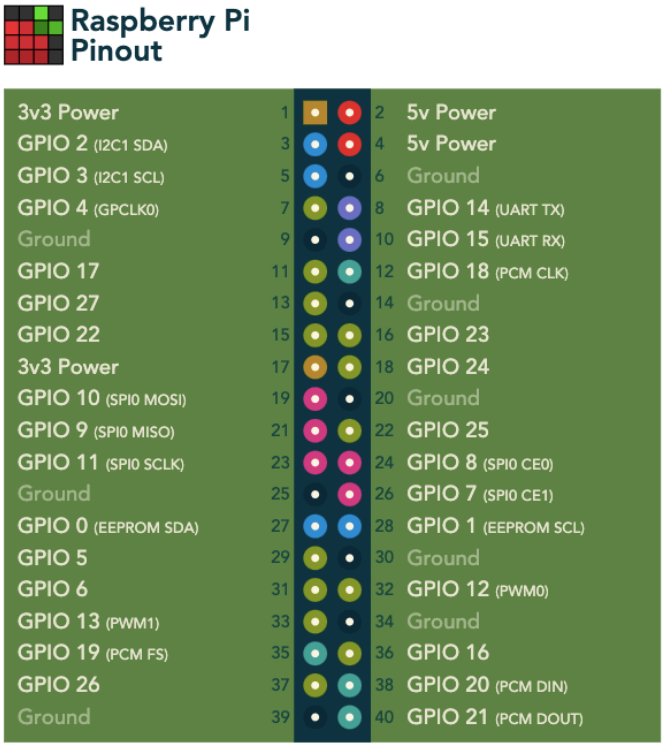
LED SubSystem With Brightness Control using PWM

Team member_1:Mukkavalli Venkata Devi Karthik Raja(2020H1400173P)
Team member_2:V.Sudheer Reddy(2020H1400179P)

Project Summary:

This driver implements the mechanism of controlling the brightness of an LED. Two push buttons are used to change the LED brightness level one for increasing and another for decreasing the brightness.This driver is compiled and tested on Raspberry Pi Model B+ V1.2. and total design is integrated by breadboard.

Hardware Design:



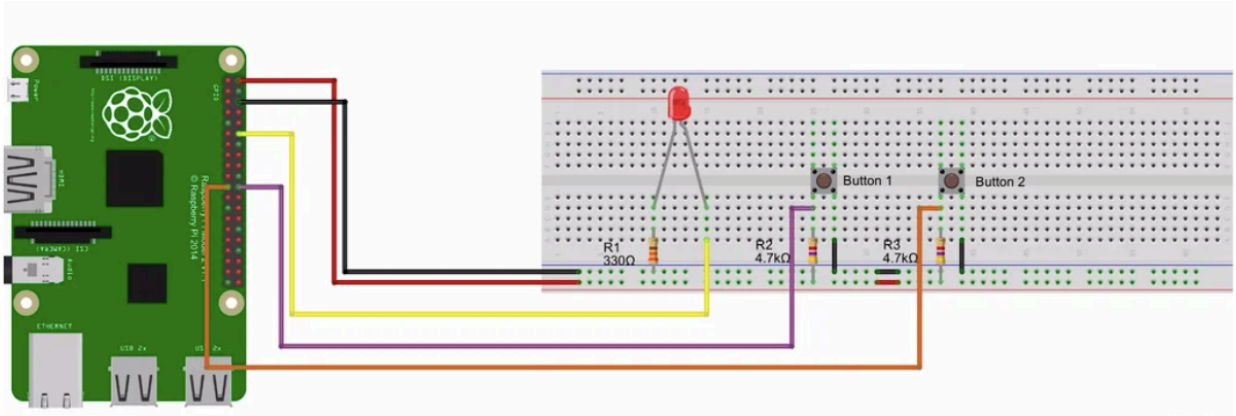
Components Used:

Raspberry Pi Model B+ V1.2
LED
2 push Buttons
Bread Board,Resistors,Jumper wires.

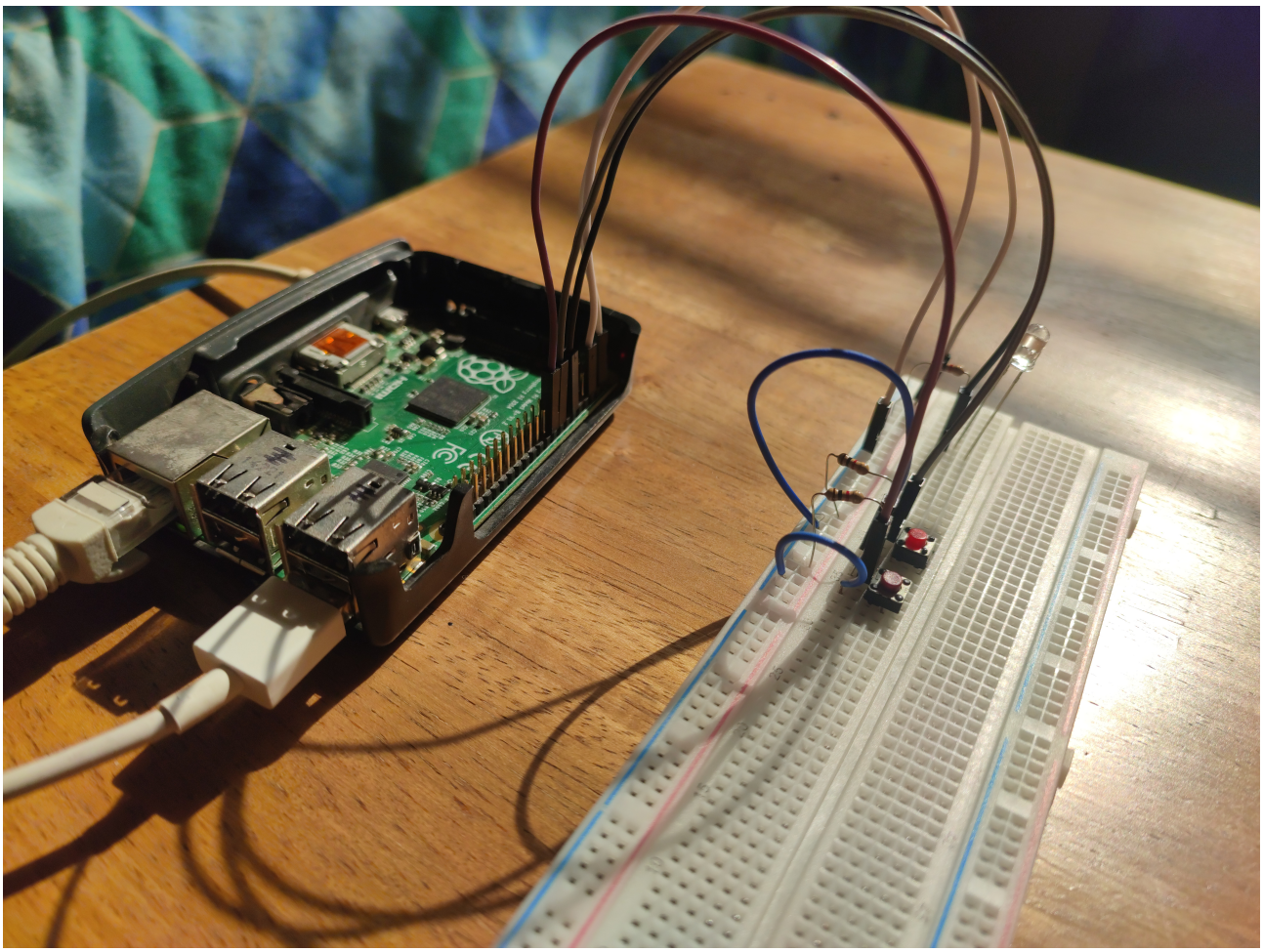
Raspberry Pi Model b+ V1.2 (2014) has BCM2835 SoC. It supports the Ethernet connection but no Wireless Support.

Ethernet port of Raspberry pi is connected to laptop ,the hardware is connected to Intenet by USB tethering hotspot. Given the power supply to the board and bread board connections were done according to the schematic shown in the below figure.

Schematic:



Actual Photograph:



Build Process:

- Complie using make command in the file directory: *make all*
- The driver should be loaded using the following command (as root):

Sudo insmod leds.ko [down_button_gpio=23] [up_button_gpio=24] [led_gpio=18] [pulse_frequency=100000] [led_max_level=5](as per the kernel driver mentioned)

After inserting the module ,generate IRQs by pressing push buttons externally and observe the brightness of led respectively.

- Use *dmesg* command to view at which brightness the led is blinking .
- The module can be unloaded using this command : *rmmod leds*

Implementation Details:

down_button_gpio, up_button_gpio and led_gpio represent the GPIOs where the components are connected.

pulse_frequency represents the amount of time (in nanoseconds) for which the proportion of LOW and HIGH signals sent to the LED is calculated. E.g. with pulse width of 100 ms and requested LED brightness of 40%, 40 ms will be spent sending HIGH signal and 60 ms will be spent sending a LOW signal to the LED.Default value is 100 000 nanoseconds (0.1 ms).

led_max_level determines the number of brightness levels the driver will support. E.g., with maximum level 2 there will be 3 distinct brightness levels - 0%, 50% and 100%.with maximum level 5 (meaning a step of 20% i.e., 0%,20%,40%,60%,80%,100%.

Interrupt Handler and Finite-State Machine

When one of the push-buttons is pressed, an interrupt handler processes the received IRQ and sets the proper FSM event (UP or DOWN) depending on which button was pressed. A work is scheduled to handle the actual LED level change.

In the work queued by the IRQ handler, the appropriate FSM function is called, then the FSM state is updated. The FSM function increases or decreases the current LED level or does nothing. The LED level is later translated to a brightness level (proportion of LOW and HIGH signals) in a separate work function.