ISSN (ONLINE): 2395-695X
ISSN (PRINT): 2395-695X

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3, Special Issue.24, March 2017

GO THROUGH ME PRO SMART LOCK

[1] D.POORNAKUMAR, [2] K.SEVVANTHY, [3] P.SHREEMATHY
[4] U.C SNEKHA, [5] N.SARAN DEVA
[1], Assistant Professors, [2][3],[4],[5] UG Scholars, K.S.Rangasamy College of Technology,Tiruchengode.

Abstract— **Mobile devices have been integrated into our everyday life. Consequently, home automation and security are becoming increasingly prominent features on mobile devices. In this paper, we have developed a security system that interfaces with an Android mobile device. The mobile device and security system communicate via Bluetooth because a short-range-only communications system was desired. The mobile application can be loaded onto any compatible device, and once loaded, interface with the security system. Commands to lock, unlock, or check the status of the door to which the security system is installed can be sent quickly from the mobile device via a simple, easy to use GUI. The security system then acts on these commands, taking the appropriate action and sending a confirmation back to the mobile device. The security system can also tell the user if the door is open. The Main Door Security System using SMS propose the uses of Short Message Service Text Messaging (SMS) as a mechanism to control the system via mobile phone to lock and unlock the door user enter the pattern or password on the Smartphone, the door will immediately opened after going through the process of identification by the system. LCD will display the status of door.unlock the door user enter the pattern or password on the Smartphone, the door will immediately opened after going through the process of identification by the system. LCD will display the status of door.A liquid-crystal display (LCD) is a flat-panel display that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly.**

Keywords:NFC, ***Android SDK, Arduino Firmware.***

I. INTRODUCTION

In day to day life security of any object or place is plays a major role. This project has considered about to create a secure access for a door which needs a pattern to open the door. In earlier days traditional lock system using mechanical lock and key mechanism are being replaced by new advanced technique of locking system. These techniques are an integration of mechanical and electronic devices and highly intelligent. The main purpose of this project is high security because in olden days the door lock system was very simple and it can be easy to break. This android lock system demonstrates a pattern based door lock system. wherein the pattern is entered, the door is opened and the concerned person is allowed to access to the secured area. Again if another person arrives it will ask to draw a pattern. If the pattern is wrong, the door would remain closed. The android lock is not a normal lock because it cannot be easily accessed by unauthorized person. The door can be accessed only by the authorized person. If any unknown person tries to access the door, the security system in the door will give alert to us and the unknown person was easily identified.

Today, the modern home integrates many conveniences of technology. Technology-based home security and automation have become commonplace. Advances in mobile technology and cloud computing allow for the management of many different aspects of the home through an internet connection offering considerable flexibility for users, such as Yale's Locks & Hardware new device. This new lock exploits Near Field Communications (NFC) to secure a home through a set of mobile keys. This system is the first to be integrated with

NFC enabled mobile devices. The most important part of every home security will be the users to enter and go out the door careful identification The way we manage a entrance guard are many, some use a password, some RFID sensor or fingerprint way, and face recognition methods. Here we are using pattern and password.

This paper will discuss the development of a security system that integrates with an Android mobile device using Bluetooth as a wireless connection protocol. Android OS is currently the lead on mobile market. This proposed system allows a user to lock or unlock a door a short range from the door. The application was designed to allow the user to also check the status of the door. The mobile device requires a password to increase the security of the system. The hardware on the door uses a microcontroller to control a linear actuator that acts as the locking mechanism. The Bluetooth protocol was chosen as a communications method because it is already integrated into many Android devices and is secured through the protocol itself. It also fit well into the design requirements of the project for a short range, wireless connection method. Bluetooth allows for the setup of a "piconet" that can be used to communicate with a device that can be up to 100 meters away. The protocol incorporates data encryption for security and interference avoidance.

II. SYSTEM DESIGN

A.Bluetooth on Android Mobile Devices

Bluetooth is a wireless communications system intended to replace cables on devices such as phones and other mobile devices. It originally was developed to replace serial connections via RS-232 cables.It is meant to unify communication protocols over short range into a single protocol. Bluetooth technology features low power consumption, low cost, and security. It operates in the ISM (Industrial Scientific Medical) Band of 2.4-2.83 GHz. Bluetooth has three different classes based on power consumption and effective range. The Bluetooth protocol allows for seven devices to be linked to a master device in a piconet based on a master/slave relationship.

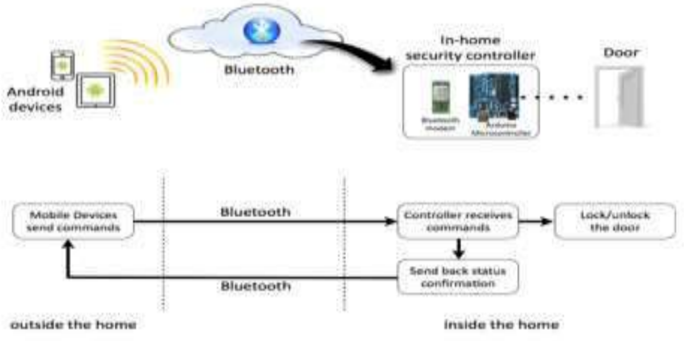

Fig 1. A diagram of home security using Bluetooth on Android mobile devices

While a piconet acts as an infrastructure mode network, multiple piconets can create an ad hoc network because the master of one piconet can be a slave in another. To avoid interference and collisions, Bluetooth uses an adaptive pseudo-random frequency hopping technology that is synchronized to the master's clock. Bluetooth offers four security modes based on use of a mutual pin entered into both devices. Authentication and encryption/decryption keys are made after devices are paired with the appropriate pin. Bluetooth development capability was officially added to the Android operating system with the release of the Android 2.1 API. Before this release, there was an API add- on for developers who wished to use Bluetooth in their applications, but has fallen out of use in favor of the official Bluetooth API software. To use Bluetooth in an application, a developer must declare the Bluetooth and Bluetooth ADMIN permissions in the application. Users will be notified of these permissions before download and must allow these permissions to continue downloading the application.

B.In-home Security Controller

Fig. 1 illustrated the diagram of a home security system using android mobile devices. This project called for an Android smartphone or tablet with Bluetooth capabilities and a microcontroller security interface on the door to be secured. The testing device used was the Motorola Backflip running the Android 2.1 update, a personally owned device that was on hand. The in-home security interface of the door was made of several major components. An Arduino Mega2560 microcontroller was chosen for this project for the ease of programming and ability to quickly prototype. The Mega uses the Atmel ATmega2560 chip. A BlueSmirf gold Bluetooth transceiver[8] was chosen for communication because it came pre-packaged on a break-out board. As illustrated in Fig. 2b, door position was determined by use of a magnetic switch similar to those found on window alarm systems that was connected to the Arduino board. In addition to using the Android device to lock or unlock the door, a switch that used a physical key was installed in the door jamb along with status lights of the system. The locking mechanism itself was a linear actuator, originally purposed for use in the power locking mechanism of a car, and a set of transistors to reverse the voltage polarity and lock/unlock the door.

C.Android SDK and Arduino Firmware

Android uses a Java based language. To develop an Android app, a tool named Eclipse is required. When creating a new application for the Android, the platform version must be selected, e.g. 1.5, 1.6, 2.1, or 2.2. However, the version can differ on the Android device depending on which Android device is being used. The three main components required in the creation of the Android app are: the java file, which is a file that contains all code required for completion of desired tasks and functions; an xml file, which contains the layout for how the application will look to a user; and a resource folder, which contains all images, sounds, and graphics files needed for the application. For Arduino firmware, the IDE is provided as the open-source by the company [10]. The tool can run on multiple platforms, e.g. Windows, OS X, and Unix. The language is a Wiring-based language which is similar to C/C++ style. Two programs were created for this project that had to be able to communicate with one another via Bluetooth channel. One was on the Android mobile device, and the other was on Arduino board, inside the in-home security controller.

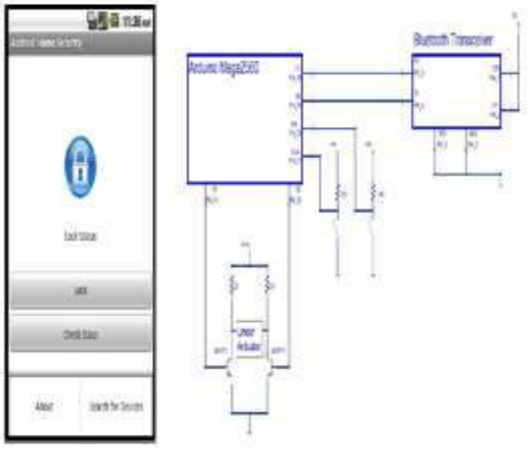

Fig 2. (a) screenshots of Android mobile application for home security,(b) The wiring circuit of in -home security controller

They needed to be able to communicate with each other in a secure manner over short range. Bluetooth fulfills both these requirements and was used as the communication protocol for the project. The Android app was created in two major development steps. The first step had the MAC address of the microcontroller directly coded into the app for initial testing, and the second was an improvement of the first in that it allowed a user to search for the device, which becomes the final design. The basic steps for connecting to Bluetooth were the same for both versions of the application. To connect to Bluetooth, first one must connect to the Bluetooth radio by calling it as an adapter in the app source code. Then, once a device's address has been obtained, a variable associated with the device is created. This device variable allows for the creation of a socket, similar to what is seen in TCP connections. The socket is yet another object in the program, and has to be connected before communications can occur. The input stream reader and output stream writer objects were used to read and write data to the other program. The snippet code of how to connect to a Bluetooth socket in Android devices is given below.

```
//after you get the address, connect to the device if
(address.length()>0) {
stopFlag = true;
   //Create a device with a certain address
   device = jnBluetoothAdapter.getRemoteDevice(address);
  //Creates connection
  try {
       //Create a socket to the device
   btsocket=
      device.createRfcommSocketToServiceRecord(MY_UUID);
  } catch (IOException e) {

block e.printStackTrace();            // TODO Auto-generated catch
  }
  jnBluetoothAdapter.cancelDiscovery(); }
```

As shown in Fig. 2a, the finished application had two buttons and two menu options. The buttons were used to unlock/lock the door and to check the status of the door. One menu option allowed the user to search for devices and the other gave more information about the application. An image of a lock that changed based on door status was also put on the application to give a visual representation of the last status of the lock. Fig. 4 shows the procedure on Android devices for checking the status and waiting for reply. The application would take two steps to lock/ unlock the door or one to check the status. When the lock/unlock button was pressed, the app would first send a check message to the door and wait for a response. The door would send back one of three responses, "locked" if the door was locked, "unlocked" if the door was unlocked, or "door" if the door was ajar. The app would then either alert the user by a toast popup message that the door was open or lock/unlock the door. The door was locked or unlocked by the commands "lock" and "unlock" respectively. Then the status of the door would be checked again to confirm the appropriate action had been taken. If the status button was pushed, then only the initial check of the door's status would be performed.

The door side programming, Arduino firmware, was by comparison much simpler because there was no graphic user interface to work with. The entire user interface consisted of the door, a keyhole, and two LEDs. One LED was a power indicator, and the other was a door ajar indicator. This program continually monitored the door ajar switch when it did not receive any commands from the app. When a command was received and the door was confirmed as closed, the appropriate action would be taken. The system would either ignore the command if it was to already in the state requested by the app or it would actuate the locking pin appropriately. This is done by connecting the transistor array's bases to two pins on the Arduino. Initially, both inputs are held low, then one is put to five volts for 500 milliseconds and returned to low. The input that is put to high depends on what action is required. After every request by the app, a confirmation message is sent after any actions are taken, as mentioned previously. Confirmation messages are not sent if the key is used to lock/unlock the door because the assumption that the owner of the key would also be the owner of the mobile device and not need to be informed about this on the device. This key was added to the system as a precautionary measure in the event that the mobile device was lost.

```
//send message out to see door status try {
    //Send status string to output stream to send to door
    outputStream.write(check_status,0,check_status.length()); long
    millis = SystemClock.uptimeMillis();
    while(SystemClock.uptimeMillis()<= millis+500);
    outputStream.flush();
} catch (IOException e1) {
    //  TODO Auto-generated catch block
    e1.printStackTrace();
}
//Wait for 800ms to get response
long millis = SystemClock.uptimeMillis();
while(SystemClock.uptimeMillis()<= millis+800);
//read in status
try {
    //See if there is anything in input stream
   if(aInputStreamReader.ready()){ //Read it into
   buffer
   aInputStreamReader.read(buffer, 0, buffer.length);
  //Make a string out of character array buffer and clean up null aString
  = new String(buffer);
   aString = aString.trim();
```

III.RESULTS

When the project was completed, the mobile device was able to communicate approximately thirty feet away from the microcontroller through concrete walls. The total time from initiation of a lock/unlock to action was approximately one second. This could be shortened through use of smaller delays in the program, but delays were left long in this project to ensure that data was sent successfully. The hardware was modified to draw power from a 120V to 12V transformer that was available, so that it could be added to any existing structure. To keep all the electronics relatively stable, the normal convention of the locking pin in the door was ignored. Instead, the linear actuator was placed in the door jamb, along with the microcontroller and key switch.

For demonstration purposes, and to avoid major construction work on the building, a mock door approximately half the size of a standard door was constructed. It consisted of several feet of a false wall, the door jamb, and the door itself. The false wall housed the microcontroller, linear actuator and other necessary circuitry. Since no door this size was available, it had to be constructed. Building both the door and the jamb led to the issue of having them both square, which was a major consideration throughout the construction process. The frame was made as square as possible, but was not entirely sturdy because it was not part of a larger structure. There was some resistance when the door was opened or closed, but this did not seem to detract from the true purpose of the project. This was thought to be the best solution for demonstration because it was very easy to install and repair the electronics and was portable, so that the project could be demonstrated anywhere with ease.

Two major problems arose in the development of this project. The original program prototype for both the mobile device and the microcontroller only communicated a single character to toggle a LED on the microcontroller protoboard. The mobile device user interface consisted on a single button to transmit the character because the MAC address of the microcontroller was hard coded in. Some problems arose when the second version of the mobile device program was developed. The mobile device was reading and writing to its buffers to quickly and losing data. An addition of a delay to the source code of the application fixed this problem. This problem did not occur in the microcontroller because delays had already been added to that code. Another problem arose in the choice of a suitable resistor value for the transistor array. It had to be a relatively small value, approximately 30 Ω, to supply enough current. While that value was on hand, it would quickly burn up because it drew too much power.

IV.CONCLUSION

The goal of this project was to create a security interface to an Android mobile device. It was also to be a short range system that was simple to use. The range and security aspects were achieved through the use of the onboard Bluetooth radio of the mobile device. Simplicity was a constant factor in design of the user interfaces. The system was able to actuate a pin to lock or unlock a door from a short distance away with the push of a button on the mobile device. It could also check the status of the door. The system also had a physical key included as a backup. Future work would include the design and building of a battery backup system. Improvements to the locking mechanism could also be another aspect for future work. This project could also be expanded to multiple doors and windows.

V. REFERENCE

[1]P.Kocher,‘‘Timing attacks on implementations of Diffie-Hellman,
RSA, DSS, other systems,’’ CRYPTO 1996, vol. 1109, N. Koblitz, Ed. Berlin, Germany: Springer-Verlag1996,pp.104–113,
ser. Lecture Notes in Computer Science.

[2]P. Kocher, J. Jaffe, and B. Jun, ‘‘Differential power analysis,’’ CRYPTO 1999, vol. 1666,M. J. Wiener, Ed. Berlin, Germany: Springer-Verlag, 1999, pp. 388–397,
ser. Lecture Notes in Computer Science.

[3]S. Mangard, E. Oswald, and T. Popp, Power Analysis Attacks: Revealing the Secrets of Smart Cards. New York, NY, USA: Springer-Verlag, 2007.

[4]C. S. Kim and M.-K. Lee, “Secure and user friendly PIN entry method,” in Proc. 28th Int. Conf. Consum. Electron., 2010, p. 5.1–1.

[5]V. Roth, K. Richter, and R. Freidinger, “A PIN-entry method resilient against shoulder surfing,” in Proc. CCS, 2004, pp. 236–245.

[6]G. T. Wilfong, “Method and apparatus for secure PIN entry,” U.S. Patent 5940511, May 30, 1997.