ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 2, February 2018

An Asset Finder Using Arduino Pro Mini

Saswati Bhattacharya [1], Haymon Rai [2]

Assistant Professor, Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology

Majitar, Sikkim, India[1]

UG Student, Department of Computer Science and Engineering, Sikkim Manipal Institute of Technology

Majitar, Sikkim, India[2]

ABSTRACT*:* Portable devices are all that technology is trying to make nowadays for the ease of using and carrying them anywhere the user wants them. This paper proposes an approach of developing a portable finding device that can be attached to anything that we use daily and keeps misplacing for eg- keys, wallet, books, etc. The proposed device can be attached to anything that the user wants to find or misplaces frequently. This project will use an android application that will help us to find the device. A Light Emitting Diode (LED) and a buzzer will be connected to the device which will blink and produce a sound when the user wants to find the thing he/she has misplaced. To find the misplaced thing, user has to use the android app which will connect the mobile app with the device via Bluetooth and the user can press the 'FIND' button to locate the device. When the 'FIND' button is pressed, the device will blink and produce a sound, then the user can follow the sound and find what he/she is looking for. It will also notify the user whenever the android app is open and the Bluetooth goes out of range. This way, a user can know that his/her device is going out of range and they can immediately look for it.

KEYWORDS: Portable finding device, Android application, Light Emitting Diode (LED), Bluetooth.

I. INTRODUCTION

A portable device is a hand-held or wearable device, For example, the walkie-talkie is a device that is hand-held when in use, and wearable when not in use. Portable telephones (cellular telephones) are also carried, or worn, on a belt, or in a pocket. Portable devices have also become usable when worn. Portable devices are all that technology is trying to make nowadays for the ease of using and carrying them anywhere the user wants them.

As we all know misplacing things is a common habit and people often get late or leave some tasks incomplete because of the things they misplace and cannot find. In case of a mobile phone, it can be dialed and found easily if misplaced but same thing cannot be done for every daily use things. It is a common thing for a person to misplace their belongings and not find them. The existing finders or trackers are limited to iOS with their battery being irreplaceable at most times.

The systems that runs under android have non-rechargeable batteries and the notification sounds produced by them are not loud enough with some of the applications having a time constraint of 15 seconds which sometimes may not be enough for the user to find the misplaced asset. Also those devices that are available do not give any notification to the user when the devices go out of certain range. We can only use the device to track the asset after the asset has been stolen or misplaced. An android application and a portable finder can be attached to the user's belonging that he/she wants to find when misplaced. The user will open the mobile app and connect the app and finding device via Bluetooth. After the connection is made, the user will use the app to find the misplaced object using the sound of buzzer and light from LED present in the finding device saving the valuable time of the user. To prevent theft within the Bluetooth range, if the device goes out of Bluetooth range, a notification will be sent to the user telling them that their device has gone out of range.

DOI: 10.15680/IJIRCCE.2018.0602029

II. RELATED WORK

Finding an asset was a major problem in the field of electronic and computer industry. Many hardware solutions are increasingly being used to provide a solution for the given problem. It is also necessary that the reliabilityof system should be measured and evaluated. Over the past 20 years numerous many papers have been published on automation system. In Bluetooth based home automation system the home appliances are connected to the Arduino BT board at input output ports using relay. The password protection is provided so only authorized user is allowed to access the appliances. In Zigbee based home automation system, the device performance is recorded and stored by network coordinators. The GSM based home automation system uses transducer which convert machine function into electrical signals which goes into microcontroller. The Home automation using Android ADK establishes the connection between the Android device and ADK. Another solution given by Raspberry pie for home automation with wireless sensors and smart phone guarantees to be an efficient platform for implementation powerful, and economic smart home automation.

III. PROPOSED SYSTEM

This paper aims at developing a portable asset finder that can be attached to any frequently misplaced things. The proposed work contains the finding device as the hardware and a mobile android application using which we can find the misplaced thing. Whenever a user wants to find the misplaced thing he/she will open the android app which will have a ‘FIND’ button. The finder will have a LED and a speaker. The user can attach this finder to his/her frequently used thing and when misplaced, the user will open the android application, he/she will connect the phone with the finder via Bluetooth and press the ‘FIND’ button. After pressing the button the finder will produce a sound along with LED blinking in case it is dark. The user can follow the sound and look for the blink to find his/her misplaced object. After he/she finds the thing, user will press the ‘FOUND button which will stop the sound and blinking.

Further, to prevent theft when the user is busy in some task and the device is in another room, a feature is available for the android app. The feature is that the user will get a notification via the android app if the device goes out of Bluetooth range. So the user can work without having to worry about their valuable asset.

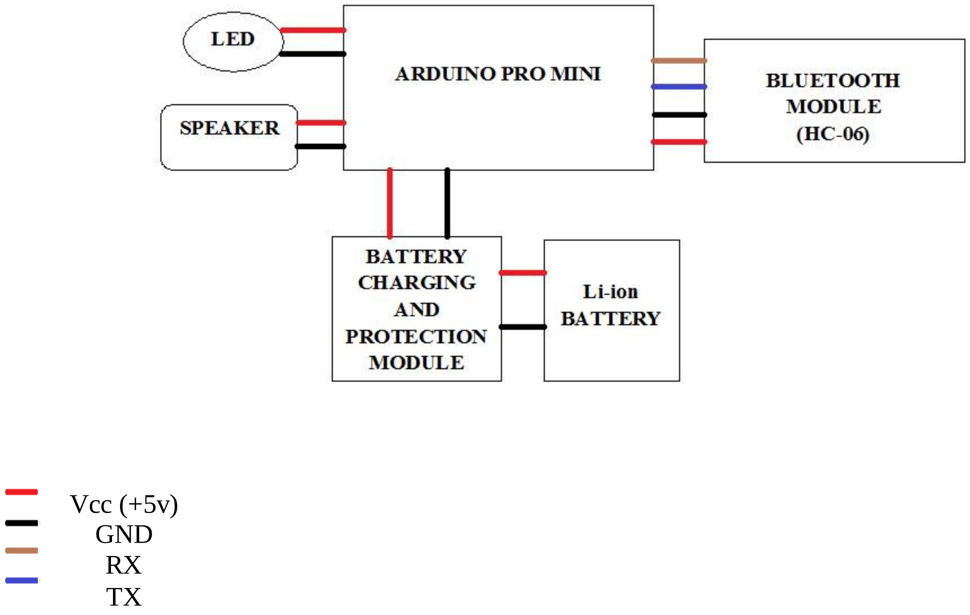


Figure.1 Block diagram of the proposed system

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 2, February 2018

The figure shows the simple architecture of the asset finder. It generally shows the placement and the connection of the LED, buzzer and the Bluetooth module with the arduino pro mini which is the heart of the device and also with the power supply which powers up the finder.

The Arduino pro mini is the heart of the asset finder which is powered by 3.7V power supply. The power supply consists of a Li-ion battery which has an output voltage of 3.7 V, a battery charging plus protection module. The battery charging and protection module is used for charging the battery and also to protect the battery from getting damaged due to overcharging and over discharging. The battery is charged up using the mini USB port available which provides 3.7V 1A supply to charge up the battery. The Arduino pro mini uses Atmega168p microcontroller.

IV. IMPLEMENTATION USING ARDUINO PRO MINI

The Arduino Pro Mini is a microcontroller board based on the ATmega168. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. A six pin header can be connected to an FTDI cable or Sparkfun breakout board to provide USB power and communication to the board. The Arduino Pro Mini is intended for semi-permanent installation in objects or exhibitions. The board comes without pre-mounted headers, allowing the use of various types of connectors or direct soldering of wires. The pin layout is compatible with the Arduino Mini.

There are two version of the Pro Mini. One runs at 3.3V and 8 MHz, the other at 5V and 16 MHz.

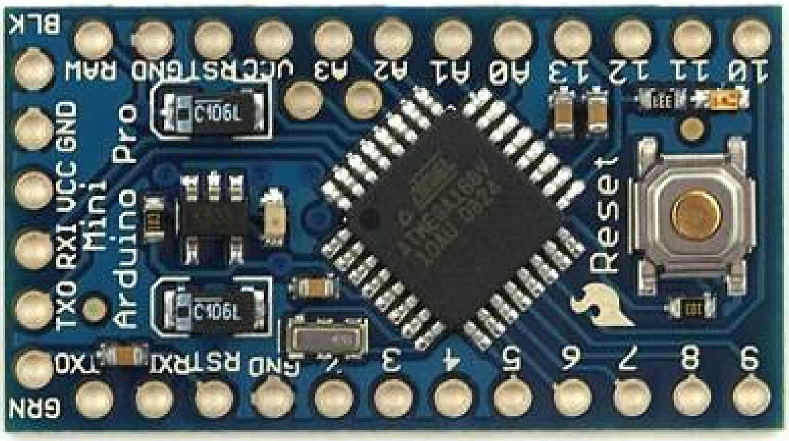

BLK
RAW GND RST VCC A3 A2 A1 A0 13 12 11 10
TXO RXI VCC GND
Arduino Pro Mini
Reset
TXO RXI RST GND 2 3 4 5 6 7 8 9
GRN

Figure.2 Arduino Pro Mini

The arduino pro mini is the heart of the asset finder which is powered by 3.7V power supply. The Arduino Pro Mini can be powered with an FTDI cable or breakout board connected to its six pin header, or with a regulated 3.3V or 5V supply (depending on the model) on the VCC pin. There is a voltage regulator on board so it can accept voltage up to 12VDC. If you're supplying unregulated power to the board, be sure to connect to the "RAW" pin on not VCC.

The power pins are as follows:

- RAW: For supplying a raw voltage to the board.
- VCC: The regulated 3.3 or 5 volt supply.
- GND: Ground pins

Each of the 14 digital pins on the Pro Mini can be used as an input or output, using pinMode(), digitalRead() and digitalWrite() functions. They operate at 3.3 or 5 volts (depending on the model). Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the TX-0 and RX-1 pins of the six pin header.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Pro Mini has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). Four of them are on the headers on the edge of the board; two (inputs 4 and 5) on holes in the interior of the board. The analog inputs measure from ground to VCC. Additionally, some pins have specialized functionality:

- I2C: A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the Wire Library. There is another pin on the board:
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

4.1 Communication:

The Arduino Pro Mini has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 provides UART TTL serial communication, which is available on digital pins 0 (RX) and 1 (TX). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board via a USB connection.

A SoftwareSerial library allows for serial communication on any of the Pro Mini's digital pins. The ATmega168 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

4.2 Programming

The Arduino Pro Mini can be programmed with the Arduino software, Arduino IDE. The ATmega168 on the Arduino Pro Mini comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).You can also bypass the bootloader and program the ATmega168 with an external programmer.

4.3 Automatic (software) reset:

Rather than requiring a physical press of the reset button before an upload, the Arduino Pro Mini is designed in a way that allows it to be reset by software running on a connected computer. One of the pins on the six-pin header is connected to the reset line of the ATmega168 via a 100nF capacitor. This pin connects to one of the hardware flow control lines of the USB-to-serial convertor connected to the header: RTS when using a
FTDI cable, DTR when using the Sparkfun breakout board. When this line is asserted the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of the reset line can be well-coordinated with the start of the upload.

This setup has other implications. When the Pro Mini is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Pro. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

4.4 Bluetooth module (HC-06)

Bluetooth can wirelessly connect devices together. Here Bluetooth SerialTransceiver Module version V2.0+EDR with backboard is used which works for Bluetooth TTL transceiver module that allows the target device to both sends and receives the TTL data. The key interface on the baseboard is the master mode button and can be controlled by high level from external MCU, and then this module will search again automatically.

In order to use Bluetooth features in our application, we must first enable the permission for accessing Bluetooth to perform any Bluetooth communication, such as requesting a connection, accepting a connection, and transferring data.

The Android platform includes support for the Bluetooth network stack, which allows a device to wirelessly exchange data with other Bluetooth devices. The application framework provides access to the Bluetooth functionality through the Android Bluetooth APIs. These APIs let applications wirelessly connect to other Bluetooth devices, enabling point-to-point and multipoint wireless features.

If we want our app to initiate device discovery or manipulate Bluetooth settings, we must declare the BLUETOOTH_ADMIN permission in addition to the BLUETOOTH permission. Most applications need this permission solely for the ability to discover local Bluetooth devices. The other abilities granted by this permission should not be used, unless the application is a "power manager" that will modify Bluetooth settings upon user request.

- Setting Up Bluetooth

  Before our application can communicate over Bluetooth, you need to verify that Bluetooth is supported on the device, and if so, ensure that it is enabled.

  If Bluetooth is not supported, then you should gracefully disable any Bluetooth features. If Bluetooth is supported, but disabled, then you can request that the user enable Bluetooth without leaving your application. This setup is accomplished in two steps, using the `BluetoothAdapter`:

- Get the BluetoothAdapter.

  The BluetoothAdapter is required for any and all Bluetooth activity. To get the BluetoothAdapter, call the static getDefaultAdapter() method. This returns a BluetoothAdapter that represents the device's own Bluetooth adapter (the Bluetooth radio). There's one Bluetooth adapter for the entire system, and the application can interact with it using this object. If getDefaultAdapter() returns null, then the device does not support Bluetooth and it is not possible to develop an app that uses Bluetooth.

  For example:

  ```
  BluetoothAdapter mBluetoothAdapter=BluetoothAdapter.getDefaultAdapter();
  if(mBluetoothAdapter==null)
  { //Device does not support Bluetooth }
  ```

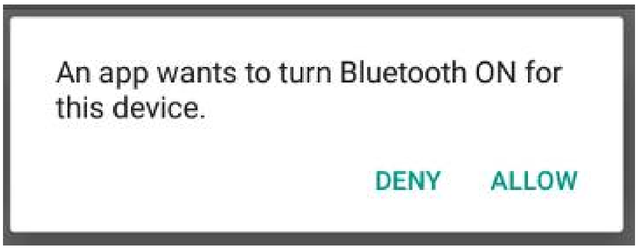
- Enable Bluetooth.

  Next, we need to ensure that Bluetooth is enabled. Call isEnabled() to check whether Bluetooth is currently enabled. If this method returns false, then Bluetooth is disabled. To request that Bluetooth be enabled, call startActivityForResult(), passing in an ACTION_REQUEST_ENABLE intent action. This call issues a request to enable Bluetooth through the system settings (without stopping your application).

  For example:

  ```
  if(!mBluetoothAdapter.isEnabled())
  {IntentenableBtIntent=newIntent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
  startActivityForResult(enableBtIntent,REQUEST_ENABLE_BT); }
  ```

  A dialog will appear requesting user permission to enable Bluetooth, as shown in Figure 1. If the user responds "Yes," the system will begin to enable Bluetooth, and focus will return to your application once the process completes (or fails).



An app wants to turn Bluetooth ON for this device.

Fig 3: The enabling Bluetooth Dialog

The REQUEST_ENABLE_BT constant passed to startActivityForResult() is a locally defined integer that must be greater than 0. The system passes this constant back to you in your onActivityResult() implementation as the requestCode parameter.
If enabling Bluetooth succeeds, your activity receives the RESULT_OK result code in the onActivityResult() callback. If Bluetooth was not enabled due to an error (or the user responded "No") then the result code is RESULT_CANCELED.

- Checking Bluetooth Connection:

The project requires the app to check the state of Bluetooth connection and take actions when Bluetooth gets disconnected. For checking the connection state of Bluetooth, i.e., whether the devices are still connected via Bluetooth or not, we need to give the following actions in the AndroidManifest.xml file.

```
<action android:name="android.bluetooth.device.action.ACL_DISCONNECTED" />
<action android:name="android.bluetooth.device.action.ACL_CONNECTED" />
```

These two actions will give us the facility to check the Bluetooth connection while developing the Android app.
We use the BroadcastReceiver to receive the actions of Bluetooth, ACTION_ACL_CONNECTED and ACTION_ACL_DISCONNECTED are used for checking Bluetooth connection and taking appropriate actions. For eg:

```
if (BluetoothDevice.ACTION_ACL_CONNECTED.equals(action))
{
Toast.makeText(getApplicationContext(),"BluetoothConnected",
Toast.LENGTH_SHORT).show(); //Bluetooth Connected will be displayed
}
else if (BluetoothDevice.ACTION_ACL_DISCONNECTED.equals(action))
{
Toast.makeText(getApplicationContext(),"Bluetooth                 Disconnected",
Toast.LENGTH_SHORT).show();//Bluetooth Disconnected will be displayed
}
```

Using these two actions, the app can be developed to check the connection of Bluetooth and notification can be sent when the connection is lost.

4.5 Light Emitting Diodes (LEDs):

It is a p-n Junction diode and emits light when an electric curresnt passes through them. LED should not be connected to a power supply directly, it must have resistor in series to limit the current to a safe value, for quick testing purposes a 1k resistor is suitable for most LEDs if your supply voltage is 12V or less.

V. RESULTS AND OUTPUT

The system makes use of an embedded system based on the Bluetooth technology. The arduino pro mini which acts as the heart/brain of the asset finder controls the Bluetooth module, buzzer and the LED connected with it. The main purpose of the asset finder is to find misplaced items instantly. Additionally, it contains a mobile android application which is used to find the misplaced item and also if the asset finder goes out of the Bluetooth range, a notification is sent to the user via the android app.

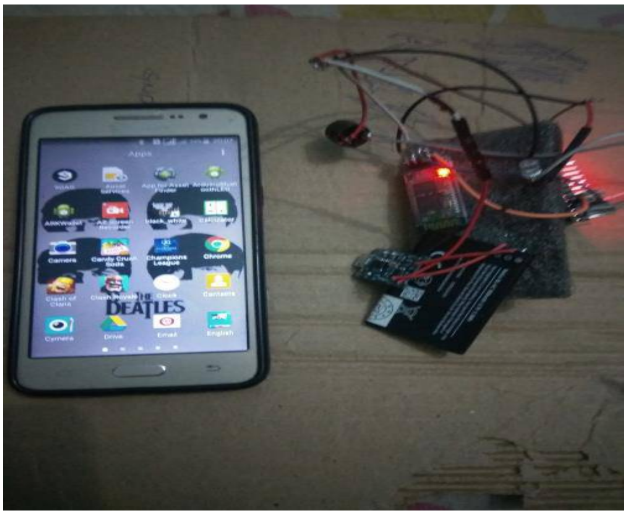


Fig. 4 Bluetooth module blinking before connecting with the Android app

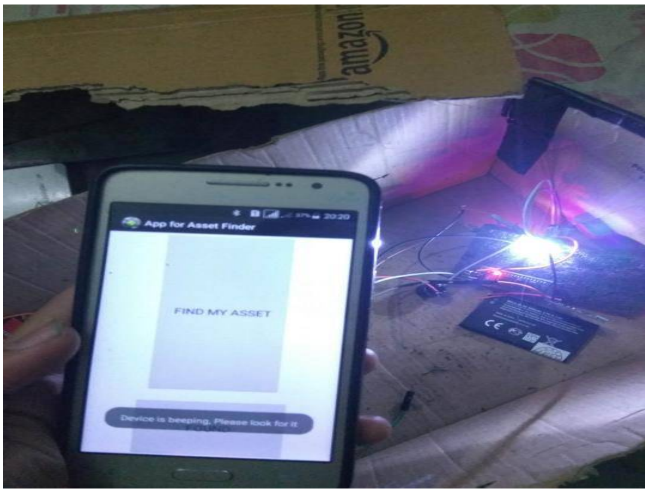


Fig.5 LED starts blinking and buzzer beeps after FIND MY ASSET button is pressed

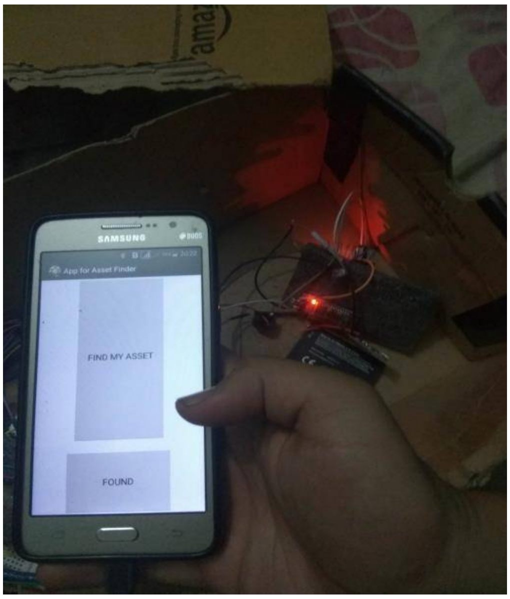
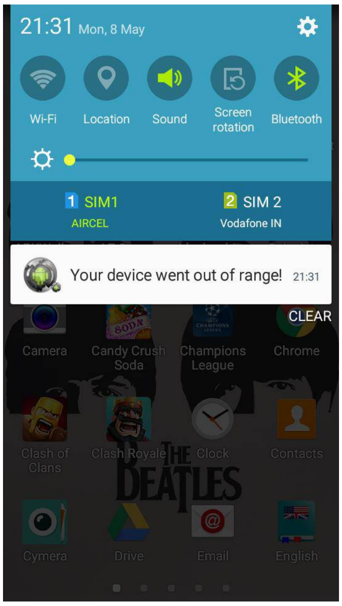


Fig. 6 Device stops blinking and beeping after FOUND button is pressed



Your device went out of range! 21:31

Fig. 7 Notification arrives within 2-3 seconds

V. CONCLUSIONS

A simple and portable asset finding device has been successfully designed and constructed. All the features of the asset finder i.e. locating the asset within the room and notifying the user when Bluetooth goes out of range have been proven to be very useful and effective in day to day life.

This finder has many advantages such as saving the user's time, valuable assets, preventing theft and it is easy to use too. Upgrading this setup is very easy which makes it open to future upgradation that makes it more efficient. The user interface that is provided, is quite simple and user friendly. This project is limited to bluetooth range only and can be further enhanced by the use of PIR sensors to make the user's asset anti-theft. A GPS module can also be added to enable tracking mechanism in the asset finder.

REFERENCES

1. Yi-Jen Mon,"Bluetooth based LED Control For Arduino Test Platform By Using Mobile APP" by, International Journal of Scientific & Technology, June 2015
2. Arduino Bluetooth Controller, Developer- broxcode, Store- Play Store, Developed in- 2015
3. https://www.arduino.cc/en/tutorial/blink
4. https://www.arduino.cc/en/Reference/Tone
5. http://forum.arduino.cc/index.php?topic=369356.0
6. https://developer.android.com/guide/topics/ui/notifiers/notifications.html
7. Brian Wirsing,"Sending and Receiving Data via Bluetooth with an Android Device" ,March 26,2014.
8. http://developer.android.com/sdk/index.html
9. Pascal Knierim, Jens Nickels, Steffen , Musiol, Bastian Könings, Florian Schaub, Björn Wiedersheim, and Michael Weber "Find My Stuff: A Search Engine for Everyday Objects" , Institute of Media Informatics, Ulm University, May 2012.