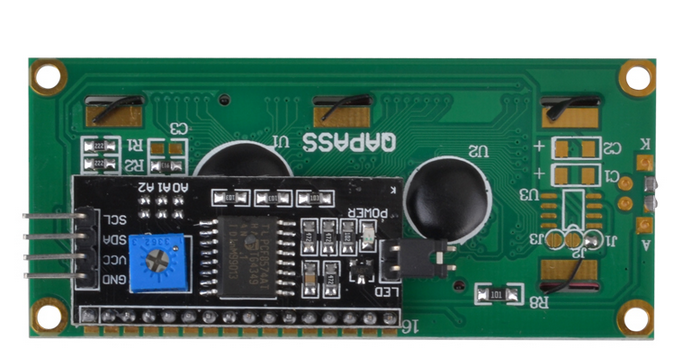
**Drive i2c LCD Screen experiment for RPI**

In previous posts We’ve driven 16×2 LCD screens with the Raspberry Pi. That project was easy to set up, however, it needs to solder many jump wires and occupy lots of GPIO ports which are valuable resources to the Pi.

To save our GPIO port resources, in this project we will connect the Pi with an I2C enabled LCD screen which only has 4 pins as shown below.  


****Caution:**** To make sure this project works properly, you should use a MicroSD card with fresh-installed Raspbian OS.  
****Step 1: Circuit Connection Graph:****

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [IMG_257](http://osoyoo.com/wp-content/uploads/2016/06/PI-LCDI2C_bb.jpg) | ****PiI2C Pin****   |  |  | | --- | --- | | GND(Pin 39 or Pin 9) | GND | | 3V3(Pin 1) | VCC | | SDA(Pin 3/GPIO 2) | SDA | | SCL(Pin 5/GPIO 3) | SCL | |

****Step 2: Enable i2c using raspi-config utility****  
In terminal, type the following command:  
*sudo raspi-config*

Select Advanced Option -> I2C -> Enable I2C ->Finish  
The Pi will reboot after you click the Finish Button

After rebooting the Pi, we need to modify the module’s config file. Type the following command in terminal:  
*sudo nano /etc/modules*  
Add following two lines in modules file if they do not exist:  
*i2c-bcm2708  
i2c-dev*  
Then Type Ctrl X and Yes to save the file.

****Step 3: Step 3 – Install smbus and i2c python library****  
Type following command in terminal:  
*sudo apt-get update  
sudo apt-get install -y python-smbus i2c-tools  
sudo reboot*

After rebooting the system, type the following command in order to check software installation:  
*lsmod | grep i2c\_*  
You should see i2c\_bcm2708 in a list, this means the library has been installed successfully. Otherwise you might need to find another Raspbian OS MicroSD card and repeat Step 2 and 3.

****Step 4:Testing Hardware****  
Depending on your Raspberry Pi version, please run****one of****following commands in terminal:

*sudo i2cdetect -y 1*or *sudo i2cdetect -y 0*  
You should see something as follows:  
0 1  2  3  4  5  6  7  8  9  a  b  c  d  e  f  
00:             — — — — — — — — — — — — –  
10: — — — — — — — — — — — — — — — –  
20: — — — — — — — — — — — — — — — –  
30: — — — — — — — — — — — — — — — 27  
40: — — — — — — — — — — — — — — — –  
50: — — — — — — — — — — — — — — — –  
60: — — — — — — — — — — — — — — — –  
70: — — — — — — — –

If you can only see “– — –” sign in the list ****without**** any numbers, it means either your circuit connection is wrong or your software is not properly installed.

****Step 5: Download Python Code and run the project:****  
in terminal window, type the following commands:

*wget http://kumantech.com/u\_file/1609/file/c59d4bbbe1.txt  
sudo python i2clcda.py*  
If there is any runtime writing error, it might be caused by the LCD I/O address mismatching. try the following commands:

*wget http://kumantech.com/u\_file/1609/file/3f262593ac.txt  
sudo python i2clcdb.py*

****Now your LCD should display the following messages:****

*Created by  
kumantech*

> Tutorial Url:  
> http:/kumantech.com

If your Pi does not show any runtime error but LCD still does not display any messages, you can use a screw driver to adjust the contrast screw on the back of the LCD until you see the message.