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

The motive of this project is to create learning materials based on Raspberry Pi. Interfacing Raspberry Pi with Firebird V Robot will enhance the capabilities of the robot to do lot more tasks.
This project includes:

- Interfacing LED, Switch, LCD and ICs on Raspberry Pi
- 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
- Documentation and video tutorials explaining individual module
- Designing PCB for the devices interfaced with Raspberry Pi

Completion status

- **Task 1:**
  Learnt, tested and 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 communication 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

- **Task 7:**
  Challenge Activity

1.1 Hardware parts

- List of hardware

  1. Raspberry Pi
     Download Datasheet Vendor Details

Figure 1.1: Raspberry Pi

2. FireBird V Robot
   Download Datasheet Vendor Details

Figure 1.2: Firebird V robot

3. LED
4. Resistor
5. Switch
6. LCD
   Download Datasheet Vendor Details

Figure 1.3: LCD

7. MCP23017 IC
   Download Datasheet Vendor Details

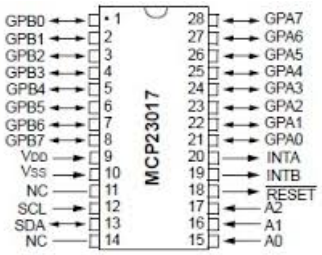

Figure 1.4: MCP23017 Port Expander IC

8. MCP3008 IC
   Download Datasheet Vendor Details

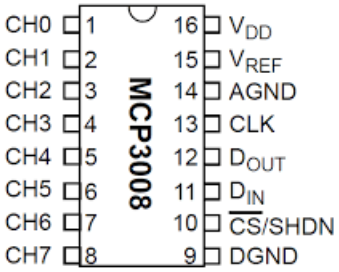

Figure 1.5: ADC MCP3008 IC

9. Sharp Sensor
   Download Datasheet Vendor Details

Figure 1.6: Sharp IR Sensor

10. LM35 Temperature Sensor
    Download Datasheet Vendor Details

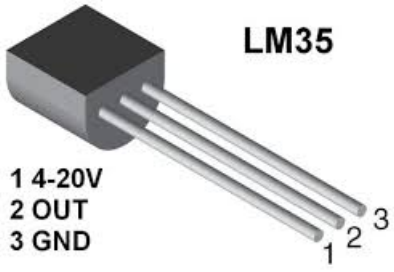

Figure 1.7: LM35 Temperature sensor

11. PCA9685 PWM Driver IC
    Download Datasheet Vendor Details

Figure 1.8: PCA9685 PWM Driver IC

12. L293D Motor Driver IC
    Download Datasheet Vendor Details

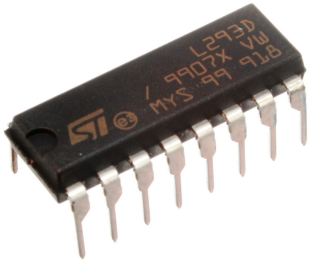

Figure 1.9: L293D Motor Driver IC

13. Capacitor
14. DC Motor
    Download Datasheet Vendor Details

Figure 1.10: DC motor

15. Servo Motor
    Download Datasheet Vendor Details

Figure 1.11: Servo motor

16. 9 V battery
17. Xbee Module
    Download Datasheet Vendor Details

Figure 1.12: Xbee

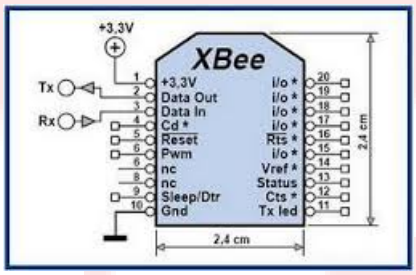

Figure 1.13: Xbee pin diagram

18. Bluetooth Module
    Download Datasheet Vendor Details

Figure 1.14: Bluetooth HC-05

19. Arduino UNO
    Download Datasheet Vendor Details

Figure 1.15: Arduino UNO

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

1.3 Accessing GPIO pins of Raspberry Pi

Description:
The Raspberry Pi 1 Model B+ board contains a single 40-pin expansion header providing access to 26 GPIO(General Purpose Input Output) pins. Remaining 14 pins are 3.3V, 5V, Ground.Out of the 26 GPIO pins there are some pins having special functions such as SPI, I2C and UART communication.

In order to refer to the R-Pi pins there exists two modes:

1. BCM mode: Referring the pins with the GPIO number.
2. Board mode: Referring the pins using the IC pin number.

Problem Statement:
LED and switch are connected to different GPIO pins.LED should get ON when switch is pressed once and should get OFF when switch is pressed again.

Circuit Diagram

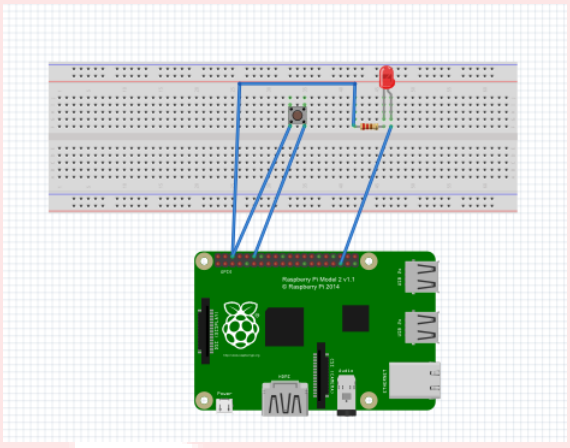

Figure 1.16:

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

Description:
I2C stands for Inter Integrated Circuit. It is a communication protocol in which many devices are connected with two signal line Serial Data(SDA) 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. Type *sudo raspi-config*. This will launch the raspi-config utility.

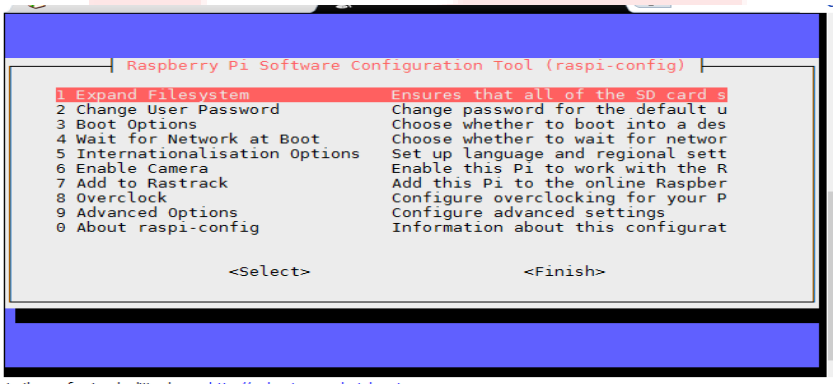

Figure 1.17: [4]

4. Select the *Advanced options*

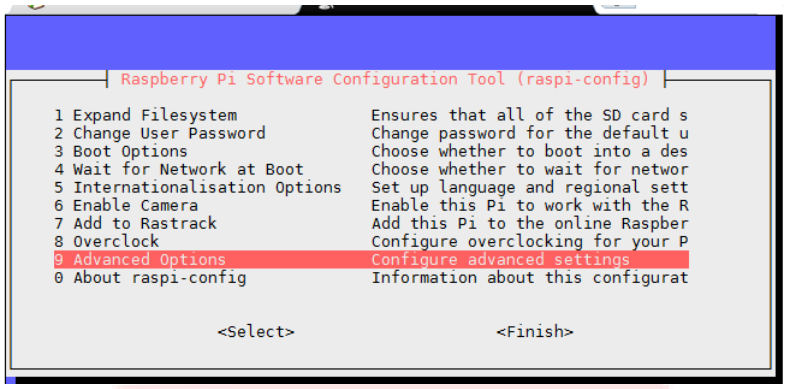

Figure 1.18: [4]

5. Then select *option A7 I2C*

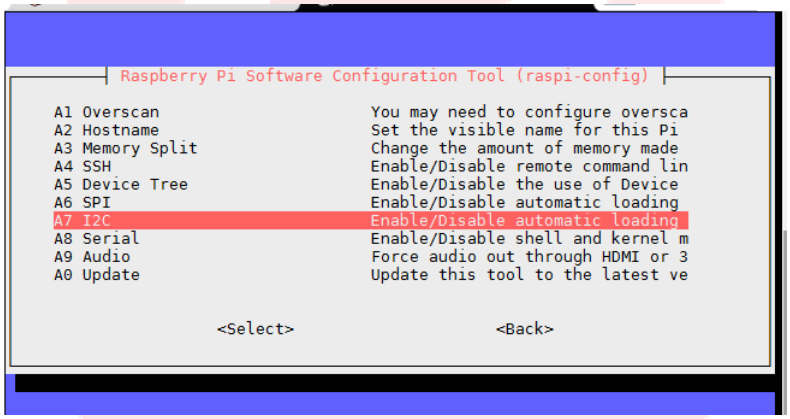

Figure 1.19: [4]

6. It will ask to enable the ARM I2C interface, click *YES*.

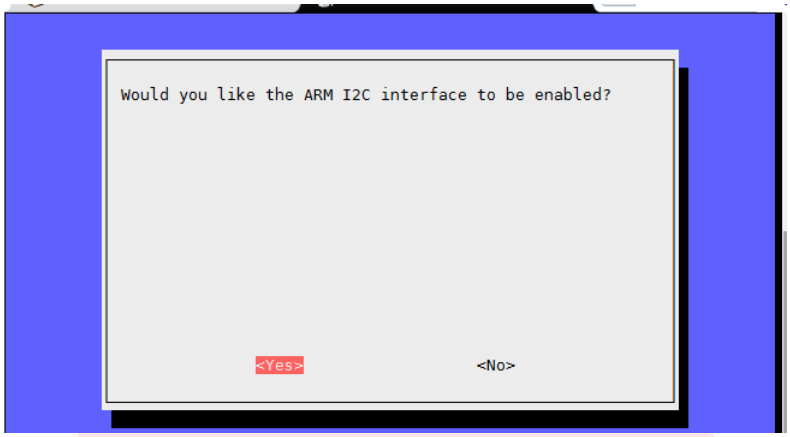

Figure 1.20: [4]

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

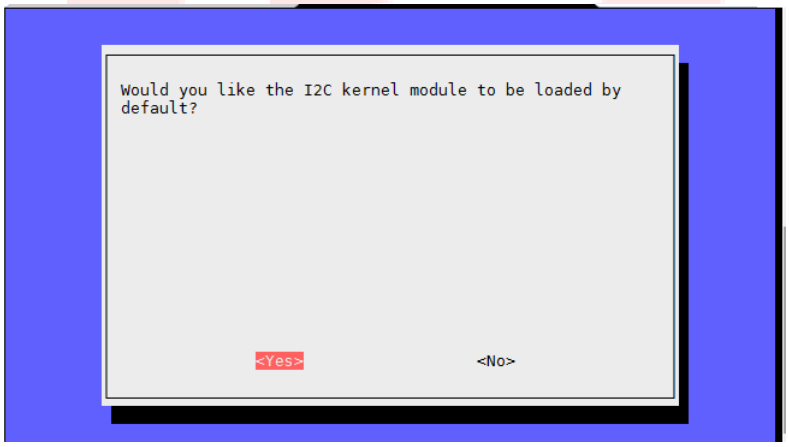

Figure 1.21: [4]

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

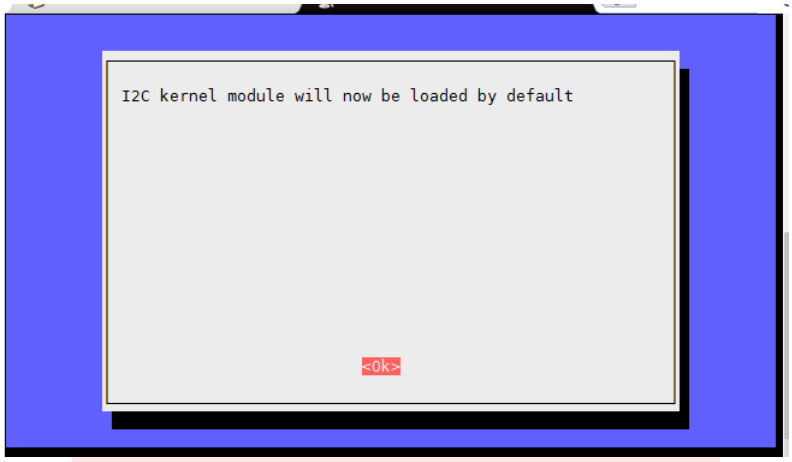

Figure 1.22: [4]

9. Select Finish to return to command line.

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

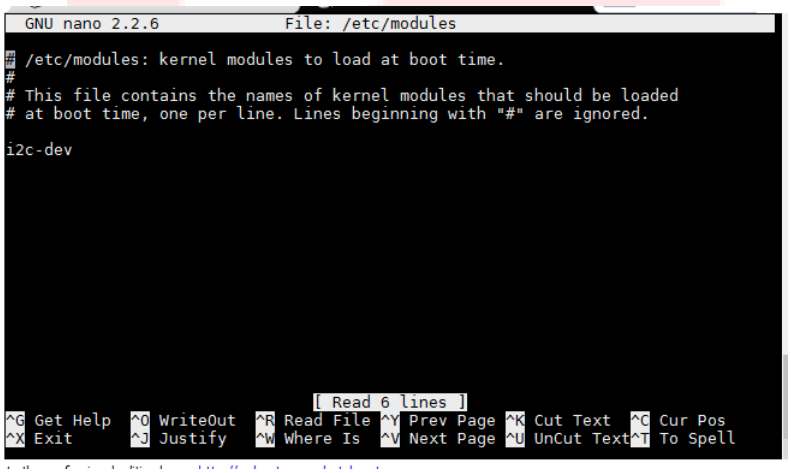

Figure 1.23: [4]

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

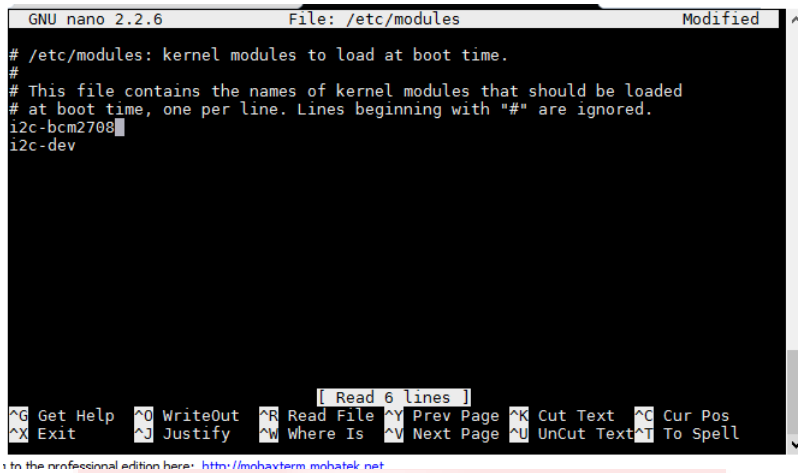

Figure 1.24: [4]

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

13. 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

14. 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.

15. 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.

16. 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.

17. Type the command:
sudo i2cdetect -y 1

18. 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 1.25: [4]

1.5 Interfacing Port Expander MCP23017 IC

Description:
There are only 26 GPIO pins on Raspberry Pi, so when we want to connect more number of actuators, LCD and other peripherals then these pins may be insufficient. So, in order to increase the number of GPIO pins a port expander IC MCP23017 is used. It communicates with Raspberry Pi through I2C protocol. It is a 28 pin IC, providing 16 additional GPIO pins.
Problem Statement:
Interfacing LCD to RPi using MCP23017 IC and display some message.
Circuit Diagram

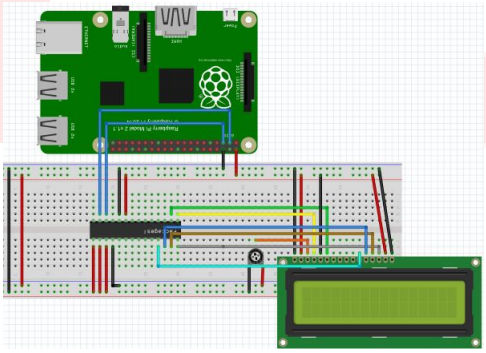

Figure 1.26: LCD connected to MCP23017 IC

Schematic Diagram

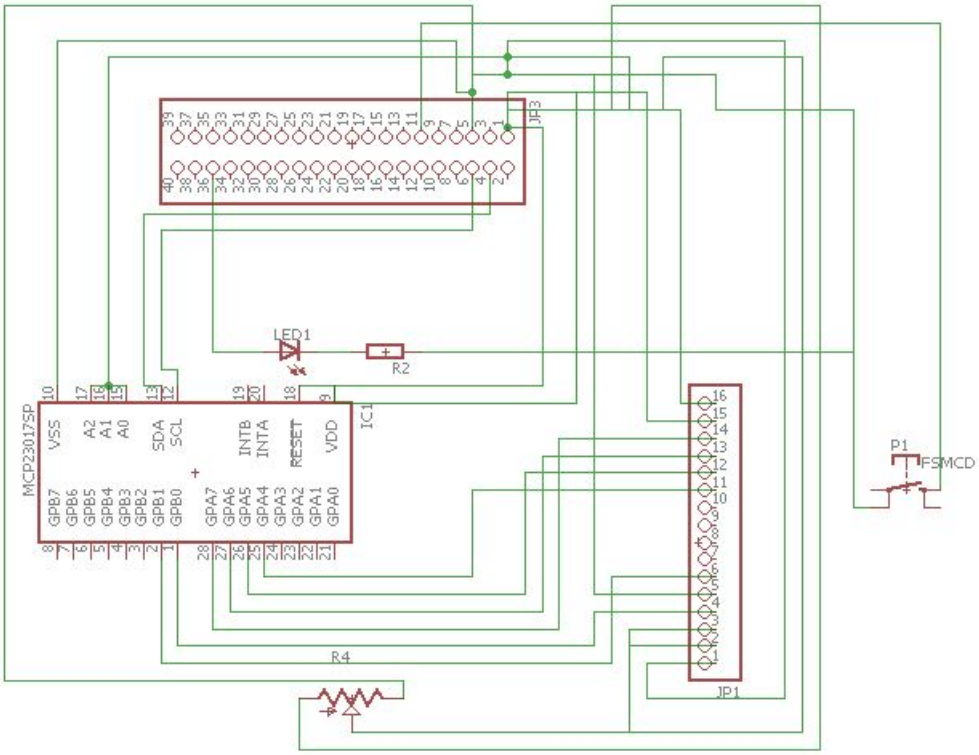

Figure 1.27: Eagle Schematic

PCB Layout

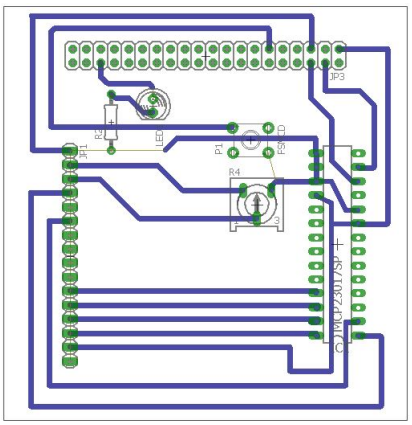

Figure 1.28: PCB Layout

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 does not require any additional components. It uses the SPI bus protocol which is supported by the Pi's 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

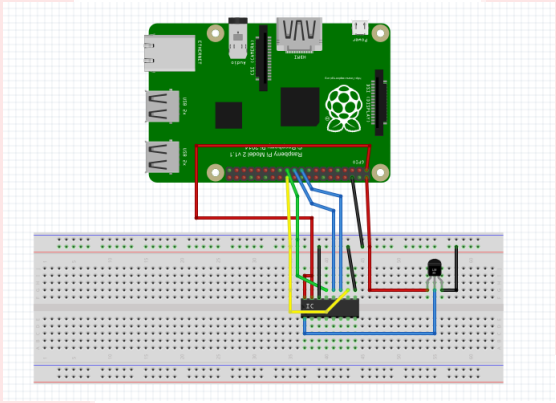

Figure 1.29: Circuit Diagram

Schematic Diagram

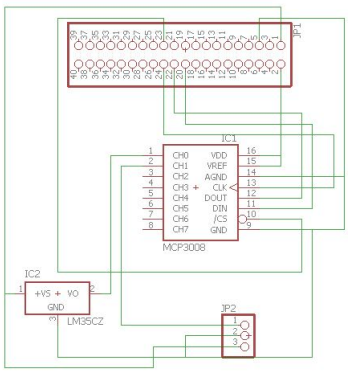

Figure 1.30: Schematic in Eagle

PCB Layout

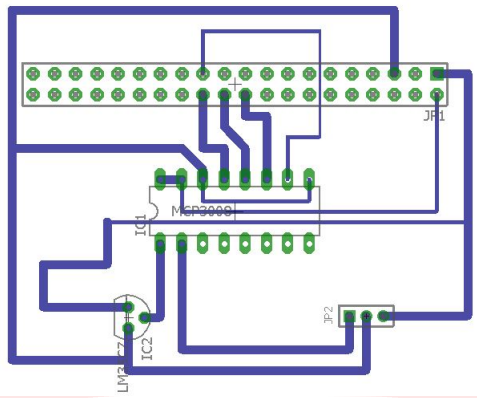

Figure 1.31: PCB Layout in Eagle

1.7 Interfacing DC Motor to RPi using MCP23017(Port Expander) IC

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:

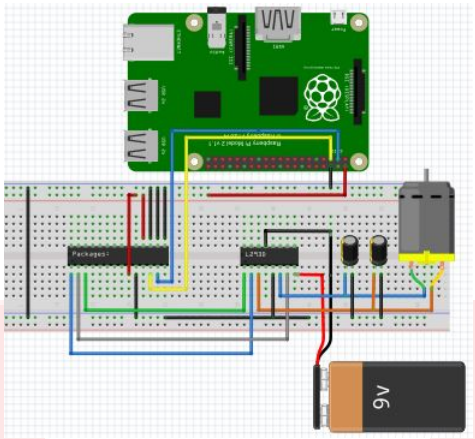

Figure 1.32: Fritzing Circuit Diagram

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 to RPi 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

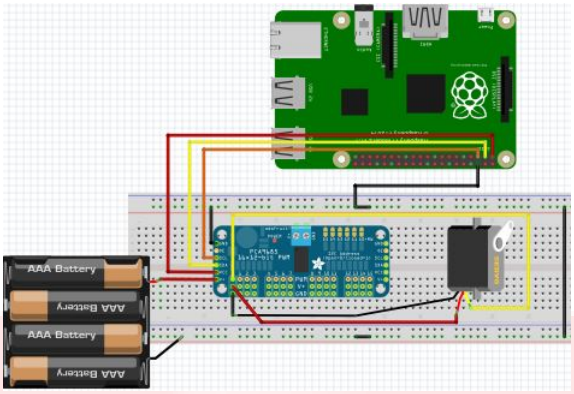

Figure 1.33: Fritzing Circuit Diagram

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

Description:
The universal asynchronous receiver/transmitter (UART) takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. Each UART contains a shift register, which is the fundamental method of conversion between serial and parallel forms.Serial communication is the process of sending data one bit at a time, sequentially, over a communication channel.Asynchronous serial data communication is widely used for character-oriented transmissions.
In the asynchronous method, each character is placed between start and stop bits. This is called framing. In data framing for asynchronous communications, the data, such as ASCII characters, are packed between a start bit and a stop bit. The start bit is always one bit, but the stop bit can be one or two bits. The start bit is always a 0 (low) and the stop bit(s) is 1 (high).

1.9.1 Bluetooth Module HC-05:

Description:
HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. It comes with a factory set Baud rate of 9600bps. It is used for serial communication for small distances of few meters.
Problem Statement:
Establish communication between Raspberry Pi and any other bluetooth enabled device(we have used android device) using Bluetooth module.Check it by sending and receiving data.

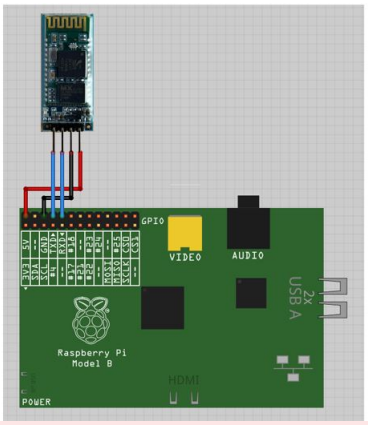

Figure 1.34: Connection of Bluetooth module with RPi

Assembly of hardware:

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.

1.9.2 Xbee Module:

Description:
Xbee can be used for wireless communication with low power consumption. A 3.6V 600mA Lithium battery may last 6 - 12 months for powering up an Xbee while the wireless range can up to 1 mile. It talks with well known UART interface and makes it easy to use. It is simple and straight forward if you only use 2 Xbee for communication. Xbee is the module using Zigbee

protocol. Zigbee is a wireless communication protocol like wifi and bluetooth.

Problem Statement:

Connect one Xbee to RPi and connect second Xbee to PC(access this module using XCTU software).Check the communication by sending and receiving data serially.

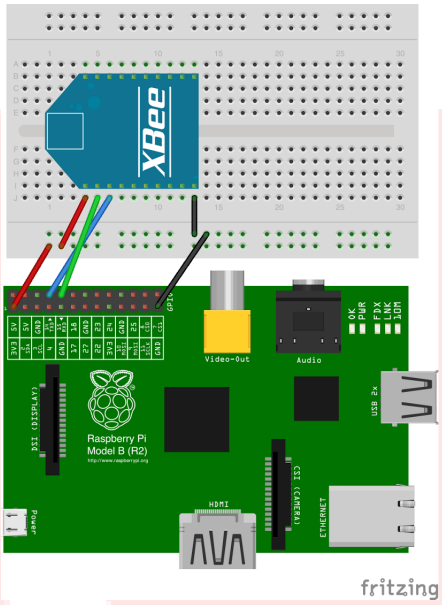

Figure 1.35: Connection of Xbee module with RPi

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.

Step 5

Connect second Xbee to PC through USB cable.
Schematic Diagram of Bluetooth and Xbee Module:

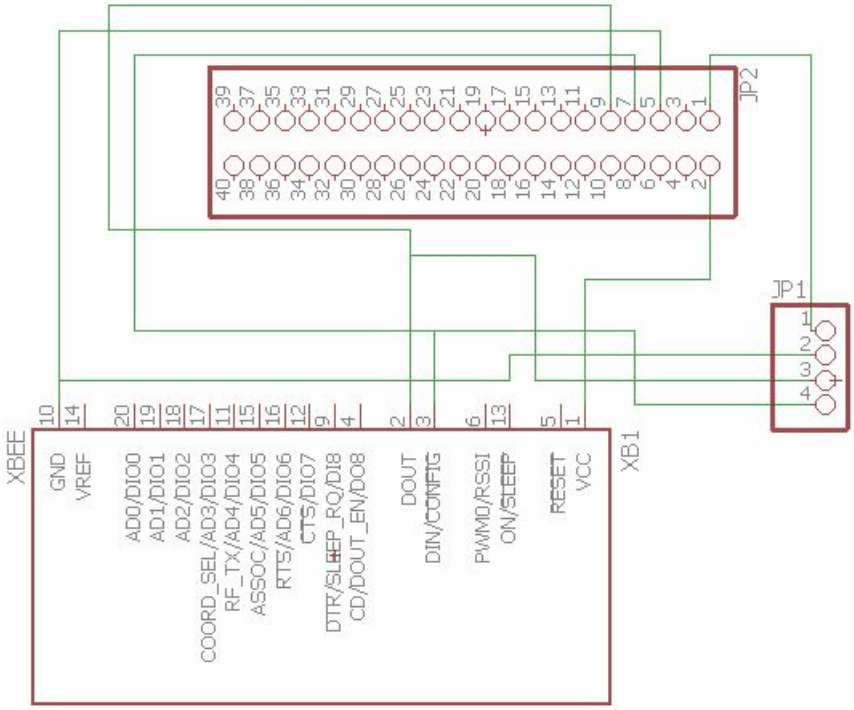

Figure 1.36: Schematic in Eagle

PCB Layout

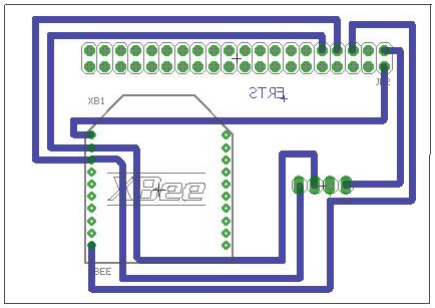

Figure 1.37: PCB Layout

1.10 Interrupt on Raspberry Pi

An Interrupt is a signal generated by some event external to CPU, which causes the CPU to stop what it is doing and go to separate piece of code known as ISR for execution. When the execution of ISR completes it starts main program from where it left. Types of interrupts are as follows:

1. Hardware Interrupt:
   Hardware interrupts are used by devices to communicate that they require attention from the operating system.These interrputs are requested by external peripherals such as keyboard, moving the moving trigger.

2. Software Interrupt:
   A software interrupt is a type of interrupt that is caused either by a special instruction in the instruction set or by an exceptional condition in the processor itself. A software interrupt is invoked by software, unlike a hardware interrupt, and is considered one of the ways to communicate with the kernel or to invoke system calls, especially during error or exception handling.

In Raspberry Pi we actually use the concept of multithreading. By multithreading multiple programs can be executed in different threads i.e., multiple tasks can be executed at the same time.

External Interrupt on RPi:

Description:
Interrupt can be generated externally on Raspberry Pi by logic devices such as switch or push button which changes the logic when they are pressed.
Problem Statement:
Connect a switch to any one of the GPIO pin and connect two LED's to the two other GPIO pins.One LED should blink continously with a time delay of 1 second and other LED should blink only when pushbutton is pressed.
Circuit Diagram

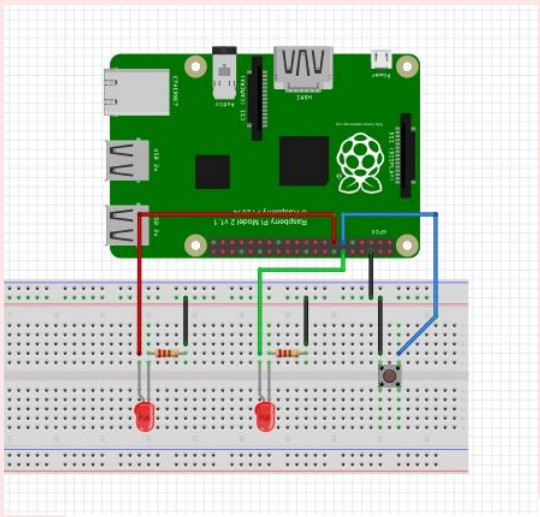

Figure 1.38: Fritzing Circuit Diagram

Assembly of hardware

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

Description:
SPI is a single-master communication protocol. This means that one central device initiates all the communications with the slaves. When the SPI master wishes to send data to a slave and/or request information from it, it selects slave by pulling the corresponding SS line low and it activates the clock signal at a clock frequency usable by the master and the slave. The master generates information onto MOSI line while it samples the MISO line.
Problem Statement:
Connect RPi as SPI master and Arduino as SPI slave and generate an SPI interrupt on RPi.
Circuit Diagram

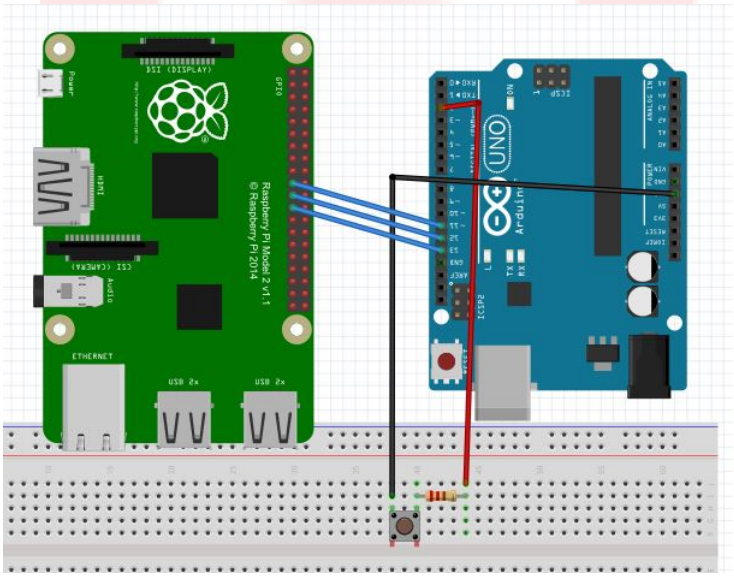

Figure 1.39: SPI Interrupt using RPi as master and Arduino as slave.

Assembly of hardware

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

Description:
The Inter-integrated Circuit (I2C) Protocol is a protocol intended to allow multiple slave digital integrated circuits (chips) to communicate with one or more master chips. I2C requires a mere two wires, like asynchronous serial, but those two wires can support up to 1008 slave devices. I2C can support a multi-master system, allowing more than one master to communicate with all devices on the bus (although the master devices can't talk to each other over the bus and must take turns using the bus lines). Whenever CPU receives interrupt request from external device using I2C protocol.Control goes to receives data from devices. After that control returns to its main program execution.
Problem Statement:
Connect RPi as I2C master and Arduino as I2C slave and generate an I2C interrupt on RPi.

Circuit Diagram

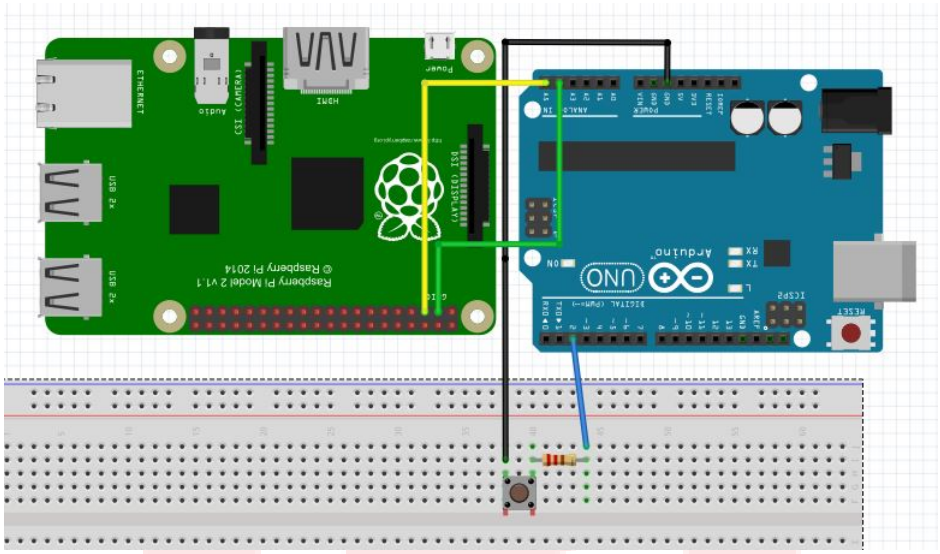

Figure 1.40: I2C Interrupt using RPi as I2C master and Arduino as I2C slave.

Assembly of hardware

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

Description:
In Serial Interrupt, whenever CPU receives interrupt request from external device serially(UART), Control goes to receives data from devices. After that control returns to its main program execution.
Problem Statement:
Connect Raspberry Pi to Firebird V through USB cable and control Firebird V by giving commands in the form of data serially from RPi.
Connection Image:

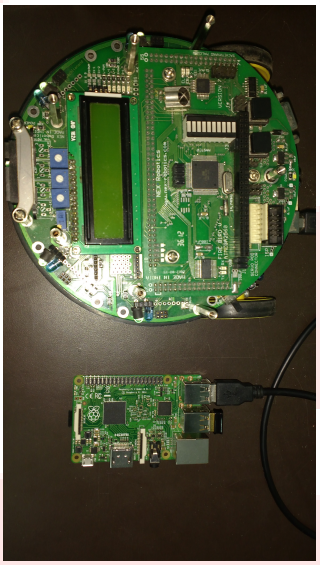

Figure 1.41: UART communication between RPi and Firebird V

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 AtMega2560 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.

    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.

- SPI interrupt and I2C interrupt on Raspberry Pi using RPi as Master and Arduino as slave.

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*

[7] Queries regarding Raspberry Pi *http://stackoverflow.com/questions/tagged/raspberry-pi*

[8] Libraries for progrmming different IC's or modules *https://github.com/adafruit*