eYSIP2016

Raspberry Pi Hardware Development & Tutorials

Aditya Kumar
Pritish Salunke
Rutuja Ekatpure
Deepa Avudiappan
Duration of Internship: 21/05/2016 – 10/07/2016

Raspberry Pi Hardware Development & Tutorials

Abstract

Interfacing LED, switch, LCD and ICs on Raspberry Pi. Create a module of each device interfaced. Communication between Raspberry Pi and other device using Zigbee and Bluetooth module.Communication between Raspberry Pi and Firebird V using SPI, I2C and UART protocol. Create documentation and video tutorial explaining individual module. Design PCB for the devices interfaced with Raspberry Pi.

Completion status

- **Task 1:**
  Developed different modules for Raspberry Pi
  E.g. PWM Driver IC PCA9685, ADC, Port Expander etc.

- **Task 2:**
  Communication between Rpi and other
  device through Xbee and Bluetooth Module

- **Task 3:**
  Interfacing LCD with Raspberry Pi

- **Task 4:**
  Communication between Raspberry Pi and
  Firebird V using UART protocol

- **Task 5:**
  Communication between Raspberry Pi and Arduino UNO using I2C and SPI protocol

- **Task 6:**
  Designed PCBs for

  1. LCD connected with port expander IC MCP23017 IC
  2. LM35 temperature sensor and Sharp IR sensor with ADC IC MCP3008
  3. PWM driver IC PCA9685 to drive DC motor and Servo motor
  4. Xbee and Bluetooth module

1.1 Hardware parts

- List of hardware

  1. Raspberry Pi
     Download link Vendor Link
  2. FireBird V Robot
     Download link Vendor Link
  3. LED
  4. Resistor
  5. Switch
  6. LCD
     Download link Vendor Link
  7. MCP23017 IC
     Download link Vendor Link
  8. MCP3008 IC
     Download link Vendor Link
  9. Sharp Sensor
     Download link Vendor Link
  10. LM35 Temperature Sensor
      Download link Vendor Link

11. PCA9685 PWM Driver IC
    Download link Vendor Link
12. L293D Motor Driver IC
    Download link Vendor Link
13. Capacitor
14. DC Motor
15. Servo Motor
16. 9 V battery
17. Xbee Module
    Download link Vendor Link
18. Bluetooth Module
    Download link Vendor Link
19. Arduino UNO
    Download link Vendor Link

1.2 Software used

- List of software used

1. MobaXterm Personal Edition
   Download link
2. Raspbian Jessie
   Download Link
3. Atmel Studio 6.0
   Download Link
4. Eagle Version 7.6.0
   Download Link
5. NEX ISP USB STK500V2 Programmer
   Download Link
6. XCTU
   Download Link
7. Arduino 1.6.9
   Download Link
8. Camtasia Studio 8
   Download Link

1.3 Accessing GPIO pins of Raspberry Pi

In this experiment we have interfaced an led and switch. Resistor is connected with led to limit the current.

Problem Statement

When switch is pressed led will turn on. Again when switch is pressed led will turn off.

Circuit Diagram

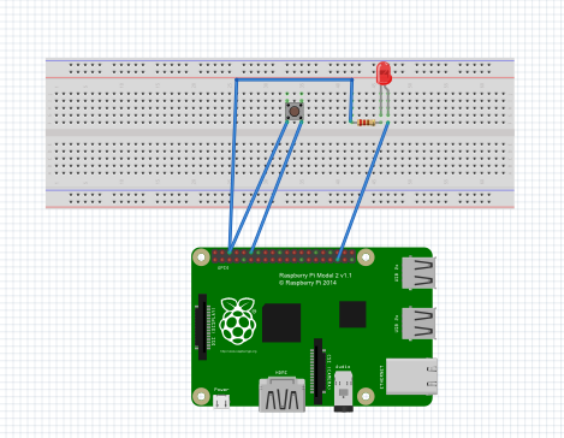


Figure 1.1:

Assembly of hardware

Step 1

One pin of the push button is connected to Ground(Pin 9).

Step 2

The other pin of the push button is connected to IC pin no. 12.

Step 3

The anode of led is connected to IC pin 35 of raspberry pi.

Step 4

The cathode of led is connected to the the resistor of 300 ohms which is then connected to the ground.

1.4 Enabling I2C interface in Raspberry Pi

I2C stands for Inter Integrated Circuit. It is a communication protocol in which many devices are connected with two signal line SDA(Serial Data) and Serial Clock(SCL). Since I2c interface is disabled by default. So, in following steps we have explained how to enable I2C interface on RPi.

Steps to enable I2C interface on RPi

1. Open MobaXterm.
2. Establish SSH connection to R-Pi.
3. Under Advanced SSH settings.

   If in remote environment you have chosen Interactive shell.

   If in remote environment you have chosen LXDE desktop then open LXTerminal.
4. Type *sudo raspi-config*. This will launch the raspi-config utility.

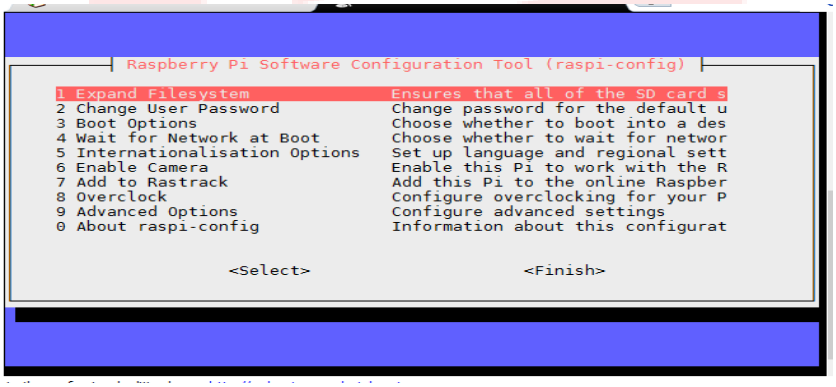


Figure 1.2: [4]

5. Select the *Advanced options*
6. Then select *option A7 I2C*
7. It will ask to enable the ARM I2C interface, click *YES*.

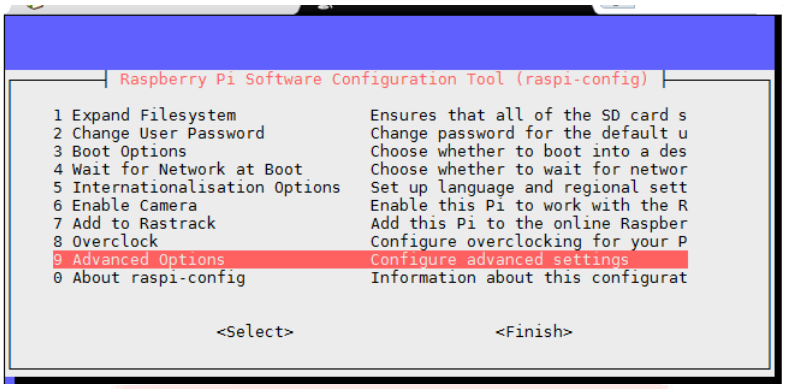


Figure 1.3: [4]

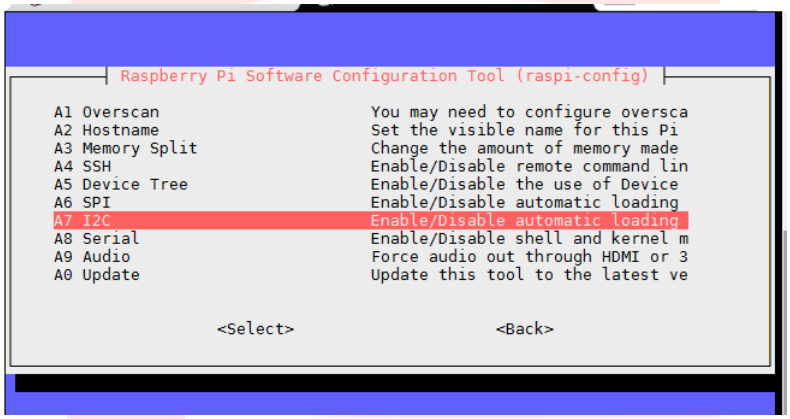


Figure 1.4: [4]

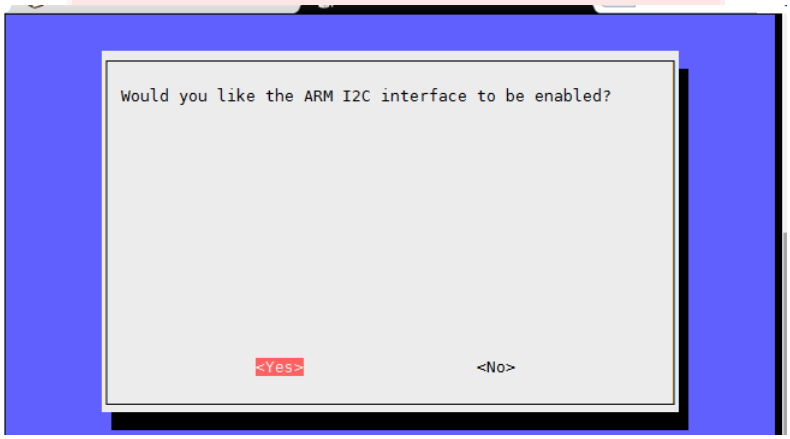


Figure 1.5: [4]

8. Then it will ask if you would like I2C kernel module to be uploaded by default. Select *YES*.

Figure 1.6: [4]

9. I2C kernel module will now be loaded by default.Click *OK*

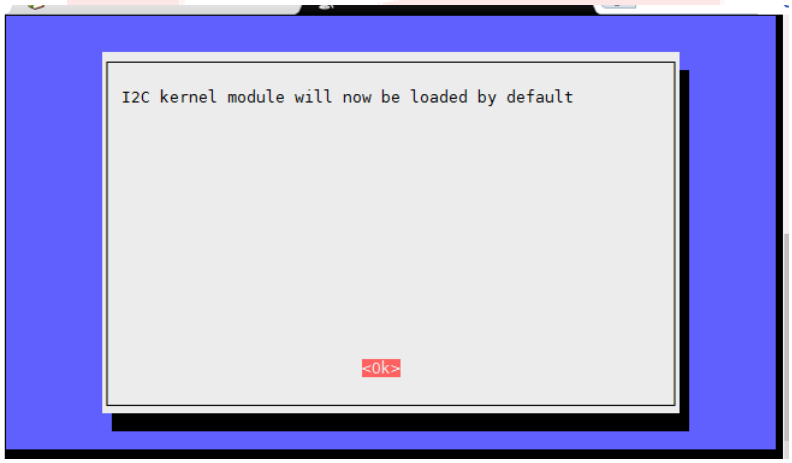


Figure 1.7: [4]

10. Select Finish to return to command line.

11. Next we need to edit the modules file using :
sudo nano /etc/modules

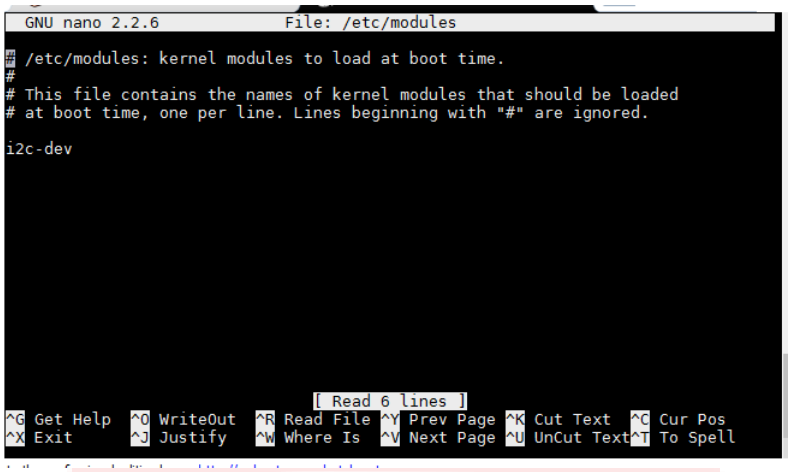


Figure 1.8: [4]

12. Add the following two lines :
i2c-bcm2708
i2c-dev

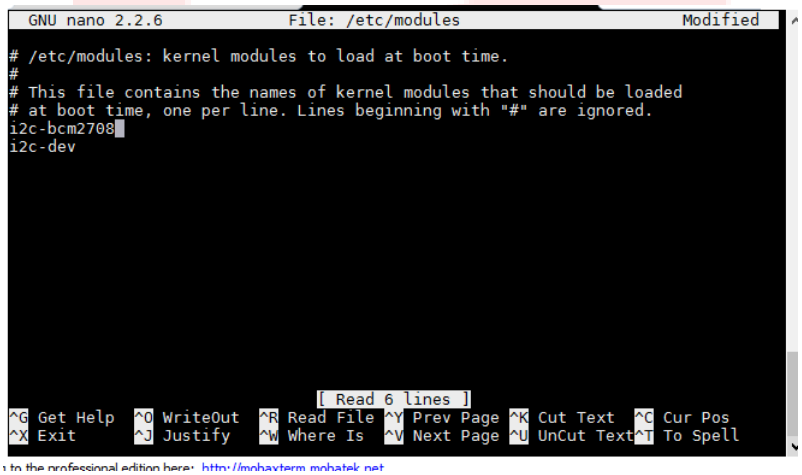


Figure 1.9: [4]

13. Use CTRL-X, then Y, then RETURN to save the file and exit.

14. To help debugging and allow the i2c interface to be used within Python we can install python-smbus and i2c-tools :
sudo apt-get update
sudo apt-get install -y python-smbus i2c-tools

15. Shutdown your Pi using :
    sudo halt
    Wait ten seconds, disconnect the power to your Pi and you are now ready to connect your I2C hardware.

16. When you power up or reboot your Pi you can check the i2c module is running by using the following command :
    *lsmod — grep i2c_*
    That will list all the modules starting with i2c_. If it lists i2c_bcm2708 then the module is running correctly.

17. Once you have connected your hardware double check the wiring. Make sure 3.3V is going to the correct pins and you have got not short circuits. Power up the Pi and wait for it to boot.

18. Type the command:
    sudo i2cdetect -y 1

19. You should the output as:

```
pi@raspberrypi ~ $ sudo i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: 20 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

Figure 1.10: [4]

1.5 Interfacing Port Expander MCP23017 IC

In this experiment we have interfaced port expander MCP23017 IC with RPi. Port expander is used to increase GPIO pins of RPi.
Problem Statement Interface an LCD with port expander MCP23017 IC and display some message.
Circuit Diagram

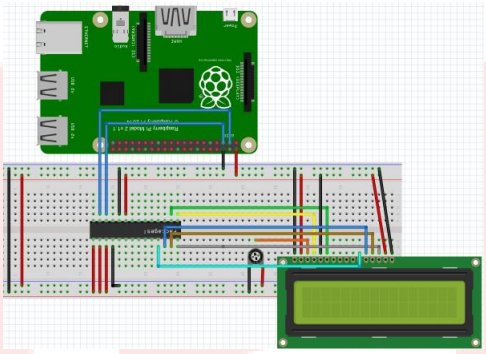


Figure 1.11: LCD connected to MCP23017 IC

Schematic Diagram

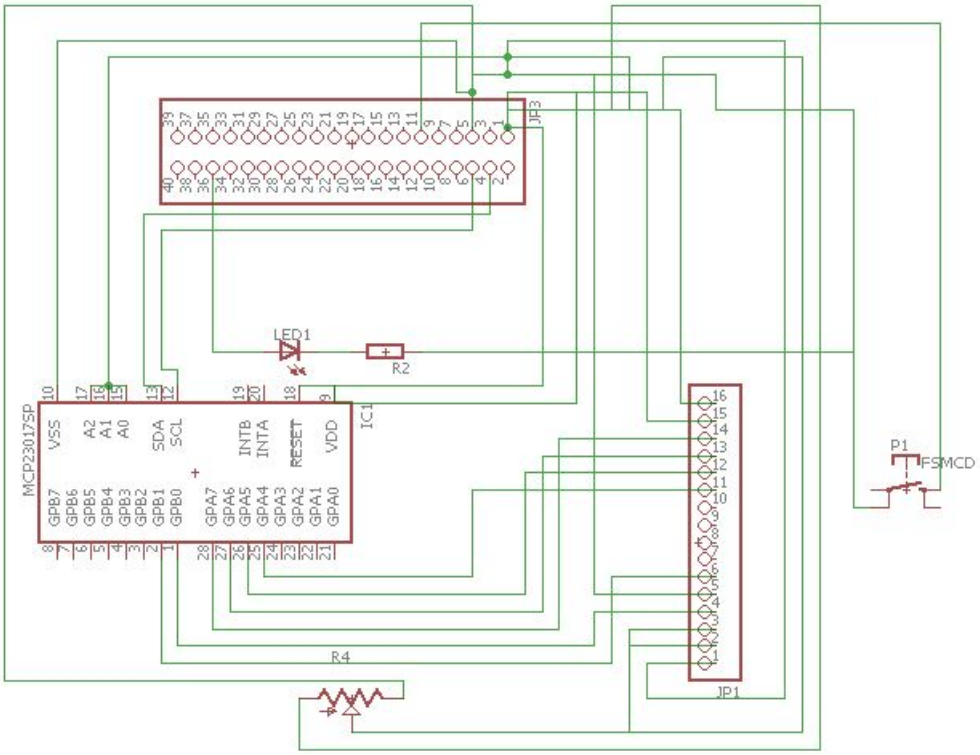


Figure 1.12: Eagle Schematic

PCB Layout

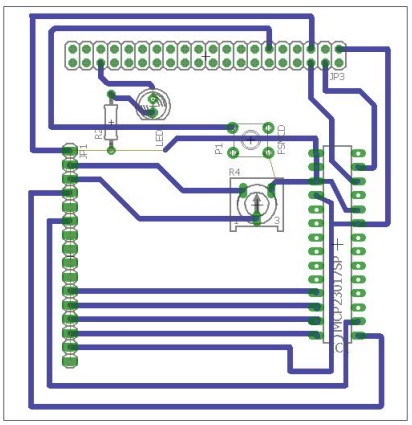


Figure 1.13: PCB Layout

Assembly of hardware

Step 1

Pin 9 (VDD) is connected to 5V

Step 2

Pin 10 (VSS) is connected to Ground

Step 3

Pin 12 (SCL) is connected to Pin 5 on the Pi GPIO

Step 4

Pin 13 (SDA) is connected to Pin 3 on the Pi GPIO

Step 5

Pin 18 (Reset) should be set high for normal operation so we connect this to 5V

Step 6

Pins 15, 16 & 17 (A0-A2) determine the number assigned to this device. We are only using one device so we will give it a binary zero by setting all three of these pins to 0 (ground)

Step 7

RS and Enable pins of the LCD are connected to GPB0 and GPB1 respectively.

Step 8

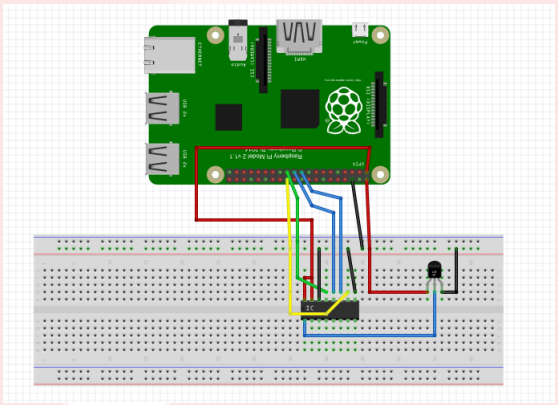
R/W pin is Grounded

Step 9

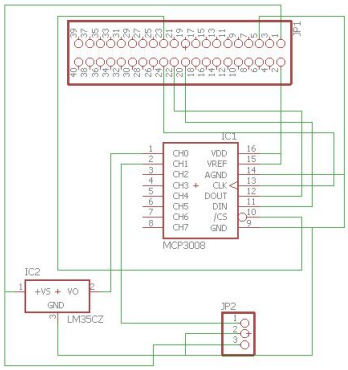
Data pins D7,D6,D5 and D4 are connected to GPA7,GPA6,GPA5 and GPA4 respectively.

1.6 Interfacing ADC IC MCP3008

Description:
Raspberry Pi does not have internal ADC. So,to read the values of analog sensors we need to provide external ADC. Here, we will be using MCP3008 IC. The MCP3008 is a successive approximation 10bit 8-channel Analogue-to-digital converter (ADC). It is cheap, easy to connect and doesnt require any additional components.It uses the SPI bus protocol which is supported by the Pis GPIO header.
Problem Statement: Read the values of LM35 temperature sensor by using MCP3008 IC(ADC) and calibrate it by programming in python.
Circuit Diagram



Schematic Diagram



PCB Layout

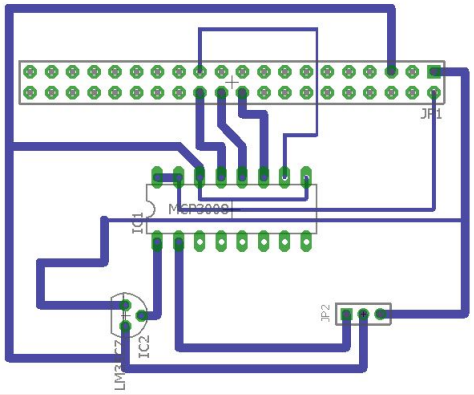


Fig. shows PCB Layout

1.7 Interfacing DC Motor

Description:
Normal DC gear-head motors require current greater than 250mA. Most of the ICs like 555 timer,74 series ICs cannot supply this amount of current.Instead if we directly connect motors to the output of any of the above IC's, they might get damaged. There is a need of a circuitry that can act as a bridge between the above mentioned ICs and the motors. This is where a motor driver plays a crucial role. It regulates the current owing through the circuit hence preventing any damage to the device. L293D is dual H-bridge motor driver ICs. Using these we can control the rotation of two motors in both clockwise and anti-clockwise direction.

Problem Statement:
Controlling the speed of DC motor by using Port expander IC MCP23017 and motor driver IC L293D.

Circuit Diagram

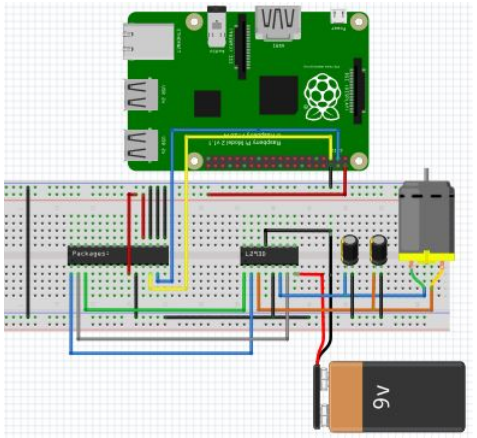


Fig. shows the conncection diagram.

Assembly of hardware

Step 1

Pin 9 (VDD) is connected to 5V

Step 2

Pin 10 (VSS) is connected to Ground

Step 3

Pin 12 (SCL) is connected to Pin 5 on the Pi GPIO

Step 4

Pin 13 (SDA) is connected to Pin 3 on the Pi GPIO

Step 5

Pin 18 (Reset) should be set high for normal operation so we connect this to 5V

Step 6

Pins 15, 16 & 17 (A0-A2) determine the number assigned to this device. We are only using one device so we will give it a binary zero by setting all three of these pins to 0 (ground)

Step 7

Input 1 and Input 2 of L293D is connected to GPB0 and GPB1 of MCP23017

Step 8

Pin 1 (enable pin) of L293D is connected to GPB2.

Step 9

Out 1 and Out 2 are connected to a DC motor.

1.8 Interfacing Servo motor using PWM Driver IC PCA9685

Description:
Raspberry Pi has only one pin for PWM generation, which is pin number 12. So, if we want to control the speed of more than one motor by changing its PWM then we need PCA9685 motor driver IC. PCA9685 is an I2C-bus controlled 16-channel PWM Driver IC.Each PWM channel output has its own 12-bit resolution (4096 steps) fixed frequency individual PWM controller that operates at a programmable frequency from a typical of 24 Hz to 1526 Hz with a duty cycle that is adjustable from 0to be set to a specific PWM value. All outputs are set to the same PWM frequency.
Problem Statement:
Controlling the angle or duty ratio of servo motor using PCA9685 PWM Driver IC.

Circuit Diagram

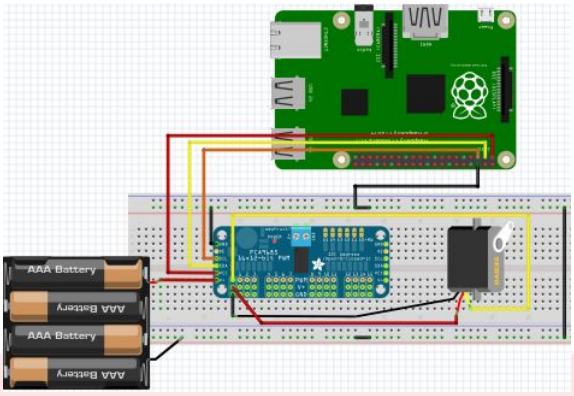


Fig. above shows connections.

Assembly of hardware

Step 1

Control pin of servo motor is connected to S pin of channel 0.

Step 2

Ground and Vcc of servo motor is connected to ground and Vcc of channel 0 of PCA9685 respectively.

Step 3

Pin 12 (SCL) is connected to Pin 5 on the RPi GPIO

Step 4

Pin 13 (SDA) is connected to Pin 3 on the RPi GPIO

Step 5

Vcc of IC is connected to pin 2 on the RPi.

Step 6

Ground pin of IC is connected to pin 6 of RPi.

Step 7

V+ of IC is connected positive terminal of 9V battery.

1.9 Serial Communication using Xbee and Bluetooth Module

Circuit Diagram of Bluetooth and Raspberry Pi

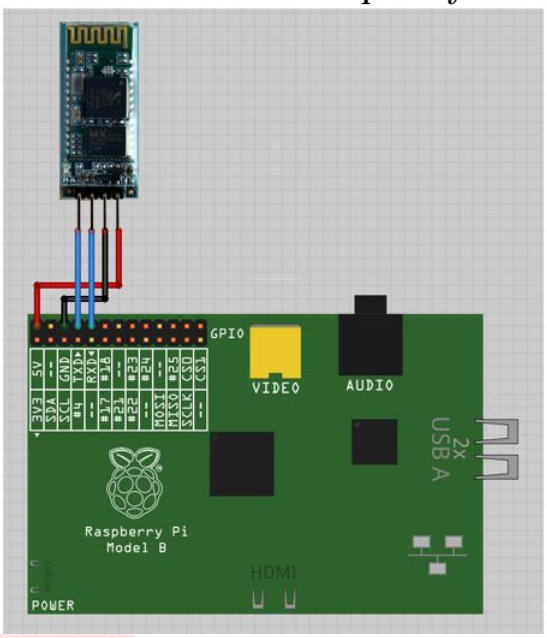


Fig. shows Connection Diagram.

Assembly of harware

Step 1

Pin 2 of RPi is connected to Vcc.

Step 2

Pin 6 of Rpi is connected to ground.

Step 3

Pin 8(TXD) of RPi is connected to RXD pin.

Step 4

Pin 10 (RXD) is connected to TXD pin of bluetooth module.

Connection Diagram of Xbee and Raspberry Pi

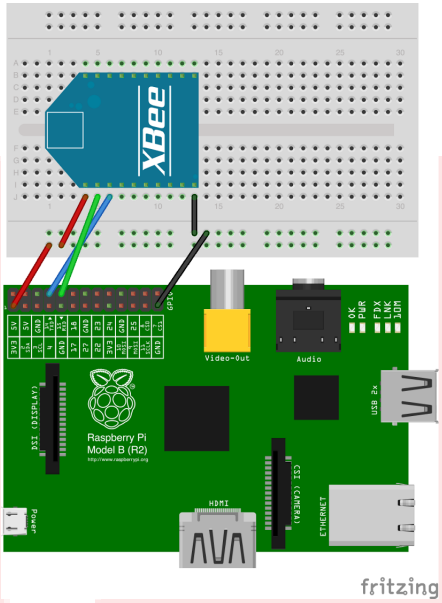


Fig. shows connection diagram.

Assembly of harware

Step 1

Connect Vcc to pin 2 (5V) of Rpi.

Step 2

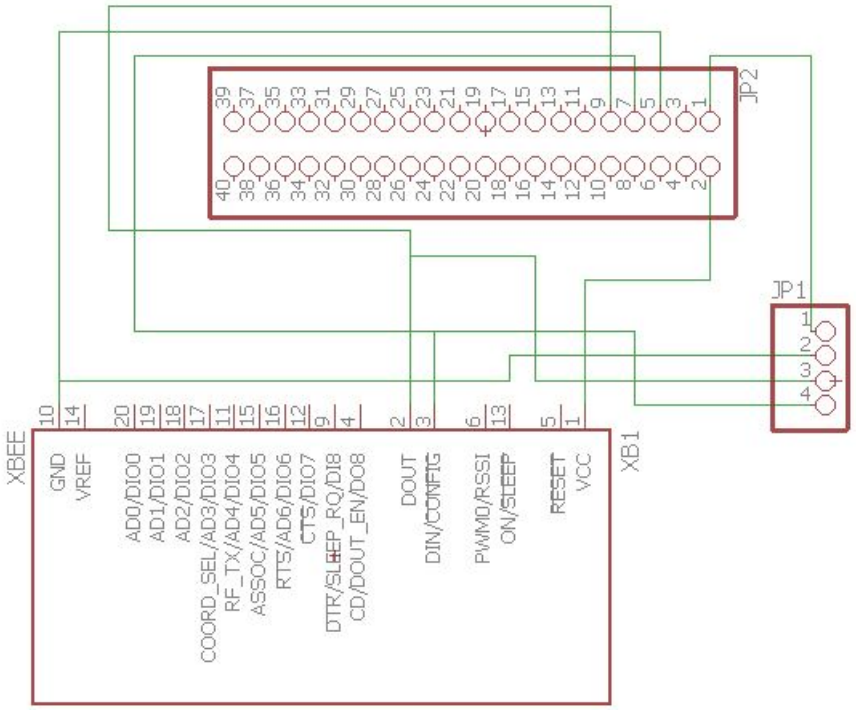
Ground is connected to pin 6 (GND) of RPi.

Step 3

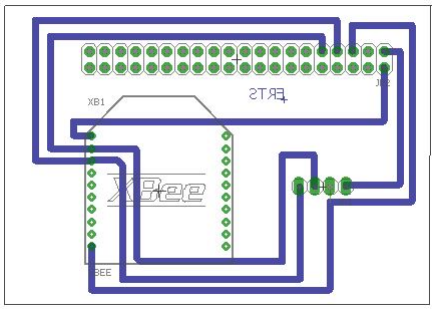
Pin TXD of Xbee is connected to pin 10 (RXD) of RPi.

Step 4

Pin RXD of Xbee is connected to pin 8 (TXD) of RPi. **Schematic Diagram of Bluetooth and Xbee Module**



PCB Layout



Above figure shows our PCB Layout.

1.10 Interrupt on Raspberry Pi

External Interrupt on RPi

Circuit Diagram

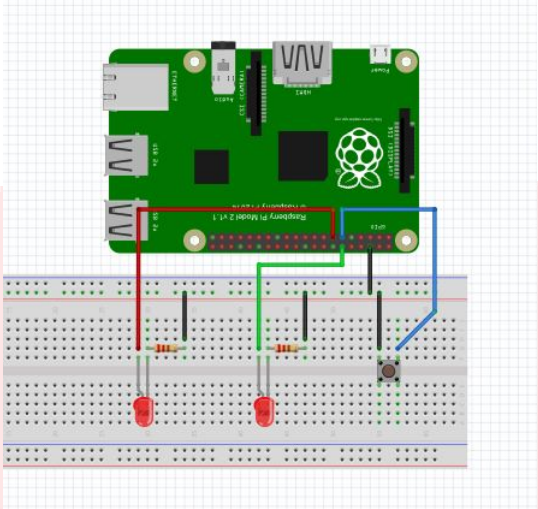


Fig. shows Connection Diagram.

Assembly of harware

Step 1

Pin 11 is connected to switch and other terminal of switch is grounded.

Step 2

Pin 12 of Rpi is connected to anode of one of the LED. Cathode of LED is connected to ground through a current limiting resistor.

Step 3

Pin 13 of Rpi is connected to anode of second LED. Cathode of second LED is connected to ground through a current limiting resistor.

Interrupt on Raspberry Pi using SPI protocol

Circuit Diagram

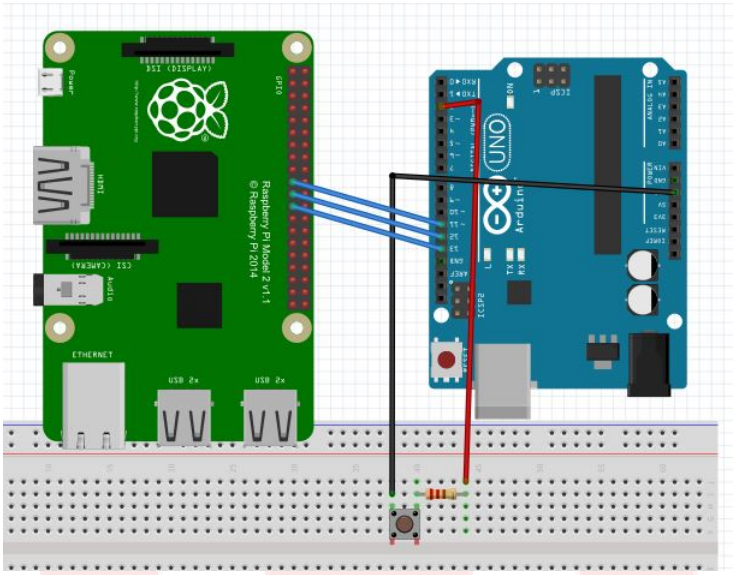


Fig. shows Connection Diagram.

Assembly of harware

Step 1

Pin 11 of arduino is connected to MOSI(i.e pin 19) of RPi.

Step 2

Pin 12 of arduino is connected to MISO(i.e pin 21) of RPi.

Step 3

Pin 12 of arduino is connected to SCLK(i.e pin 23) of RPi.

Step 4

One terminal of switch is connected to resistor and other terminal to ground.

Step 5

Other terminal of resistor is connected to digital pin 2 of Arduino.

Interrupt on Raspberry Pi using I2C protocol

Circuit Diagram

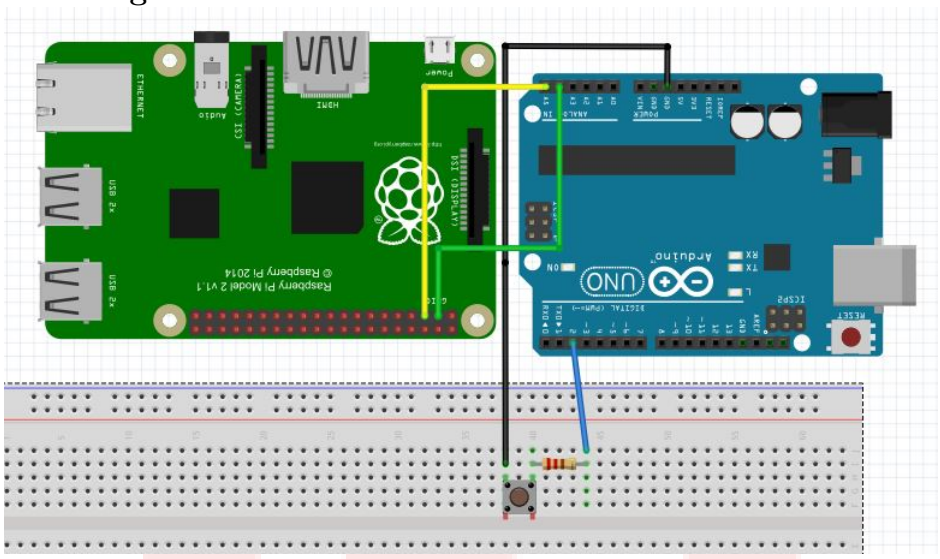


Fig. shows Connection Diagram.

Assembly of harware

Step 1

Pin 3(SDA) of Rpi is connected to analog pin A4 of Arduino.

Step 2

Pin 5(SCL) of Rpi is connected to analog pin A5 of Arduino.

Step 3

One terminal of resistor is connected to switch and other terminal to digital pin 2 of Arduino.

Step 4

One terminal of switch is connected to resistor and other terminal to ground.

1.11 UART Communication between Firebird V and Raspberry Pi

Connections
Firebird V is connected with Raspberry Pi using USB cable.
CODE
Code for Firbird V and Raspberry is available on Github repository.

1.12 Software and Code

Github link for the repository of code

1.13 Use and Demo

Youtube Link of Accessing GPIO pins of RPi.
Youtube Link of External Interrupt on Raspberry Pi.
Youtube Link of I2C Enabling on RPi.
Youtube Link of SPI Enabling on RPi.
Youtube Link of Interfacing Port expander MCP23017 to RPi.
Youtube Link of ADC MCP3008.
Youtube Link of Bluetooth communication.
Youtube Link of PWM driver PCA9685 IC.

1.14 Future Work

Make a PCB Shield for Raspberry Pi which will enhance capability of Firebird V robot.
Raspberry Pi is an Excellent tool for Dynamic and Real time Image Processing. So, by using Raspberry Pi instead of AtMega 2560 on Firebird V we can implement much complex themes comprising image processing, as the programming in python makes it much easier.

1.15 Bug report and Challenges

Communicating Raspberry Pi with ATMega2560 through UART communication protocol causes a lot of delay which can make the programming task much complicated and the program becomes inefficient.

Any failure or challenges faced during project

1. Object Tracking of Firebird V Robot using 3 Sharp IR Sensors using UART communication between Raspberry Pi and ATMega2560.
2. Courier Service theme of eYRC+ using UART communication between Raspberry Pi and ATMega2560.
3. Obstacle Avoidance Robot using UART communication between Raspberry Pi and ATMega2560.

These ideas did not work due to delay in communication because of which line following is not at all effective.

Bibliography

[1] Raspbian Jessie OS Installation, *https://www.engadget.com/2012/09/04/raspberry-pi-getting-started-guide-how-to/*

[2] SPI and I2C communication protocol *www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/*

[3] I2C configuration *https://learn.adafruit.com/ adafruits-raspberry-pi-lesson-4-gpio-setup/ configuring-i2c*

[4] Pulse Width Modulation *http://www.electronics-tutorials.ws/blog/ pulse-width-modulation.html*

[5] Software Interrupt *https://www.techopedia.com/de nition/22195/software-interrupt*

[6] Port expander MCP23017 IC *https://www.mathworks.com/examples/matlab/4547-add-digital-i-o-pins-to-raspberry-pi-hardware-using-mcp23017*