PiCalc: Raspberry Pi Pico Based Pocket Calculator

by Makertronics

It's quite rare these days to find someone carrying a pocket calculator but school students who are don't allowed to carry phones may use it, or why don't they create their own pocket calculator.

In this Instructable, I will show you how I made a pocket calculator using raspberry pi pico and that's where it gets its name PiCalc.

Supplies:

Raspberry Pi Pico

SSD1306 OLED display module 128^64

20 X SMD Momentary Switches (6x3.5mm) for Keypad

Slide Switch for Power On/Off

Lipo Battery smaller than 30mmx30mmx4mm

TP4056 Lipo Charge/Protection Circuit

Some 3D Printed Parts

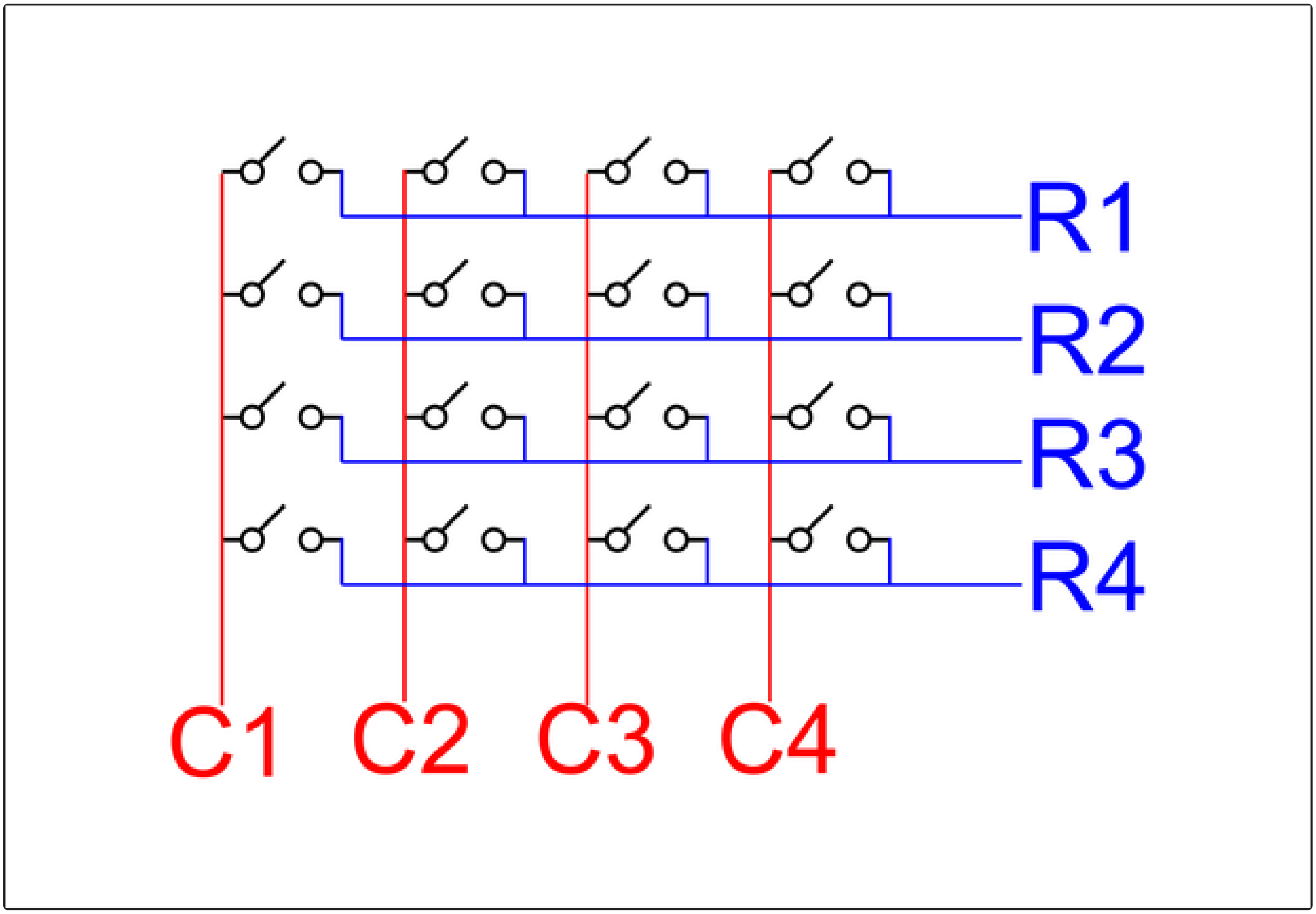
Copper Clad + Glossy Paper + Ferric Chloride if you are going to make PCB by yourself

https://youtu.be/ITYfOPvW5O8

Step 1: Principle Behind Keypad Matrix (Optional)

Before making the calculator, we need to understand the input mechanism of the calculator.
It is based on a keypad matrix when the input is multiplexed.
In simple terms, a grid of buttons/keys is arranged such that the terminals are connected row-wise and column-wise, as can be seen in the image.
All the buttons' right terminals are connected to their respective rows and all the left ones to their respective columns giving out eight connections, namely R1, R2, R3, R4, and C1, C2 C3, C4.

Now when a user pressed the button, we need to find out which column and row that press corresponds to.
And to do that, the microcontroller reads each row one by one and while turning on/off each column's supply one by one for each row. i.e., first, all the input lines are turned high by input pullup. The microcontroller reads that input line let suppose row is the input so, we will start reading from R1 then we output a logic low on C1 then C2, C3, and then C4. If R1 turns low for any of that, it will confirm the pressing of that button.



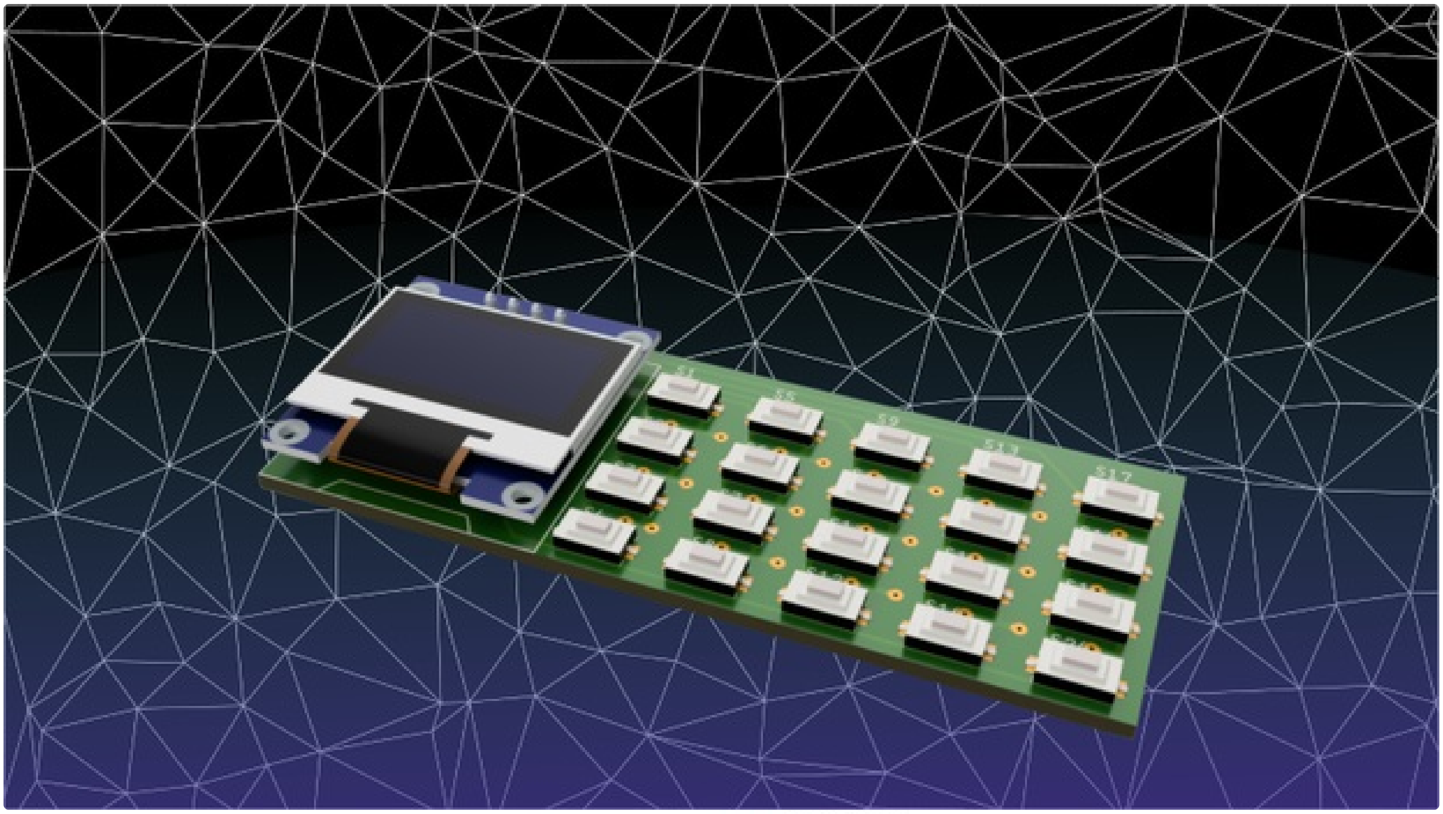
Step 2: Circuit & PCB Design

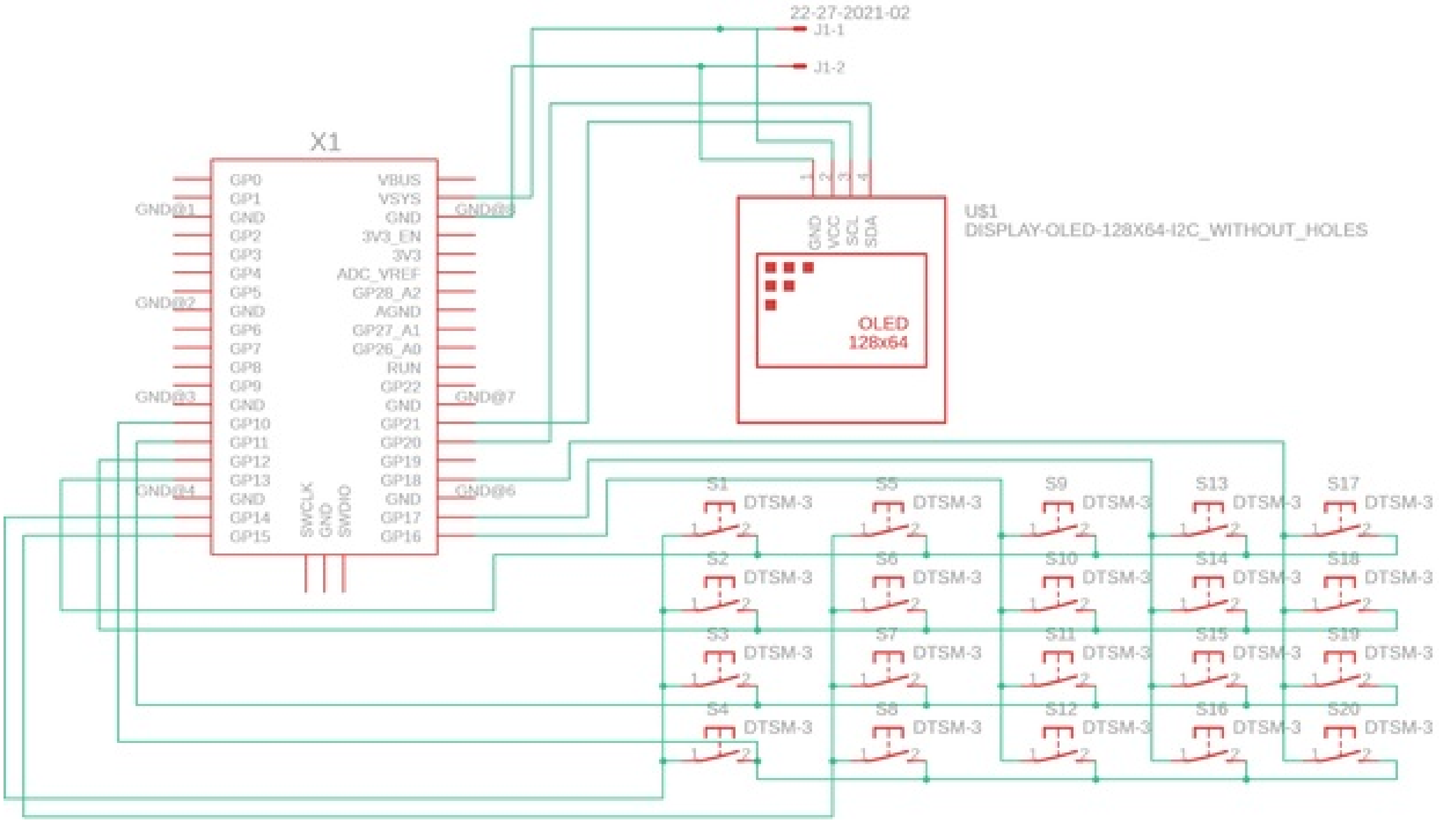
The Circuit will consist of 20 buttons arranged in a 4X5 keypad matrix, an SSD1306 OLED display, and a raspberry pi pico on the backside.

I am going to design a single-layer PCB and that's why the column wires for the keypad will be connected externally by soldering bridge wires also the connection of the keypad matrix on the top layer to the raspberry pi pico on the bottom layer will be done by connecting vias from pico to array of vias on the left side of the PCB.

Also, the Gerber File for the Double Layer PCB can be found on this link.

You can use that file to get PCB manufactured from **Seeed Studio Fusion PCB**. They take care of the entire fabrication process from PCB manufacturing, parts sourcing, assembly, and testing services, so you can be sure that they are getting a quality product. After gauging market interest and verifying a working prototype, Seeed Propagate Service can help you bring the product to market with professional guidance and a strong network of connections.





Step 3: PCB Manufacturing

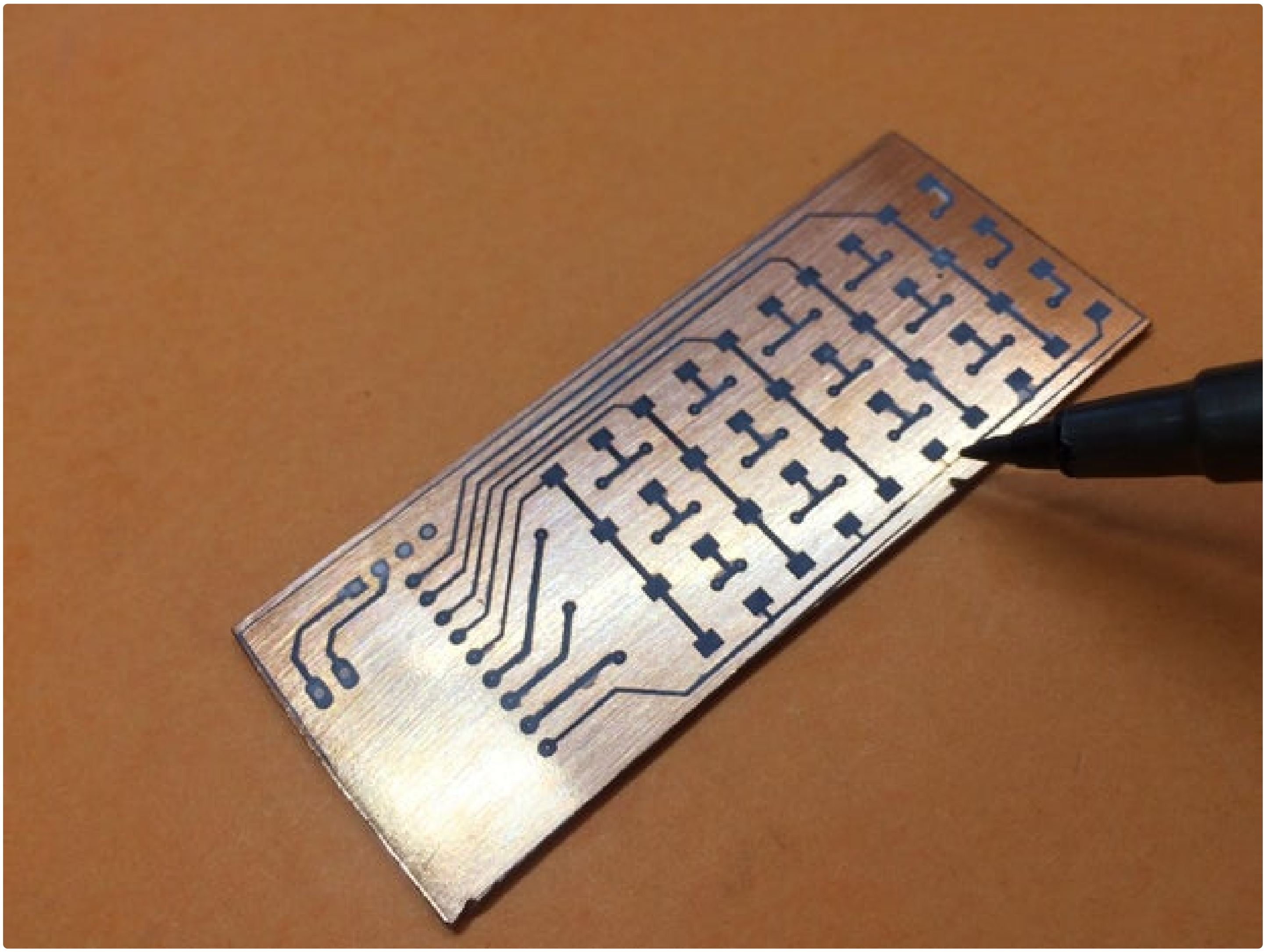
For making PCB we are going to use Toner Transfer Method.

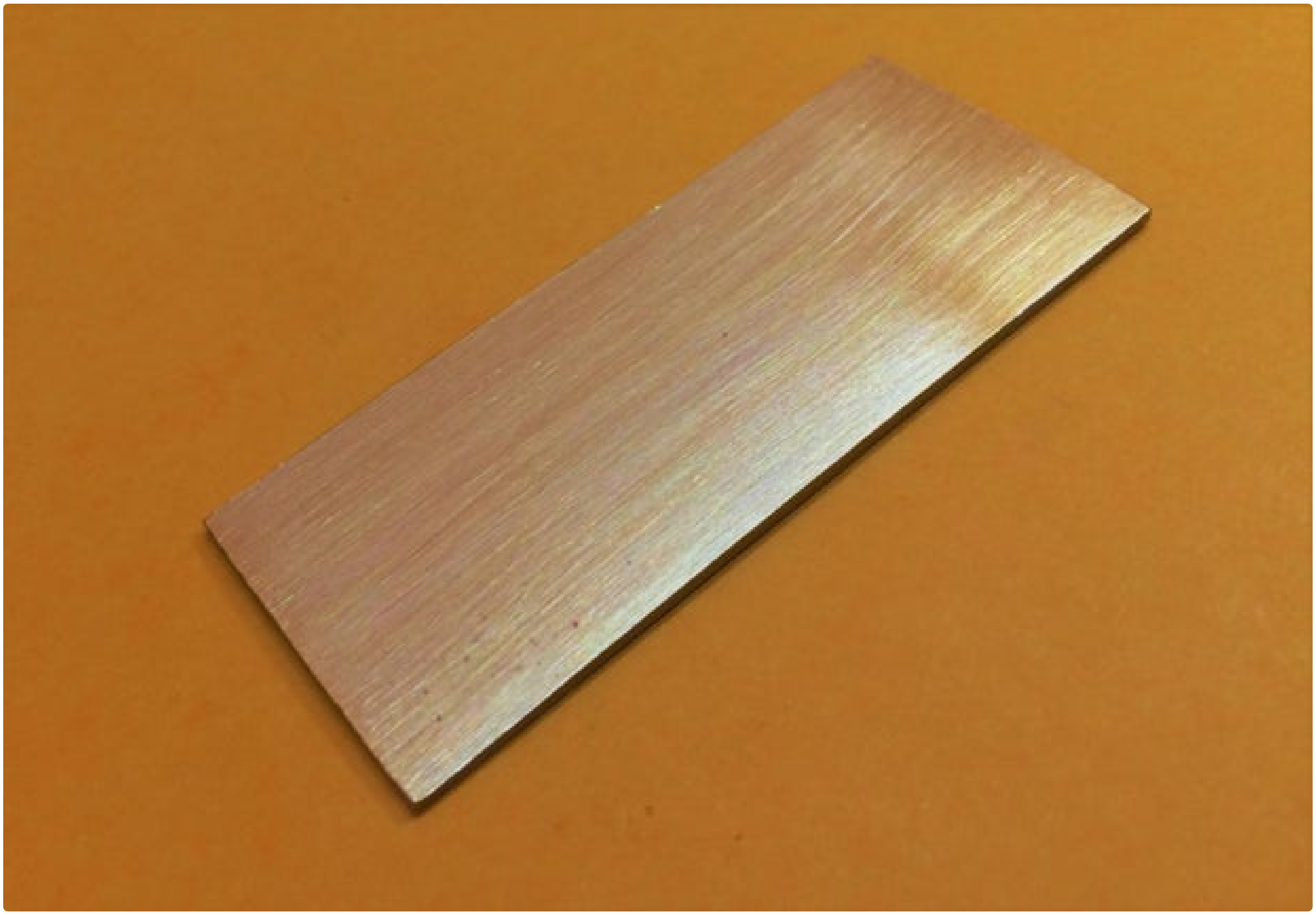
We will take a Copper Clad of dimensions 30 mm by 78 mm and clean it using some high grit sandpaper to remove the oxidation layer.

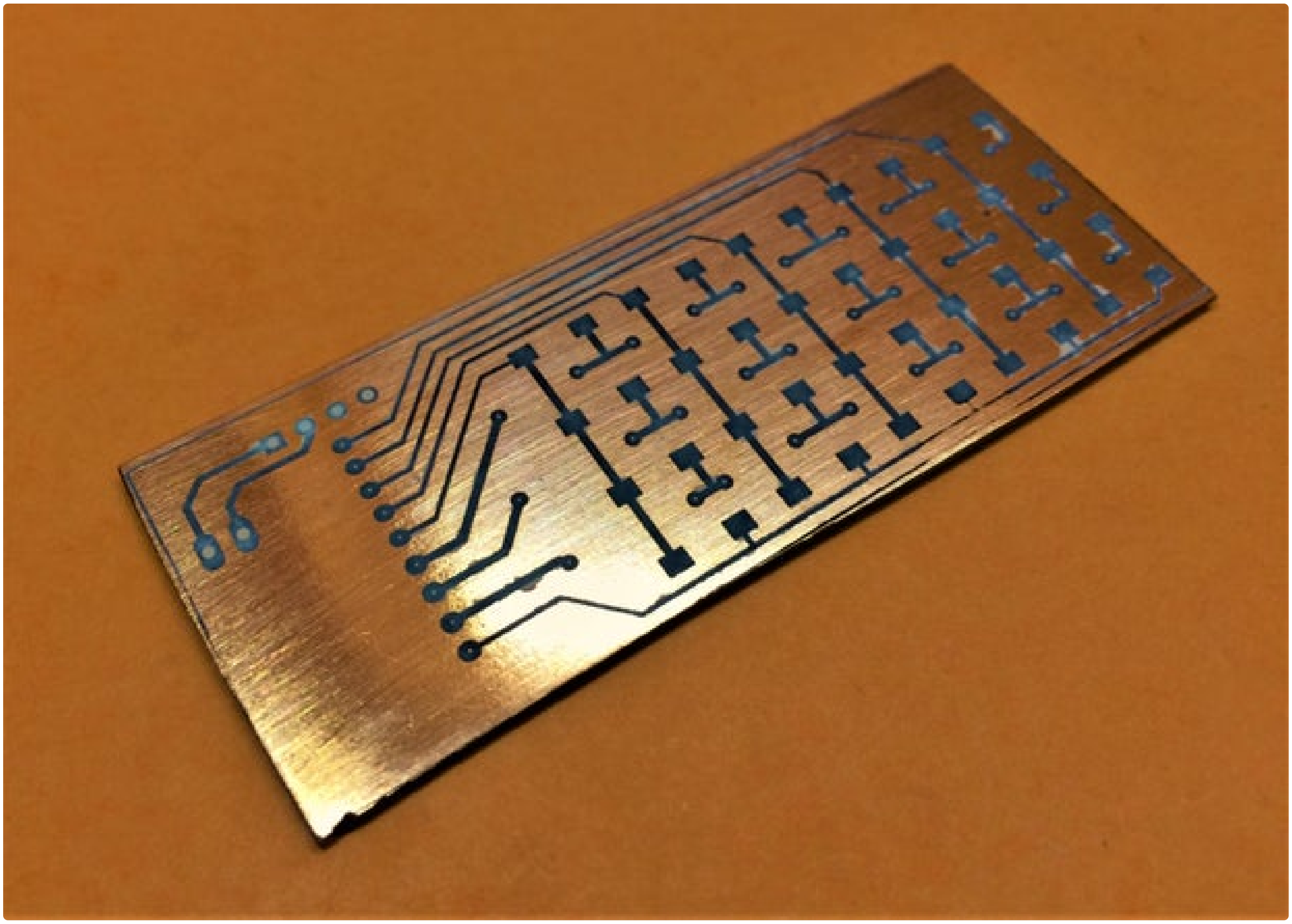
Then print the PCB Design(as given below) on glossy paper and align it to the copper. Use some hot press or iron to heat for 5 minutes to transfer the tonner on a glossy paper to copper clad which will later be used as a template to etch away the copper from where the tonner isn't there.

Soaking the PCB + glossy paper in warm water will help in easily removing the glossy paper.

If any trace gets damaged during the removal process, use a permanent marker to redraw the trace.





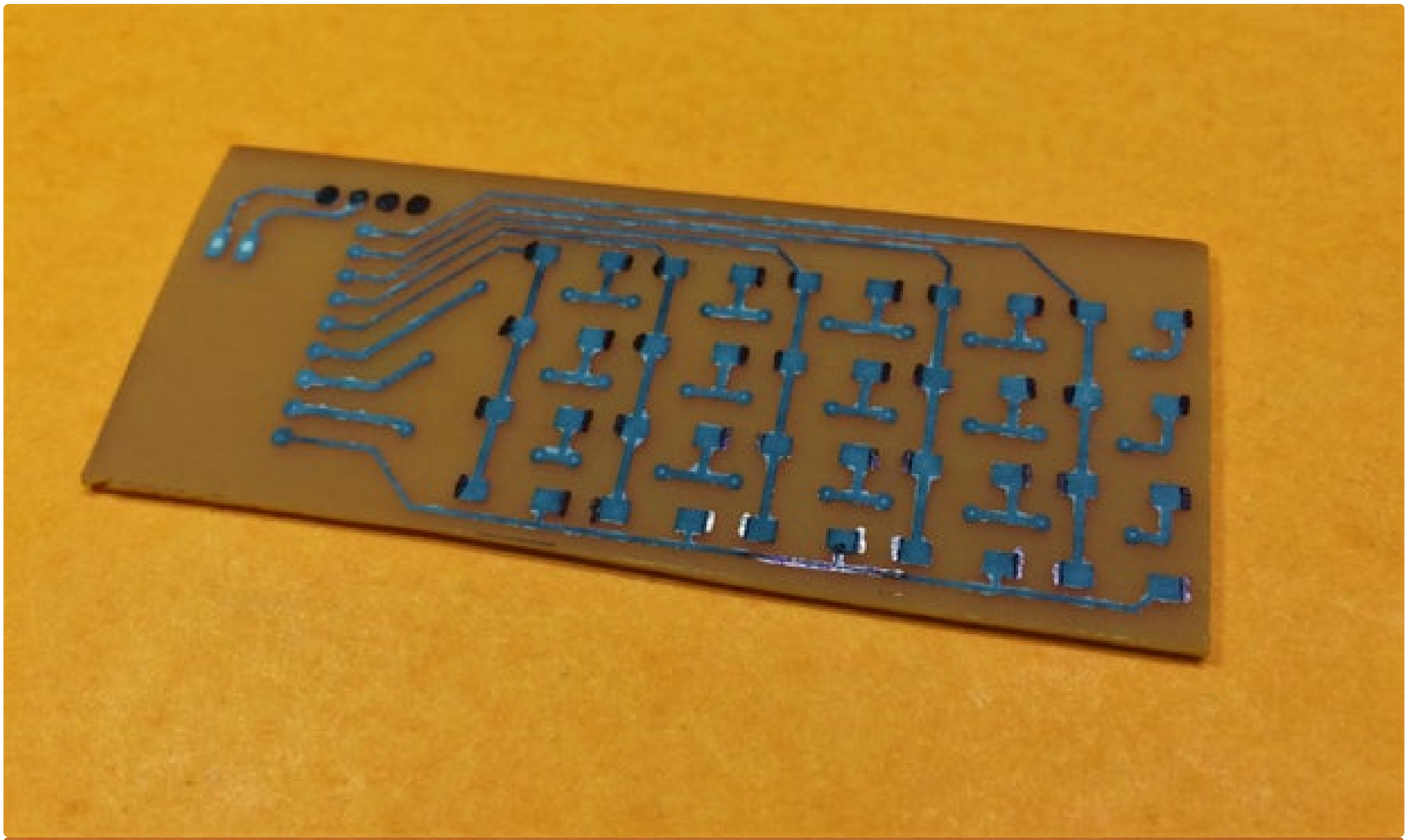


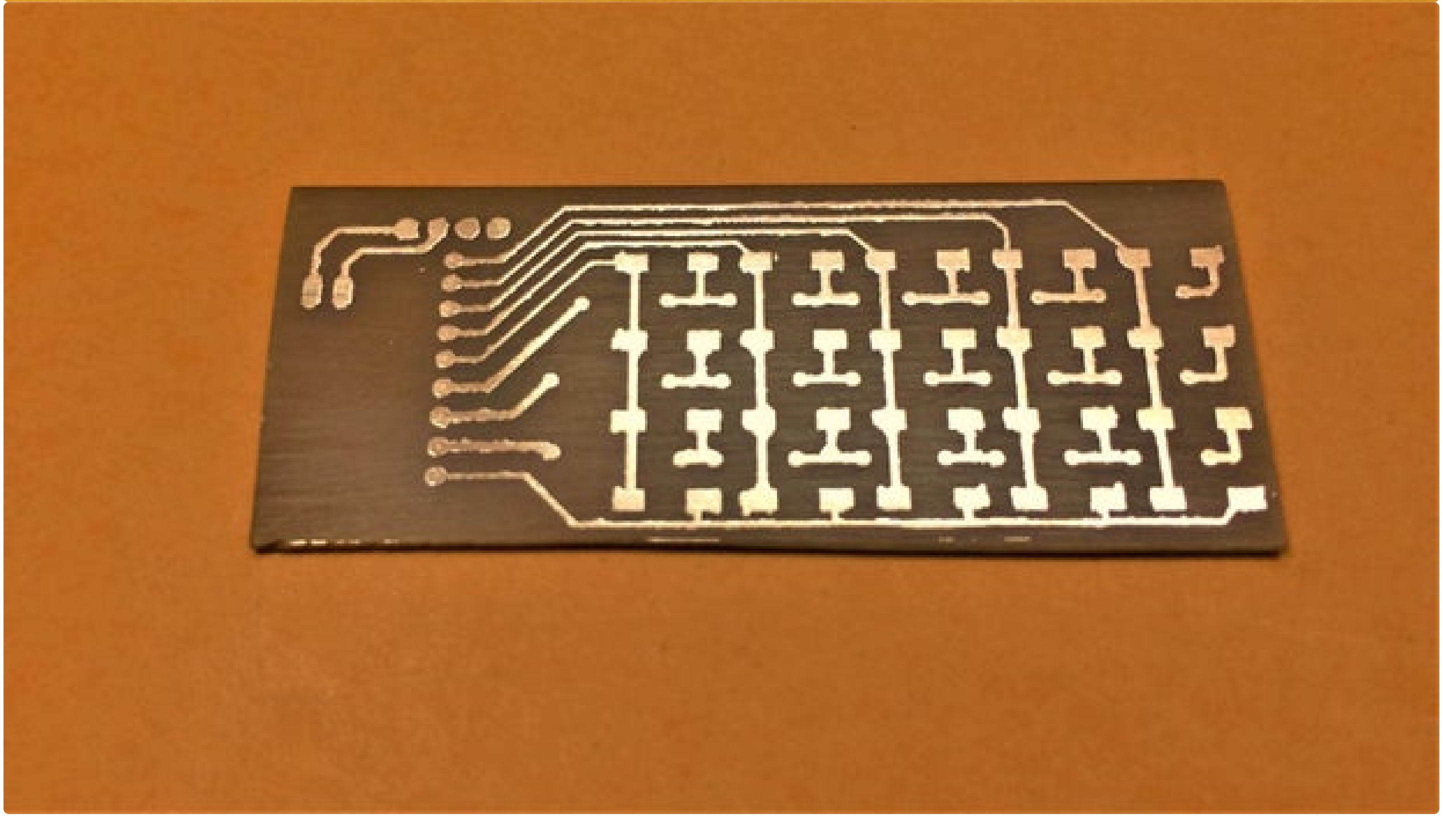
Step 4: Etching the PCB

Now drop the PCB in Ferric Chloride solution and aggregate the solution for around 5 minutes to etch away the unwanted copper.

Use some Acetone to remove the tonner from the etched PCB and a shinny layer of copper will emerge out of it.

Drill the holes (0.8mm diameter) for OLED displays and vias for connecting keypad matrix to raspberry pi pico.



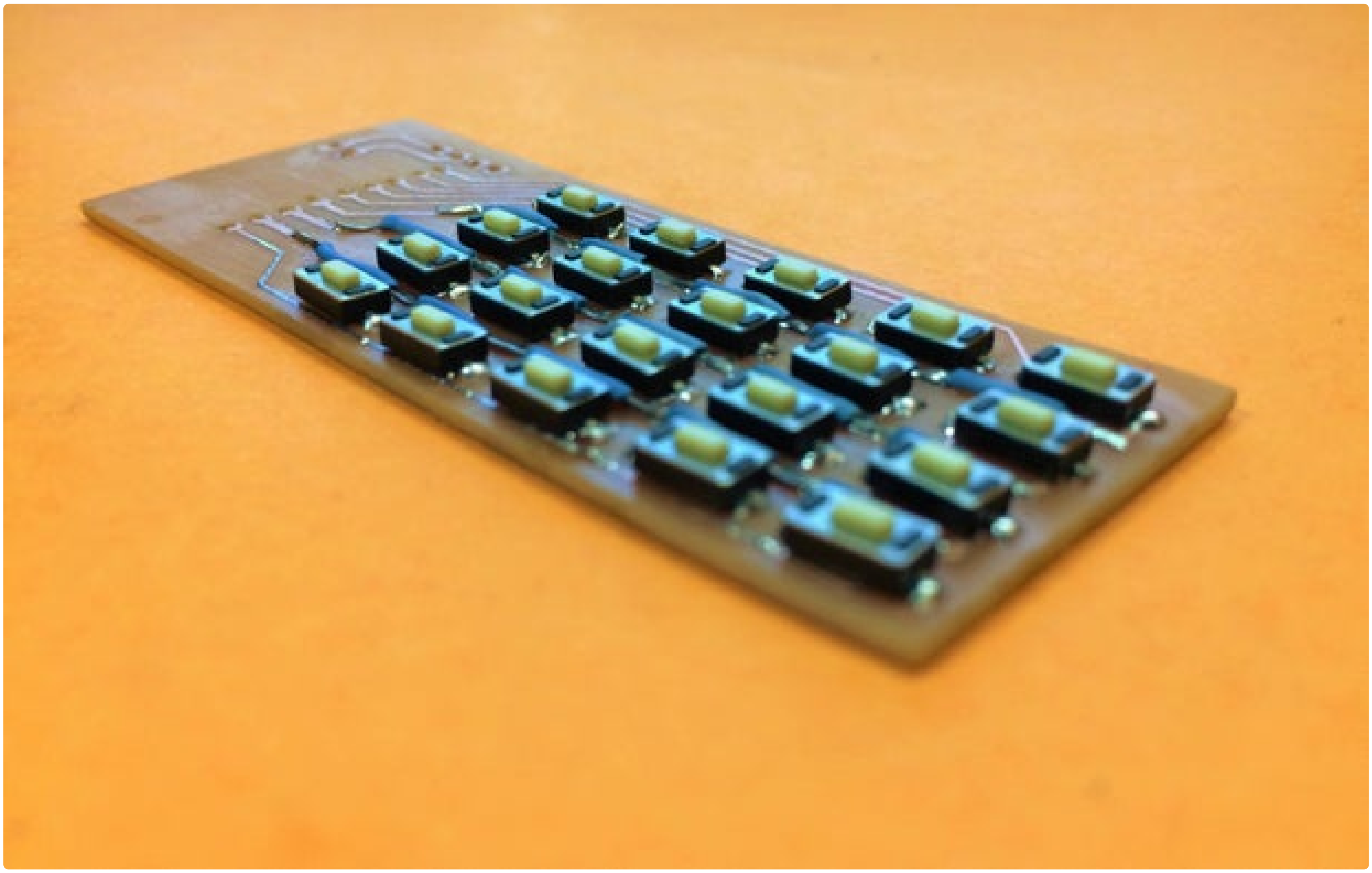


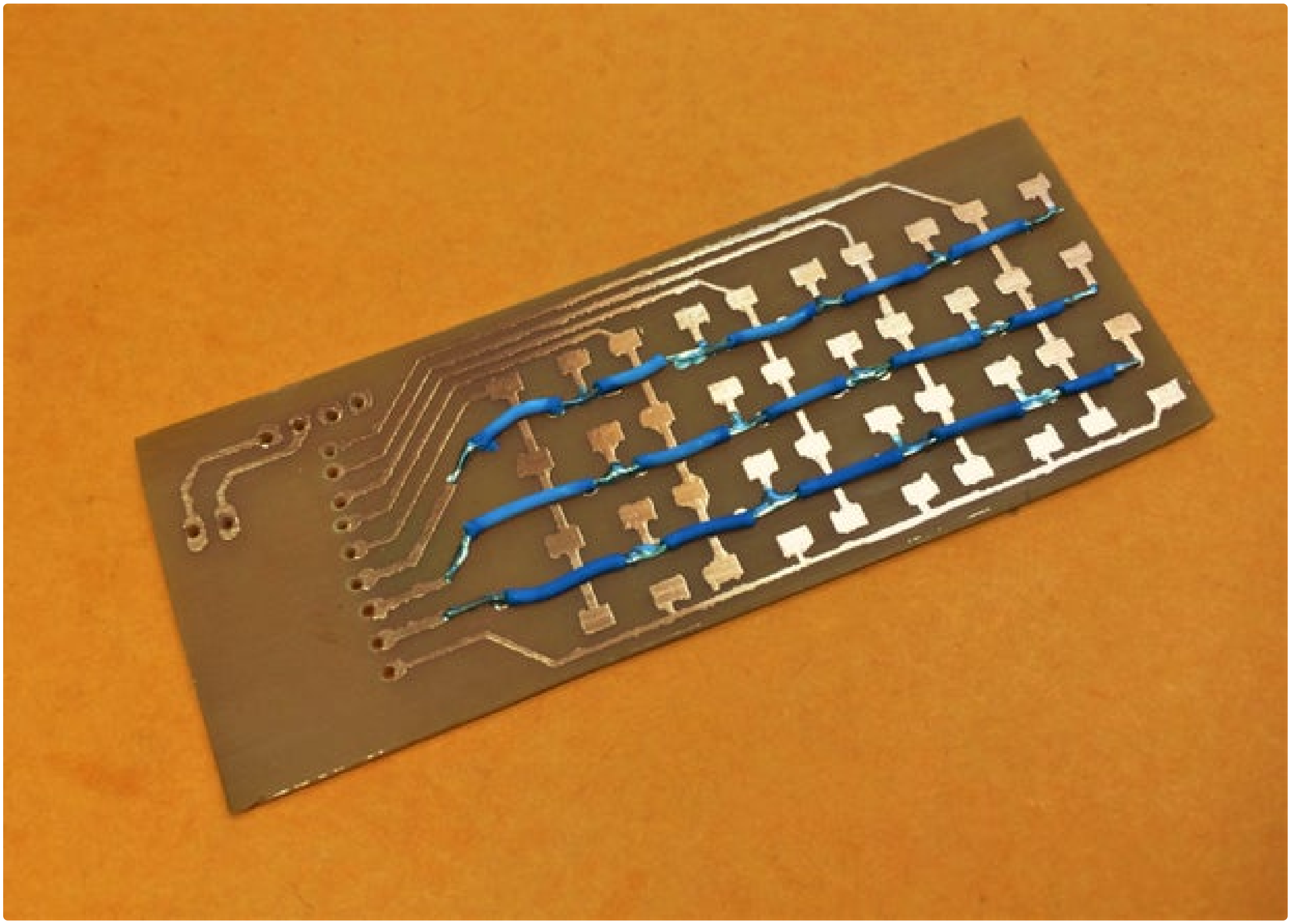
Step 5: Soldering the Parts

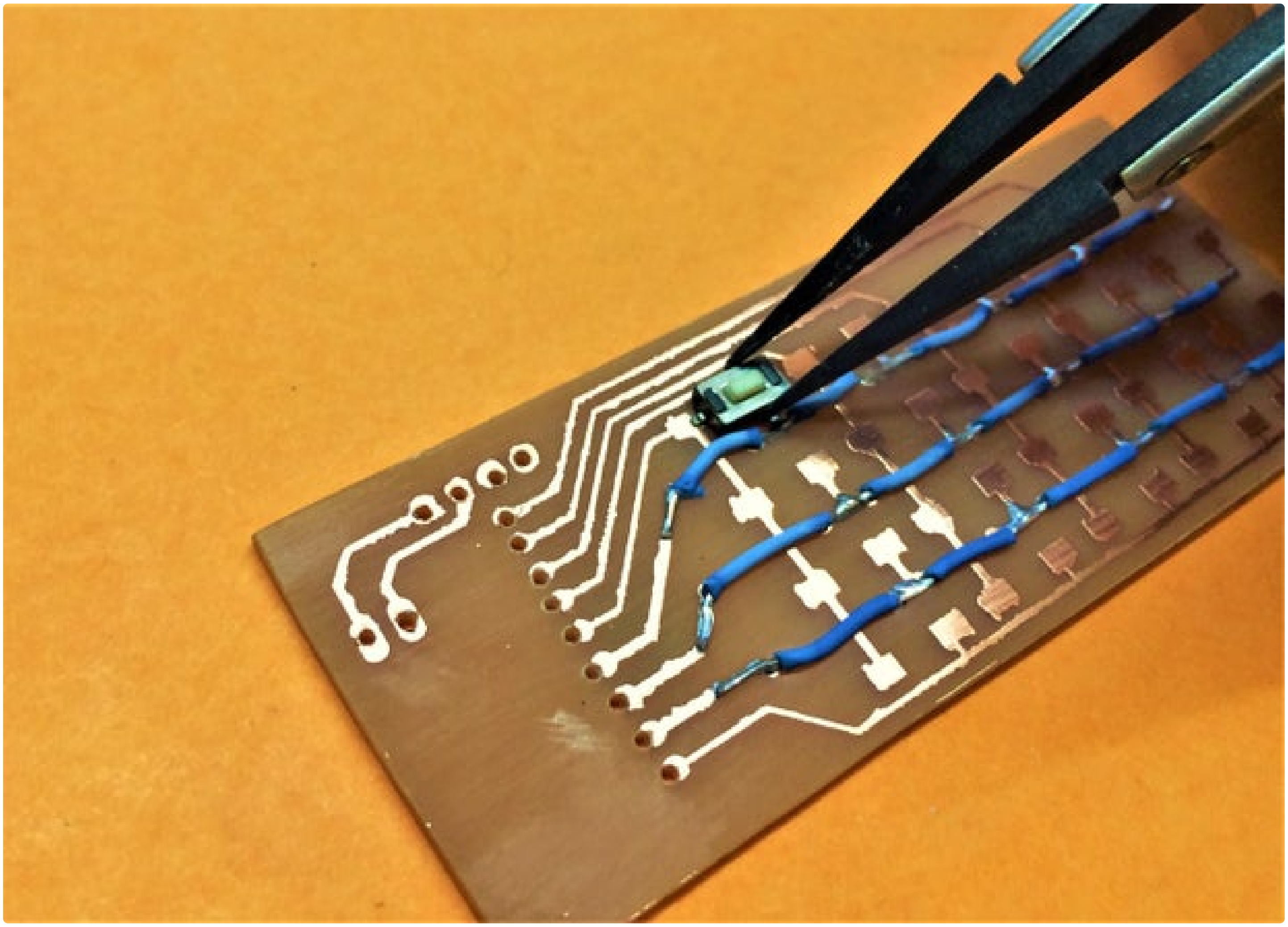
Now it's time to solder the parts.
If you have manufactured your own PCB like me then you first have to solder the bridge wires to make the connections for the column of the keypad matrix.

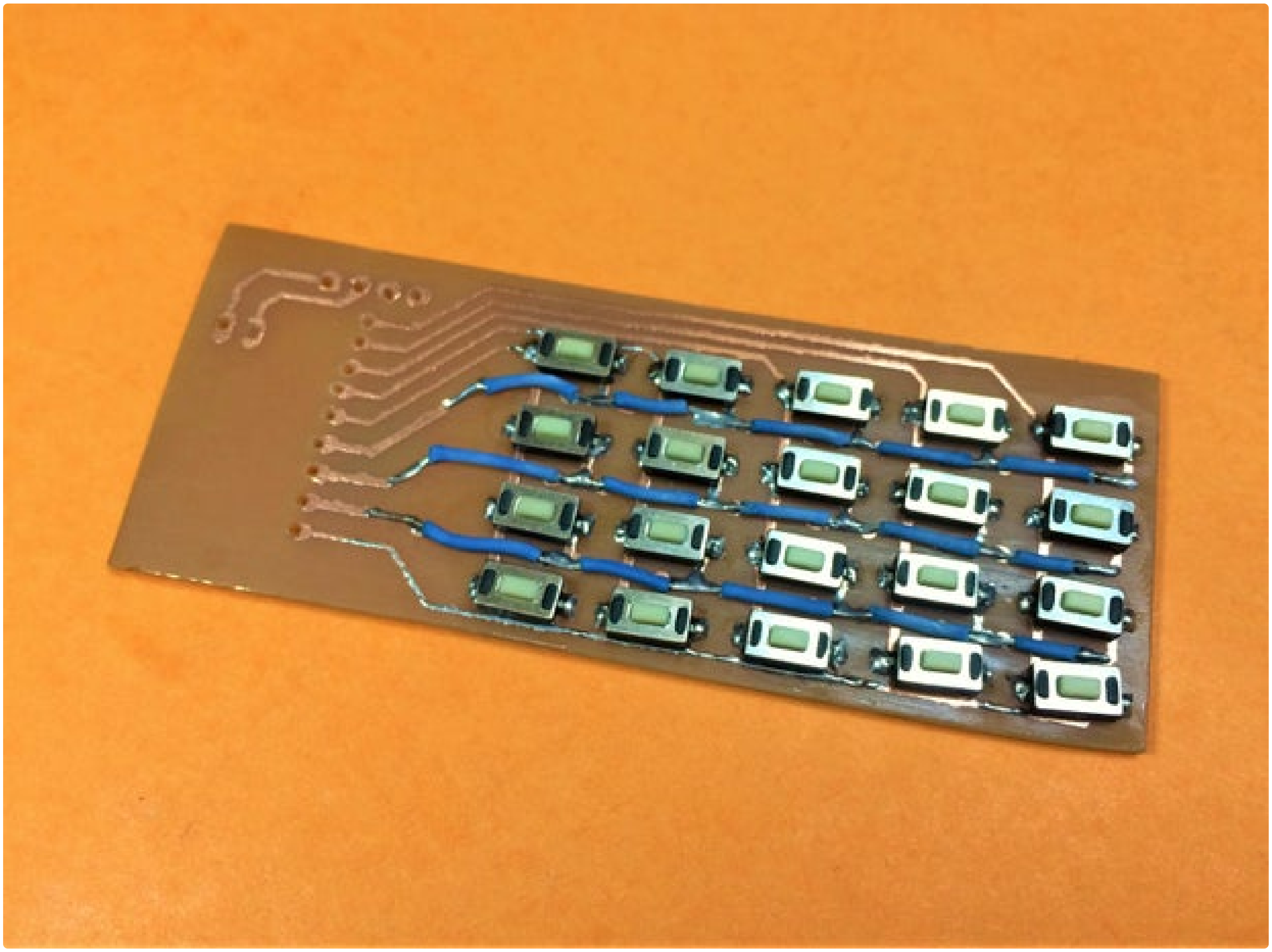
Out of 4 rows, you need to do that only for 3 rows as the last row is already traced out with copper.

After that, we can go forward with soldering the buttons.





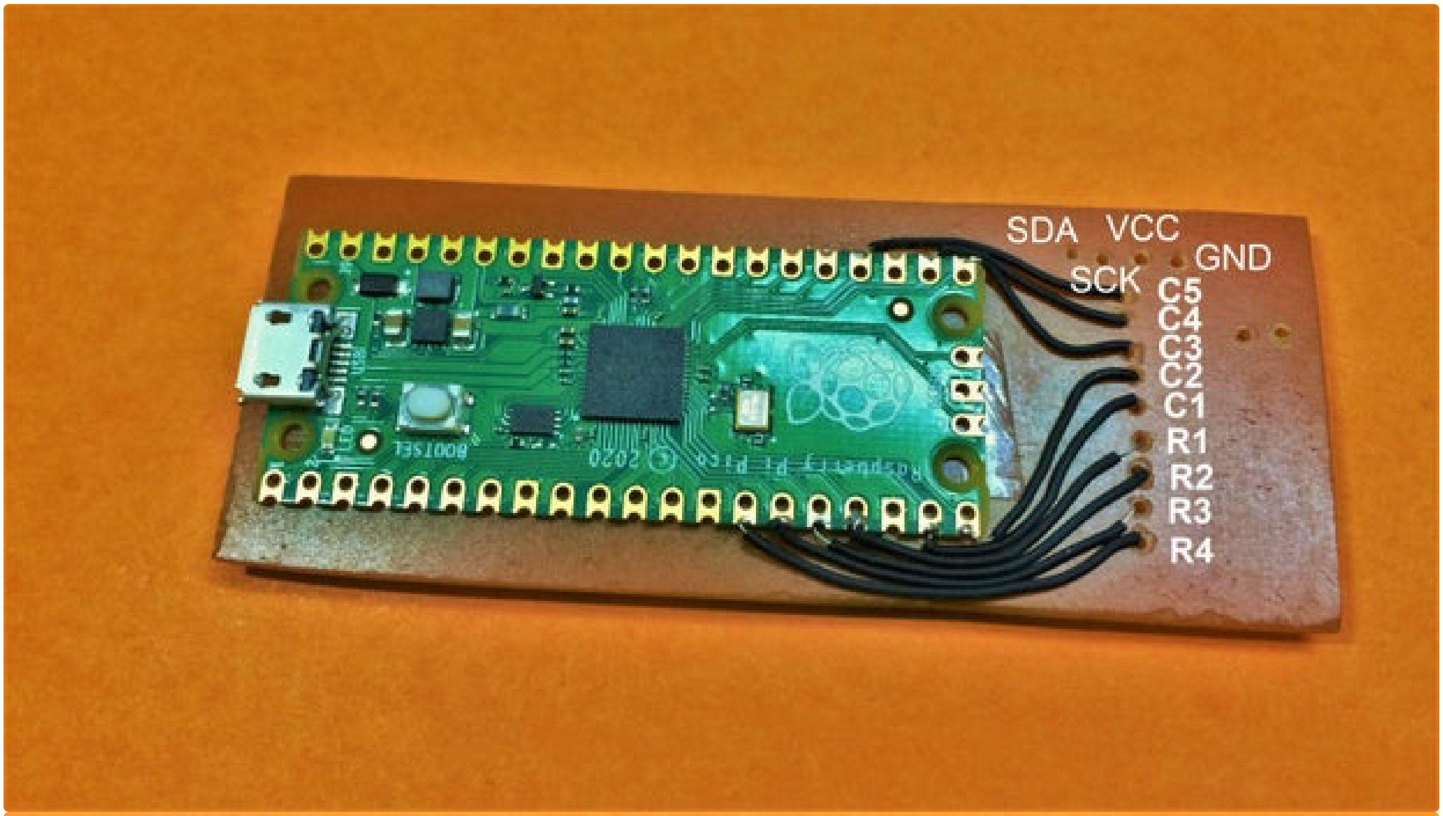




Step 6: Connection to Raspberry Pi Pico

After soldering all the buttons on the top layer we can stick the RPi Pico on the bottom layer using some double-sided tape and then solder the wires from RPi pico to the array of vias on the right of the pico as in the image according to the schematic.

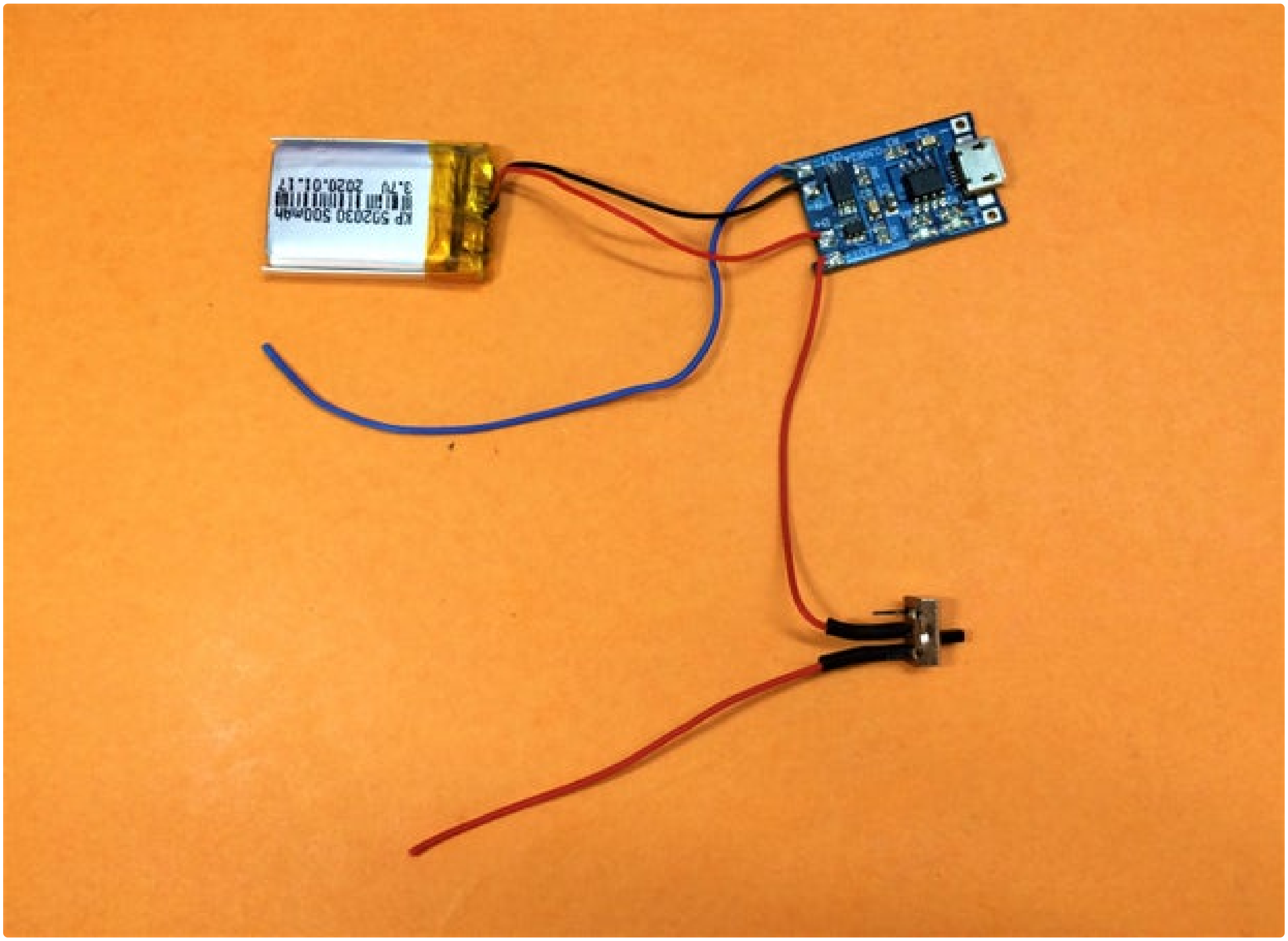
Only after these connections, we can attach the OLED display as it will cover the holes for soldering these wires.

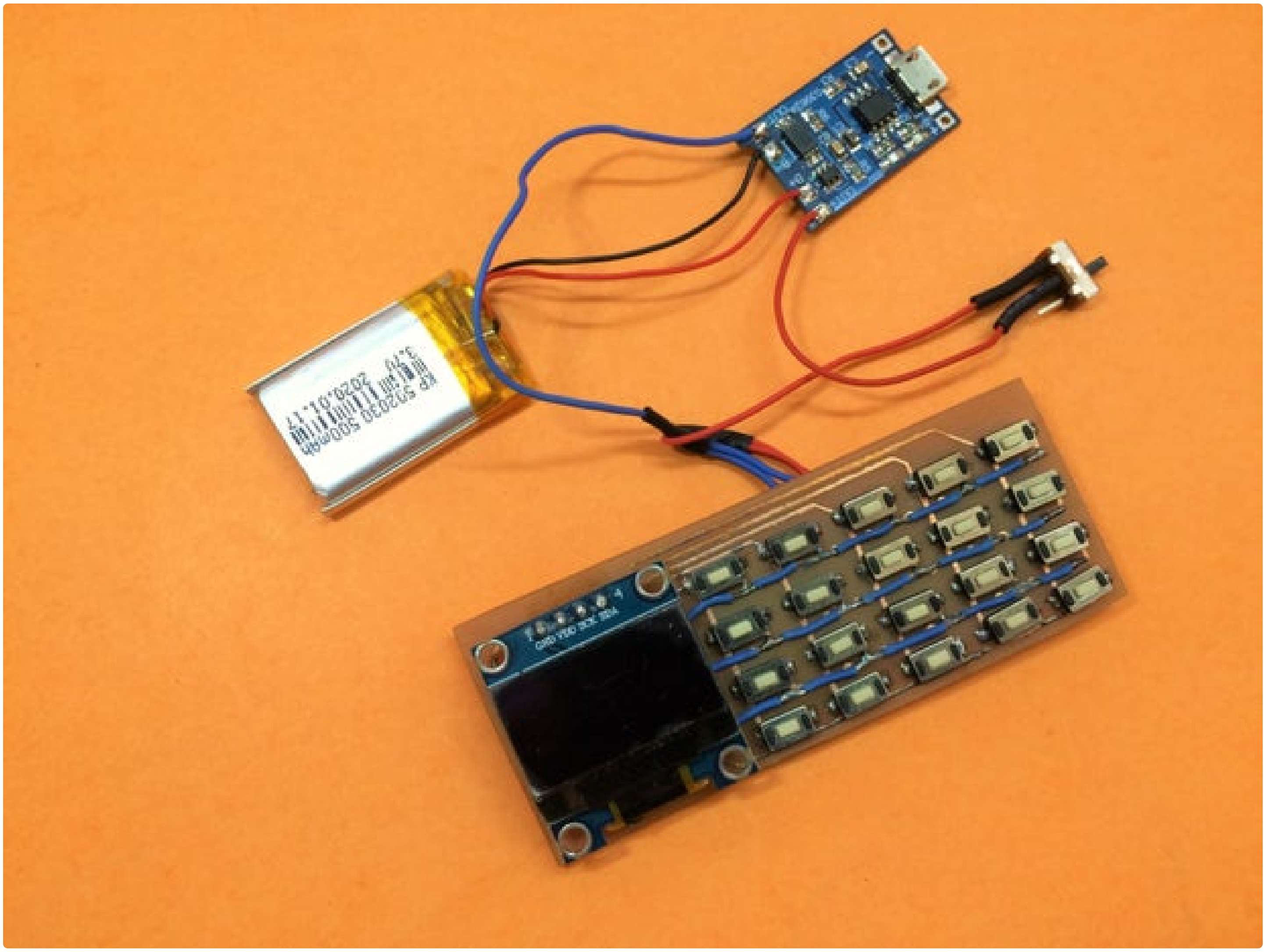

SDA
VCC
GND
SCK
C5
C4
C3
C2
C1
R1
R2
R3
R4

Step 7: Connecting Battery and Charging Circuit

After the OLED display its time to connected the battery and LiPo protection and charge circuit TP4056 along with an on/off switch.

properly insulate all the connection using heat shrink as all will be going to enclosed in an enclosure later.





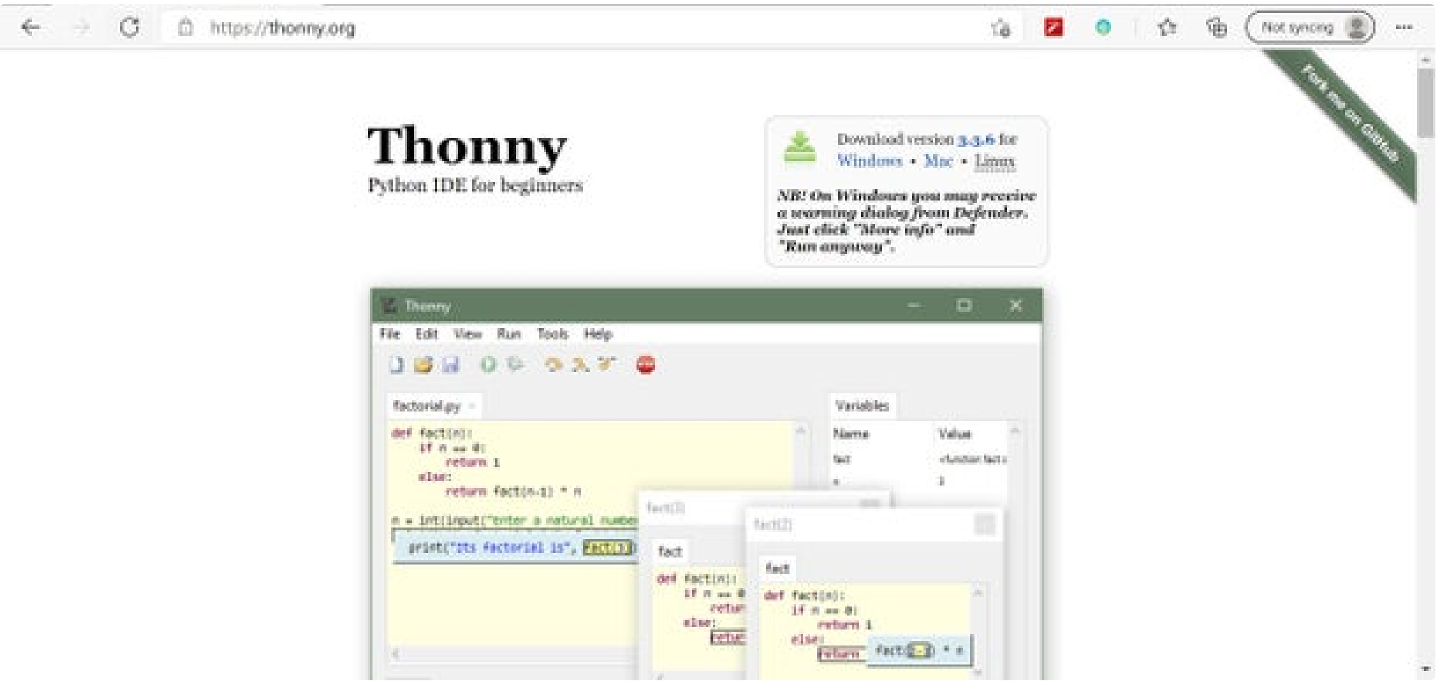
Step 8: Installing Thonny IDE

I have programmed the Raspberry Pi Pico using Micropython SDK, which can be easily done through Thonny IDE. It is the same IDE which some of you might remember from traditional Raspberry Pi Linux Computers.
You can get one from Windows as well as Mac from https://thonny.org/.

After downloading and installing the IDE, connect the pico to the computer while holding the Boot select button on pico which will power up it in programming mode.

A Folder will pop up on the screen, you can install the micropython firmware using the file from this link by dragging and dropping it into the folder that popped up.

Or you can also do it through Thonny IDE, By going into Run->Select Interpreter, there you will find micropython for RasberryPi Pico and at the bottom, on the window, you will see a blue text button "Install or Update Firmware".



Thonny
Python IDE for beginners

Download version 3.3.6 for
Windows • Mac • Linux

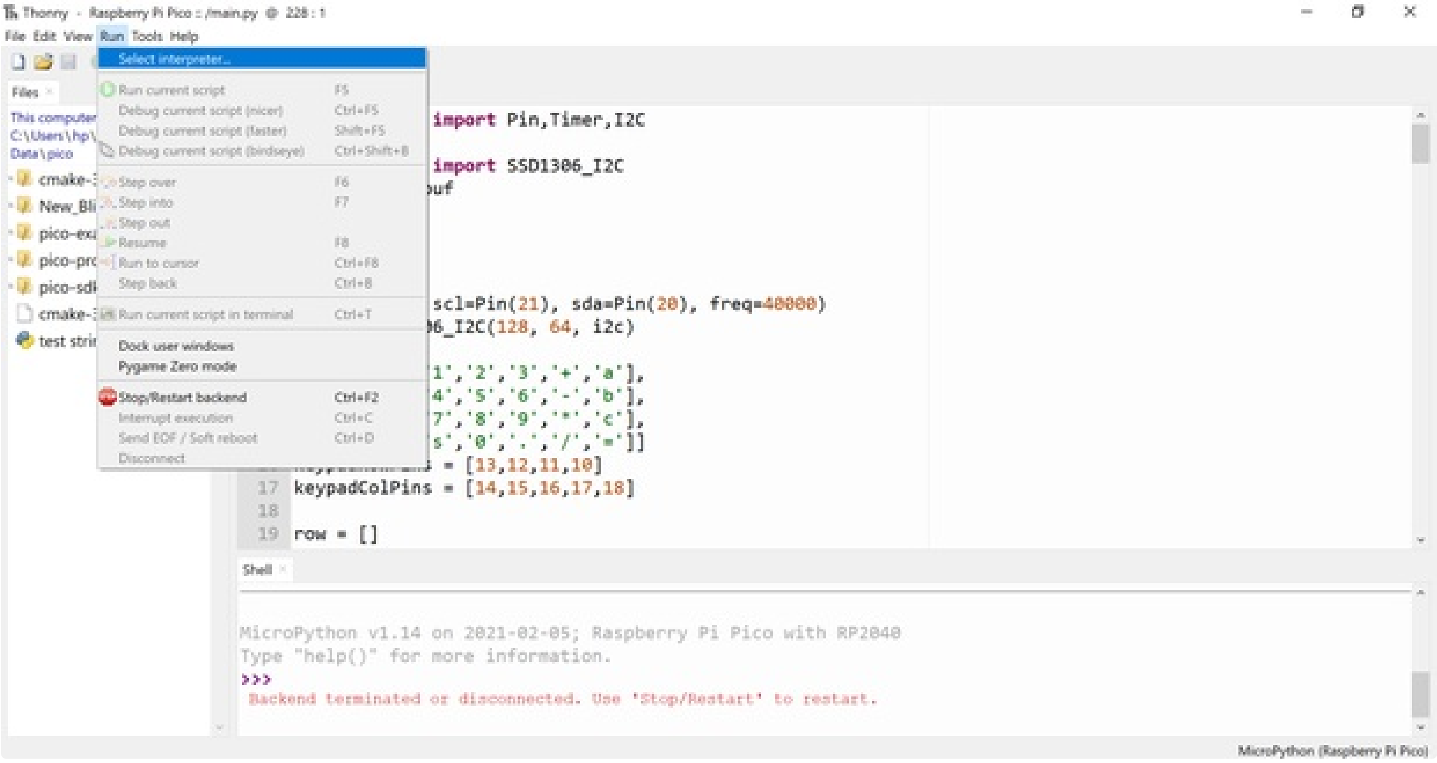
NB! On Windows you may receive a warning dialog from Defender. Just click "More info" and "Run anyway".

Step 9: Programming Raspberry Pi Pico

Before programming the Pico we need to install the library for OLED display, to do so go to the Tools Menu -> Manage Packages.
there you can search for SSD1306 and install the first one. It will be saved directly onto your PICO.

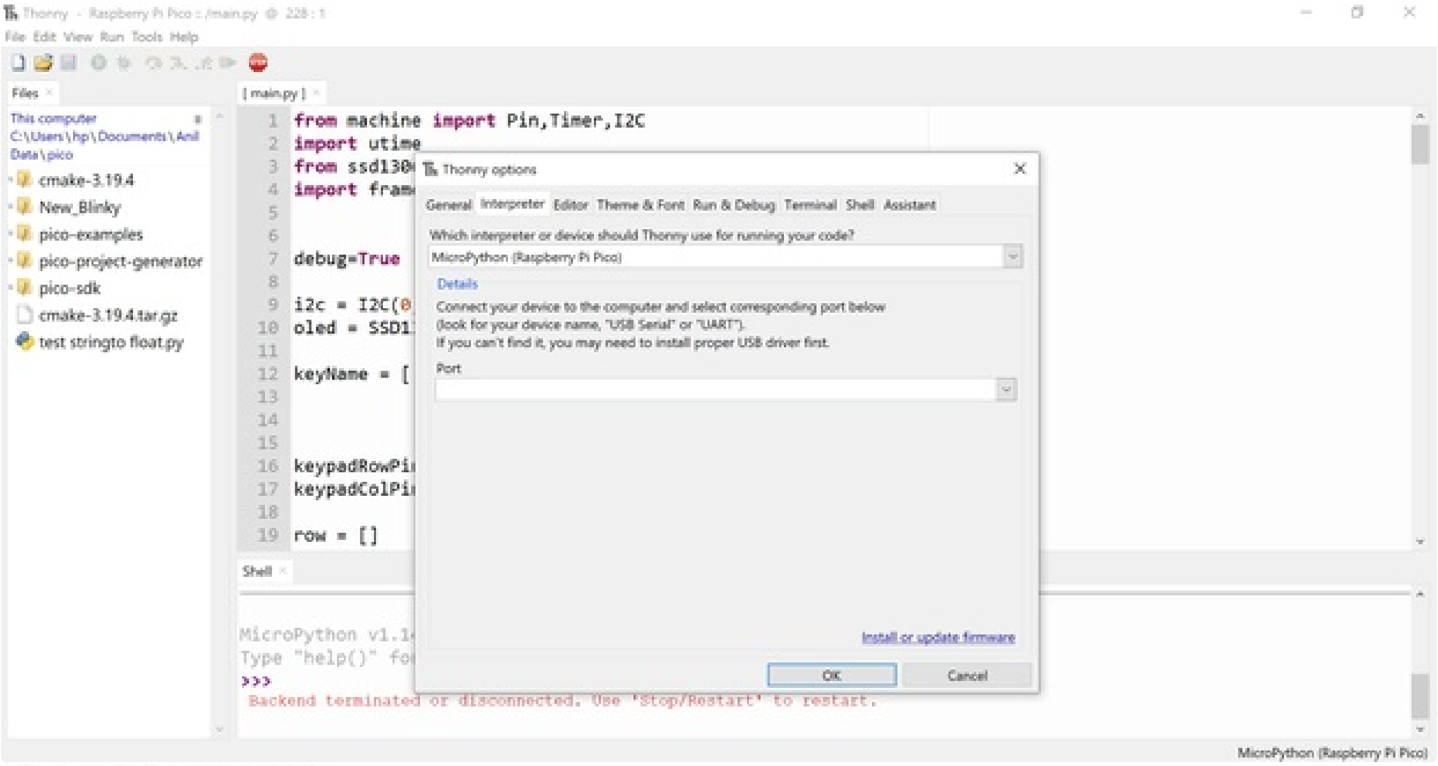
And then you can use the code given below to flash the PiCalc Firmware.

Now you have a barebone working calculator, to make it move friendly to sue we will enclose it in a 3D Printed Cover.


Thonny - Raspberry Pi Pico :: /main.py @ 228 : 1
File Edit View Run Tools Help
Select interpreter...
Run current script F5
Debug current script (nicer) Ctrl+F5
Debug current script (faster) Shift+F5
Debug current script (birdseye) Ctrl+Shift+B
Step over F6
Step into F7
Step out
Resume F8
Run to cursor Ctrl+F8
Step back Ctrl+B
Run current script in terminal Ctrl+T
Dock user windows
Pygame Zero mode
Stop/Restart backend Ctrl+F2
Interrupt execution Ctrl+C
Send EOF / Soft reboot Ctrl+D
Disconnect
Files
This computer
C:\Users\hp\
Data\pico
cmake-
New_Bli
pico-exa
pico-pro
pico-sdk
cmake-
test strin
Shell
MicroPython (Raspberry Pi Pico)

```
import Pin,Timer,I2C

import SSD1306_I2C
uf


scl=Pin(21), sda=Pin(20), freq=40000)
6_I2C(128, 64, i2c)

1','2','3','+','a'],
4','5','6','-','b'],
7','8','9','*','c'],
s','0','.','/','=']]
s = [13,12,11,10]
keypadColPins = [14,15,16,17,18]

row = []
```

```
MicroPython v1.14 on 2021-02-05; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
 Backend terminated or disconnected. Use 'Stop/Restart' to restart.
```



```
from machine import Pin,Timer,I2C
import utime
from ssd130
import fram

debug=True

i2c = I2C(0
oled = SSD1

keyName = [

keypadRowPi
keypadColPi

row = []
```

Which interpreter or device should Thonny use for running your code?

MicroPython (Raspberry Pi Pico)

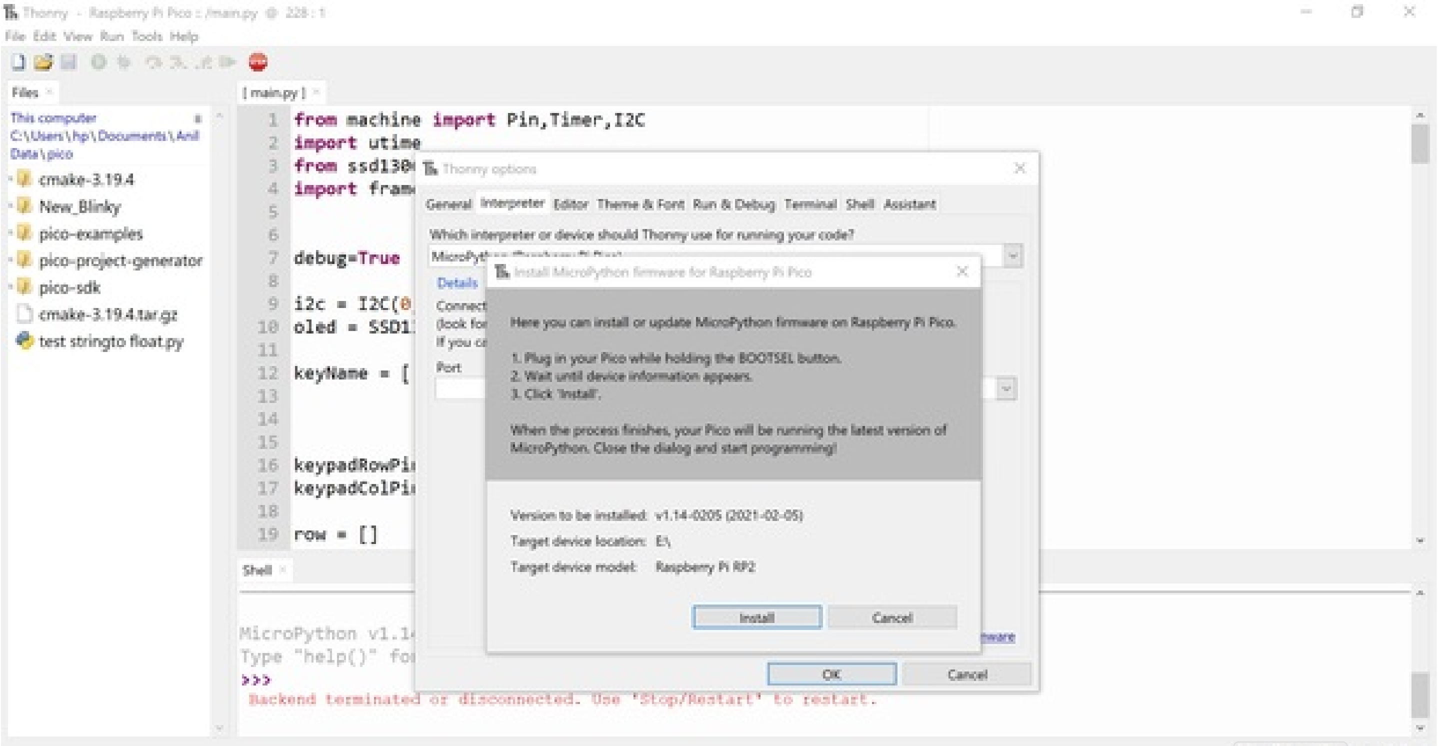
Details

Connect your device to the computer and select corresponding port below
(look for your device name, "USB Serial" or "UART").
If you can't find it, you may need to install proper USB driver first.

Port

Install or update firmware

```
MicroPython v1.1
Type "help()" fo
>>>
 Backend terminated or disconnected. Use 'Stop/Restart' to restart.
```


Thonny - Raspberry Pi Pico :: /main.py @ 228 : 1
File Edit View Run Tools Help
Files
[main.py]
Shell
Thonny options
General
Interpreter
Editor
Theme & Font
Run & Debug
Terminal
Shell
Assistant
Install MicroPython firmware for Raspberry Pi Pico
Install
Cancel
OK
Cancel

Here you can install or update MicroPython firmware on Raspberry Pi Pico.

1. Plug in your Pico while holding the BOOTSEL button.
2. Wait until device information appears.
3. Click 'Install'.

When the process finishes, your Pico will be running the latest version of MicroPython. Close the dialog and start programming!

Version to be installed: v1.14-0205 (2021-02-05)

Target device location: E:\

Target device model: Raspberry Pi RP2

Download

https://www.instructables.com/ORIG/FLC/FCHT/KMALLMW1/FLCFCHTKMALLMW1.py

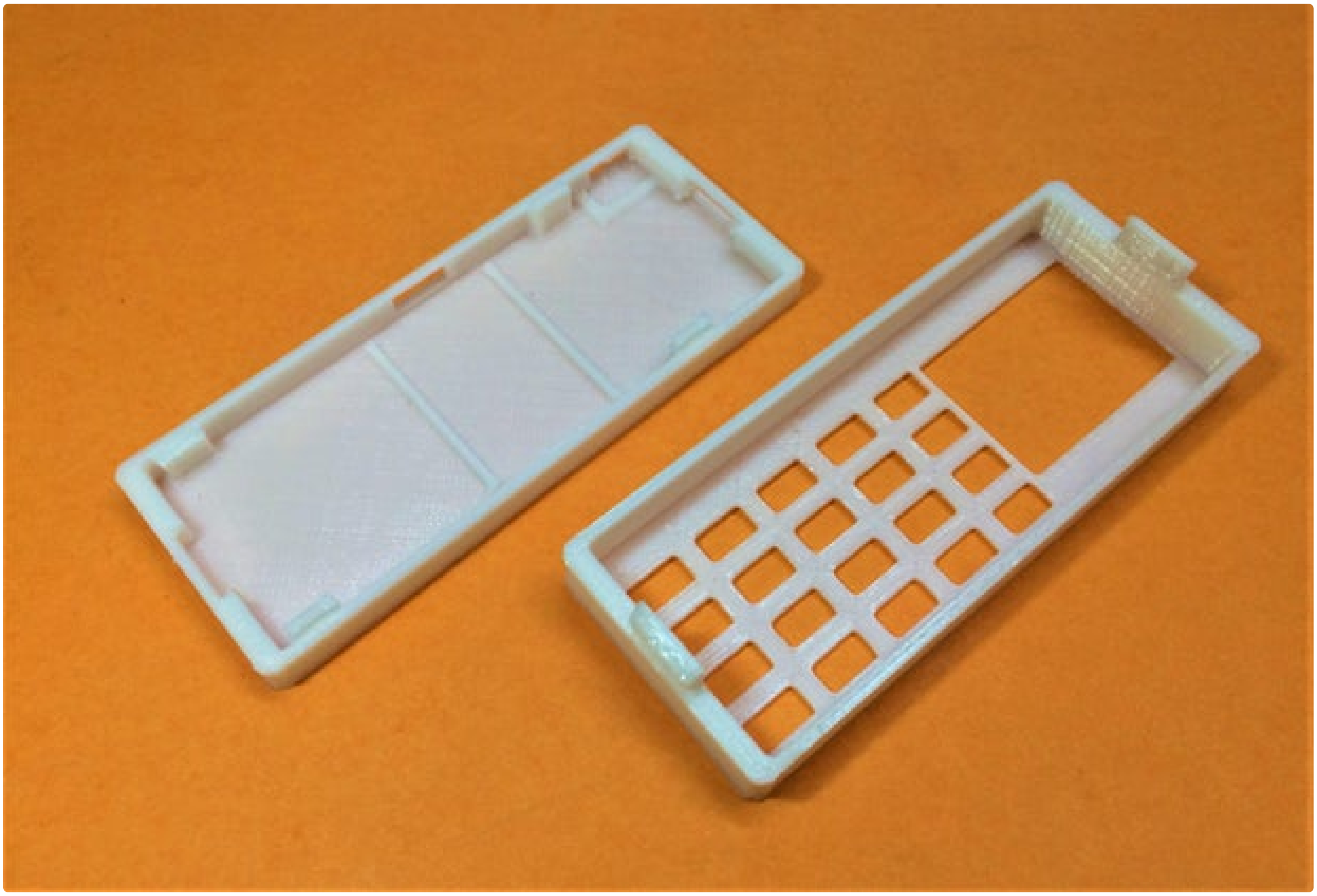
Step 10: 3D Printing Enclosure

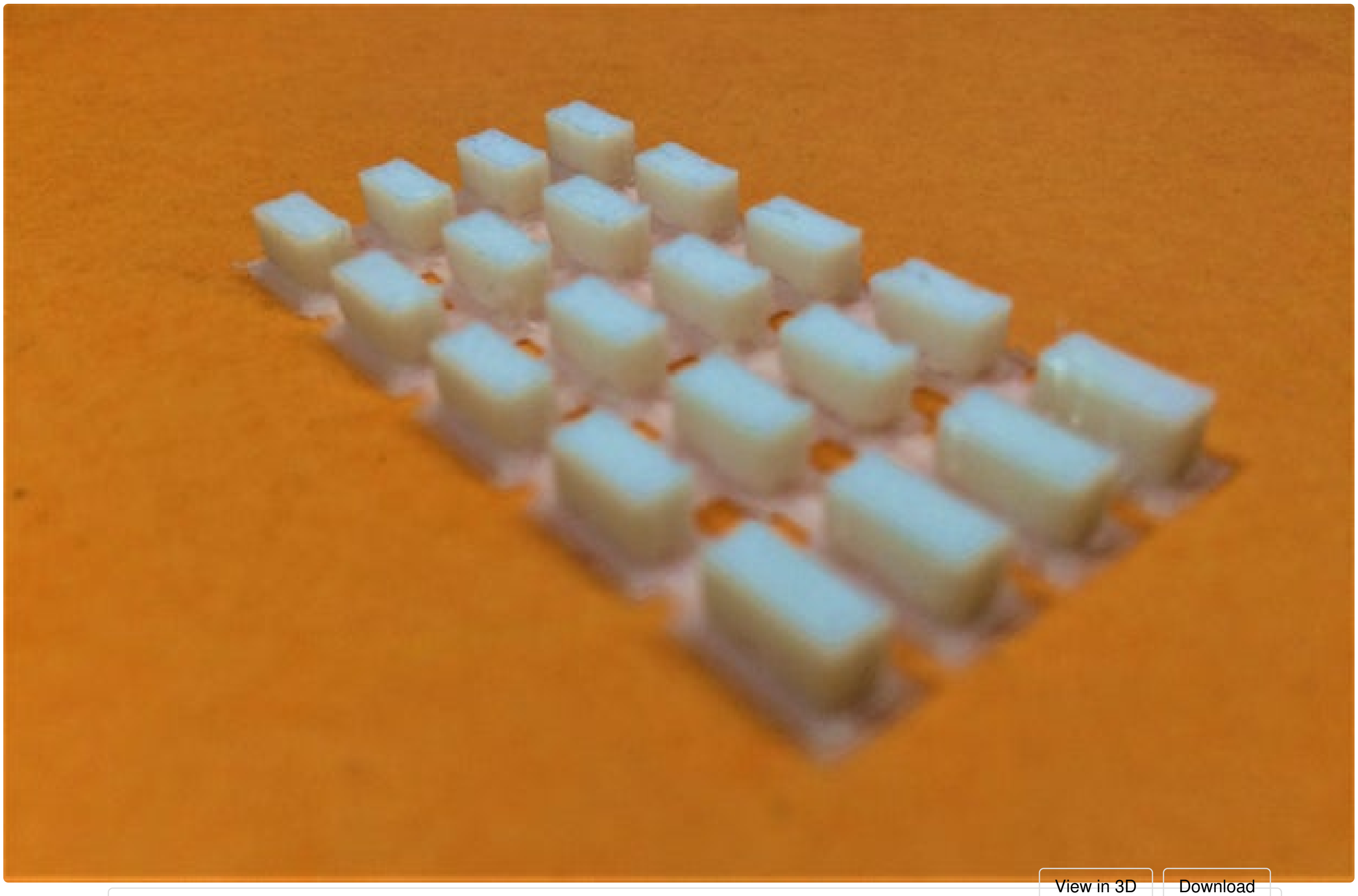
using the Eagle CAD's Fusion 360 Sync feature I transferred the 3D file of the PCB to Fusion 360 and then create an enclosure around it.

There are mainly 3 parts needed to be printed, a keypad matrix, top and bottom cover.

The 3D Print is done with 2 wall perimeter and 20% infill, there is no need for support.
There are some places where bridging will happen so keep it in mind and adjust your 3D printer accordingly.

I have also included the Step file for while construction if you wish to tweak some parameters.





https://www.instructables.com/ORIG/FI2/S2VO/KMALLN26/FI2S2VOKMALLN26.stl View in 3D Download

https://www.instructables.com/ORIG/FZ6/52CW/KMALLN25/FZ652CWKMALLN25.stl View in 3D Download

https://www.instructables.com/ORIG/FAB/I5MI/KMALLN23/FABI5MIKMALLN23.step View in 3D Download

https://www.instructables.com/ORIG/F8E/LZM4/KMALLN24/F8ELZM4KMALLN24.stl View in 3D Download

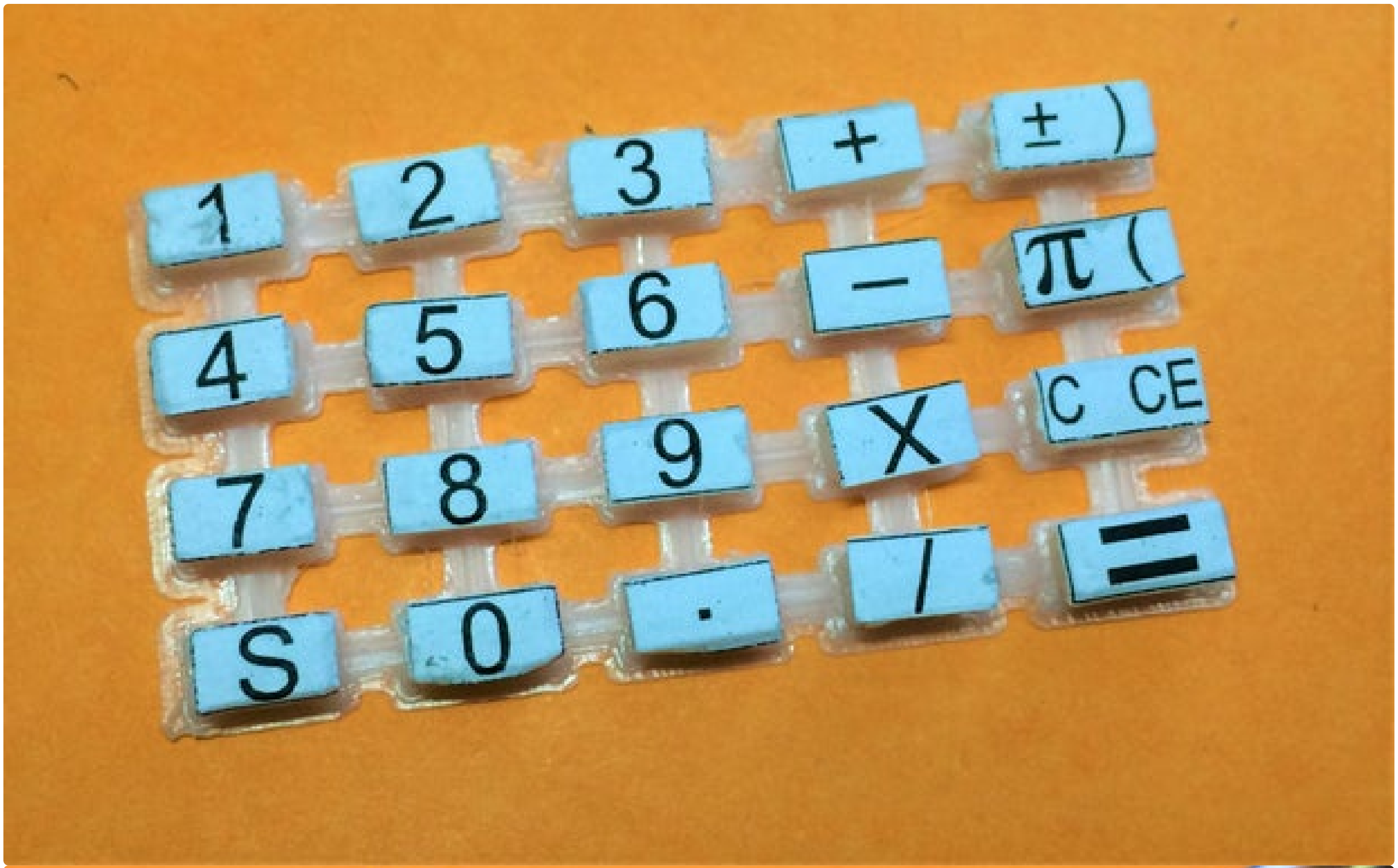
---

Step 11: Preparing Keypad

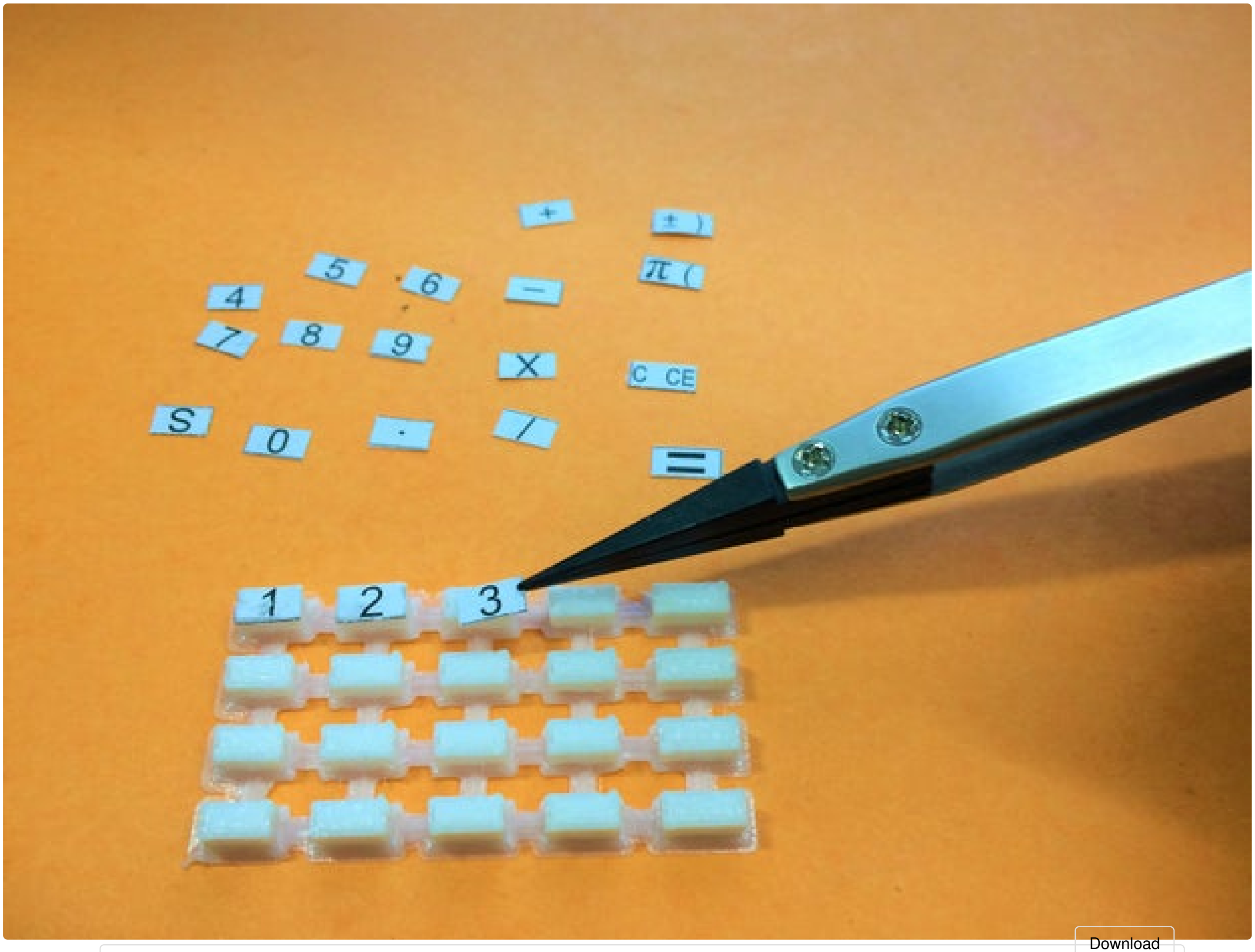
For a better understanding of which button to be pressed, all the buttons need to be labeled.
To do so, I have printed the numbers and symbols and cut them out.

I have attached the pdf file which you can print and stick as well.

Using some paper glue I stuck them onto the 3D printed Part.
As all the prints are done in PLA, paper Glue holds good onto it.


1 2 3 + ±)
4 5 6 – π (
7 8 9 X C CE
S 0 . / =

1 2 3 + ±)
4 5 6 – π (
7 8 9 X C CE
S 0 . / =

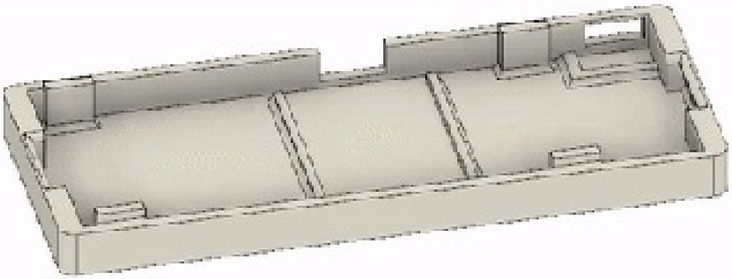


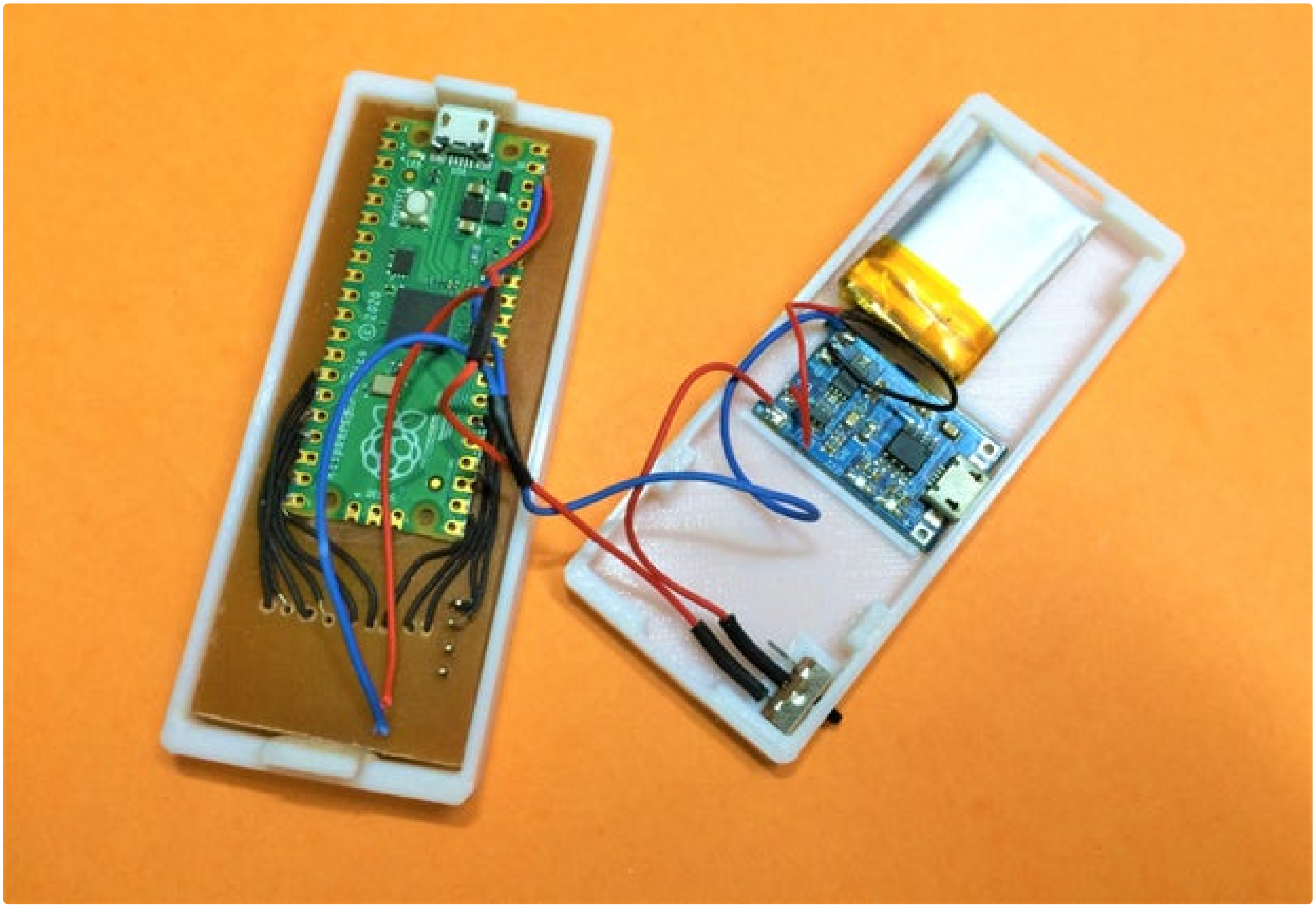
Download

https://www.instructables.com/ORIG/FM1/DGK2/KMALLONL/FM1DGK2KMALLONL.pdf

Step 12: Assembly

This animation is for a reference purpose to show how assembly will proceed.



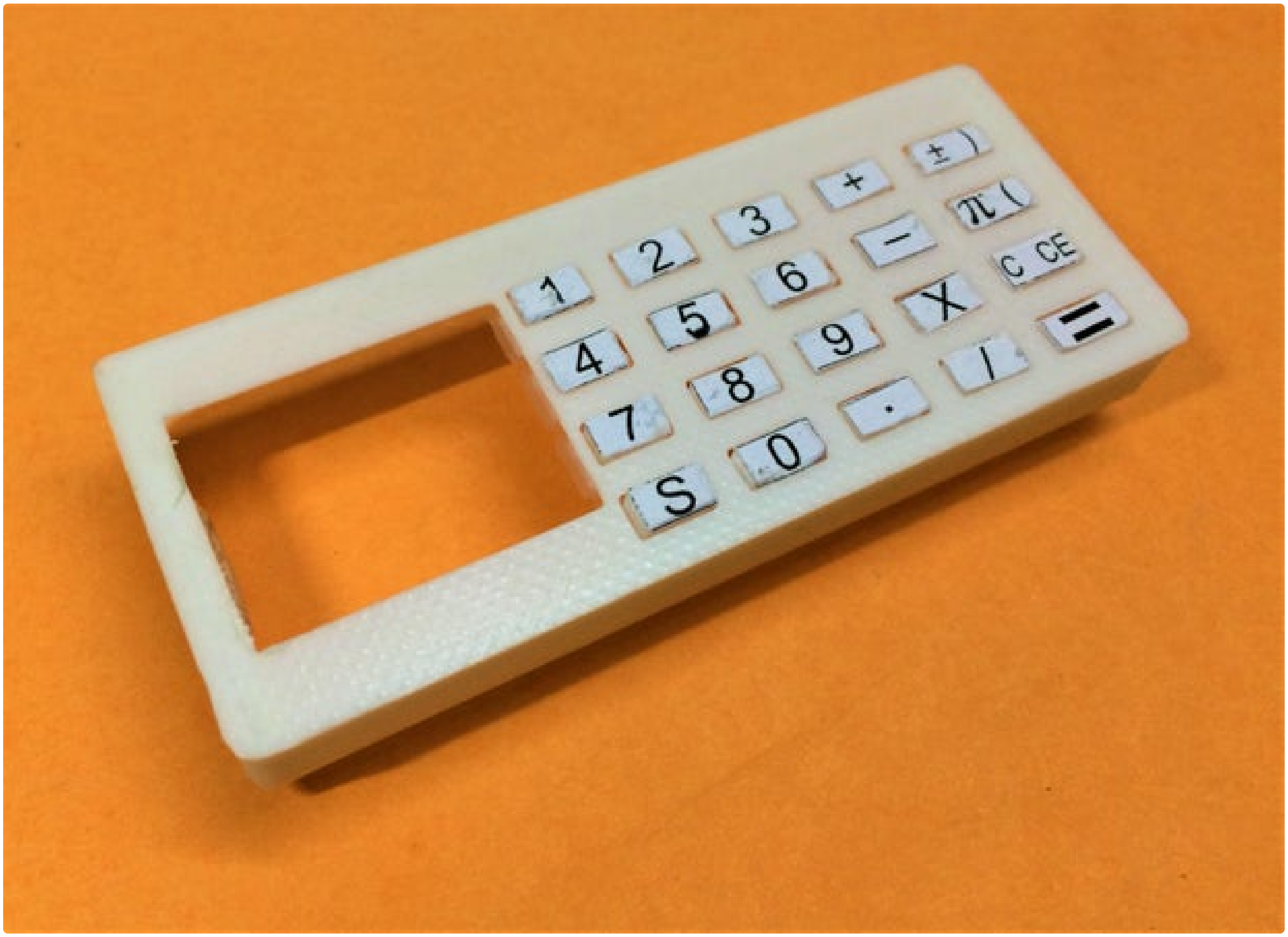


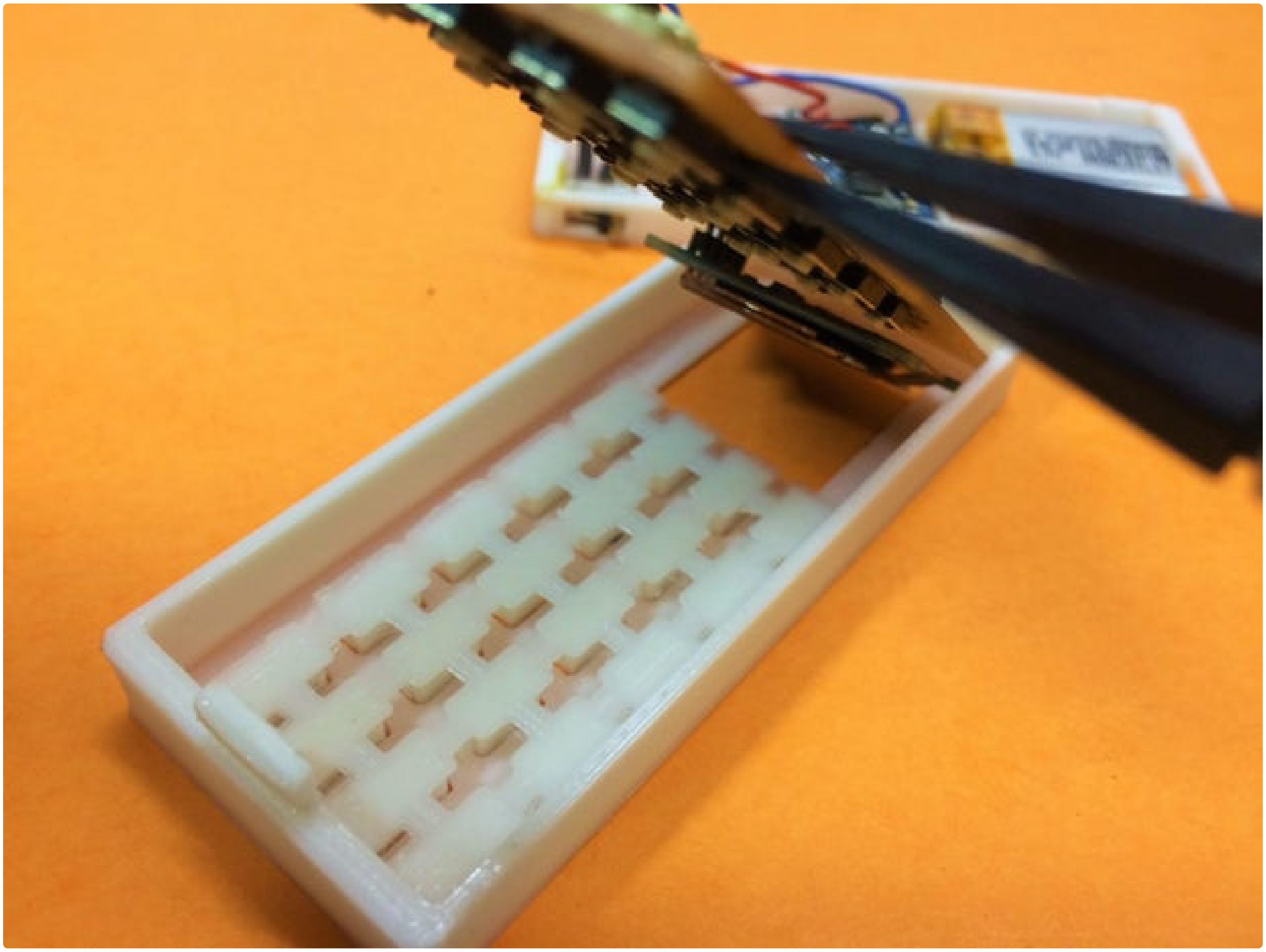
Step 13: Enclosing Into Single Unit

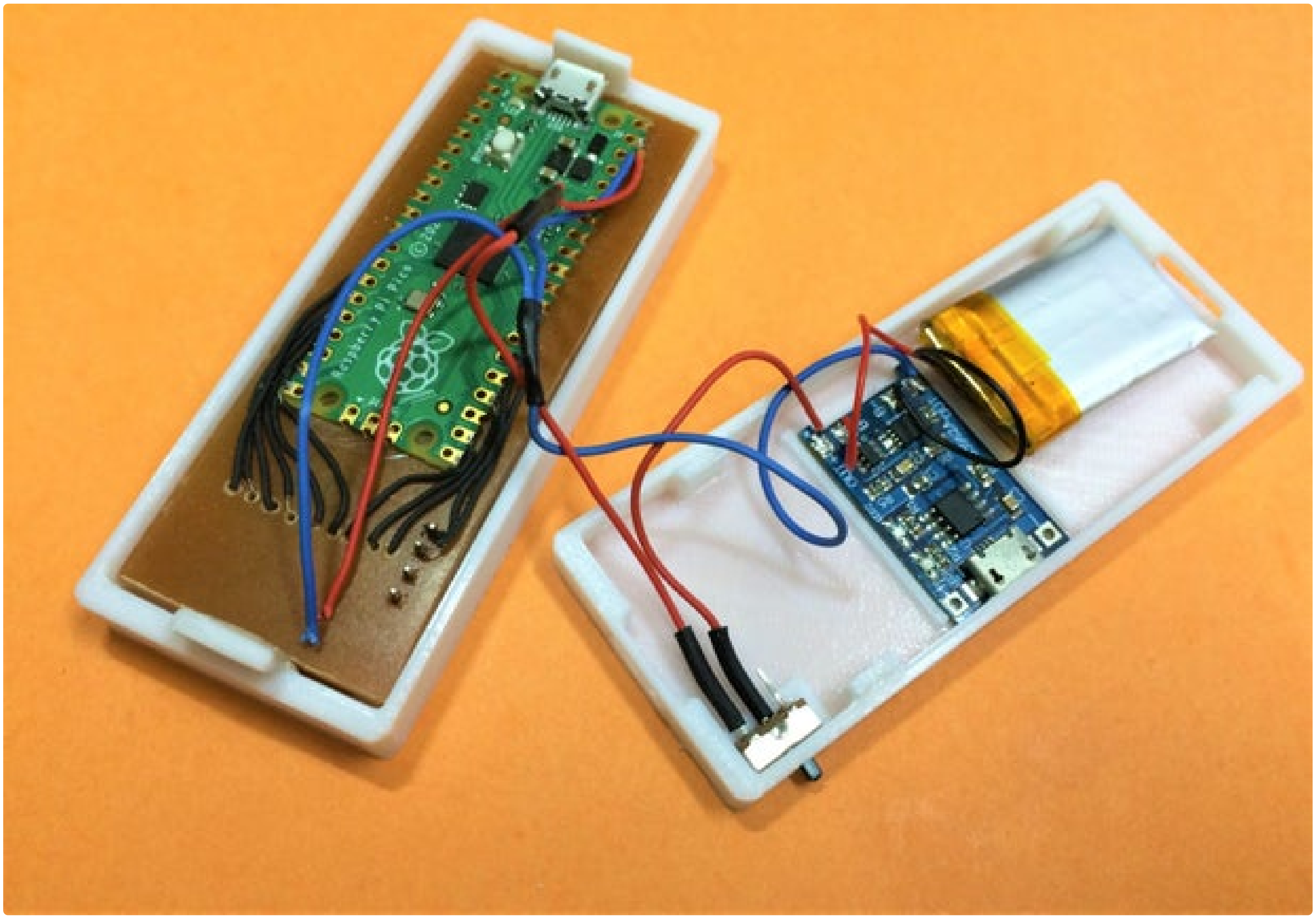
There are indents for switch and TP4056 (charge/protection circuit) and slots for USB input for charging the on/of the slide switch.
Stick then using a glue gun and stick the battery using double-sided tape as using hot glue on a lipo battery isn't recommended.

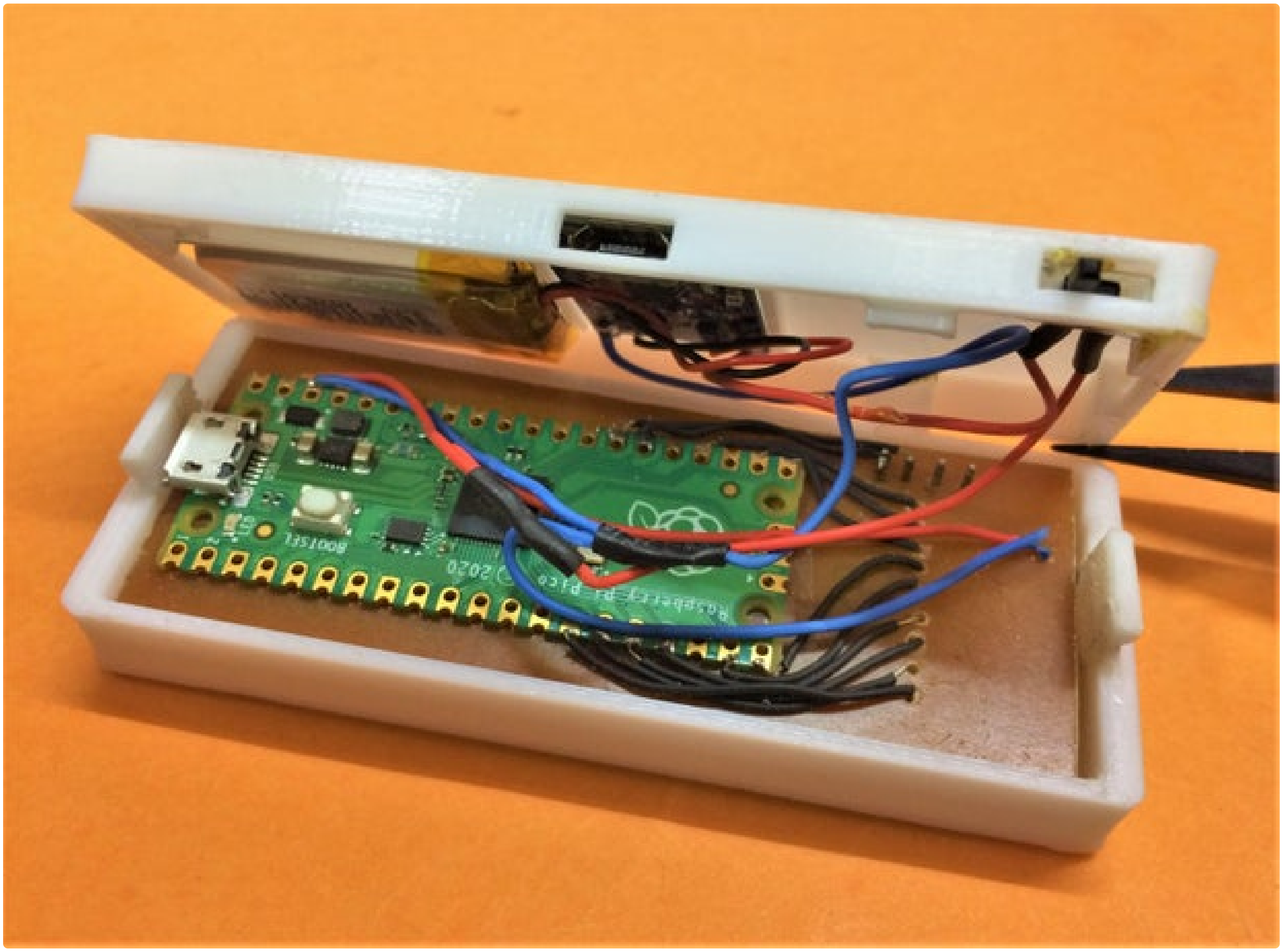
For the top part first, insert the keypad into the grid and the PCB on top of it.

The enclosure is designed for a snap fit.









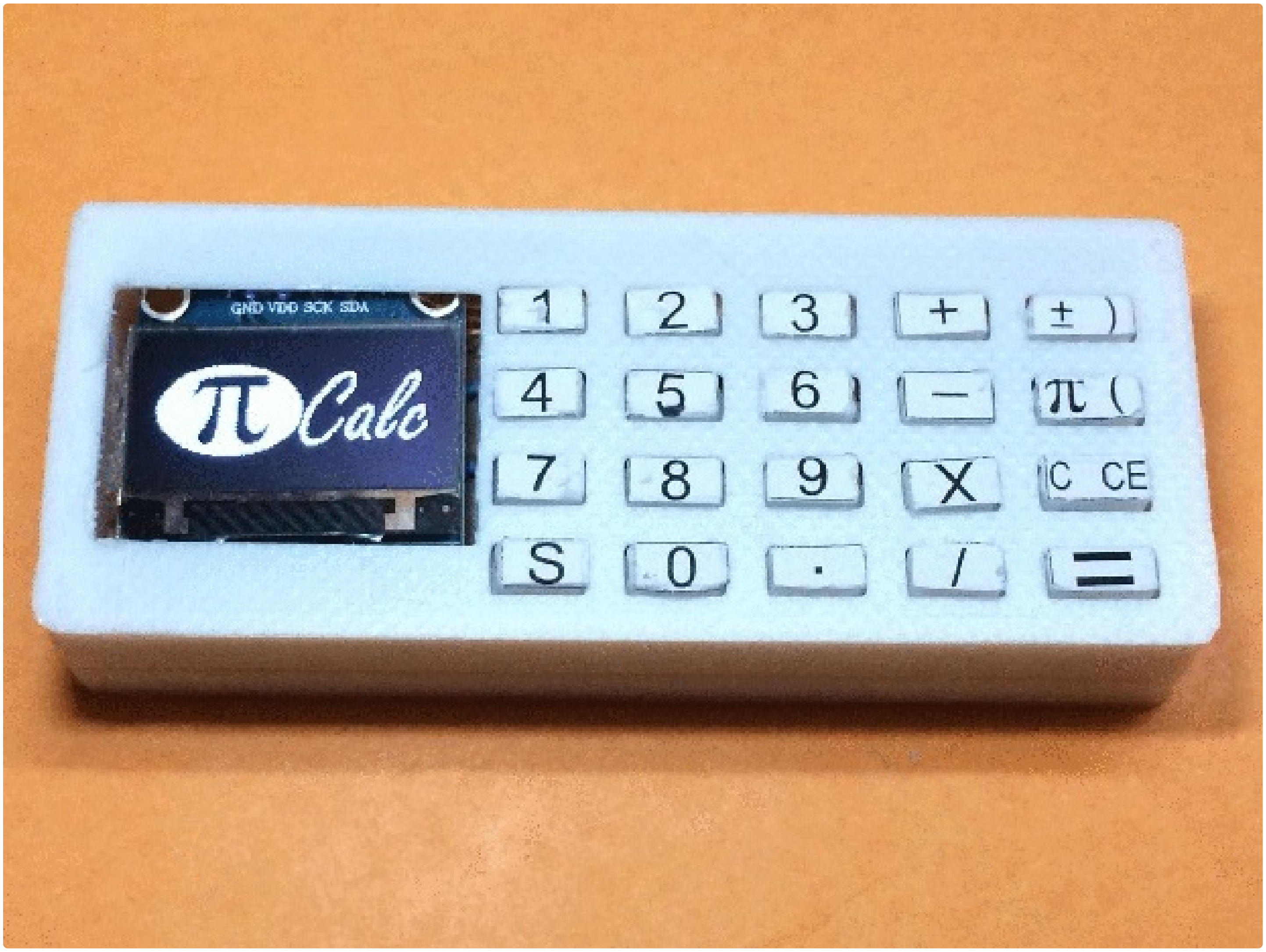
Step 14: Features of PiCalc

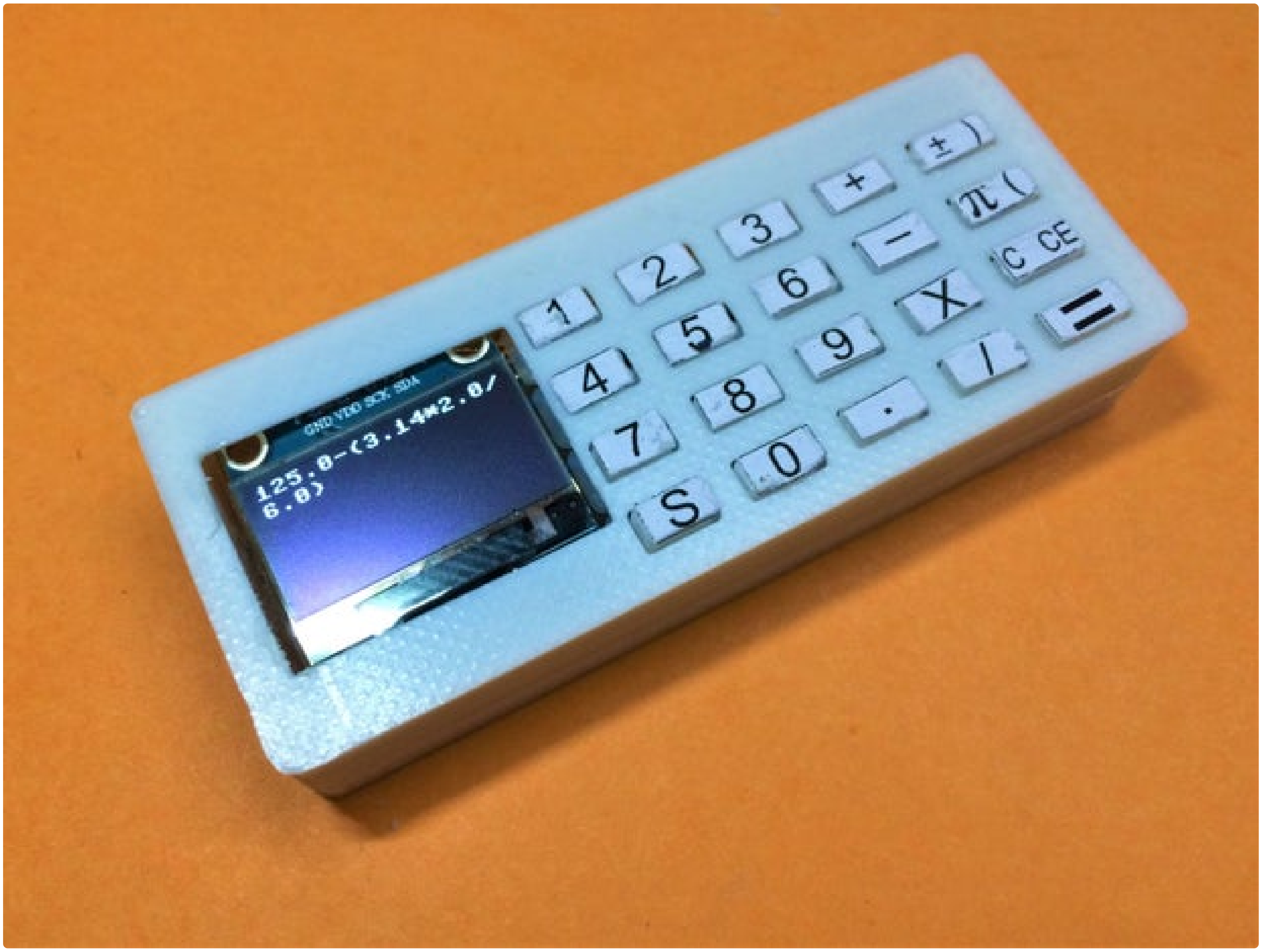
The PiCalc is quite versatile and can be used with long equations with decimal numbers and brackets to define execution priority. By default, the priority of execution is similar to that of the BODMAS rule.

The right side button consists of dual usage with a shift key on the bottom left corner.
For example, to use the open bracket '(' first, click the 'S' button on the bottom left corner, then click the second from the last column's top.
The left symbol on the last column can be used directly without any shift key and to use the right, one shift key is to be clicked before.

Similarly 'C' is for clearing a single element at a time, and 'CE' is to clear all history.

If an equation is written that can't be processed, 'error' will be shown on the screen and to proceed further you need to click the clear button.


GND VDD SCK SDA
π Calc
1
2
3
+
±)
4
5
6
−
π (
7
8
9
X
C CE
S
0
·
/
=

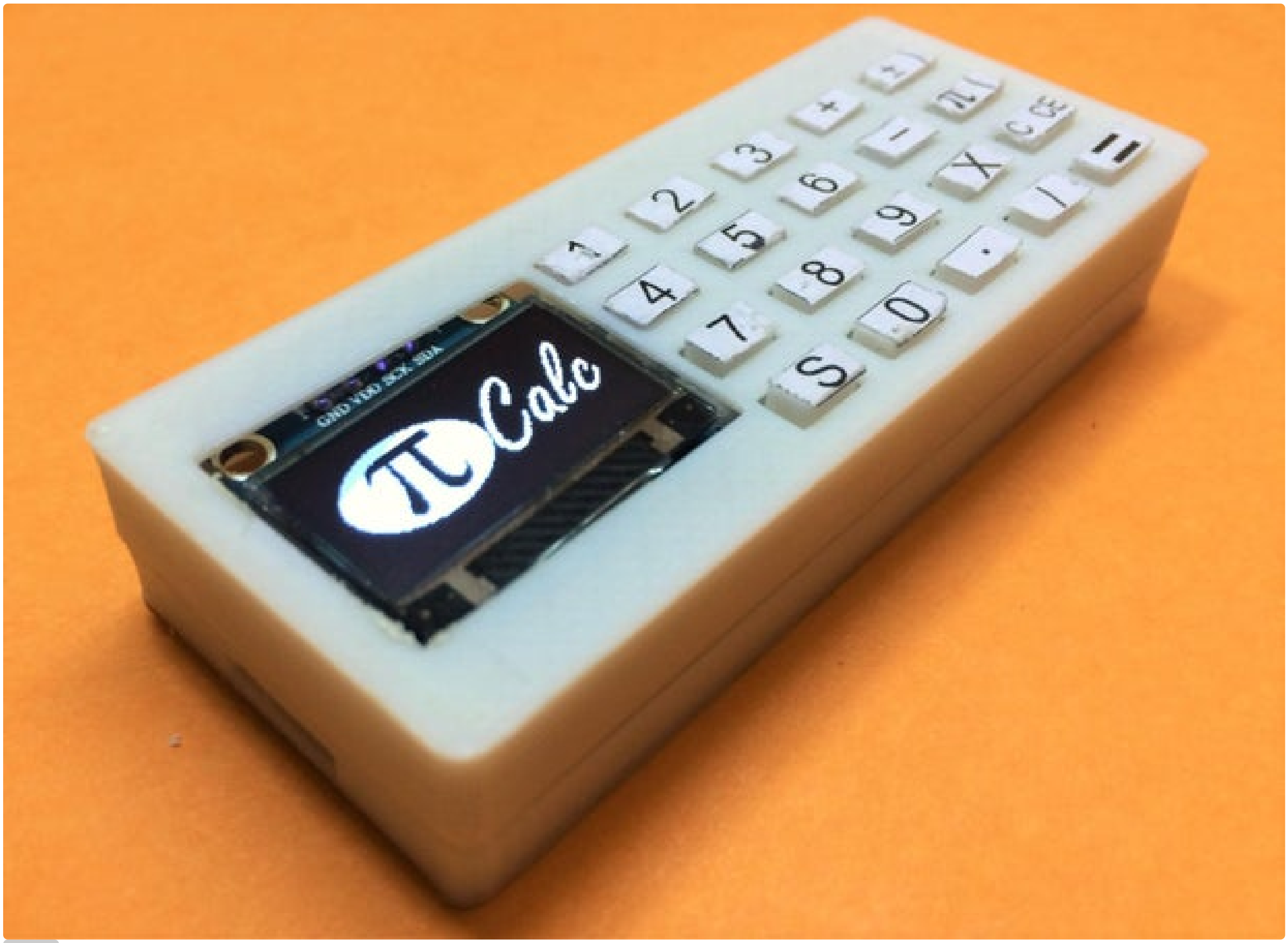

GND VDD SCK SDA
125.0-(3.14*2.0/
6.0)
1
2
3
+
±
)
4
5
6
−
π
(
7
8
9
X
C
CE
S
0
·
/
=

Step 15: Enjoy Your Own Pocket Calculator!

Woohoo! you created your own pocked calculator.

If you face any difficulties or you want to provide any suggestions, do comment below.

πCalc
1
2
3
+
±)
4
5
6
−
π (
7
8
9
X
C CE
S
0
.
/
=



This looks amazing, great idea and great job!!

Nice project and the shift key is a good idea to reduce the number of buttons but.....

Most calculators use the C for "clear (all)" and CE for "clear entry" but you have them reversed ("Similarly 'C' is for clearing a single element at a time, and 'CE' is to clear all history.") I'd switch those around to be more standard.