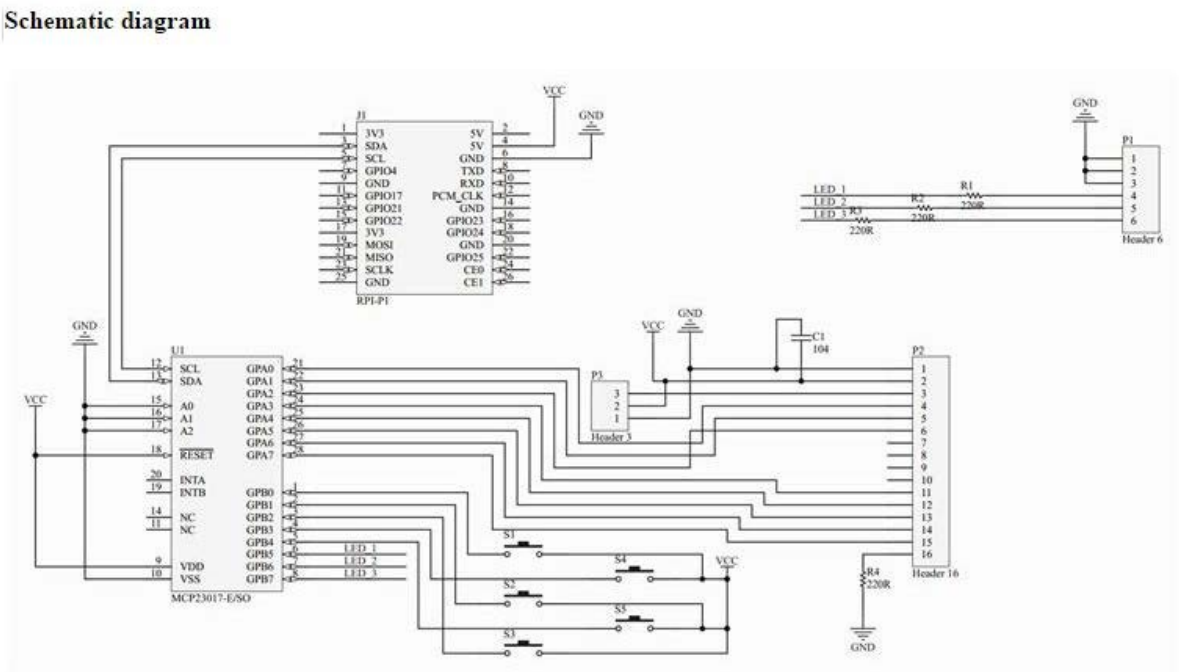
A fix for the missing backlight on Chinese 1602 I2C LCD keypad shield for Raspberry Pi

I bought this LCD/keypad shield off the internet for a very attractive price of less than $10. It plugs right into the GPIO buss of all models of the Raspberry Pi computer and communicates through the I2C bus so it does not tie up a bunch of GPIO pins. It says it also has RGB (red, green, blue) lighting. Several Chinese venders on eBay seem to offer it. When it came it did not come with any software or hardware information. It uses the same chips as the Adafruit RGB LCD Plate so I thought I would try their software. I was in for a surprise when I ran the Python RBG LCD Plate test program which comes bundled in Adafruit's WebIDE real-time python interactive code developer environment. It runs the Pi from a browser on your PC and communicates over your local area network. All the Pi needs is a network connection. It is a free download and includes the complete Adafruit support library for their real-time products. While the 3 colors on the Adafruit shield control the backlight for the LCD display, the 3 colors on the Chinese board simply light a high-intensity LED on the board above the display (the red circle on the picture of the shield below). The second surprise was that **the backlight did not light up** making it almost impossible to read the characters on the screen. I found **a schematic of a similar shield** on the internet and it is shown below.



The MCP23017 chip interfaces the I2C bus to the 1602 LCD module which is piggy-backed on the shield main board with 16 pins. The 23017 takes I2C serial commands and decodes them to control two 8bit I/O ports which connect to the 1602 display. The first 8 bits control character generation on the display. The second 8 bits are used to sense the 5 push buttons as inputs and control the 3 colored LEDs as outputs. That uses all 16 I/O lines and leaves none for the backlight. The backlight in the schematic is controlled by a 220 ohm resistor (R4) from pin 16 on the LED header to ground (lower right corner of the schematic). That means their backlight is on all of the time! I could not find any such resistor on my board. **However, when I jumpered a 330 ohm resistor from pin 16 to pin 1 on the header (Vss is ground) my backlight did turn on**.

I thought that maybe they used one of the GPIO pins to light the backlight under program control. I tried blinking all of the GPIO output pins from 1 to 26 but none of them lit the backlight so that was not it either. It looks like there is no provision for turning on the backlight on this shield which seems weird. If I jumpered the resistor from pin 16 on the 1602 header to pin 12 on the Pi's GPIO bus and set it low the backlight again turns on. Finally, I tried connecting the jumper to one the 3 resistors that limit current to the RBG LED at the top right of the board. Notice that they are labeled GRN, RED, and BLUE. The right side of each resistor is the connection to the 23017 and it goes to 0V to turn on the corresponding LED. Connecting pin 16 through the 330 ohm resistor to that point again caused the backlight to light when that LED was turned on. So, **I could see three ways to turn on the backlight:** 1. have it always on when the Pi is powered up, 2. dedicate a GPIO pin to turn it on or off, or 3. have it light along with one of the LEDs. I chose the simplest method and simply soldered a resistor between pin 1 and pin 16 of the 1602 header as can be seen in the picture above. I used a 330 ohm resistor as the backlight seemed plenty bright with it and it saves a few millamps over the 220 Ohm resistor in the diagram. Finally, notice the small silver potentiometer at the top center of the board. This is the contrast control (P1 in the schematic) and it must be adjusted for the best contrast. Once the backlight is working it is a very nice shield and I have bought several.

I found the shield to be almost **100% compatible with the python subroutines for the Adafruit RGB LCD Plate.** They have functions both to send text to the display and to read the keys. Below is a program I wrote to display the IP address of my Pi2 when it is turned on. I put a sudo python command to run the program in the last line of etc/rc.local file. That causes the Pi to automatically display its IP address on startup. Once I know its IP, I can run the Pi either through VNC or WebIDE from my PC.

```
#!/usr/bin/python
from time import sleep
from Adafruit_CharLCDPlate import Adafruit_CharLCDPlate
import os
    #
    # Initialize the LCD plate. Should auto-detect correct I2C bus. If not,
    # pass '0' for early 256 MB Model B boards or '1' for all later versions
lcd = Adafruit_CharLCDPlate()
    # Clear display and turnoff all 3 LEDs
#Note-the backlight function controls the LEDs, not the backlight
lcd.backlight(lcd.OFF)
    # Pi 2 running Jessie is slow getting an ip address so delay for 20 seconds
sleep (20)
f=os.popen('hostname -I')
lcd.clear()
lcd.message("ip=" + f.read())
sleep(10)
lcd.stop()
```

I hope you found this information useful. If you have any questions or comments you can contact me at jasdowney@gmail.com .