PROJECT

INVOLVING - ARDUINO UNO

RFID READER

BLUETOOTH MODULE

ANDROID APP

BY NITIN GAURAV SINGH , B.tech (CSE) , 2nd year, IITR

MAHENDRA KUMAR , B.tech (CSE) , 2nd year, IITR

UNDER SUDIP ROY, PhD
Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology (IIT) Roorkee

INTRODUCTION

Technology changes the way we live our daily lives, the way we learn, and the way we use our faculties of attention .It has profound effects on our memories (particularly the short-term, or working memory), altering and in some cases impairing its function.
The implications of a poor working memory on our brain functioning and overall intelligence levels are difficult to over-estimate.

MOTIVATION

- Information overload makes it harder to retain information.
- Distraction makes it more difficult to form memories.
- Information overload causes us to lose sight of the big picture (and then the small picture).

Packing for the trip is too important to put off until the day before one leaves because if one is rushing around, he/she is more likely to forget things that will cost a lot of money to replace during the trip.

It's almost inevitable that you're going to forget something when you're going out somewhere, but save for personal identification, most can be replaced with little irritation or expense.

Whether its our first trip or the hundredth, it’s always helpful to have a rundown of what items we may want to pack so that we have a travel checklist.

PROBLEM STATEMENT

The main idea of our app is to ensure that we carry the essential things and not forget them. For instance, if we are going out somewhere and would like to have things like wallet, power bank,charger, earphones, camera and so on, with us, using this project we could know the things which we are carrying and the things we are missing without any effort of opening the bag and searching out for each and every item.This project almost reduces the effort required to zero and ensures that the things are handy whenever they are needed.

MAIN COMPONENTS USED

Arduino Uno board

RFID MFRC 522 reader

HC 05 Bluetooth module

Buzzer

Red LED

PLAN AND IMPLEMENTATION OF THE WORK

We started the project work on 29th November, 2016. Then we had very little idea about the arduino board and the components in the arduino kit.

So first we planned to work with the componenets that we got in the arduino kit and understand their use.

So we started learning about how to code in the arduino IDE and upload it to the board.

DAY 1

So our first ever code was to light up an LED.

As we finished the coding part of it, we went about to make the required connections on the board.

Although it took a while for us to understand the working of the bread board, we managed to do it. And finally, we made the proper connections to the LED(red), resistor and the board and successfully lit it up.

Then we improvised on it and added another LED (green) and lit it up along with the other one.

After that we modified the code for the LEDs to glow one after the other.

A traffic light controller, being the most basic thing to learn as far as connections to the board and its code is concerned ,was our motive to implement on the first day.

So, having added the red and green LEDs to the board already, we added a yellow LED to the bread board and made its connections to the arduino board too.

After adding the proper delays to the LEDs and the sequence in which they glow, we completed the traffic light controller.

DAY 2

We then thought of adding a push button to the existing traffic light controller that can be used by the pedestrian to stop the traffic for some time in order to cross the road. In other words, whenever the switch would be pressed, the red LED would go HIGH and the remaining would go Low for some amount of time.

Thus we added the switch and it worked well.

We also had a Buzzer in the kit. So we thought to use it to produce a sound whenever the switch is pressed i.e, whenever the pedestrian wants to cross.

So we added the code for the buzzer also, to send a high frequency tone for the same amount of delay the traffic light would be red, as the switch is pressed.

DAY 3

Now, we somewhat started to feel comfortable with the arduino IDE and the connections as we finished the traffic light controller essentially.

Then we went on to explore the other components present in the kit.

We worked with the temperature sensor and recorded the temperature on the serial monitor.

Then we also learnt about the servo motor and its use. We then connected it to our board and ran it.

DAY 4

After running the servo motor, we then decided to run the DC motor which is trickier than the servo motor as we had to deal with the back voltage of the DC motor which could have harmed the board.

So we planned to add a transistor and a diode in the circuit to run the motor properly .

After uploading the code and making the proper connections, we managed to run the motor also.

We then thought to add a code that would vary the speed of rotation of the motor. However, even after a lot of research and check, we were not able to do that.

As a matter of fact, since the DC motor would not be in our main project, we were not much worried about it.

DAY 5

Then we started working on the LCD display. We searched for the component all over the internet to gather information about it and almost everywhere it was asked to connect using a potentiometer.

We did not have the potentiometer so we searched for the exact use of this component for the case of LCD display.

And we found that it was just used to control the voltage supply to the LCD display which could in turn vary the brightness of the display.

So, since the potentiometer was not a necessary component for the connections, we connected the display without it and it worked out well.

We also had a RGB LED. So we connected the required pins and on uploading the code that would change the voltage supply to the R , G, and B pins, we managed to get different colours.

DAY 6 and 7

So, we had more or less worked with all the components of the kit.

So we started working on a basic android app that could send some data to the arduino via Bluetooth.

Our purpose was to turn ON and turn OFF an LED using the app.

This was just a small step involved in our main project.

This application details a method to utilize Bluetooth in order to communicate with the microcontroller. This method includes verifying Bluetooth support and

status, pairing and connecting to the microcontroller's Bluetooth module, and sending and receiving data serially.

There are several issues that must be overcome before the Android device can successfully transmit and receive data via Bluetooth. First, the Android must determine if it supports Bluetooth, and if it does, if Bluetooth is turned on. Then, it must pair and connect with the Bluetooth module on the Arduino. Finally, the Android must actually send and receive data.

In order for this method to work, the Android will have to be paired with the Arduino's Bluetooth module. Also, to ensure that the two devices connect correctly, unpair with all other Bluetooth devices.

So we successfully made an app that had two buttons that would turn On and turn Off the led respectively.

DAY 8 and 9

Now , we were able to send data to the arduino but we also had to receive data back to the phone.

So we started to work on this part. We planned to get the UID (Unique Identification Number) of an RFID tag to be displayed on our app beside the TAG. The RFID reader would scan the tag and via bluetooth module, it would send the UID of the tag to the phone.

But before that we had to first understand the RC 522 reader and how it would be connected.

We had to solder the pins into the reader and then we worked on it.We managed to get the UID printed on the serial monitor.

And we also successfully got the UID transferred to the app.

DAY 10

We had made two different apps, one for sending the data from app to the arduino and the other for receiving the information from the arduino to the app. So we then improvised and made an app that would send a particular data from the app to the arduino such that the reader would scan the tag and send the UID of the tag back to the app and display it there.

In other words, we just combined the two apps in to one.

DAY 11 and 12

After having made the app that could send and receive the data, we had to add a database into it which would store the items in it.

SQLite is that default feature which is used as a database and also used as a local database for any application.

We created a database that would be stored in the Phone's internal memory. Its container table has 4 columns that would store the Item's ID, Name, Tag, and whether it is available or not i.e. a "Yes" or a "No".

DAY 13, 14 and 15

- Final modifications in the app and arduino (adding a buzzer, notification sound etc)
- Handling of exceptions and some conditions
- Enhancing the layout of the app
- Making the code more efficient
- Adding toast messages

WORKING OF THE PROJECT

We make all the required connections and give a power supply to the arduino board.

Then we connect and pair our device with the bluetooth module HC05.

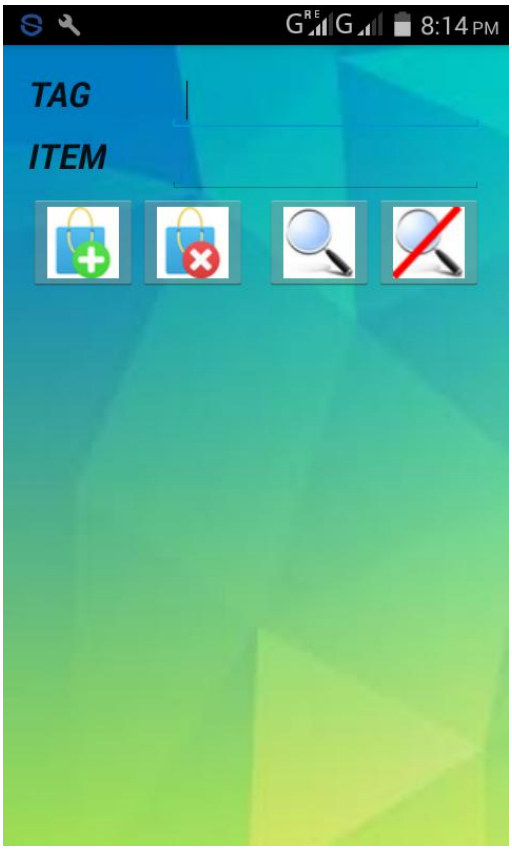
Once we get connected we can notice a change in the blinking rate of the LED present on the bluetooth module.

We also need to associate all the essential items with an RFID tag that could be sensed by the reader.

Now we are ready to use this system.

The name of the app is MITEMZ.

The basic Layout of the app is as follows-



TAG shows the TAG ID of the RFID card associated with the item.

To Add a new item in the list one needs to enter the name of the Item in the ITEM, the TAG ID of the tag associated with the item in the TAG, and then press the ADD button - >

To Delete an item , one can either just type the item's name in the ITEM and press the DELETE Button - >

or just look up for that particular item in the list and press the garbage bin button there corresponding to it.

SCANNING is switched ON when the scan ON button is pressed and switched OFF when the scan OFF button is pressed. When we press the scan ON button, the RFID receives the information that it needs to start scanning and the scan OFF button to stop the scanning.

When there's a notification sound beeped in the phone while scanning , it tells us that there is an item which the user has not included in the list .That item can either be named and added in the list or just be removed from the bag if it is not required.

The empty area is a LISTVIEW and shows the list of items that we need to carry.

Now, firstly we need to add items into the list. As we know, to add, we need to enter both the name of the item and the tag ID of the tag associated with it.

Some of the tags have their tag ID printed on them.In those cases we can simply enter that number.

But all tags don't have the tag ID printed on them. Then in those cases we take the help of the scan ON button that allows the reader to scan the tag and get its UID to be printed beside the TAG.Once we get that ID we can switch off the scan by pressing the scan off button.

This way is recommended also for the cases when we already know the tag ID because we may type it incorrectly.

To add any item,we need not always scan (unless we do not know the tag ID).

Once we enter the information and press the add button, the item adds into the database and is displayed in the listview below.

Similarly, we could add any number of items in the list and they would be displayed in a listview. We could scroll to view the items.

Before we leave for the journey, we place the bag containing all the items in front of the reader. Then we press the scan On button.

As the scan On button is pressed, the red LED turns high and the buzzer produces a sound indicating that the reader can now scan the bag.

If there is an item found in the bag by the reader that is not present in the list of required items, the notification sound of the mobile would play and would toast a message to either add that item in the list or remove it from the bag.

Scan time would depend on the volume of the bag because the reader scans only one item at a time. So it needs to cover every area in the bag. It hardly takes a second to scan an item.

As soon as the reader scans an item that is present in the list, a tick appeares beside that item in the list.

After scanning every area, we can stop the scan by pressing the scan Off button.

As we press this button, the led produces a sound different from the previous one and the LED turns low indicating that the reader can no more scan.

Also the items beside which the tick did not appear (since they were not present in the bag), have a cross beside them.

So,we need to go and find for those items having the crosses.

After finding them,we would place them in the bag and re-scan the bag just for satisfaction.

As we press the scan On button again, the previous crosses and tick dissapear since we would want to know the updated version of those.

And, as the scanning takes place and we see all the items having a tick beside them, we could stop the scan.

Thus we have ensured that we have the items that are required with us ; neither less nor more(no extra items also).

Any item could be added or deleted at any point of time. Also the list can be updated as per our wish.

DEMO VIDEO OF OUR PROJECT -

https://www.youtube.com/watch?v=7b322_8L7a4

FUTURE SCOPE OF THE WORK

- Multiple databases could be added for different journeys
- The storage of the data could be done on a server rather than in the phone's internal memory
- Syncing facilities could be added for all the devices using the app from a particular account
- Usage of a reader with a better range that could make the product feasible and efficient.

CONCLUSION

Thus we have a system that could help us plan the journeys and not miss out on carrying the required things and also help us avoid carrying the unwanted things.

The project was interesting since it involved the concepts of software and hardware.

Although this app is made for packing our things correctly, it could well be used in the security entrances or for taking the attendance. Fundamentally, they are the same. Just the difference is that instead of "items" being associated with the tag, there would be people, cars, ID cards etc associated with the tag.

REFERENCES

Arduino board -

http://www.makeuseof.com/tag/arduino-traffic-light-controller/

Bluetooth -

https://developer.android.com/guide/topics/connectivity/bluetooth.html

Data transfer using bluetooth -

http://www.egr.msu.edu/classes/ece480/capstone/spring14/group01/docs/appnote/Wirsing-SendingAndReceivingDataViaBluetoothWithAnAndroidDevice.pdf

RFID -

http://www.instructables.com/id/Arduino-and-RFID-from-seeedstudio/?ALLSTEPS

SQLITE -

http://www.w3schools.com/sql/sql_intro.asp

https://www.youtube.com/watch?v=PrMhjEh2ySk&index=54&list=PL6qx4Cwl9DGBsvRxJJOzG4r4k_zLKrnxl

Listview android studio-

https://www.youtube.com/watch?v=8K-6qdTlGEA