**\*\*STEPS TO MAKE 16X2 LCD, 4X4 MATRIX KEYPAD & ADAFRUIT THERMAL PRINTER WORK TOGETHER AS A UNIT (EMBEDDED SYSTEM) WITH THE HELP OF RASPBERRY PI WHERE ANYTHING THAT IS BEING INPUT FROM THE KEYPAD GETS DISPLAYED ON THE LCD AND PRINTED OUT OF THE PRINTER IN REAL-TIME. I HAVE ALSO INCLUDED THE GOOGLE FIREBASE SECTION WHEREIN ONE CAN PUSH THE INPUT FROM KEYPAD ON THE FIREBASE ALSO THAT TOO IN REAL-TIME. \*\***

1. Install Raspbian OS on a 16GB micro-SD card(preferred). Complete installation procedures can be found here:

<https://www.raspberrypi.org/documentation/installation/installing-images/>

and

<https://www.raspberrypi.org/downloads/raspbian/>

1. Now to connect matrix keypad to raspberry pi and LCD to your Raspberry PI, follow the link:

<https://www.instructables.com/id/Interface-16x2-Alphanumeric-LCD-And4x4-Matrix-Keyp/>

I have changed the GPIO pins connected to Keypad slightly. The same pins will be used for the rest of setup and are as follows:

COL\_PINS = [17, 15, 14, 25] # BCM numbering

ROW\_PINS = [24,22,27,18] # BCM numbering

I request you to make necessary changes and connect pins accordingly.

I have provided with a different code with changes already made with the name “lcd\_keypad.py”

1. Now, after making all the connections properly, to make sure that keypad and LCD together work fine, run the code “lcd\_keypad.py” from the command line in terminal using “python3 lcd\_keypad.py”. If connections are proper, code should work absolutely fine.
2. Now to add Adafruit printer to above designed system, give the power supply to your printer using VCC and GND pins on the printer. To connect the printer with your raspberry pi board, connect it using a USB and corresponding pins on the printer. For complete setup, follow the link:

<https://learn.adafruit.com/networked-thermal-printer-using-cups-and-raspberry-pi/overview>

1. I have provided with a separate printer test in addition to the tests mentioned in the above link. You can run “printer\_test.py” from the command line in terminal using “python3 printer\_test.py”
2. To club LCD, Keypad and Printer together, I have provided with a code “final.py” wherein instead of keyboard interrupt which was used to stop program in “lcd\_keypad.py” has been replaced by pressing “#” from the keypad. But before all of that, follow the below mentioned steps to execute:
   * 1. In command line type:

nano command.sh

* + 1. In GNU nano terminal , type:

#!/bin/bash

python3 final.py > text.txt &&

cat text.txt >> store.txt &&

echo -e “$(<text.txt)”> /dev/ttyUSB0

* + 1. Save the file using Ctrl + x followed by “y” for yes.
    2. Now create a two different blank text files with name “text.txt” and “store.txt” in your pi where your python scripts are contained. (I prefer keeping all of them in “/home/pi”)
    3. Now from the command line in terminal, run “./command.sh.save” to execute (end by pressing “#” from the keypad).
    4. To repeat the process, run “while true; do ./command.sh.save; sleep 2; done”. This will keep repeating the execution after every 2 seconds of the completion of the process.
    5. The results will be stored in the text file. Open store.txt to see your results. File text.txt shall remain blank because it is used in printing process.

1. To send the data to the google firebase, I have provided with the script “firebasepython.py”. Create a blank text file with name “data.txt” in pi folder. Follow the below mentioned steps to execute:
2. In command line type:

nano command1.sh

1. In GNU nano terminal , type:

#!/bin/bash

python3 firebasepython.py > text.txt &&

cat text.txt > data.txt &&

echo -e “$(<text.txt)”> /dev/ttyUSB0

1. Save the file using Ctrl + x followed by “y” for yes.
2. Now from the command line in terminal, run “./command1.sh.save” to execute (end by pressing “#” from the keypad). Setup your firebase before executing this command. See point vi)
3. To repeat the process, run “while true; do ./command1.sh.save; sleep 2; done”. This will keep repeating the execution after every 2 seconds of the completion of the process. Setup your firebase before executing this command. See point vi)
4. Open data.txt to see your results and you can check it on firebase by opening Console>Database>Realtime Database and under <yourdatabasename> click on “+” and in Name section type “user” and leave value section blank and click on “+” again. Now, add any random name and add any random value, it doesn’t matter what you add.
5. When you implement iv) or v), you’ll get the data on your firebase in real-time.

**CHEERS!!**