Smart Pass:
LNMIIT Bus Pass System

Dhruvraj Singh Rawat
15uec022

Department of Electronics and Communication Engineering
The LNM Institute of Information Technology, Jaipur

May 2018

Outline

- Introduction
- Literature Survey
- Proposed Work
- Implementation
- Results
- Discussion and Future Work

Problem Statement

- In LNMIIT we always carry our Institute ID with us thus if we need to board a college bus from the main gate we have to show our bus pass and well as our college ID.
- We can make our bus pass redundant if we can make smart use of our college ID.
- In this project, I intend to build an RFID Powered Smart Pass Access Control System using Raspberry Pi microprocessor.
- We have already seen this kind of services implemented in big hospitals or companies and even in different colleges across the world, e.g, in National Ilan University, we might have seen how they used RFID based locks to restrict access to certain areas.
- The same system can be used in other systems like RFID based door lock or RFID based access management system and even in Smart Card payment systems.

Introduction:

Proposal

Proposal

- At present, there are generally two methods for access control: recording on the paper or managing with technology.
- However, managing with paper-based technology has a critical drawback, they can get lost, can get easily destroyed with external conditions and cost an environment damage with cutting down of tree for making pulp to produce paper.
- RFID technology is stronger in its ability to withstand harsh environment, fast access time are its key component.
- This project is clearly useful in the scenario of LNMIIT, where there is a huge rush outside the gate whenever the college bus arrives from the city to get their bus pass punched.
- Our college ID's are embedded with a RFID identification chip inside with a unique code allotted to all the students thus if we combine with server containing the details to process students faster.

Literature Survey

Existing Recent Solutions

- The paper from Robert Waszkowski and his colleges in the paper “**Evaluating the Impact of Testing Document Management System with RFID Tags Software on the Level of its Reliability**” attempts to describe a method of using formal evaluating the impact of testing of software on the level of its reliability. Presented method has been developed on the basis of formal model of software testing process that has been constructed in the paper for some assumed testing scheme. The method under the investigation has been proposed for the mean of planning the software testing process of document management system with RFID tags.
- Furthermore, paper from Mehak Tanveer from University of Gujrat showed the **use of RFID based technology on smart traffic management system using the Internet of Things** and a decentralized approach to optimize traffic on the roads and intelligent algorithms to manage all traffic situations more accurately. This proposed system is overcoming the flaws of previous traffic management systems. The system takes traffic density as input from RFID sensors which is then abstracted using different technique and sensors data, resultantly giving output as signals management.

Proposed Work

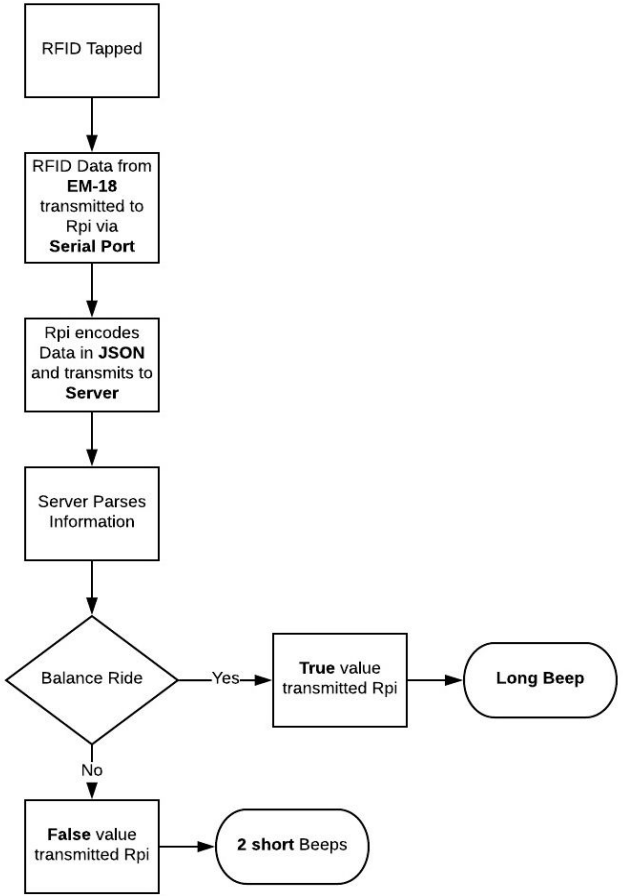
Proposed Solution

- Build a real-time system RFID powered smart pass access control system using EM-18 RFID Module and Raspberry-pi.
- Also, to implement a server in FLASK framework with dedicated SQL database maintaining all the user's information.
- Establish the interfacing between the Raspberry pi and the Remote Server.

Solution Control Flow

- Student taps its RFID card in the RFID scanner.
- The Information (unique rfid_no) is transmitted to the Raspberry pi via serial port.
- The Raspberry pi encodes the information (rfid_no) into the Json format and transmit it to the Server.
- The Server parses the information and check the statement balance regarding that rfid_no in SQL database and return a Boolean response signifying whether to grant access or not.
- The response is transmitted back to the Raspberry pi via WIFI and it parses the information and takes the appropriate step.

Proposed Work

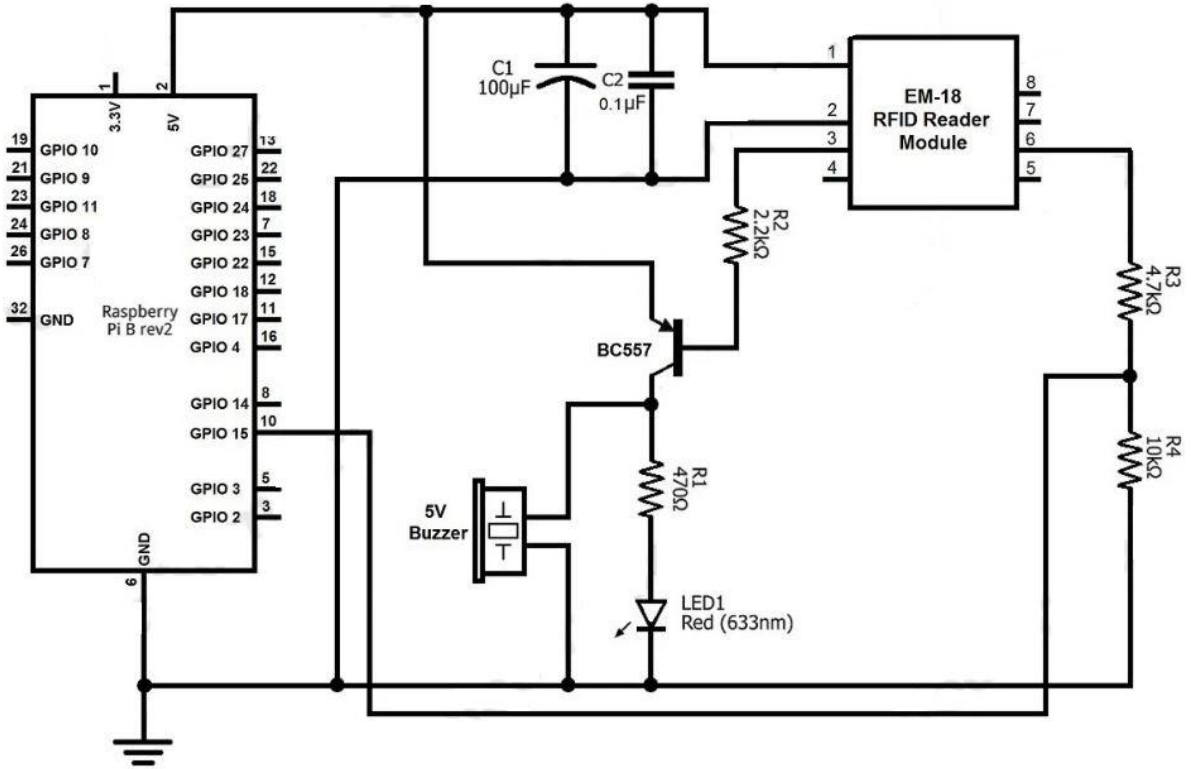


Control Flow Chart

Components

- **Raspberry Pi v3:** The Raspberry Pi 3 Model B+ is the latest product in the Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT.
- **EM - 18:** EM-18 RFID reader is one of the commonly used RFID reader to read 125KHz tags. It features low cost, low power consumption, small form factor and easy to use. It provides both UART and Wiegand26 output formats. It can be directly interfaced with microcontrollers using UART and with PC using an RS232 converter.

Proposed Work



Circuit Diagram

Technologies Used

Client Side

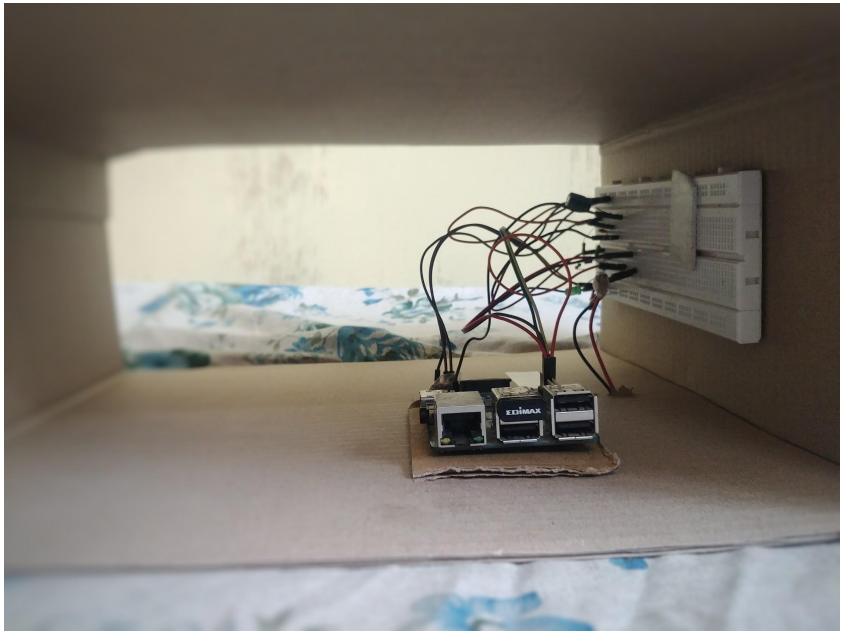
- **Python 2.7**
- ***Request*:** Http Library for Python
- ***Python Json*:** Encoder and Decoder for parsing Json requests &response

Server Side

- ***Python 2.7***
- ***Flask Framework***: Python micro web framework
- ***SQLAlchemy***: Python SQL database framework
- ***Bootstrap Framework***: WebUI creation

Simulation and Results

System Setup



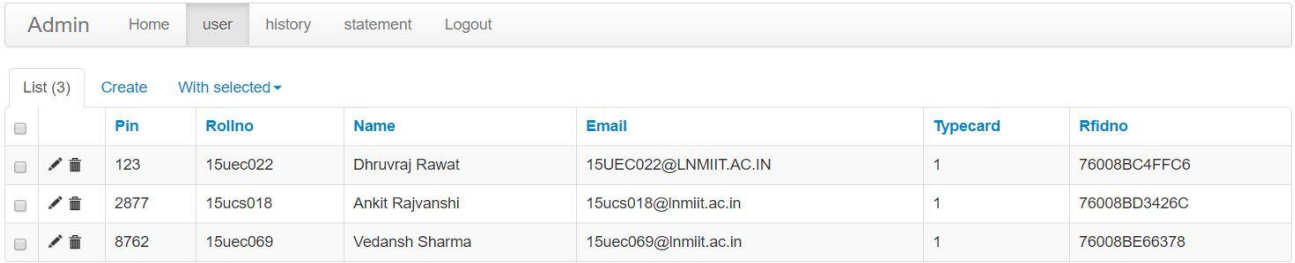
Database Creation Using SQLAlchemy

```
01. class user(db.Model):
02.
03.
04.     pin = db.Column(db.String(10))
05.     rollno = db.Column(db.String(10))
06.     name = db.Column(db.String(250), nullable=False)
07.     email = db.Column(db.String(250), nullable=False)
08.     typeCard = db.Column(db.String(5))
09.     rfidno  = db.Column(db.String(50))
10.
11.     id = db.Column(db.Integer, primary_key = True)
12.
13.
14. class history(db.Model):
15.
16.     id = db.Column(db.Integer, primary_key = True)
17.     name = db.Column(db.String(250), nullable=False)
18.     rfidno  = db.Column(db.String(50))
19.     day = db.Column(db.String(50))
20.     time = db.Column(db.String(50))
21.
22. class statement(db.Model):
23.
24.     id = db.Column(db.Integer, primary_key = True)
25.     balance_trip = db.Column(db.String(8))
26.     rfidno  = db.Column(db.String(50))
27.     name = db.Column(db.String(250), nullable=False)
```

Server API Ends for Communication with Client Side

```python
@app.route("/api_responce", methods = ['GET', 'POST'])
def api_responce():

    if request.method=='POST':

        rfidno=request.args.get('rfid_no')


        user_balance = db.session.query(statement).filter_by(rfidno=rfidno).one()
        user_info=db.session.query(user).filter_by(rfidno=rfidno).one()

        balance_available=int(user_balance.balance_trip)


        if balance_available>0:
            balance_available=balance_available-1
            user_balance.balance_trip=balance_available
            db.session.commit()

            day=datetime.datetime.now().strftime("%d/%m/%y")
            timee=time.strftime("%H:%M:%S")

            trip_update=history(name=user_balance.name,rfidno=rfidno,day=day,time=timee)
            db.session.add(trip_update)
            db.session.commit()

            mailer_server(user_info.email,day,timee,balance_available)

            output={
                'status':1,

            }
            return json.dumps(output)
        else:
            output={
                'status':0,
            }
            return json.dumps(output)



    elif request.method=='GET':

        response=make_response(json.dumps('GET is working',200))
        response.headers['Content-Type']='application/json'
        return response
```

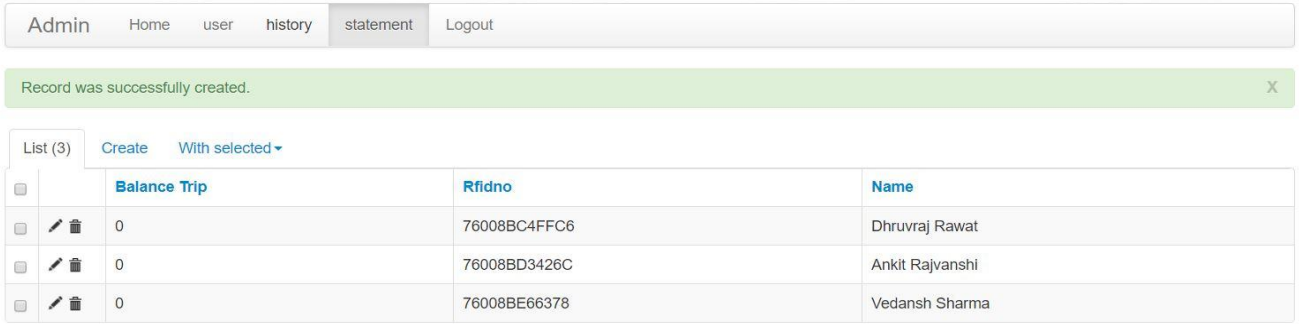
Results



In admin panel, showing User-Table

In admin panel, showing History Table

Results



In admin panel, showing Statement Table

Discussions and Future Work

Discussion on Results

Goals:

To implement a RFID based access control system powered by raspberry pi with server implementation that can securely verify each passing student in timely and fast fashion.

Functions:

- Retrieval RFID information from the user via sensor.
- Transmission of the information to the server via raspberry pi.
- On server, Validation of the information weather the RFID bearer is authorized to enter or not.
- Returning back of the output to the Raspberry pi.

Discussion on Results

Challenges Faced :

- Interfacing EM-18 with Raspberry Pi
- Establishing secure connection between Server and Client
- SSHing into the raspberry pi wirelessly
- Enabling the remote start of python script on raspberry pi

Quality of Results :

- Results shown by the proposed system are very good considering the components used.
- Latency in getting the response from server is also in few seconds.
- The susceptance range of rfid reader is less but it can be increase by using a powerful receiver.

Future Work

- A custom hardware-based implementation of the circuit can be done, by first prototyping it in FPGA then making an ASIC implementation.
- The server-side code can also be improved to detect fraud by using a Machine learning model.
- Furthermore, the RFID reading range can be increased by using the powerful reader by supply power via external power source.

Bibliography

· An intelligent management system based on RFID technology

https://ieeexplore.ieee.org/document/6011807/

· Interfacing EM-18 *RFID* reader with Raspberry Pi

https://electrosome.com/em-18-rfid-reader-raspberry-pi/

· Smart traffic management system using Internet of Things

https://ieeexplore.ieee.org/document/8323770/

· Flask framework Documentation

http://flask.pocoo.org/docs/1.0/

· SQLAlchemy Documentation for Database Creation

http://docs.sqlalchemy.org/en/latest/

Thank You.