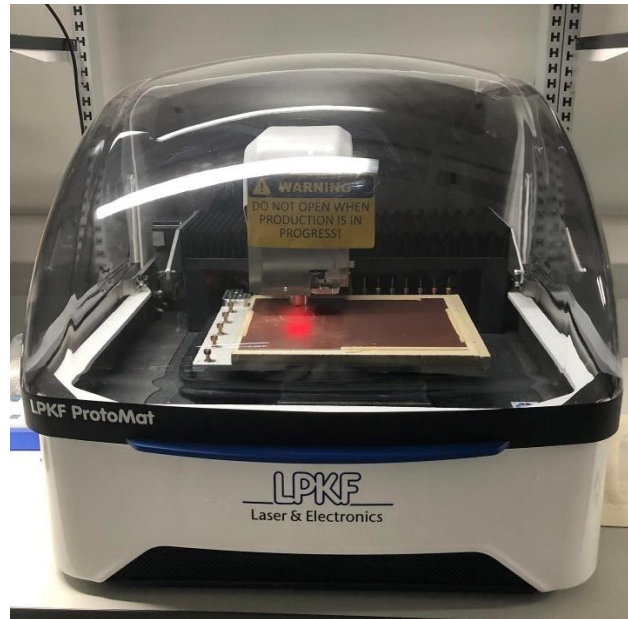


Figure 47. Protomat S104

ProtoMat S104 (*Figure 47*) is an advanced machine that we use to making the PCB from Kicad file we just created. This machine is particularly fast and precise, thanks to its 100,000 Round Per Min spindle motor which enable faster travel speed with a mechanical resolution of only 0,5um. This guarantees the best precision possible when drilling and milling even extremely find structures. Both high speed milling motor and milling depth sensor are low maintenance because they can self-clean.

Sensor-controlled measurements of the material and copper thickness allow for accurate determination of the necessary milling depth. The machine, which has 20 tool locations, automatically switches the appropriate tool as the production process progresses. Different insulating channel are produced by the conical milling cutter depending on the depth of the incursion. The milling contour has a consistent and accurate width thanks to the automatic milling

width adjustment. We can obtain quick process time because of the quick set up and operator free machining.

Processing procedures:

- Switching on the system, on the machine, there is an ON/OFF (*Figure 47*) switch that we can turn on and start the PC that connected to the machine.

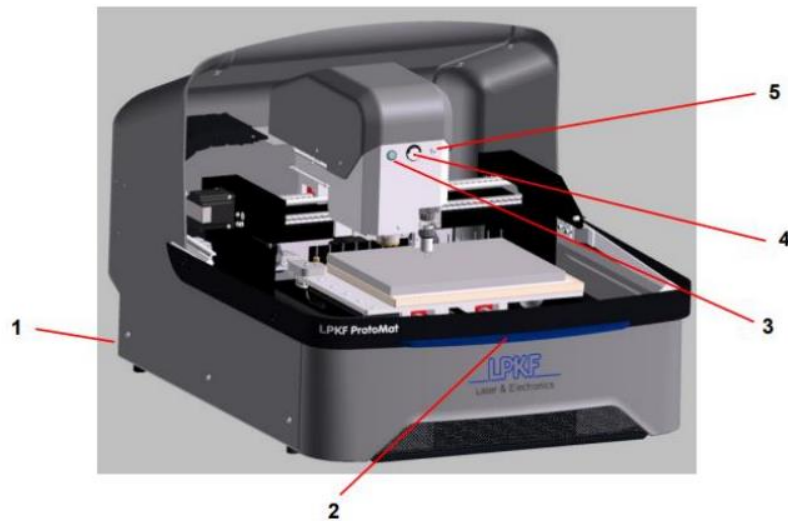


Figure 47. Layout of ProtoMat S104 machine

1. On/off switch
 2. Handle
 3. Status LED/Button for vacuum
 4. Pressure gauge for dispenser pressure
 5. Pressure regulator for dispense
- We have to make sure the compressor is turn On



Figure 48. Vacuum compressor

- The last step is to have all the required files in the PC connected to the machine, we need to generate Fabrication files (**Figure 49**). Open the Kicad project and choose PCB editor, on PCB menu, click on the top Right choose File then Plot.

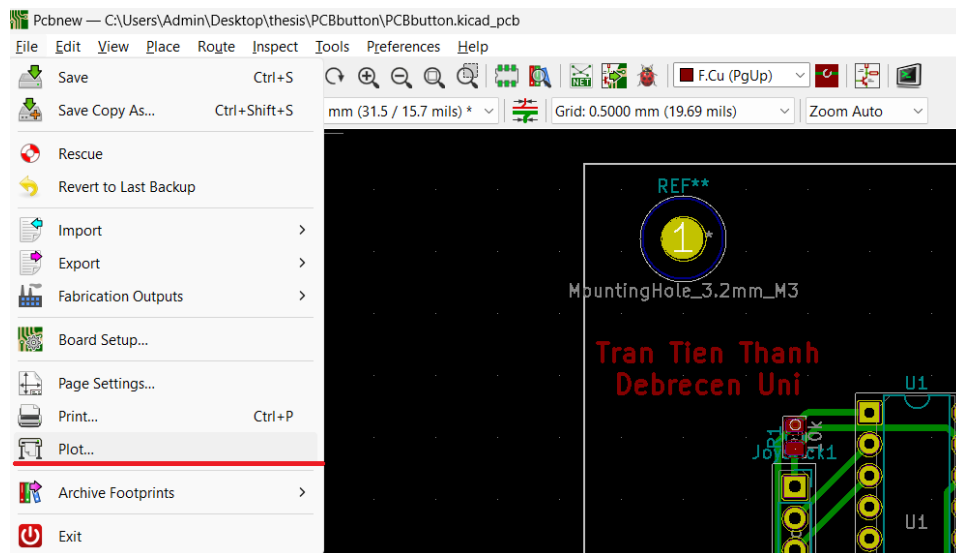


Figure 49. Plot option

On Plot option, we choose F.Cu and Edge.Cuts file from the list, these files are the most important file that contain the information of the PCB that will be created, after that we choose Generate Drill Files, the files will be generated in the project Folder.

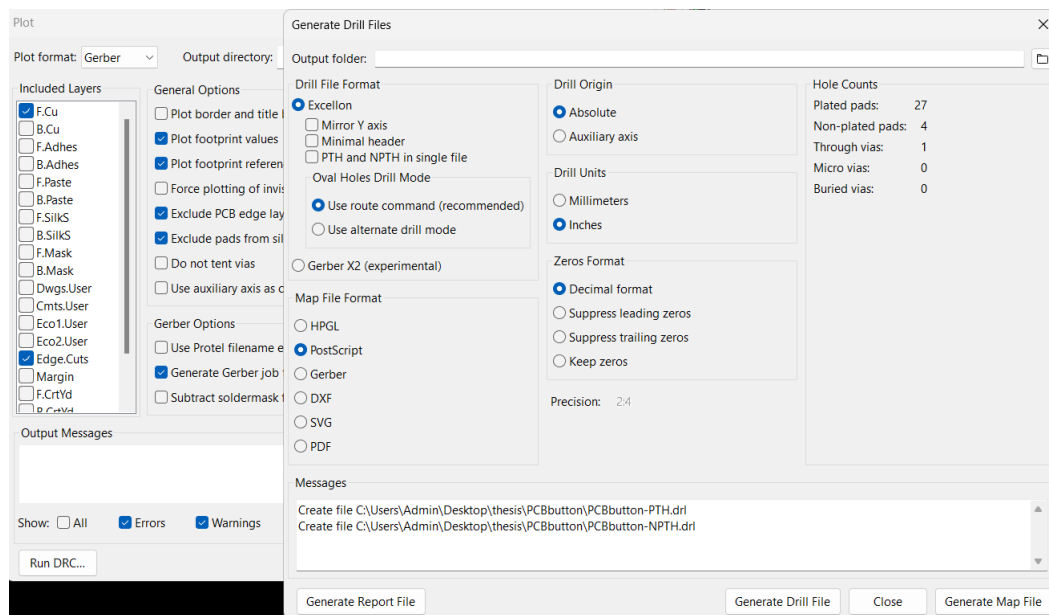


Figure 50. Generate require files

The software we use is CircuitPro PM, this software will import the data from Kicad, use it to generate the individual processing step, prepare for production of PCB, and guides us through each step of manufacturing process.

Start the software, there are steps we have to complete to get the final production. At the beginning we have to select a template to create a single-sided PCB, then import 3 files “F_Cu.gbr”, “Edge_Cuts.gbr”, and “PTH.gbr”, we can see those files in Cam view.

The next step is multiplying the layer and generating toolpaths, we have to set all the configurations to the required setting, then check whether all the tools in the machine are available. After checking everything, we can place the material inside the machine and fasten it on the processing table with adhesive tape.

Start the machine and after a period of time, the initial plate has been cut and drilled into our desired layout. The final product:



Figure 51. PCB has been printed out

3.5.3 Autodesk Fusion 360

Now, we have a Raspberry Pi, an analog joystick and a PCB for the connection, for everything to be portable and easily attached to the touchscreen, I have created a 3D box to integrate all the element into a rectangle box to easily carry and controlled.

The software I use is Fusion 360, it is a cloud base 3D modeling, I can design and manufacture my product according to aesthetic, form fit and functional aspect.

My initial idea is because the raspberry Pi has already come with its cover box, which made of 6 pieces that can assembled together and create a box surround the Pi, I will modify the top of the box (the Lid), to put the joystick on it and the PCB can fit inside the box.

To start, I create new Project in Fusion 360, New Design (*Figure 52*) and start to design my Lid. I saved the file and name it PiLid. In this design page we can see there is a plane surface and it represent XY axis plane.

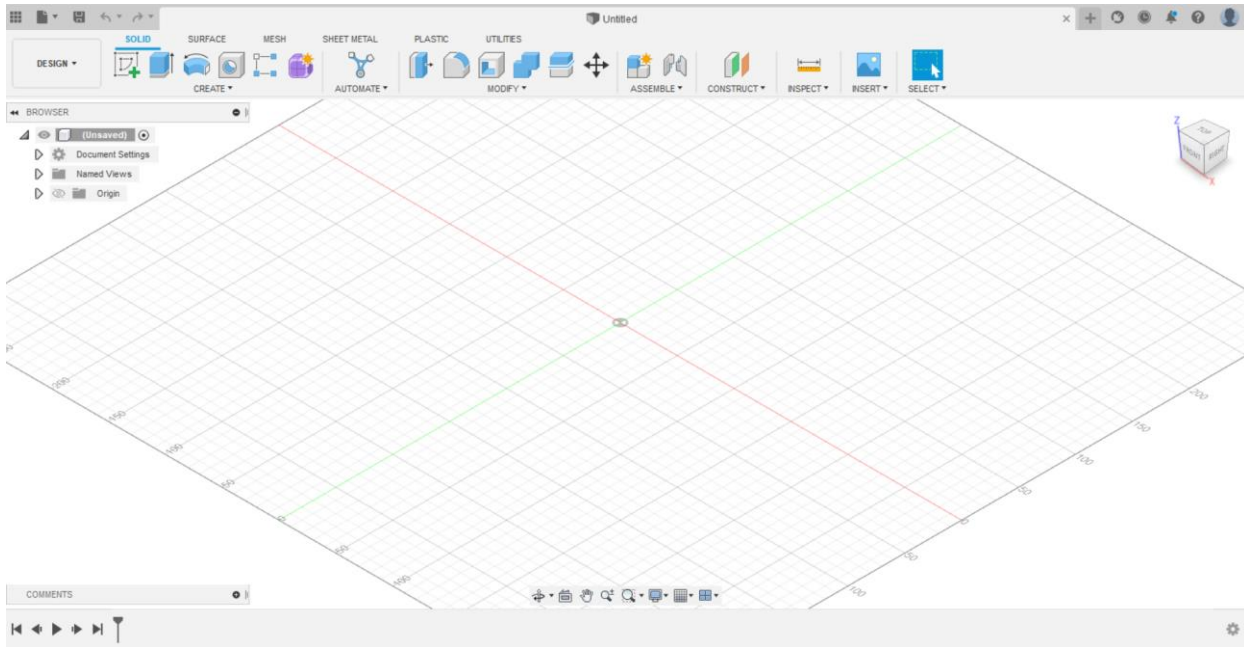


Figure 52. New design

On top right we can see the perspective box (**Figure 53**) and we can rotate it. Which include X,Y and Z, each combination will give us a side of a perspective box that we can see, for example, Z and X will create a Front side, Z and Y will create a Left side,



Figure 53. Point of view rotation

In order to make things more clear and easy to imagine, I have found a 3D model CAD file of Raspberry Pi 4 Model B on CAD forum (**Figure 54**). After I download it and upload the CAD file into my project folder, I can import the model into the design sheet use it as a reference.

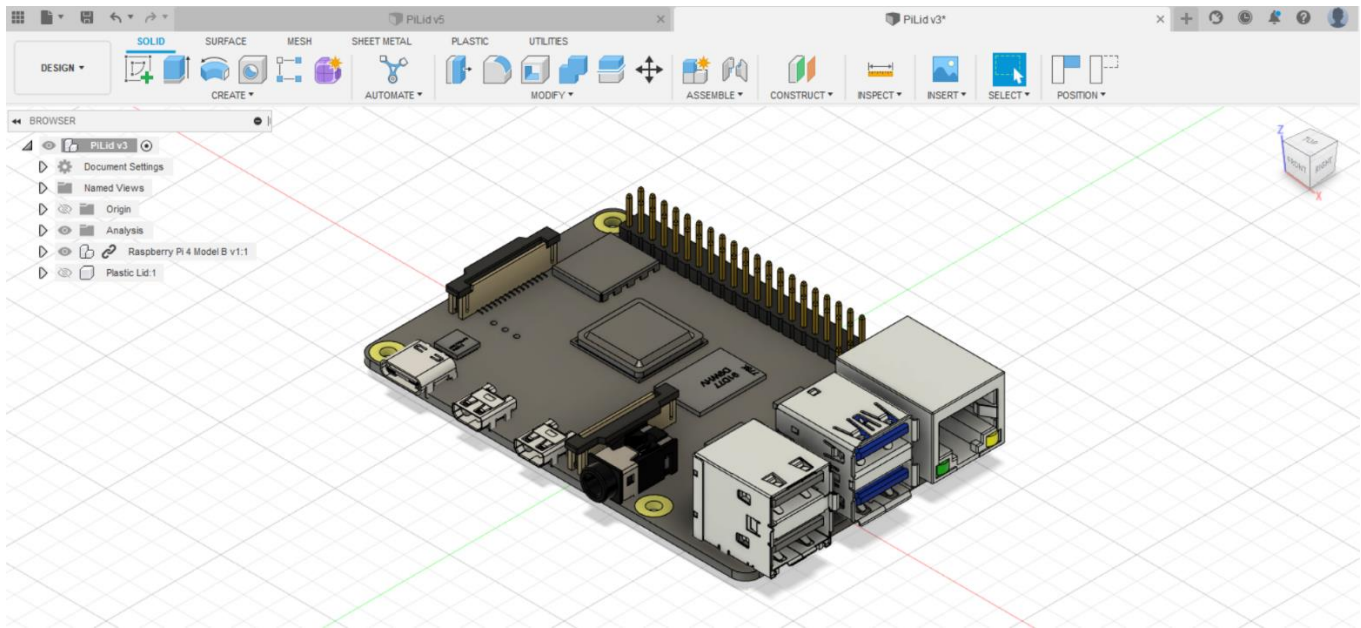


Figure 54. Pi model in side view

The next step is to create a Lid that fit the Pi, there are 4 yellow pins on Pi and I use it as a reference point. After determine the point I can continue create holes where screw and wire can go through and end up with this model.

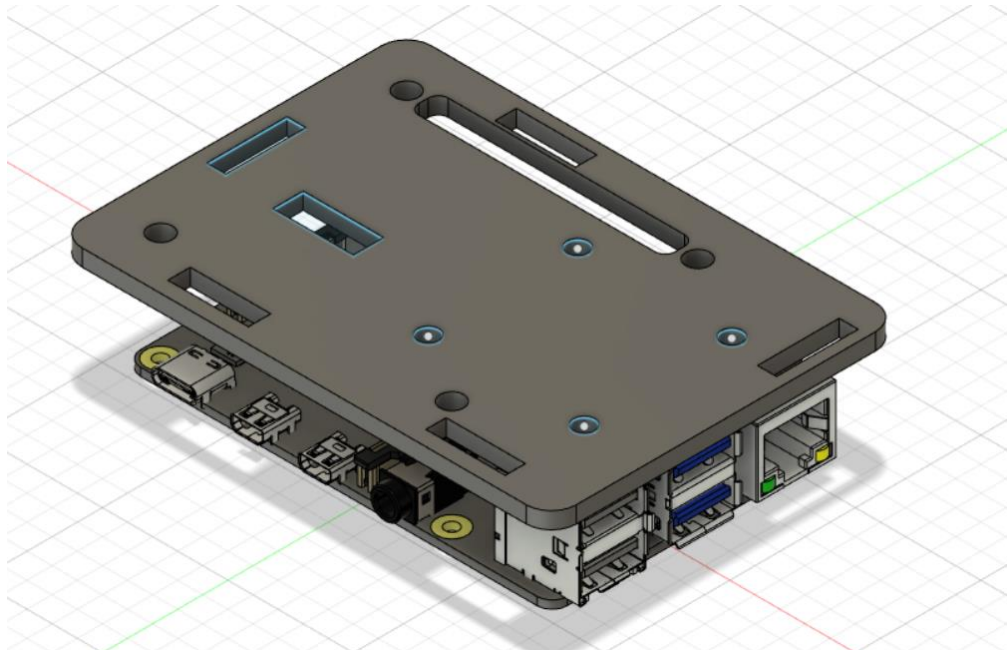


Figure 55. Side view of the lid and Pi model

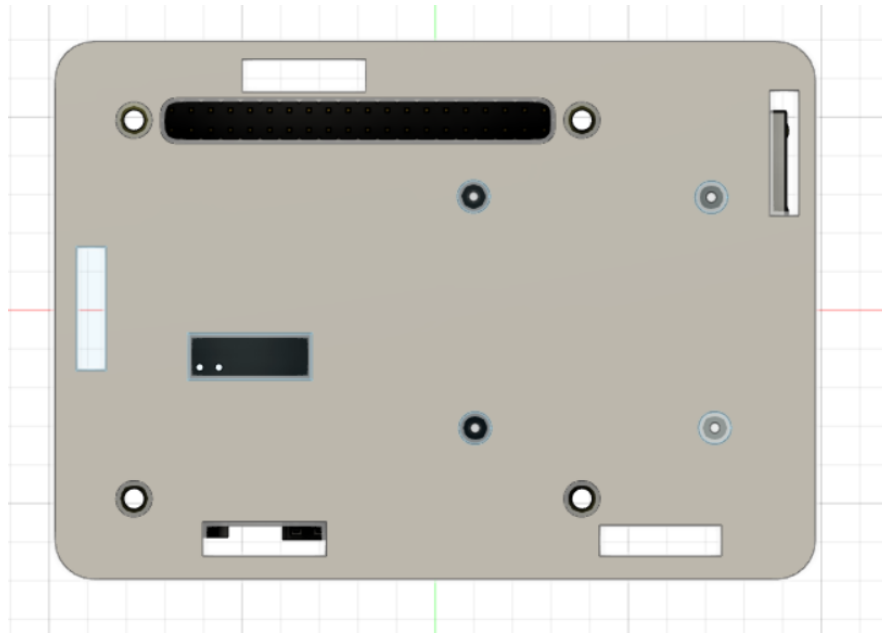


Figure 56. Top view with the lid cover

3.5.4 Witbox2

With the model in Fusion 360, I can import “.stl” file and upload it to the software that compatible with Witbox2 (**Figure 57**). Witbox2 is a second generation 3D printing machine from a Spanish manufacturer, it can print maximum volume of 297 x 210 x 200mm from PLA plastic filaments.

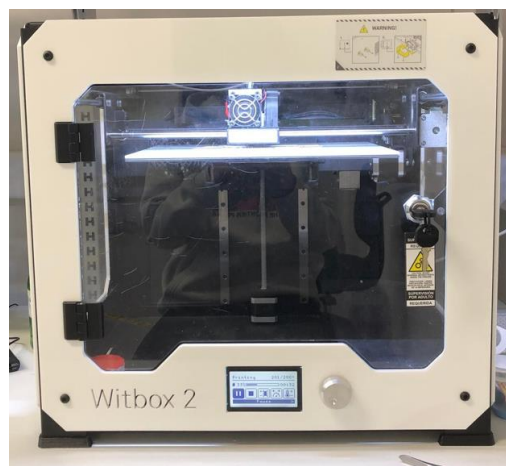


Figure 57. Witbox2

PLA plastic is the material that the machine use to print my 3D model. PLA (polylactic acid) is made from natural resources, specifically from starch that is taken from corn, beets, and wheat. It fit for the machine because it doesn't require hot bed so the printing process not consume a lot of time.

The software we use is Prusa Slicer (*Figure 59*). This software will converse a 3D object model into specific instructions for the printer. In particular, the conversion from a model in “.stl” (*Figure 58*) format to printer commands in “g-code” in fused filament fabrication.

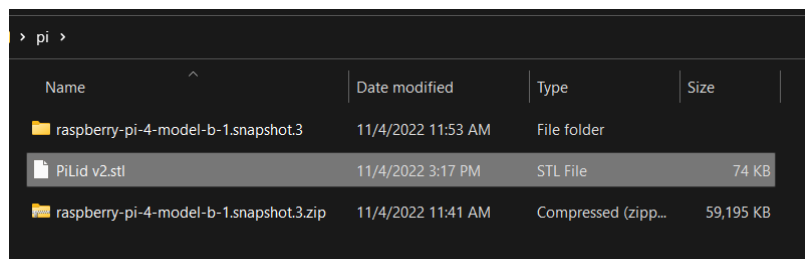


Figure 58. stl file

The working principle is it will divide the object into a stack of flat layer, then describes these layers as linear movement of the 3D printer extruder, all of the movements, together with some specific printer command are finally written in the “g-code” file and we use the file to upload to Witbox 3D printer.

First, we upload the “.stl” file in to Prusa, after setting up the printer Configuration Wizard, we can see the model appear on the screen. We can adjust the placement of the model in the plane and set the correct parameter for the slicing process. After slicing the model, we can have a result of slicing, we can see the model layer by layer.

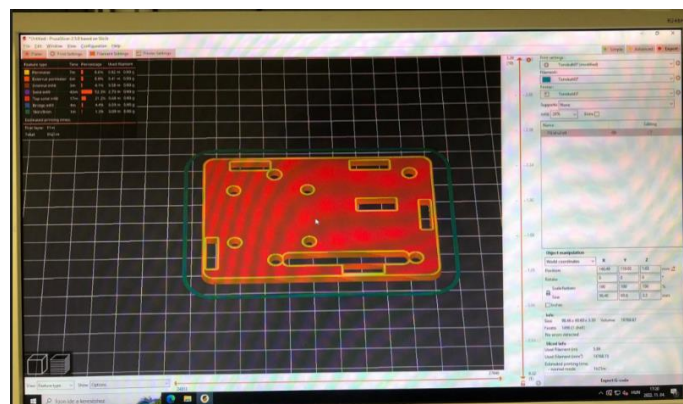


Figure 59. Prusa window

On top left we can see different layers and information about the model, at lower right corner, we can see how long the printing will take and how much material will be used. The final step is to export this “g-code” and put the file on an SD card and insert it into the machine.



Figure 60. Printer main menu

After inserting the SD card and turn on the machine, in the main menu of the machine (**Figure 60**) we can choose the SD card option, choose the correct “g-code” file that we want to print and press OK. My model takes exactly 1h20min to be fabricated and the final result is worth every second.

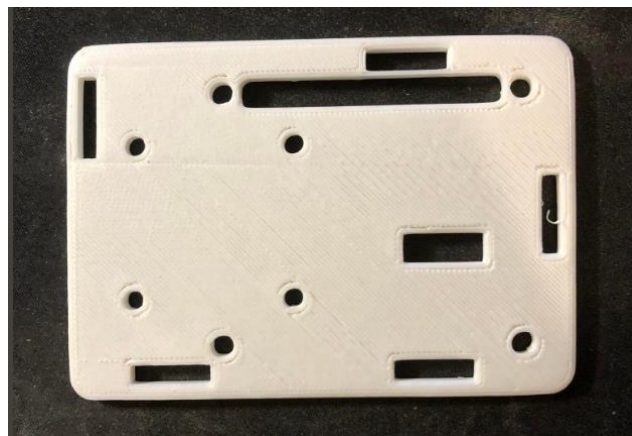


Figure 61(a). Final result of the lid

Now I can assemble the 3D printed lid with the cover box.



Figure 61(b). All the components have been assembled

3.6Google survey

After setting up, connect the peripheral with the touchscreen, we must decide what to put on the display screen. The aim is to make students in the institute access useful information easily, I have come up with several ideas and gathered them into a Google Form. With Google Forms, we can create and edit text documents right in the web browser and the best is multiple people can work or edit at the same time and every change is saved automatically. Of this, I have made the Google Form into a survey.

The survey is about deciding which kind of information will be shown on the touchscreen, and the most suitable decider is none other than the student themselves. This form has been open for 1 week and filled by 82 students in the University. The answers are very positive it is a big contribution for this project to be completed.

First in the form, I must introduce my project as well as my identity so that the form will be more legit and trustworthy.

Digital bulletin

Dear Students, I'm an electrical engineer Bsc student at Institute of Physics, University of Debrecen. I currently working on my thesis project in my close-to-final year.

There will be a micro-pc connected to a touch screen and students can make interaction with the screen, the touch screen will be placed around the campus or a specific place that we will decide later.

So, I make this survey to gather what information you want to have on the screen.

All data collected will be used for research purposes only. Your participation in the survey is voluntary and can be terminated by you at any time and without giving reason. Your participation in the research does not expose you to any significant risk.

The survey will take less than 5 minutes and your input is greatly appreciated.

Now, let's go to the survey :)

Figure 62. Introduction

First question is gather the information where the majority of the students at and its answer.

Which campus are you going to the most?
82 responses



Figure 63. Chart of destination

Secondly, to know specifically types of information to shown on the screen and its answer.

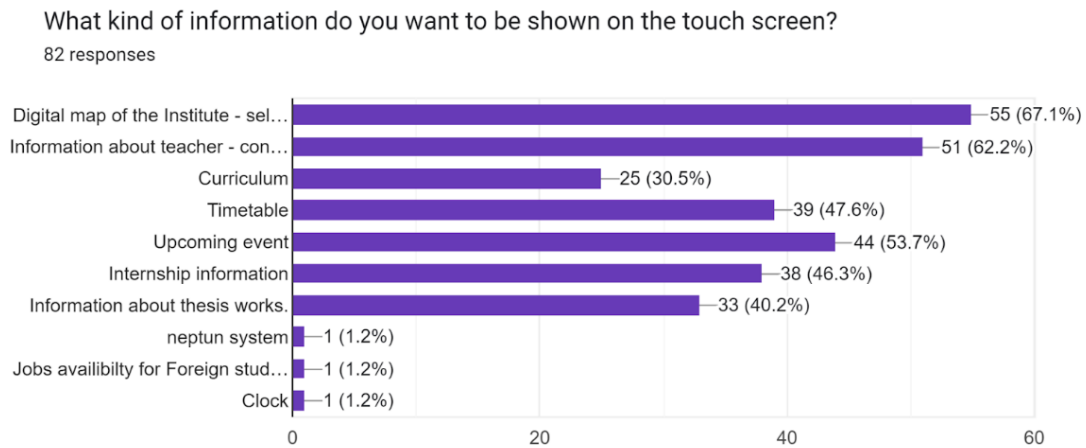


Figure 64. Basic Information

Third, the websites that appropriate to shown on the screen and the option to contribute other ideas and its answer.

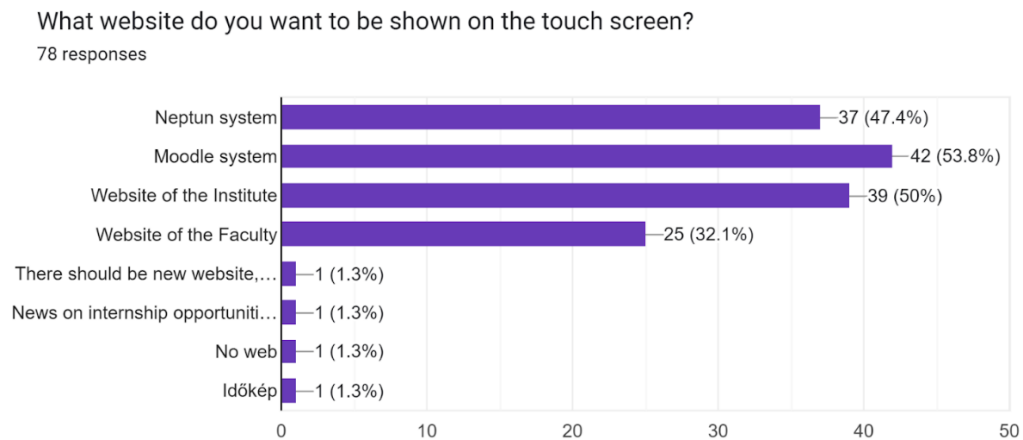


Figure 65. Website

Other possible advice for the system

5 responses

Show some jokes too

multiple language options

It should be beginner-friendly and definitely with the choice of different languages.

Awesome idea but i fear that this device is gonna be vandalized :(Please make sure that the device is only placed out around the receptionist or other personel who can supervise that this thing os gonna be used properly and with care. Also use as durable parts as you can. Good luck for the tinkering and have fun.
A hungarian EE bsc student

Provide as much information about locating classes and teachers offices

Figure 66. other answer for the survey

Now after collecting all the information, we can see that the majority of the students who filled out the survey was studied in Institute of Physic at Bemter. Based on their decision, I can make a conclusion that the bulletin board will contain information about the map of the Physic institute, the professors who teach and have their office room in the institute, the website related to the institute, and the upcoming event.

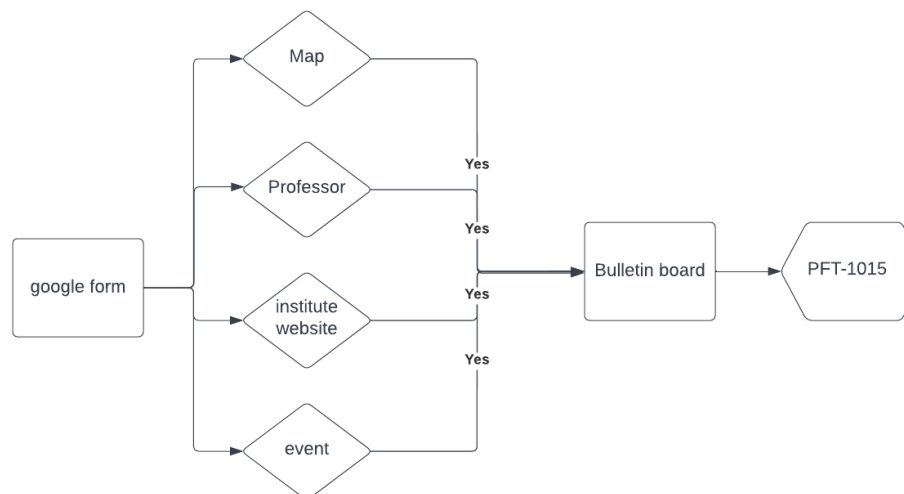


Figure 67. Block diagram of website system