Indian Academy of Sciences, Bengaluru
Indian National Science Academy, New Delhi
The National Academy of Sciences India, Allahabad

SUMMER RESEARCH FELLOWSHIPS — 2017

Format for the final Report*

Name of the candidate : UTKARSH SINGH

Application Registration no. : ENGS6904

Date of joining : 15/05/17

Date of completion : 10/07/17

Total no. of days worked : 57

Name of the guide : KVS HARI

Guide's institution : IISC, BANGALURU

Project title : STUDY OF RASPBERRY PI EMBEDDED SYSTEM FOR BIOMEDICAL APPLICATION

Address with pin code to which the certificate could be sent:

BANYAN ROAD 12/12, SHIPRA SUNCITY, INDIRAPURAM, GHAZIABAD, UP-201014

E-mail ID: utkarsh.singh.eee15@itbhu.ac.in

Phone No: 9839746773

TA Form attached with final report : YES ✓ NO ____

If, NO, Please specify reason ____

Utkarsh Singh
Signature of the candidate
Date: 10/07/17

Hari
Signature of the guide
Date: 10 Jul 17

*The final report could be anywhere between 20 and 25 pages including tables, figures etc.
This format should be the first page of the report and should be stapled with the main report.

(For office use only; do not fill/tear)

Candidate's name:	Fellowship amount:
Student: Teacher:	Deduction:
Guide's name:	TA fare:
KVPY Fellow: INSPIRE Fellow:	Amount to be paid:
Others	A/c holder's name:

IMPORTANT NOTES:
A soft copy of this report should be uploaded in the online page of our website by making use of the userid/password provided to you.

A Project report on

Study of Raspberry Pi based embedded system for biomedical applications

Summer Research Intern Project

Under the guidance of

Prof. KVS Hari

Statistical Signal Processing lab
Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore

Indian Institute of Science, Bangalore

By

UTKARSH SINGH

SRFP -2017 Application No-**ENGS6904**

Department of Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY (B.H.U)-VARANASI

DECLARATION

I declare that this indicted submission represents my conceptions in my own words and where other's conceptions or words have been included, I have adequately cited and referenced the pristine sources. I additionally declare that I have adhered to all principles of academic veracity and integrity and have not misrepresented or fabricated or falsified any conception/data/fact/source in my submission. I understand that any infringement of the above will be cause for disciplinary action by the Institute and can withal evoke penal action from the sources which have thus not been properly cited or from whom opportune sanction has not been taken when needed.

ABSTRACT

Artificial Pancreas is a wearable medical device used by diabetes 1 patients to maintain the desired insulin level in their body. Artificial pancreas uses a Continuous Glucose Monitor to Continuously monitoring the glucose level in body and communicate the data to microprocessor with in turn communicate with insulin pump interfaced with it to deliver appropriate amount of insulin in the body. The project's aim is to study about the microprocessor (Raspberry Pi) and how to communicate with its input/output ports , and the communication between Raspberry Pi and Android device through a Wi-Fi network for larger distance communication having Raspberry Pi hosting a server and Android Device acting as a client.

CERTIFICATE

This is to certify that project entitled “Study of Raspberry Pi based embedded system for biomedical application” is a certified work of Utkarsh Singh, Department of Electrical Engineering. IIT(BHU)-Varanasi and carried out at Indian Institute of Science, Bengaluru under my guidance during the period of 15 May-10 July-2017 in fulfilment of Summer Research Fellowship.

Prof. KVS Hari

Statistical Signal Processing Lab

Department of Electrical Communication Engineering

Indian Institute of Science

Bengaluru-560012-INDIA

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my guide Prof. KVS Hari for providing me the opportunity to work with him. His approach has always inspired me and motivated me to learn more. His idea of reading any research paper or thesis helps me a lot to gain a nice approach for the project. His insistence to explore new things helped me a lot. I would also like to thank Bhavisya sir for helping me throughout the project. I am also grateful for all the love, support and encouragement that I received from my family. And last but not the least, I would like to thank all my friends without whom I would never have come so far in.

CONTENTS

1. INTRODUCTION

1.1 MOTIVATION

In type 1 diabetes patients , insulin is not produced by pancreas due to absence of insulin producing beta cells . Artificial pancreas (External Wearable medical device) can be used for the continuous management of type 1 diabetes . Continuous Glucose Monitors (CGMs) monitors the glucose level in the patient's body , communicate the data to a microprocessor which processes the data and enable insulin pump interfaced with it to deliver continuous subcutaneous insulin infusion (CSII) therapy to patient. Insulin pumps deliver rapid- or short-acting insulin 24 hours a day through a catheter placed under the skin . Its doses are separated into basal rate, bolus doses to cover excess carbohydrate and Correction or Supplement doses. A Smartphone app is employed to get full details and report of insulin level in patients body for more robust examination.

One of the important part of Artificial pancreas is micro processor as it establishes the communication between CGM and insulin pump enabling these devices to work with each other in synchronous way . This project focuses on study of raspberry pi (microprocessor used) and its communication ports.

1.2 GOALS AND OBJECTIVES

- The objective of the project is to study about the Raspberry Pi 3's Input/output ports and how to communicate with the GPIO ports.
- How to read and write from secondary memory devices like flash drive , hard drive in raspberry pi.
- Wireless communication between Raspberry Pi and Android Smartphone .This consists of transferring of data using Wi-Fi between Raspberry and android app and storing the data on database created on raspberry pi(acting as a server).

2. RASPBERRY PI 3

Raspberry Pi 3 is a microprocessor in succession to the series of the small single Board computers developed by Raspberry Pi foundation .

Raspberry Pi 3 has a quad core processor with clock speed of 1200 MHz .It has 1 GB RAM. It consists of four USB 2.0 ports ,HDMI port , combined 3.5 mm audio jack and composite

video, Ethernet port , Micro SD card slot ,Camera Interface(CSI),Display Interface(DSI),Video Core 3D graphics core . Raspberry Pi has an added advantage of having BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board over its previous instalments.

Raspberry Pi 3 supports various operating Systems- Raspbian (Debian based Linux OS), Android Things, Windows 10 IoT Core , Fedora , CentOS , Ubuntu MATE , Arch Linux ARM , kali Linux , FreeBSD.

Raspberry Pi 3 consists of 40 GPIO (General Purpose Input/output Pins).The pin configuration is-

- Power- 4 pins (2 5V & 2 3.3v)
- Ground- 8 pins
- UART (Universal Asynchronous Receiver/Transmitter)- 4 Serial pins for communicating with other devices.
- I^2C (Inter-Integrated Circuit)- 2 pins that allow pi to connect and talk to hardware modules that support I2C protocol.
- SPI (Serial Peripheral Interface Bus) – 5 Pins that allow you to connect and talk to hardware modules that support SPI protocol.
- GPIO (General Purpose Input / Output) - Standard pins that can be used to turn devices on and off.

2.1 SETTING UP RASPBERRY PI 3

Format the Micro SD card using SD card Formatter. Install the NOOBS on micro SD card and save all NOOBS folders on SD card. During first boot install the downloaded Operating System on Raspberry Pi. Connect the monitor with Raspberry Pi using HDMI to VGA converter. For using the Pi without monitor use SSH client like putty to run pi terminal on laptop. Connect other peripherals devices(mouse, keyboard) using USB ports.

For wireless Communication and transfer of data with peripheral devices and raspberry pi, built-in Wi-Fi and Bluetooth is used.

Setting up Bluetooth on Raspberry Pi 3:

- In the terminal type ‘bluetoothctl’ to set the on board Bluetooth.
- Then ‘agent on’ and ‘default agent’ and scan for the available Bluetooth devices nearby using ‘scan on’.
- Pair with the desired Bluetooth device using connect command followed by the Bluetooth device address.

Setting up Wi-Fi on Raspberry Pi 3:

- " sudo iwlish wlan0 scan " scan for the available Wi-Fi networks.
- Open wpasupplicant configuration file, add ssid and password at wpa-supplicant configuration file.Add the network configuration in the file depending on the prioritises provided to the wifi networks.

2.2 COMMUNICATING THROUGH I/O PINS

Raspbian comes with a pre-installed library 'RPio.GPIO' which enables the code to control the functioning of GPIO pins.

- Import the 'RPio.GPIO' library in the python sketch.
- GPIO pins can be referred in setmode in two different ways-Physical Numbering (GPIO.BOARD) and GPIO numbering i.e. referring pins as Broadcomm SOC channel (GPIO.BCM).
- GPIO.HIGH sets pin at high potential and GPIO.LOW sets pin at zero potential.

2.3 Mounting External Hard Disk on Raspberry Pi:

New Technology File System is the file system that windows OS use for storing and retrieving files on hard disk.

- Plug in USB HDD/Drive to Raspberry Pi and install ‘ntfs-3g’ package for using a NTFS formatted drive.
- Create a location for mount point and give proper permission.
- Mount USB Drive and then check for its accessibility at /mnt/volume
- Configure RaspPi to do this after every reboot. Take backup of current fstab and add the mount information in the fstab file and then reboot.

3. Raspberry Pi & Android App Communication

Smartphones are now used almost everywhere and has wide range of application including field of Biomedical. The data can be transferred from smartphone to raspberry pi and vice-versa. Like in artificial pancreas ,Smartphone app is employed to get full details and report of insulin level in patients body for more robust examination.

This project comprises of building an android app to take input from the user and transfer the input data to MySQL database hosted on raspberry pi. Smartphone acts as a client with raspberry pi acting as a server.

Things required for the project:

- A smart phone or an emulator running on Android Studio.
- Raspberry Pi
- An internet connection with port 80 forwarded to the IP of your Raspberry Pi .

3.1 Creating Database Using MySQL and PHPMyadmin

For Server hosting on raspberry pi Apache2,MySQL,PHPmyadmin software are used. After installing these software on raspberry pi, Open up web browser and type localhost/phpmyadmin which opens a PHP home page prompting user to login with its credentials. Create a MySQL database table by selecting create option in Database .In this project ‘Parts’ database has been created.

Server Side Scripting:

On Raspberry Pi , create four php files in var/www/html directory.

Configuration file: File contains the login credentials having user id, password, database name and server name.

Connection file: This file consists of function that connects the database to android app on Wi-Fi network.

Create file: Creating new tuples in database and inserting the transferred data to the created tuple.

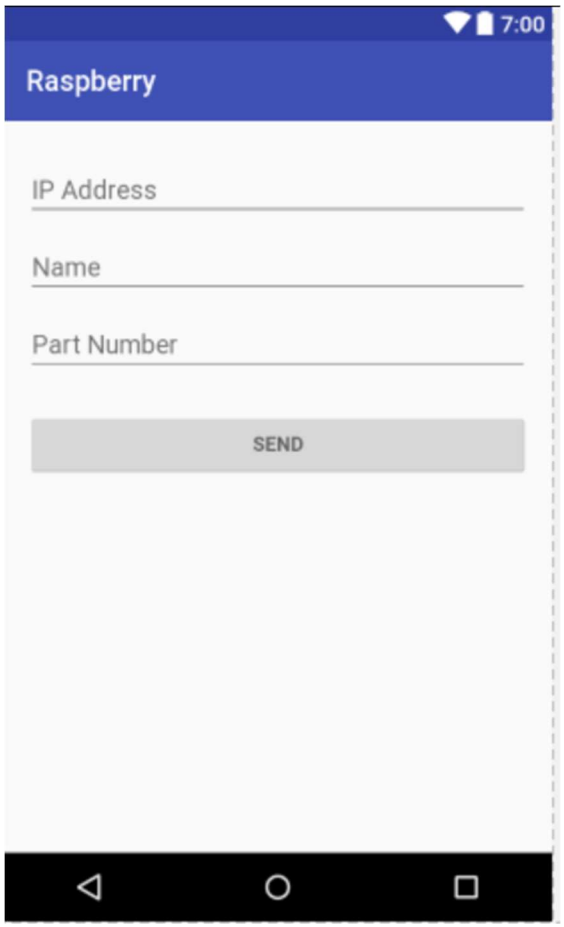
View file: Shows the records which are inserted in database in phpmyadmin web page.

3.2 Writing An App

Android Studio is used for developing the app for this project. App consists of two text field, one for string input and the other for numerical input. Send button is used to transfer the data to MySQL.

In the Application activity_main.xml contains the xml code for design layout of app and LoginActivity.java contains the back end code .

- Design Layout of the App:



- Java Code Snippet for Android App:

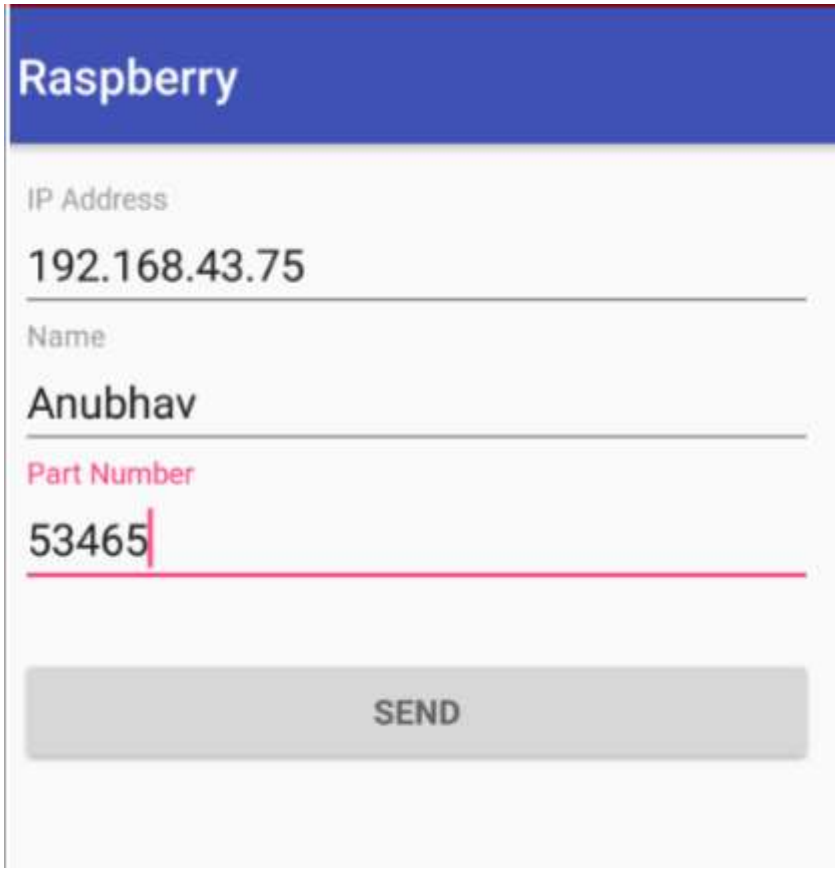
```
public class LoginActivity extends AppCompatActivity {

    EditText name;
    EditText part_no;
    Button bSend;
    TextView registerLink;
    RequestQueue queue;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        // Session Manager
        queue = Volley.newRequestQueue(getApplicationContext());

        name = (EditText) findViewById(R.id.name);
        part_no = (EditText) findViewById(R.id.partNumber);
        bSend = (Button) findViewById(R.id.bSend);


        name.addTextChangedListener(new TextWatcher()  {

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count)
{
            }

            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int
after) {
            }

            @Override
            public void afterTextChanged(Editable s)  {
                if (name.getText().toString().length() == 0){
                    name.setError("Please enter Name");
                }
                else{
                    name.setError(null);
                }

            }
        });

        part_no.addTextChangedListener(new TextWatcher()  {

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count)
{
            }

            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int
after) {
            }

            @Override
            public void afterTextChanged(Editable s)  {
```

```
                if (part_no.getText().toString().length() == 0){
                    part_no.setError("Please enter Part Number");
                }
                else{
                    part_no.setError(null);
                }

            }
        });

        bSend.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(final View v) {
                if(name.getText().toString().trim().equals("")){
                    Snackbar.make(v, "Please specify Name",
Snackbar.LENGTH_SHORT).show();
                }
                else {
                    if (part_no.getText().toString().equals("")) {
                        Snackbar.make(v, "Please specify Prt Number",
Snackbar.LENGTH_SHORT).show();
                    } else {
                        final String text = name.getText().toString().trim();
                        final String text2 = part_no.getText().toString();


                        StringRequest loginRequest = new
StringRequest(Request.Method.POST, "http://" +
((EditText)findViewById(R.id.ip)).getText() +
getResources().getString(R.string.Login_url),
                                new Response.Listener<String>()
                                {
                                    @Override
                                    public void onResponse(String response)
                                    {
                                        try{
                                            final JSONObject jsonResponse = new
JSONObject(response);
                                            boolean success =
jsonResponse.getBoolean("success");
                                            if (success) {



                                                Toast.makeText(getApplicationContext(),
"Request Successful", Toast.LENGTH_SHORT).show();


                                            } else {

                                                Toast.makeText(getApplicationContext(),
"Request Failed", Toast.LENGTH_SHORT).show();

                                            }
                                        }catch (JSONException e) {
                                            e.printStackTrace();

                                        }

                                    }
                                }, new Response.ErrorListener()
                                {
                                    @Override
                                    public void onErrorResponse(VolleyError error) {
                                        // error
```

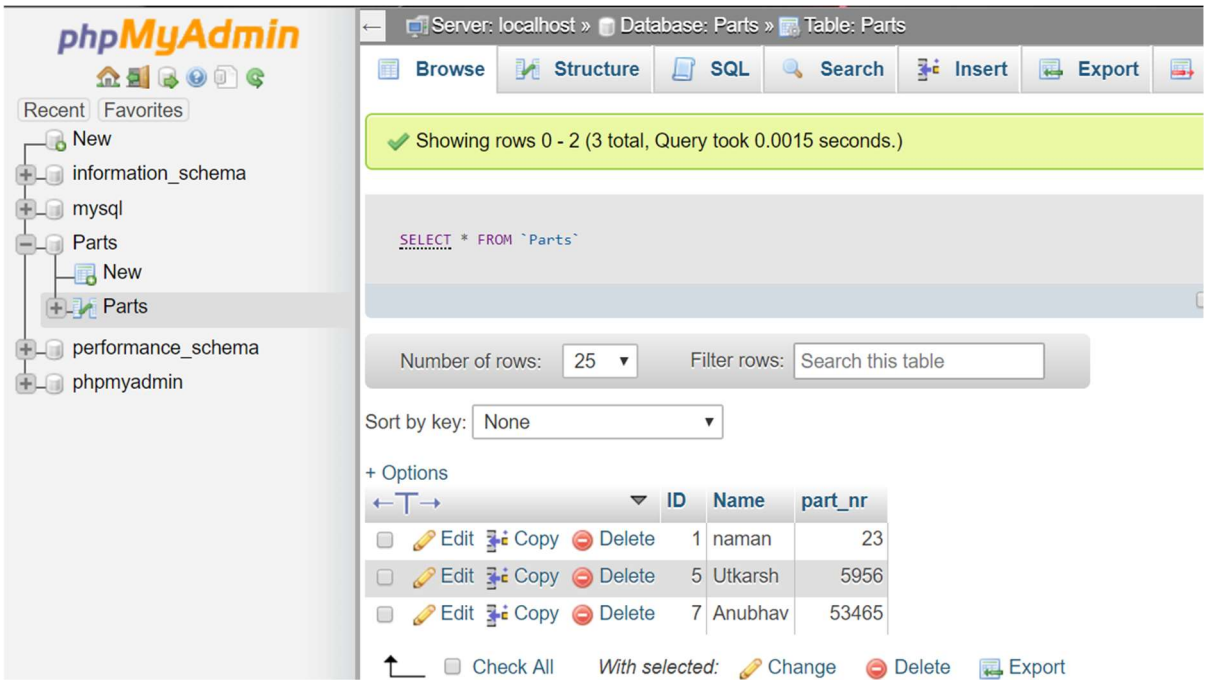
```
                                        Snackbar.make(v, "Couldn't connect to
internet",Snackbar.LENGTH_SHORT).show();
                                    }
                                }
                            ) {
                                @Override
                                protected Map<String, String> getParams()
                                {
                                    Map<String, String>  params = new HashMap<String,
String>();

                                    params.put("text", text);
                                    params.put("text2", text2);

                                    return params;
                                }

                            };
                            queue.add(loginRequest);
                        }
                    }

                }
            });

    }


}
```

- Snapshots : Transferring Of data between Android App and Raspberry Pi:

Android App



PHPMyAdmin web page hosted by a browser



This project is about transferring of data from android to raspberry pi. This provides with an idea of general communication between raspberry pi and Android App through a Wi-Fi network which enables transferring data over a longer distance. For Artificial Pancreas Application, it is required to retrieve and display the stored data in database in Android App for more robust analysis of data retrieved over a period of time.

4. BIBLIOGRAPHY

- Raspberry Pi Foundation , www.raspberrypi.org
- Raspberry Pi Android App Communication , www.projects.raspberrypi.com
- Android App Development Related Stuffs, www.github.com/namangl