

ROBOTICS CLUB, INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

Club Automation Project Report

BY

Shruti Sagar Garvit Darpan Jeevan Aakash Subham Rahul

20th June, 2017

Aim and Abstract

In this project, we look to automate various parts of Robotics Club Room, like lights, fans, switches, door lock etc., and develop a user interface and web server for use. The work has been divided among the team members in several modules like creating app for control, developing server for working of Raspberry Pi, triggering light, fan circuits using relay, designing door lock etc, with each module having its separate work guidelines. At the end, a comprehensive integration will be undertaken.

Contents

| | |
|--|----------|
| 1 Module I – Working with R-Pi | 4 |
| 1.1 Introduction | 4 |
| 1.2 Objectives | 4 |
| 1.3 Work Done | 4 |
| 1.4 Problems Faced and their Solutions | 4 |
| 1.5 Unresolved Problems | 4 |
| 1.6 Future Works | 5 |
| 1.7 Links of Tutorials | 5 |
| 2 Module II – Developing Web Server | 5 |
| 2.1 Introduction | 5 |
| 2.2 Objectives | 5 |
| 2.3 Work Done | 5 |
| 2.4 Unresolved Problems | 7 |
| 2.5 Future Works | 7 |
| 2.6 Links of Tutorials | 7 |
| 3 Module III – App Development | 7 |
| 3.1 Introduction | 7 |
| 3.2 Objectives | 7 |
| 3.3 Work Done | 7 |
| 3.4 Problems faced and their solutions. | 8 |
| 3.5 Unresolved Problems | 8 |
| 3.6 Future Works | 8 |
| 3.7 Links of Tutorials | 8 |

| | |
|--|-----------|
| 4 Module IV – Designing Relay Circuit | 9 |
| 4.1 Introduction | 9 |
| 4.2 Objectives and Work Done | 9 |
| 4.3 Problems Faced and their Solutions | 10 |
| 4.4 Unresolved Problems and Future Works | 10 |
| 4.5 Links of codes and tutorials | 11 |
| 5 Module V – Designing Door Lock | 11 |
| 5.1 Objectives | 11 |
| 5.2 Work Done and Problems Faced. | 11 |
| 5.3 Future Works | 11 |
| 5.4 Links of tutorials | 11 |

1 MODULE I - Working with R-Pi

1.1 Introduction

The **Raspberry Pi** is a small single board complete Linux computer and can provide all the expected abilities that implies, at a low-power consumption level. We will be using it to control different parts of Club Automation module, as a control system

1.2 Objectives

To send an http request (using Unirest library in python) from R-pi to web server and check the response to find out if there is any change in the database of the web server.

1.3 Work Done

We installed an operating system (Raspbian) on R-pi and successfully created a connection between our laptop and R-pi. Then we installed various packages like node.js, python. Studied python and tried to understand Unirest library through examples and successfully sent http request to some websites and got response in HTML format.

1.4 Problems faced and their solutions

- Establishing connection between R-pi and laptop
Sol: used terminal commands ping <ip address> and ssh pi@<ip address>
- Problem faced in reading data from web server
Sol: Using http request.

1.5 Unresolved Problems

- Error while installing Unirest library.
- Not completely comfortable in using Unirest library.
- Unable to get response from the server in json format and use it further.

1.6 Future Works

- Try to successfully send an http request to the server, get response in json format and convert the data to array/variables for further use.
- Send the output to Arduino using R-pi.

1.7 Links of tutorials

- For python:
<https://www.codecademy.com/learn/python>
- For installation of R-pi OS:
<https://www.raspberrypi.org/learning/noobs-install>
- For Unirest library:
<http://unirest.io/python.html>

2 Module II – Developing Web Server

2.1 Introduction

A **web server** is a computer system that processes requests via HTTP, the basic network protocol used to distribute information on the World Wide Web. The term can refer to the entire system, or specifically to the software that accepts and supervises the HTTP requests.

2.2 Objectives

- To develop a web server and create a database in it to store the current status of appliances.
- To respond to the requests sent by Android app and raspberry pi.

2.3 Work Done

- Launched an ubuntu instance on amazon ec2 web server and installed various packages like node.js, mongoDB, python,unirest library.

```

response = unirest.post("http://httpbin.org/post", headers={ "Accept": "application/json"
}, params={ "parameter": 23, "foo": "bar" })

response.code # The HTTP status code
response.headers # The HTTP headers
response.body # The parsed response
response.raw_body # The unparsed response

```

Figure 1: Getting Response from Unirest

```

C:\Users\ACER\Downloads\index.js - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

index.php x index.js x
1 var MongoClient = require('mongodb').MongoClient, assert = require('assert');
2 const express = require('express')
3 const app = express()
4 const MONGO_URL = 'mongodb://127.0.0.1:27017/robo';
5
6
7 app.get('/', function (req, res) {
8   res.send('Hello World!')
9   console.log("yoi!")
10 })
11
12
13 app.get('/update',function(req,res){
14   console.log(req.query)
15
16   var name = req.query.name;
17   var requiredStatus = req.query.requiredStatus;
18   MongoClient.connect(MONGO_URL, function(err, db) {
19     assert.equal(null, err);
20     console.log("Connected successfully to server");
21     if(err){
22       res.send('Some error');
23       db.close();
24     }else{
25       var switches = db.collection('switches');
26       switches.find({name:name}).toArray(function(err, docs){
27         if(err||docs.length==0){
28           console.log(err,docs.length,'err'||docs.length==0');
29           res.send('inserted successfully');
30           switches.insert({name:name,requiredStatus:requiredStatus});
31           db.close();
32         }else{
33           switches.updateOne({name:name},{ $set:{ "requiredStatus":requiredStatus}});
34           res.send(docs);
35           db.close();
36         }
37       });
38     }
39   });
40 });
41
42

```

Figure 2: Using MongoClient

- Studied javascript, node.js, mongoDB.
- Created a local database using mongoDB with few entries of appliances.
- Established connection between amazon ec2 web server and laptop.

2.4 Unresolved problems

Unable to bring the local database on amazon ec2 web server.

2.5 Future Works

Make some changes in the code so that the server responds to request sent by R-pi only when there is a change in a database entry.

2.6 Links of tutorials

- Java script: <https://www.w3schools.com/>
- Mongo: <https://docs.mongodb.com/#using-mongodb>
- youtube video lectures for Node: [Net ninjas](#)

3 Module III – App Development

3.1 Introduction

Android Studio is the official integrated development environment (IDE) for the Android platform. Android Studio provides the fastest tools for building apps on every type of Android device.

3.2 Objectives

To develop an android app as an interface to control the appliances of robotics club.

3.3 Work done

- Learnt Java and Android Studio.
- Made an app which can send http request to a site and display the response by using volley, an http library
- Learnt how to use listview and HashMap class.

- Made an app which consists of togglebuttons for all the appliances in the club. The current state of the appliance is displayed in the app. Whenever the togglebutton for an appliance is clicked, it can send an http request to the server.

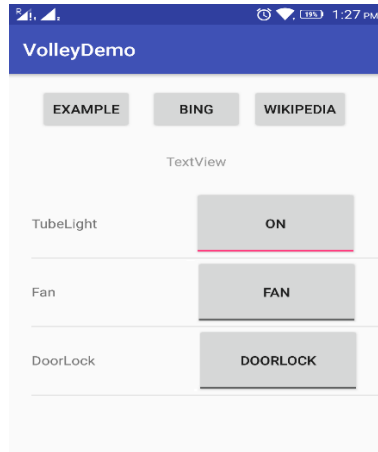


Figure 3: Screenshot of Demo App

3.4 Problems faced and their solutions

- Unable to display the response after sending an http request to a site.
Solution: Used the Volley library.
- Unable to use listview efficiently to display current state of appliance.
Solution: Learnt how to use the HashMap class for displaying current state.

3.5 Unresolved Problems

We have to see that how to use the app for opening and closing the door and make this more efficient by creating a password for the door lock in the app.

3.6 Future Works

Complete the app and add designs to it.

3.7 Links of Tutorials

- https://www.youtube.com/watch?v=QAbQgLGKd3Y&list=PL6gx4Cwl9DGBsvRxJJ0zG4r4k_zLKrnxl

4 Module IV - Designing Relay Circuit

4.1 Introduction: Why we need a Relay?

A relay is an electrically operated switch, which when connected to OS like Arduino or At-Mega, can be used to automate electric circuits which switch on and off on simple binary commands through connected web server.

4.2 Objectives and work done: Designing Circuit

The input to relay channel board will be directly connected to the Arduino and will be getting its DC input power supply from it. The Arduino will then be connected to the Rpi using a USB Cable. The output will be to the AC device that needs to be controlled. This connection will be achieved by cutting the live wire of device and connecting the new two ends with relay.

The commands to relay circuit will be given through RPi to the Arduino, so, Arduino IDE was downloaded that works through RPi.

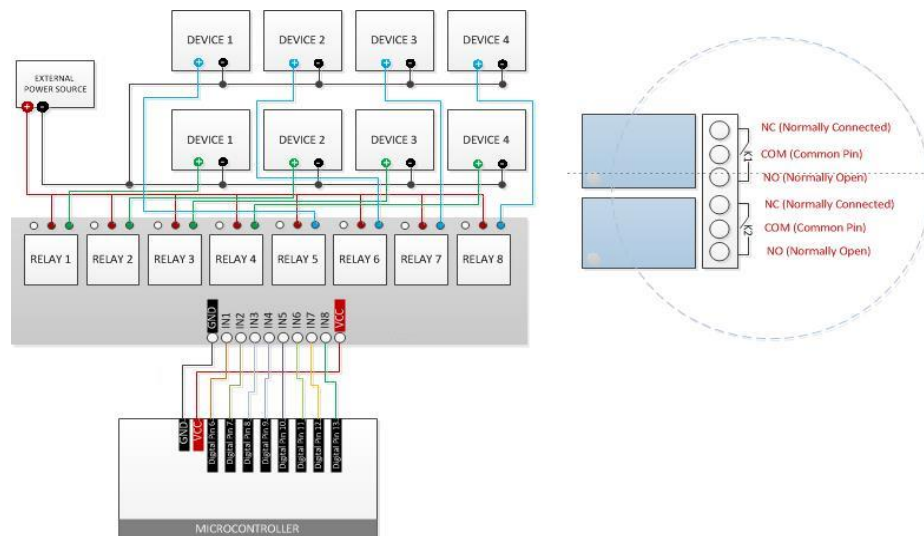


Figure 4: Circuit Diagram For 8 Channel Relay

The work done thus include the following:

- Studying Relays and how to control them using Arduino.
- Learning how to control Arduino through RPi.
- Learning Python and writing code scripts.
- Designing the comprehensive circuit to be used.
- Installing Ubuntu, creating interface for controlling RPi.

4.3 Problems faced and their Solutions

Deciding what type of relay we should use and whether to control directly using RPi, or use an Arduino.

There are a host of Relays available in market for use. Majorly we had to decide whether to use a Solid State Relay (SSR) or an Electromagnetic Relay (EMR) with optocoupler, with difference in their working mechanism, durability and range of Input-Output Methods available. Earlier, it was decided that the relay would directly be connected to RPi (as it is faster) and power input to relay would be through AC supply, so we were looking to use an SSR with AC-AC I/O Module. However, this method was found to have many inefficiencies like danger to RPi (any small wrong connection could blow up the entire circuit) and more cost, so it was finally decided to use 5V DC input - AC output EMR with optocoupler, with input from Arduino and then connection to RPi, with optocoupler improving reliability and safety to device.

4.4 Unresolved Problems and Future Work

There are still some issues with running Arduino IDE through RPi, which will be rectified after reading few more tutorials on the net.

Future works include making real connections of the above and trying out the codes. In the next week, Arduino will be controlled through the Laptop, and in the following week, it is aimed to finally use it though RPi.

4.5 Links of Codes and tutorials

- Deciding on the Relay :
<http://www.electronicdesign.com/components/electromechanical-relays-versus-solid-state-each-has-its-place>
- Connecting Arduino with RPi:
<http://conoroneill.net/connecting-an-arduino-to-raspberry-pi-for-the-best-of-both-worlds/>
- Controlling Relay with Arduino :
<http://www.instructables.com/id/Controlling-AC-light-using-Arduino-with-relay-modu/>
- Home Automation Tutorial :
<https://www.youtube.com/watch?v=xc1daIb1LVc>
- Sample Simulation in circuits.io : (This includes the sample code we have written to control few bulbs)
<https://circuits.io/circuits/3401399-club-automation>
- Sample Codes on github for reference:
<https://github.com/uugear/UUGear>
<https://github.com/ledongthuc/example/blob/master/relay/src/src.ino>

5 Module V – Designing Door Lock

5.1 Objectives

To design a mechanical lock for the door which can be controlled using the android app.

5.2 Work done and Problems Faced

While designing the lock, problems were faced regarding how the lock will be positioned on the door and the design of the lock was also not up to the mark.

5.3 Future Works

Get the lock assembled, successfully install it on the door and connect it with the electrical components for its functioning.

5.4 Links of Tutorials

- <http://makezine.com/projects/automatic-door-lockunlock-for-homeoffice/>