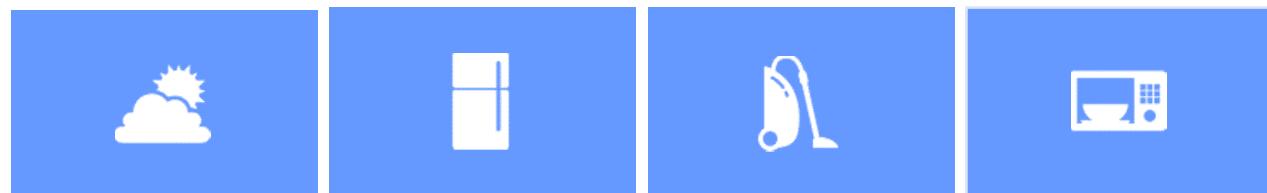


Capstone Project

INFRARED RAY SMART CONTROL

Supervisor: Hoang Xuan Son





Team Members

Lê Quang Đức

Phạm Văn Anh

Nguyễn Thị Dụ

Trịnh Xuân Trường

Nguyễn Thành Tùng



Contents

1. Introduction
2. Overview of Project
3. Main
 1. IR Smart Control System Overview
 2. Communication (Software & Hardware)
 3. Account management (Login, Change user name, Wi-Fi name)
 4. Learning, control, delete device.
 5. Sync
 6. IRSC-Box hardware design
4. Test & Develop
5. Product Demo
6. Q&A



Introduction

1

Idea & existing product

2

Scope of Project

3

Roles and Responsibilities





Disorder ?

Forgettable ?

Inconvenience ?

A l l - i n - O n e

How to solve those problems?



Mobile phone

Smart Phone

Android

Cheaper

Infrared Ray Smart Control System



Existing Product



Broadlink RM-Pro



Scope of Project

Hardware:

- Learning Infrared signal from remote and controlling devices through IRSmartControl's own wireless.
- Area of using system **in a room**, and each type of devices **exist only one** in this place.
- System support the following types of Infrared protocol: NEC, Sony, RC5, RC6, DISH, Sharp, JVC, Mitsubishi, Samsung, LG and some of undefined Infrared protocol.

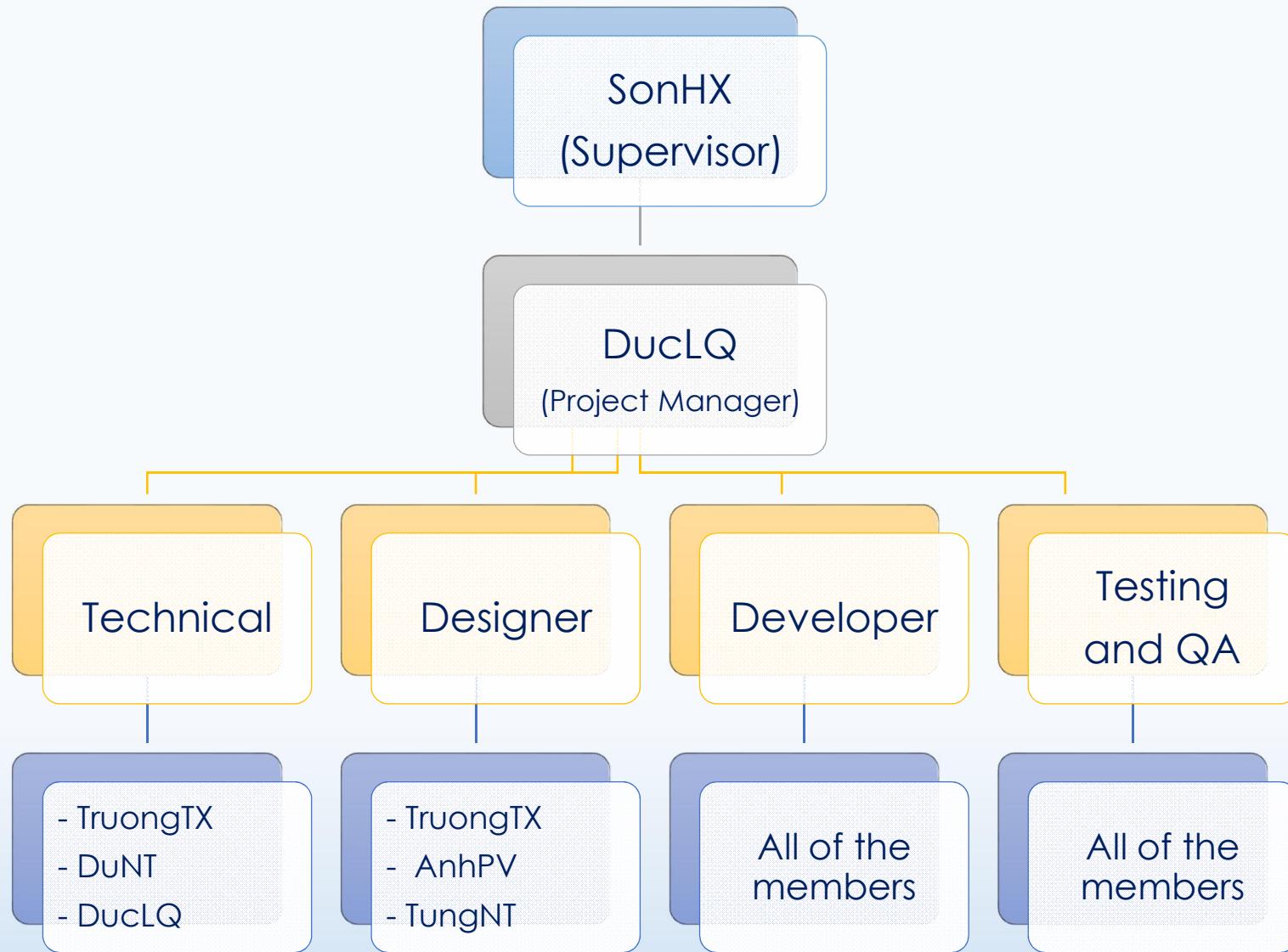


Scope of Project

Software:

- Using Android Operating system (4.1 or higher).
- Using IRSC's wireless to connects with IRSC-Box.
- Supporting most common devices: **Fan, TV, Projector, CD-Player, Air Conditioner, Door.**
- With Other devices: support only On/Off event
- Best display on 6.0-inches **1440x2560:560dpi**

Roles and responsibilities



1

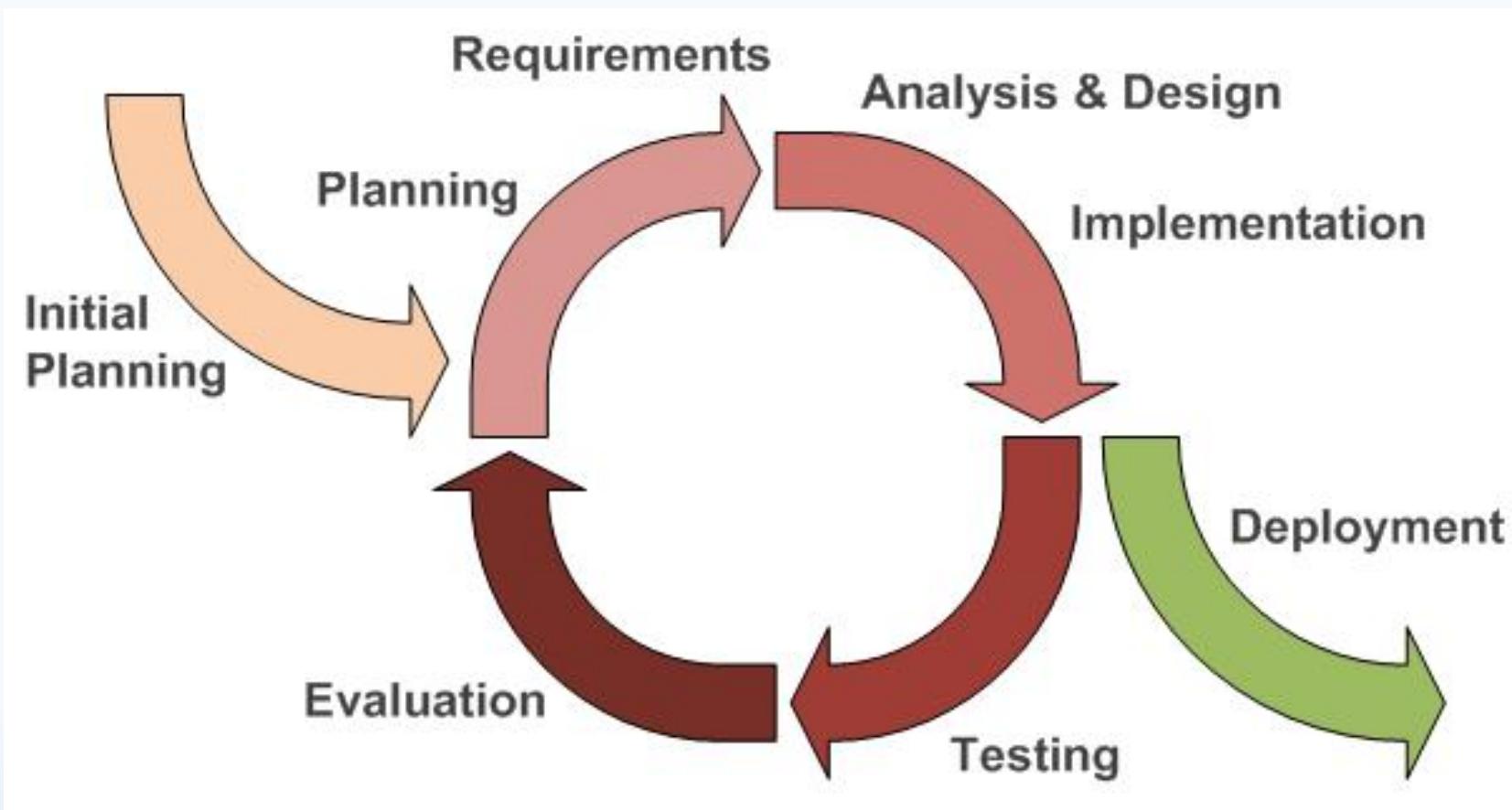
Process model

2

Development Tools

3

Overview of product



Iterative and incremental development process model



Development Tools



Google Drive

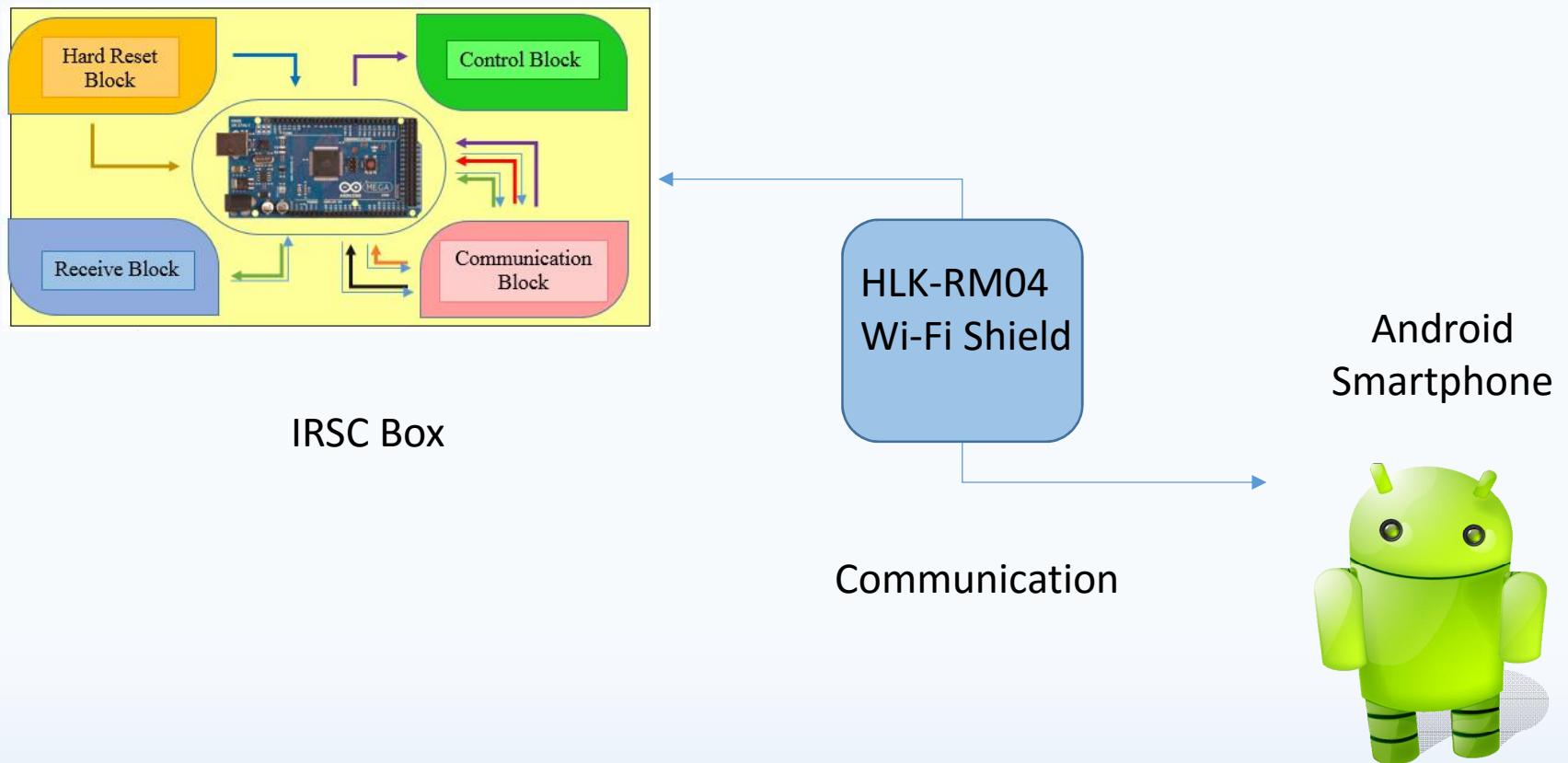




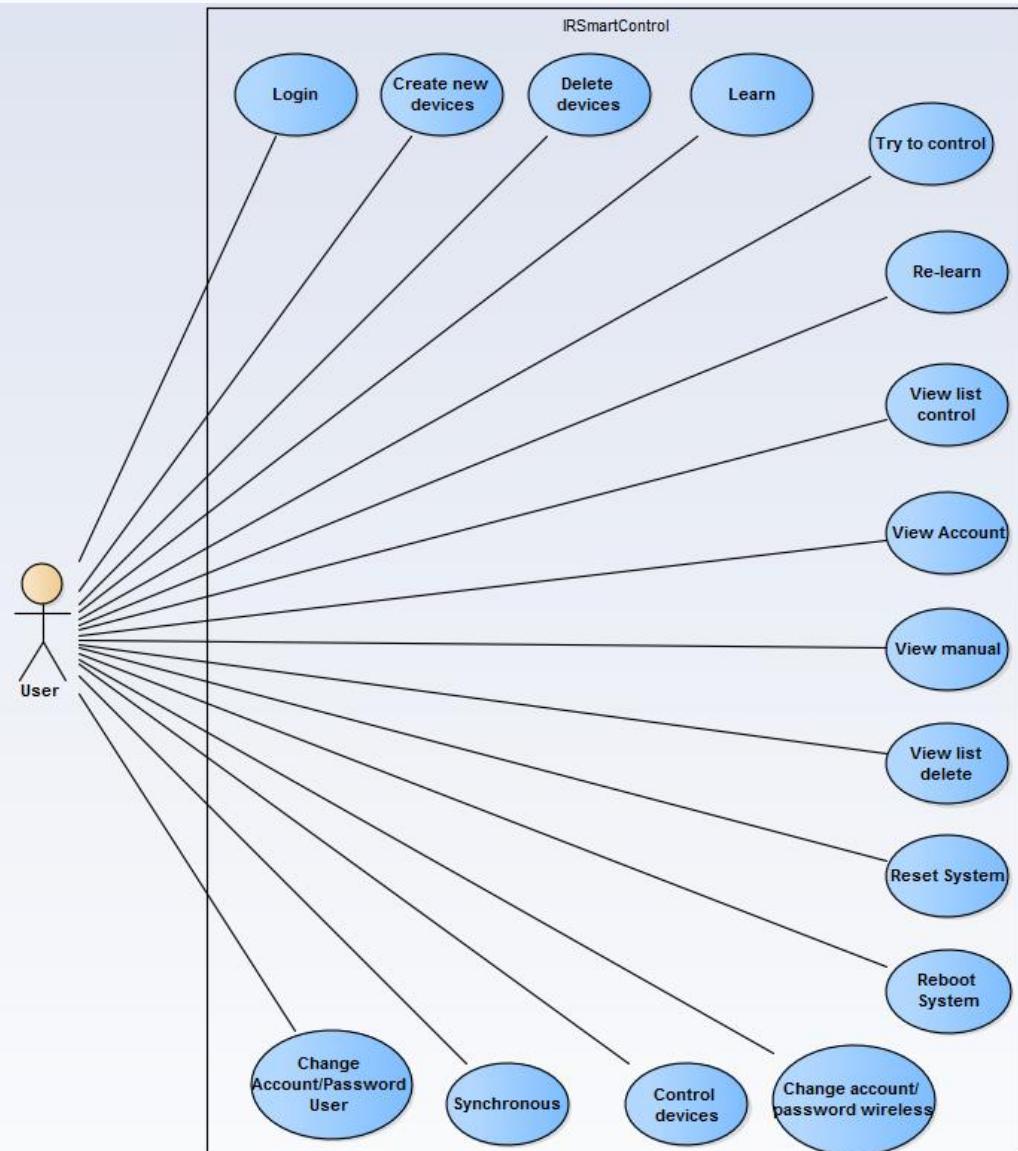
Risk Schedule

4	Hardware Failure	Low	Check all of hardware before buying. Being sure and testing about current and volt of this hardware before using.	Creating a list of store that is selling this hardware. Checking it exist if having plan goes to buy.	High
5	Requirement change	Medium	Carefully brainstorm system's features among team members. Regularly hold meeting to define and discuss all the features of systems. Design system carefully. Analyze all the possible cases to minimize the change	Team meetings with supervisor to determine whether new feature should be implemented or not. Team leaders create implementation plan for implemented features and sent to team members.	High

Overview IRSC system



Functional Requirement



Functional Requirement
of IRSC System



Non-functional Requirement

Reliability

Learning Infrared signal is correct up to 95- 97%. Immediately responses error message if system has error.

Safety

The supply voltage of hardware is 5V- 1A, be safe with user.

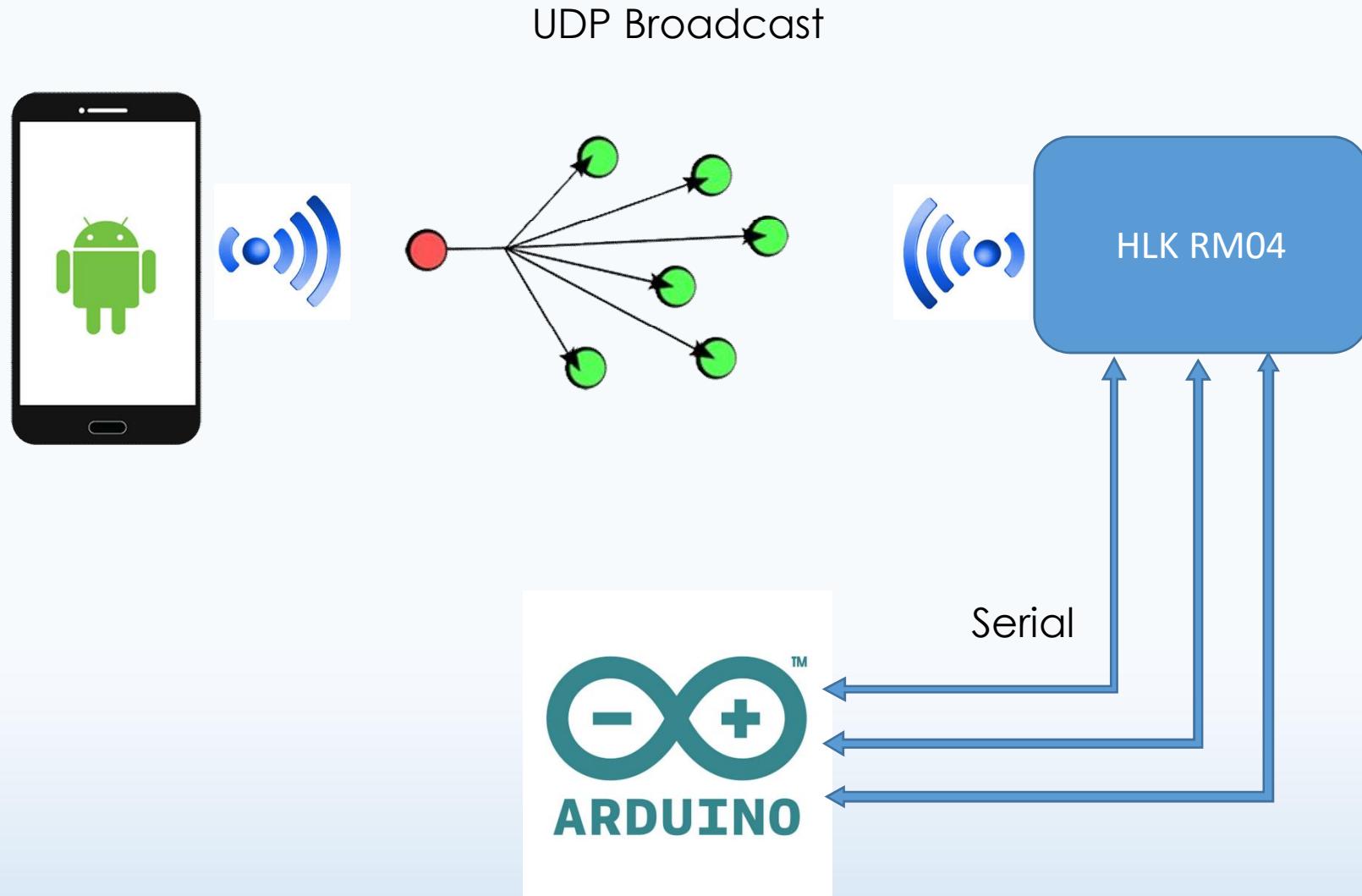
Usability

The interface of Android app should be elegant and simple.
Hardware is small and compactable

Maintainability

The IRSC system should be easy to modify and add features.

Communication





UDP Broadcast

```
/**  
 * Calculate the broadcast IP we need to send the packet along. If we send it  
 * to 255.255.255.255, it never gets sent.  
 */  
  
private InetAddress getBroadcastAddress() throws IOException {  
    Context context;  
    DhcpInfo dhcp = mWifi.getDhcpInfo();  
    if (dhcp == null) {  
        Log.d(TAG, "Could not get dhcp info");  
        return null;  
    }  
  
    int broadcast = (dhcp.ipAddress & dhcp.netmask) | ~dhcp.netmask;  
    byte[] quads = new byte[4];  
    for (int k = 0; k < 4; k++)  
        quads[k] = (byte) ((broadcast >> k * 8) & 0xFF);  
    return InetAddress.getByAddress(quads);  
}
```



SendTask

```
public class SendTasks extends AsyncTask<String, Integer, String> {

    private Context context;

    @Override
    protected String doInBackground(String... params) {
        String data =params[0];
        new UdpBroadcast((WifiManager) context.getSystemService(Context.WIFI_SERVICE), data.getBytes()).run();
        return data;
    }

    protected void onPostExecute(String result) {
        Log.e("DuNT", "Result:" + UdpBroadcast.s);
        Toast.makeText(context, UdpBroadcast.s, Toast.LENGTH_LONG).show();
        if (UdpBroadcast.s.equals("1")) {
            Toast.makeText(context,"Result is:"+UdpBroadcast.s,Toast.LENGTH_LONG).show();
        }
        else {Toast.makeText(context,"Invalid",Toast.LENGTH_LONG).show();}
    }
}
```



Encapsulation & Send

1. new SendTasks().execute(PR + "00");

2. new SendTasks().execute("");

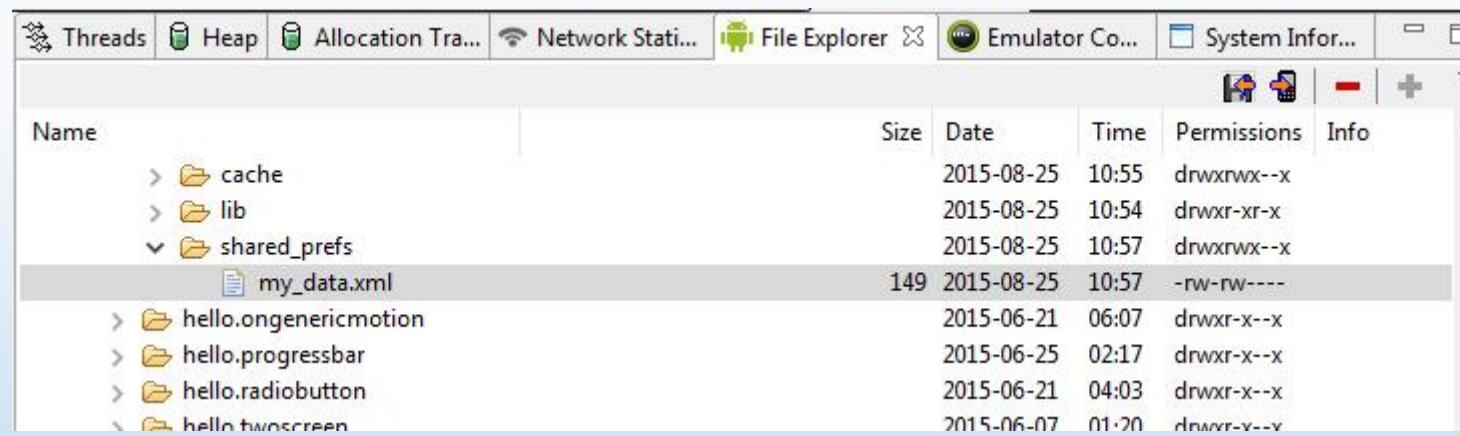
```
protected String doInBackground(String... params) {  
    String data ;  
    if ((acc.length()+pass.length()+3) >= 10) {  
        data = Integer.toString(acc.length() + pass.length()+3) + "11"+acc.getText().toString()  
+"~"+ pass.getText().toString();  
    } else {  
        data= "0"+Integer.toString(acc.length()+pass.length()+3) + "11"+acc.getText().toString()  
+"~"+ pass.getText().toString();  
    }  
    new UdpBroadcast((WifiManager)  
getApplicationContext().getSystemService(Context.WIFI_SERVICE), data.getBytes()).run();  
    return data;  
}
```

1. Receive

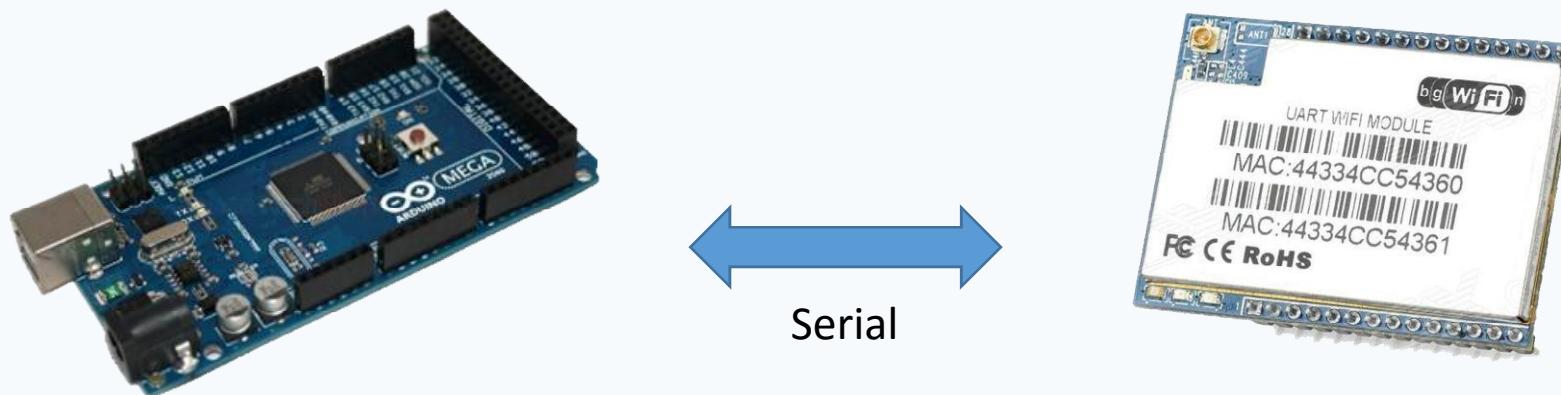
```
protected void onPostExecute(String result) {  
    if (UdpBroadcast.s.equals("1")) {  
        Intent it = new Intent(Login.this, Home.class);  
        startActivity(it);  
    }  
    else {Toast.makeText(getApplicationContext(),  
        "Please try Again " , Toast.LENGTH_LONG).show();}  
}
```

1. Save data

SharedPreferences pre=getSharedPreferences("my_data", MODE_PRIVATE);



Name	Size	Date	Time	Permissions	Info
> cache		2015-08-25	10:55	drwxrwx--x	
> lib		2015-08-25	10:54	drwxr-xr-x	
shared_prefs		2015-08-25	10:57	drwxrwx--x	
my_data.xml	149	2015-08-25	10:57	-rw-rw----	
> hello.ongenericmotion		2015-06-21	06:07	drwxr-x--x	
> hello.progressbar		2015-06-25	02:17	drwxr-x--x	
> hello radiobutton		2015-06-21	04:03	drwxr-x--x	
> hello_twoscreen		2015-06-07	01:20	drwxr-x--x	



Arduino Mega 2560	HLK-RM04
5V	PIN1 - 5V
GND	PIN2 - GND
PIN18 - (TX)	PIN20 - (RX)
PIN19 - (RX)	PIN21 - (TX)



Hardware Communication(2)

❖Serial Configuration:

- Baud rate =9600;
- Data bits = 8;
- None parity check bit;
- Stop bit = 1;

❖Network mode : Server

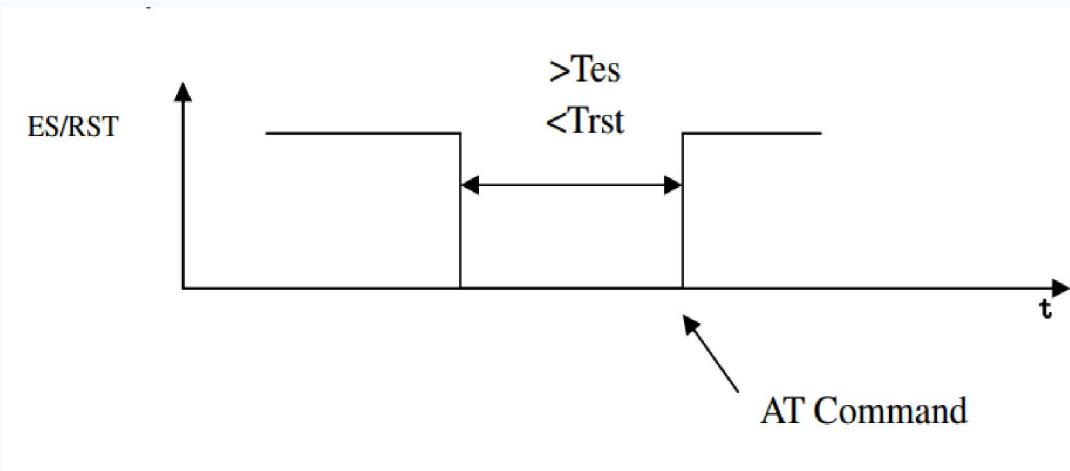
❖Locate remote port: 2390

❖Protocol: UDP

❖Network timeout:10 sec

	Current	Updated
Serial Configure:	9600,8,n,1	<input type="text" value="9600,8,n,1"/>
Serial Framing Lenth:	64	<input type="text" value="64"/>
Serial Framing Timeout:	10 milliseconds	<input type="text" value="10"/> milliseconds (< 256, 0 for no timeout)
Network Mode:	server	<input type="button" value="Server ▾"/>
Remote Server Domain/IP:	192.168.11.245	<input type="text" value="192.168.11.245"/>
Locale/Remote Port Number:	2390	<input type="text" value="2390"/>
Network Protocol:	udp	<input type="button" value="UDP ▾"/>
Network Timeout:	10 seconds	<input type="text" value="10"/> seconds (< 256, 0 for no timeout)

- Configure serial by AT command mode



```
//1. Switch to AT command Mode
bool switchToAtMode(){
    bool mode = false;
    clearSerialRxData();

    digitalWrite(ESCAPE_PIN, LOW);
    delay(400);
    digitalWrite(ESCAPE_PIN, HIGH);
    mode = true;
    return mode;
}
```



Hardware Communication(4)

Instruction format is as follows:

At+[command]=[value]\r

According to the different command, module will return a different return value.

For example :

"**at+remoteip=192.168.11.254\r**" set remote ip address as 192.168.11.254.

"**at+remoteip= ? \r**" Inquiry remote ip address.

Type of AT command code

wifi_conf	Wifi configuration
dhcpc	Dhcp client configuration
net_ip	Network ip address
net_dns	Network dns address
dhcpd	Dhcp server configuration
dhcpd_ip	Dhcp server ip address
dhcpd_dns	Dhcp server dns address
dhcpd_time	Dhcp sever time allocation
net_commit	Submit network configuration
out_trans	Exit transparent transmission mode
remoteip	Remote server domain name or IP address
remoteport	The local or distal port number
remotepro	Network Protocol type
timeout	Network timeout



Account Management

1

Log In

2

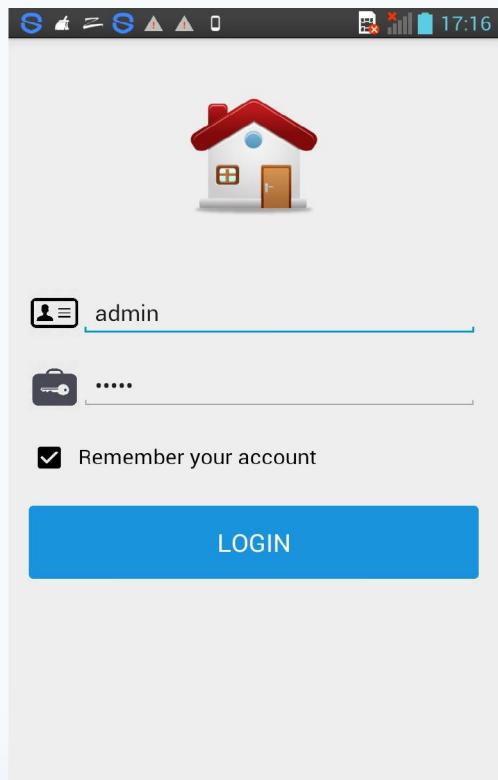
Change User Information

3

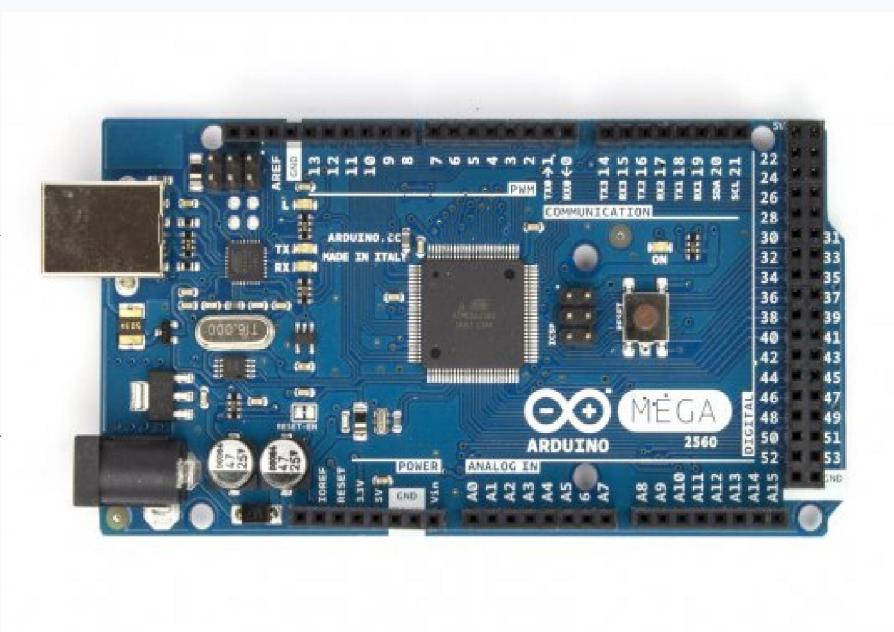
Change Wi-Fi Information



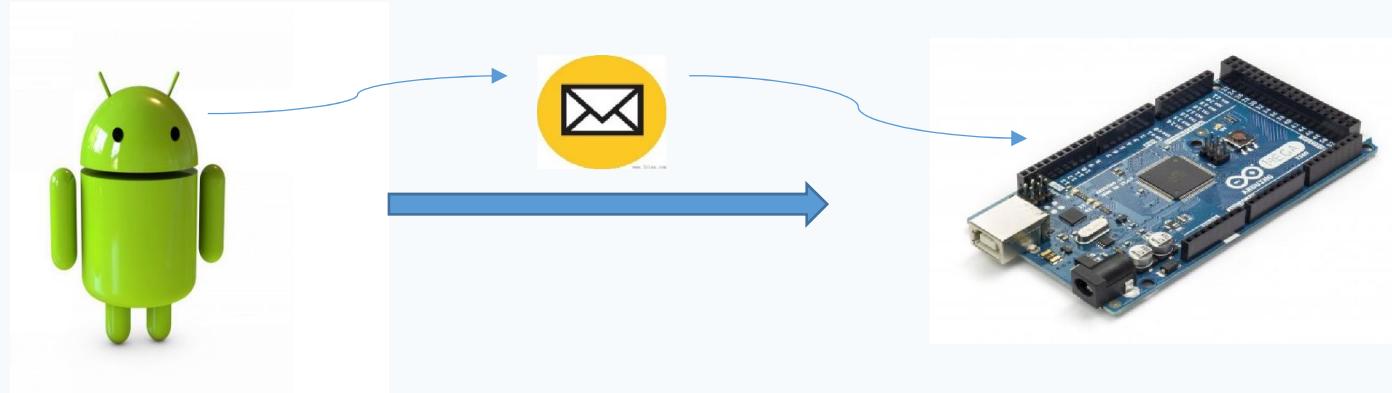
LOG IN



Send data



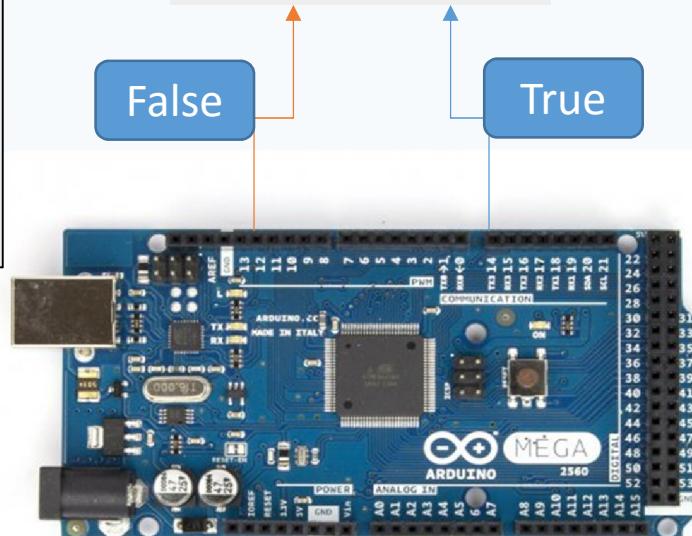
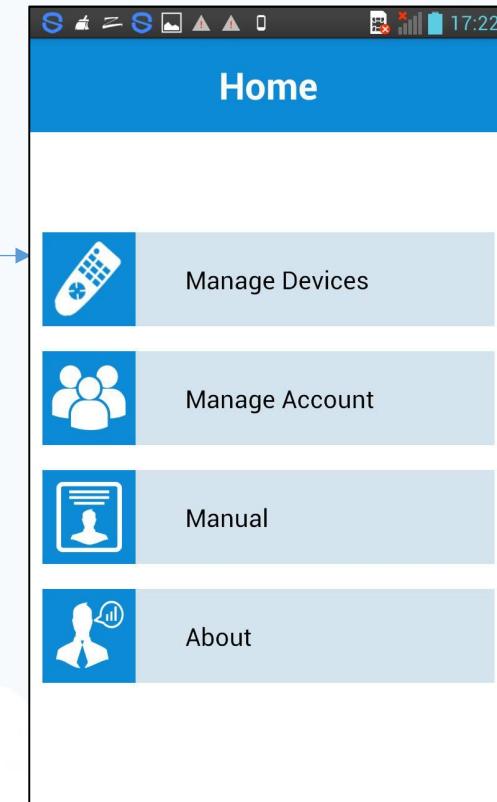
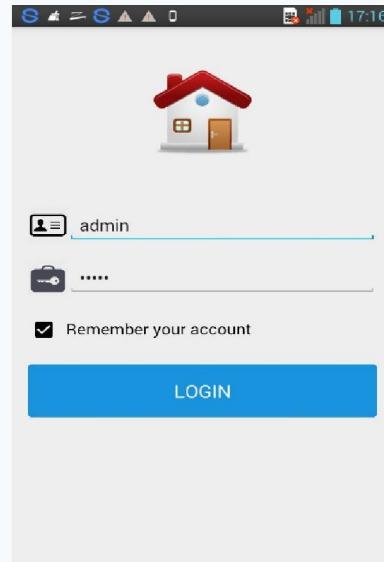
LOG IN



```
protected String doInBackground(String... params) {  
    String data ;  
    if ((acc.length()+pass.length()+3) >= 10) {  
        data = Integer.toString(acc.length() + pass.length()+3) + "11"+acc.getText().toString() +"~"+ pass.getText().toString();  
    } else {  
        data= "0"+Integer.toString(acc.length()+pass.length()+3) + "11"+acc.getText().toString() +"~"+ pass.getText().toString();  
    }  
    new UdpBroadcast((WifiManager) getApplicationContext().getSystemService(Context.WIFI_SERVICE), data.getBytes()).run();  
    return data;  
}
```

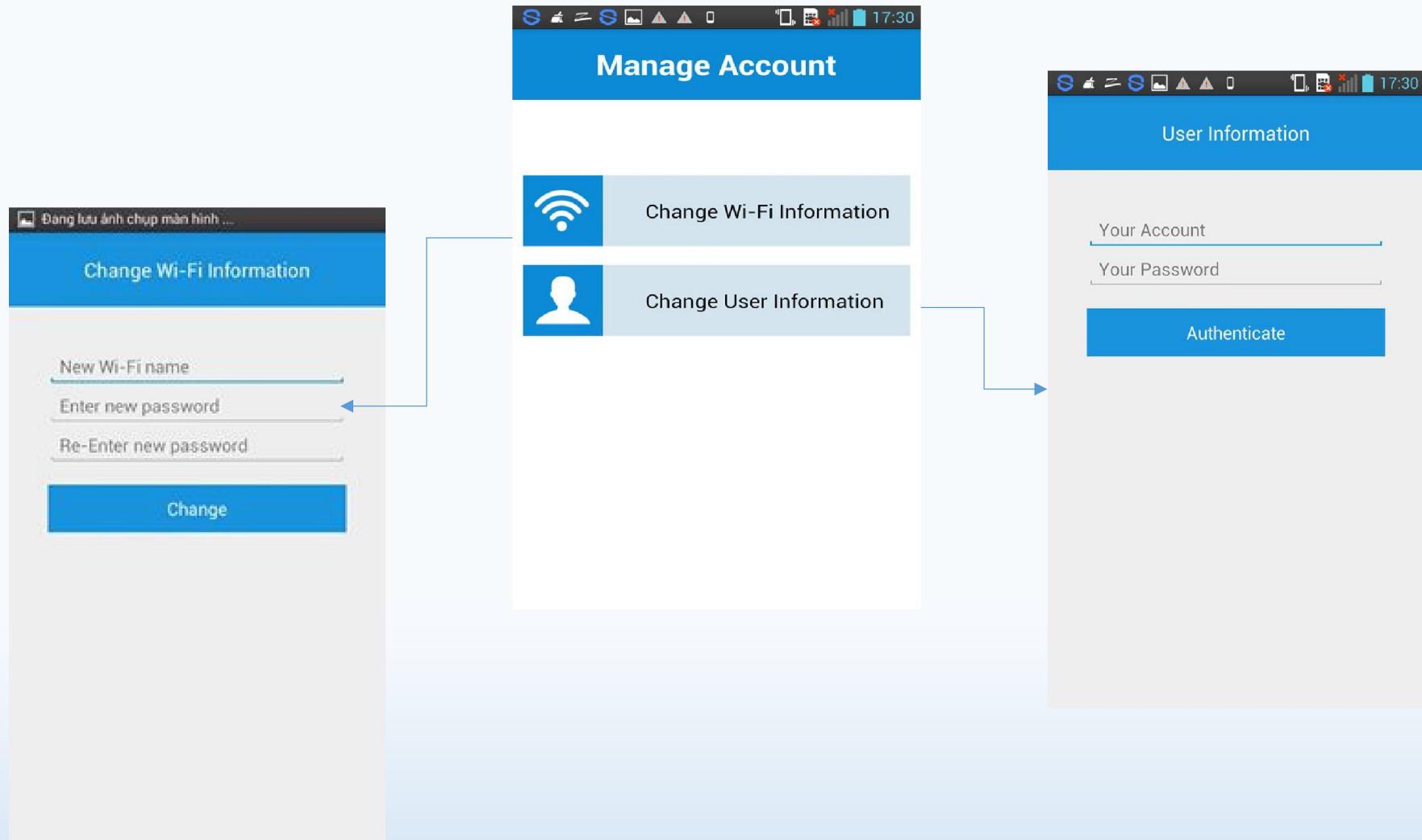


LOG IN





Change Wi-Fi & User





Change Wi-Fi & User

Đang lưu ảnh chụp màn hình ...

Change Wi-Fi Information

New Wi-Fi name
Enter new password
Re-Enter new password
Change

17:30

User Information

Your Account
Your Password
Authenticate



```
protected String doInBackground(String... params) {
    String data;
    if ((acc.length() + pass.length() + 3) >= 10) {
        data = Integer.toString(acc.length() + pass.length() + 3) + "33" + acc.getText().toString() + "\n" + pass.getText().toString();
    } else {
        data = "0" + Integer.toString(acc.length() + pass.length() + 3) + "33" + acc.getText().toString() + "\n" + pass.getText().toString();
    }
    new UdpBroadcast_Device((WifiManager) getApplicationContext().getSystemService(Context.WIFI_SERVICE), data.getBytes()).run();
    return data;
}

protected String doInBackground(String... params) {
    String data;
    if ((acc.length() + pass.length() + 3) >= 10) {
        data = Integer.toString(acc.length() + pass.length() + 3) + "99" + acc.getText().toString() + "\n" + pass.getText().toString();
    } else {
        data = "0" + Integer.toString(acc.length() + pass.length() + 3) + "99" + acc.getText().toString() + "\n" + pass.getText().toString();
    }
    new UdpBroadcast_Device((WifiManager) getApplicationContext().getSystemService(Context.WIFI_SERVICE), data.getBytes()).run();
    return data;
}
```



Change Wi-Fi & User

```
protected String doInBackground(String... params) {
    String data;
    if ((acc.length() + pass.length() + 3) >= 10) {
        data = Integer.toString(acc.length() + pass.length() + 3) + "99" + acc.getText().toString() + "~" + pass.getText().toString();
    } else {
        data = "0" + Integer.toString(acc.length() + pass.length() + 3) + "99" + acc.getText().toString() + "~" + pass.getText().toString()
    }
    new Udpbroadcast_Device((WifiManager) getApplicationContext().getSystemService(Context.WIFI_SERVICE), data.getBytes()).run();
    return data;
}
```

```
protected String doInBackground(String... params) {
    String data;
    if ((acc.length() + pass.length() + 3) >= 10) {
        data = Integer.toString(acc.length() + pass.length() + 3) + "33" + acc.getText().toString() + "~" + pass.getText().toString();
    } else {
        data = "0" + Integer.toString(acc.length() + pass.length() + 3) + "33" + acc.getText().toString() + "~" + pass.getText().toString()
    }
    new UdpBroadcast((WifiManager) getApplicationContext().getSystemService(Context.WIFI_SERVICE), data.getBytes()).run();
    return data;
}
```



Manage Devices

1

Add new device and Learning

2

Control Devices

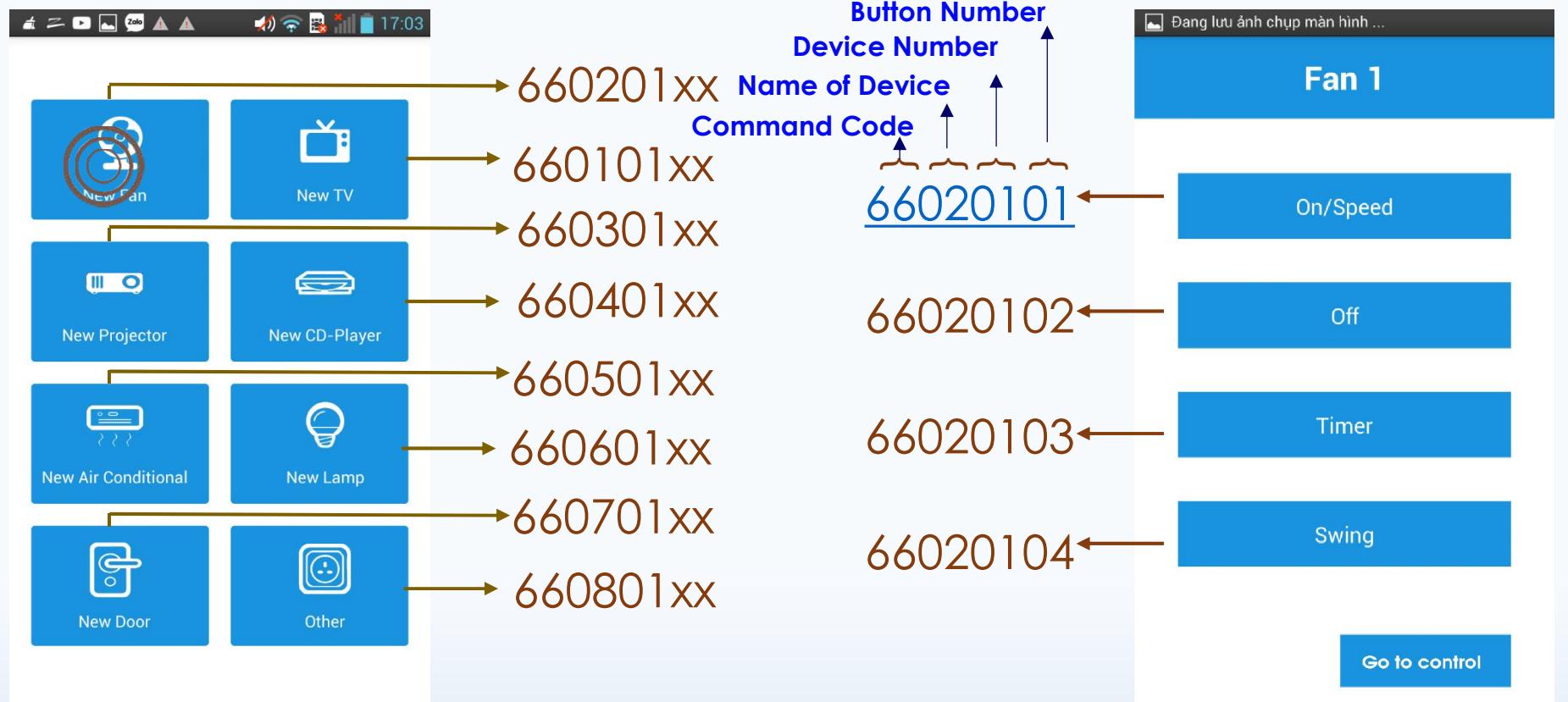
3

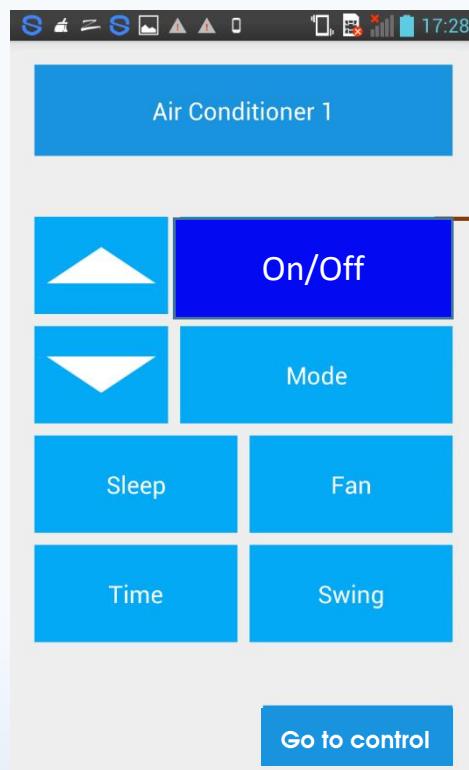
Delete devices



Add Device

new SendTasks().execute(PR + "00");



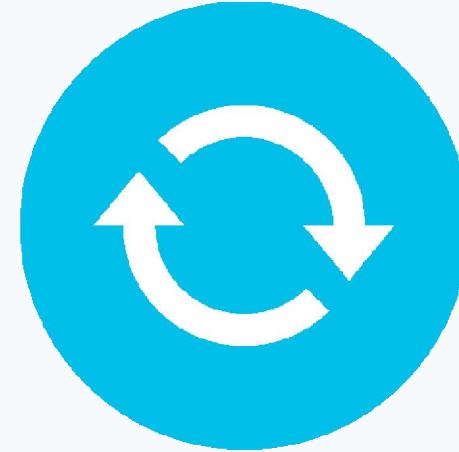
Example: Learning Air Conditioner

Example: Control Air Conditioner

```
new SendTasksSyne().execute("0221");
    new SendTasksSyne().execute("0222");
    new SendTasksSyne().execute("0223");
    new SendTasksSyne().execute("0224");
```

```
protected void onPostExecute(String result) {

    int count = 0;
    char[] temp = UdpBroadcast.s.toCharArray();
    Log.e(UdpBroadcast.s, "RRRR");
    int i = 0;
    for (i = 0; i < UdpBroadcast.s.length(); i++) {
        if (temp[0] == '0') {
            if (i >= 1 && i <= 5) {
                if (temp[i] == 'a') {
                    mEditor.putBoolean("btntv" + Integer.toString(i), true);
                    mEditor.commit();
                } else {
                    mEditor.putBoolean("btntv" + Integer.toString(i), false);
                    mEditor.commit();
                }
            }
        }
    }
}
```



1

Data Structures

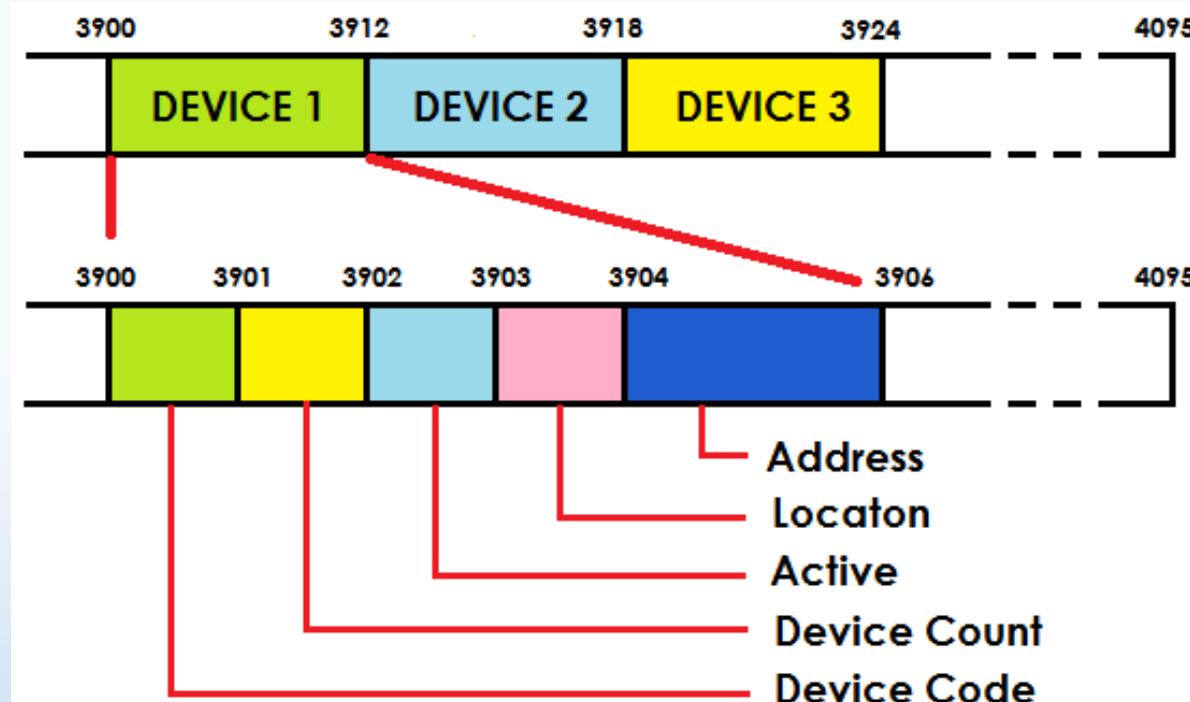
2

Setup Function

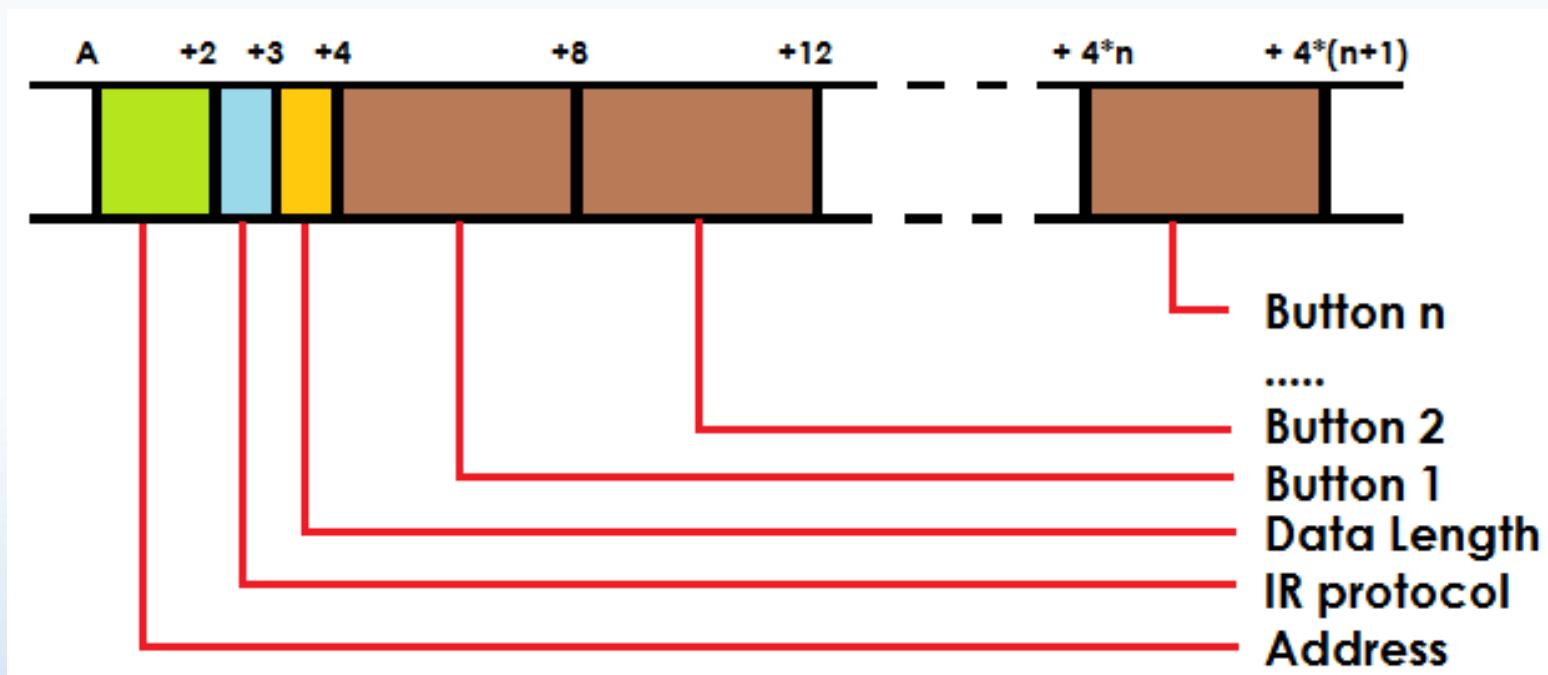
3

Infinite Loop Function

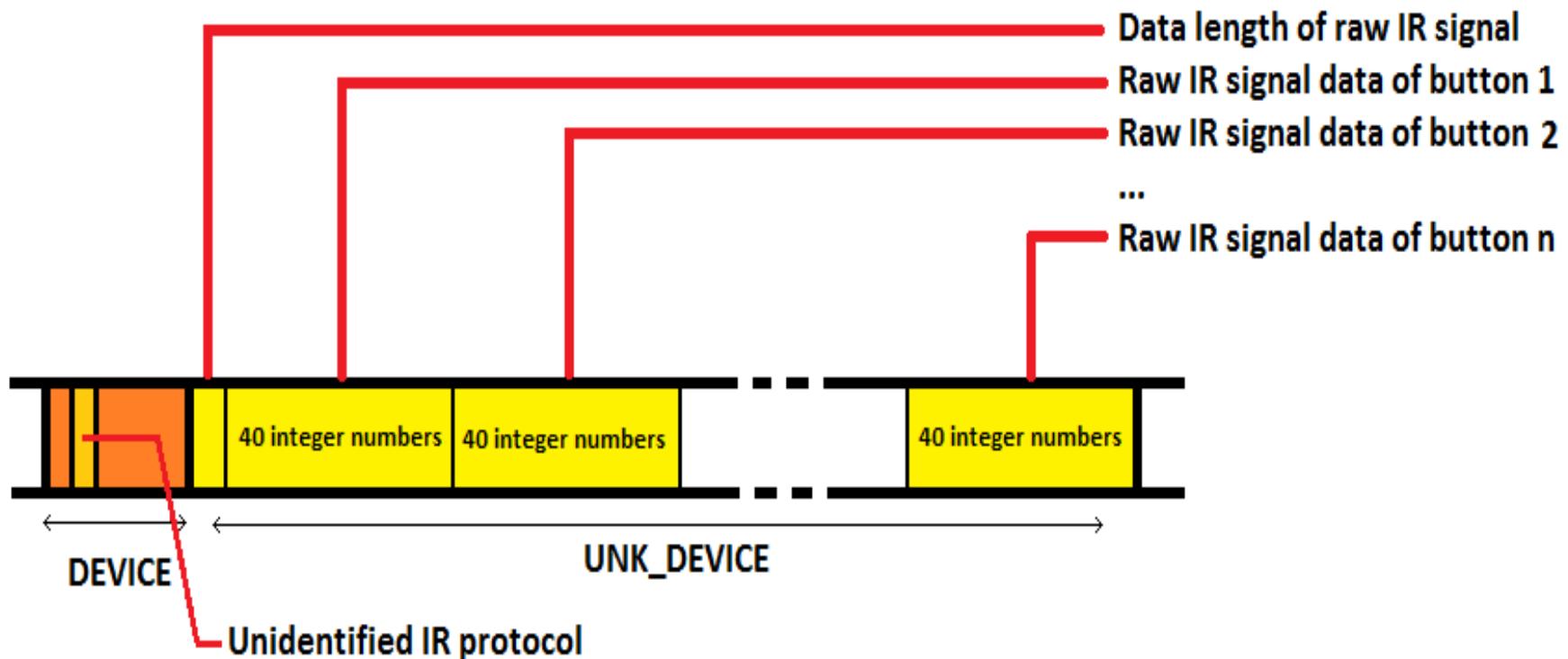
```
struct Devices {  
    byte Device; // Device Code  
    byte DC; // Device Count  
    byte active;  
    byte location;  
    int Address;  
};
```

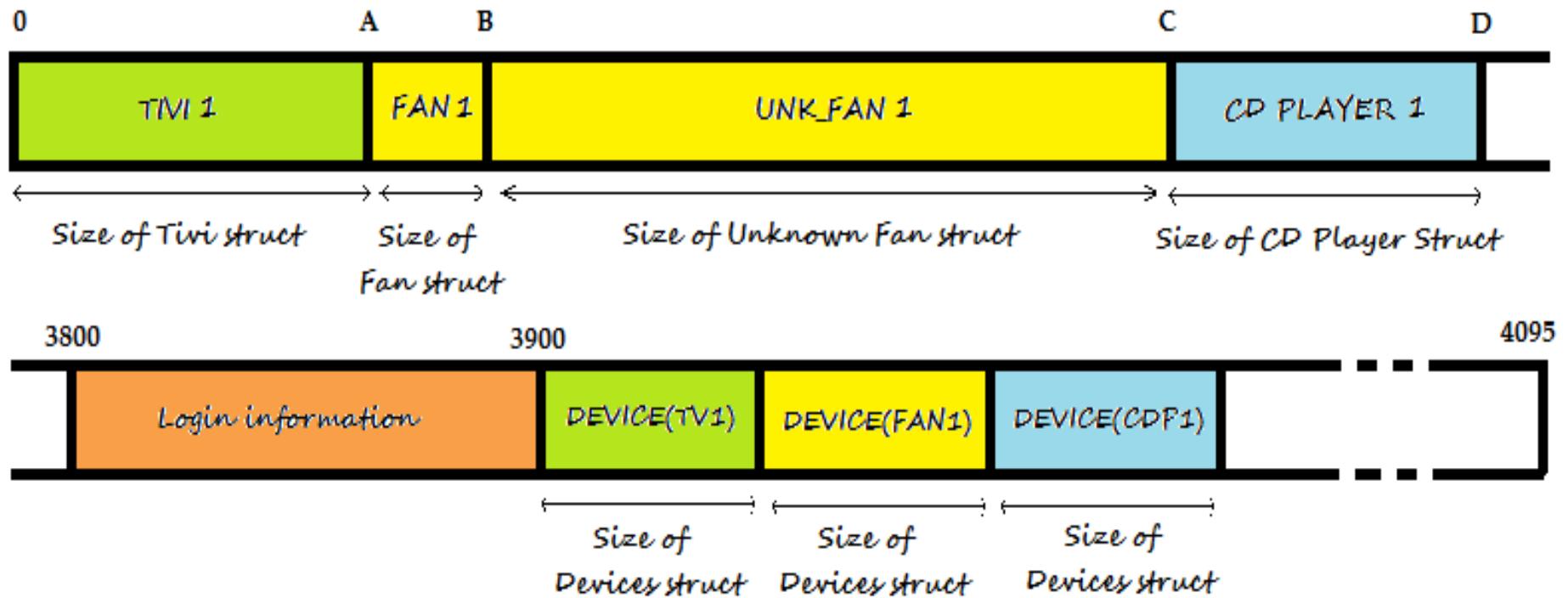


```
struct Tivi {  
    int Address;  
    byte Type;  
    byte DLength;  
    unsigned long BC[B_TV];  
};
```



```
struct UNK_Tivi {  
    byte DTLength;  
    unsigned int BC[B_TV][UNK_rec];  
};
```





Note: TV1: Address = 0;

FAN1: Address = A; Infrared protocol is unknown type

CDP1: Address = C;

Address variable in DEVICE(TV1) = 0

Address variable in DEVICE(FAN1) = A

Address variable in DEVICE(CDP1) = C

Address A = 0 + Size of Tivi struct

Address B = A + Size of Fan struct

Address C = B + Size of UNK_Fan struct

Address D = C + Size of CDP struct

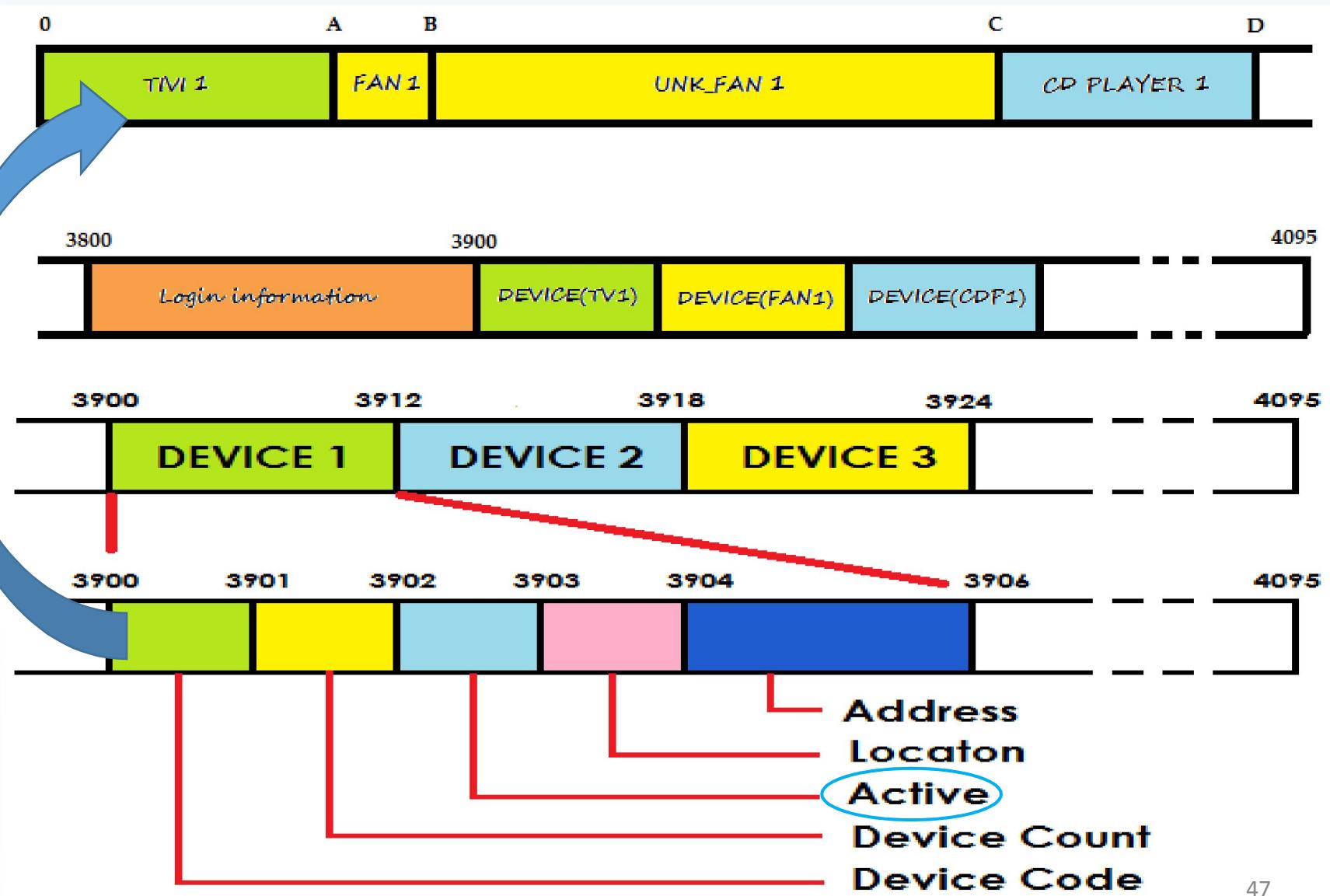


Setup Function

- Set BaudRate for serial connections
- Enable IR receive module
- Set mode for Arduino pins
- Initialize some value

```
void setupBegin() {  
    Serial.begin(9600);  
    Serial1.begin(9600);  
    irrecv.enableIRIn();  
    pinMode(ledGreen, OUTPUT);  
    pinMode(ledRed, OUTPUT);  
    pinMode(buttonPin, INPUT);  
    deviceCount=0;  
    locationCount=0;  
}
```

Reading EEPROM



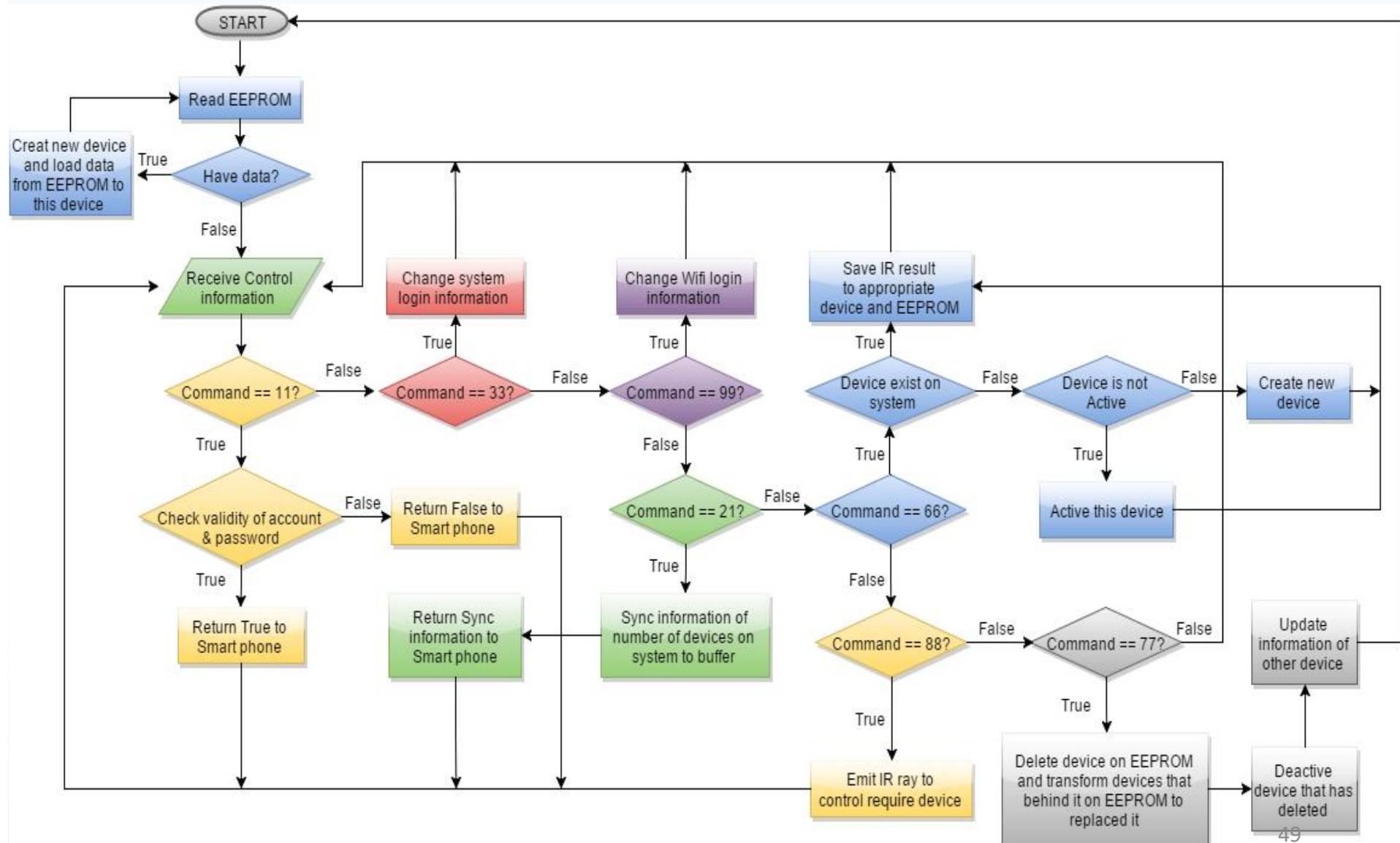


Infinite Loop Function

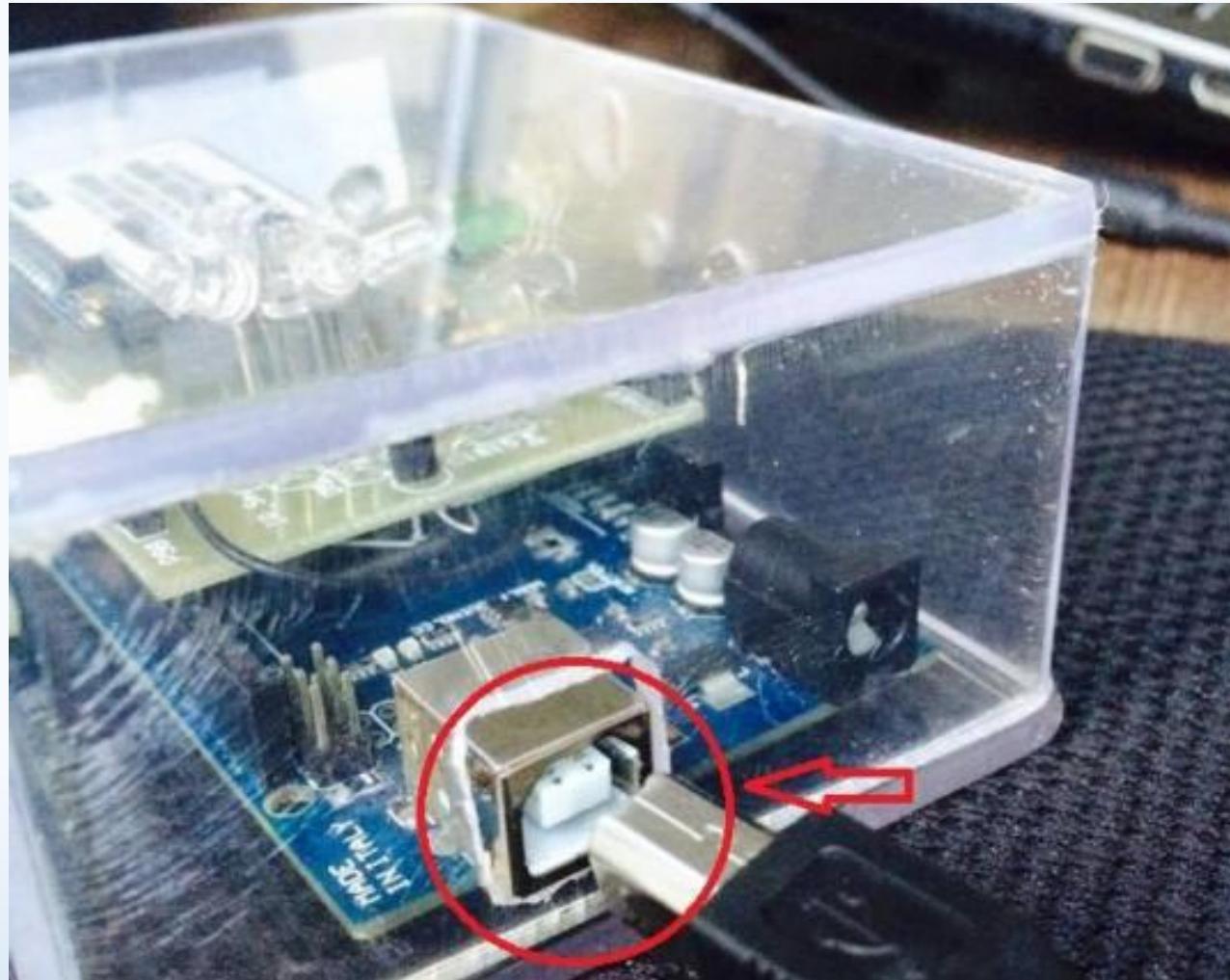
```
void loop() {
```

- **Wait** for what commit **command** from IRSC-App
 - If IRSC-App sent a command to IRSC-Box, **following the command code**, Arduino will do a below task:
 - ✓ Check validity of account & password
 - ✓ Change system login information
 - ✓ Change Wifi login information
 - ✓ Sync information of number of devices on system to buffer
 - ✓ Add new device or learn new IR signal
 - ✓ Emit IR ray to control require device
 - ✓ Delete device on EEPROM and transform devices that behind it on EEPROM to replaced it
- ```
}
```

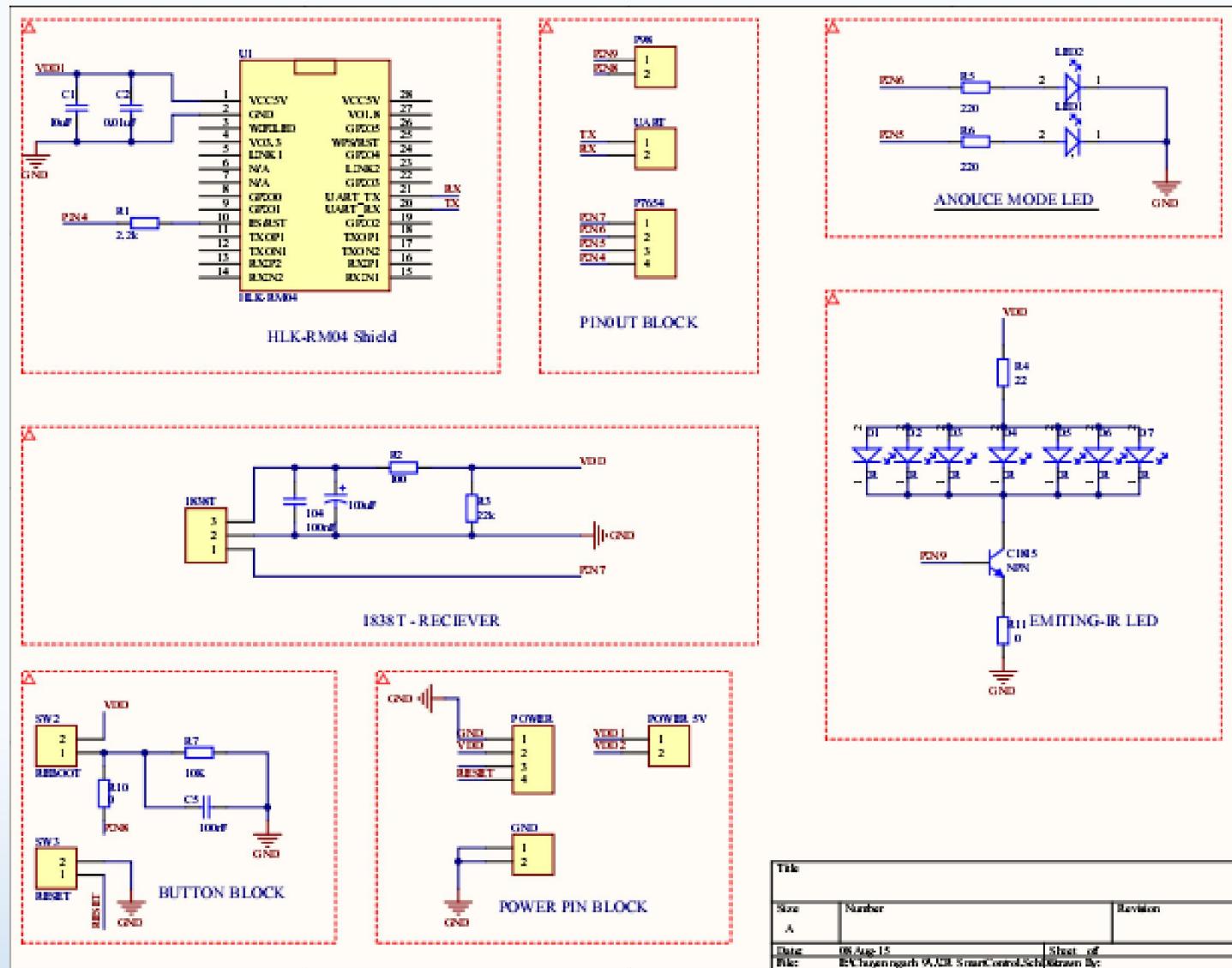
# Infinite Loop Function



# Hardware design

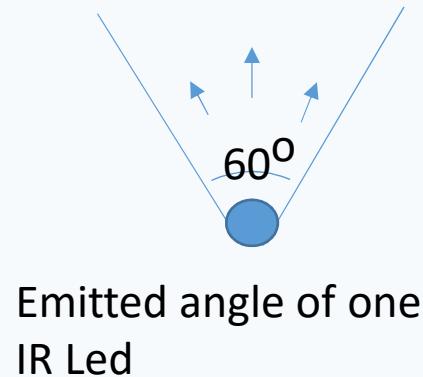


# Hardware design

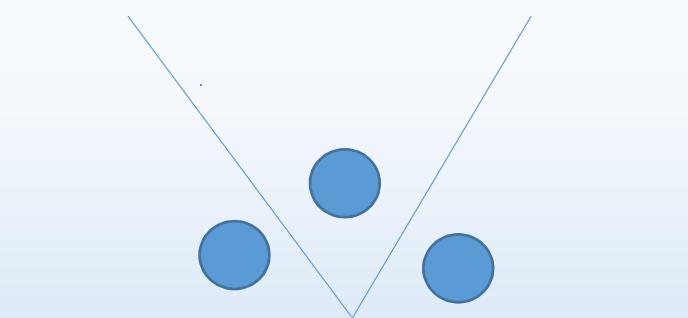
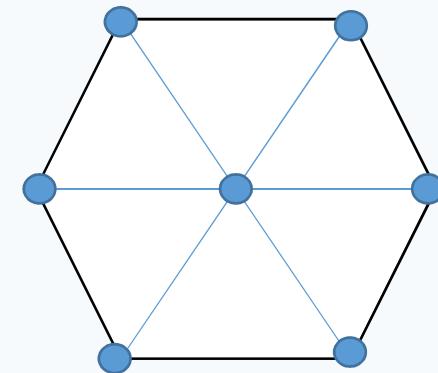


| Title                                                                          |            |          |
|--------------------------------------------------------------------------------|------------|----------|
| Size                                                                           | Number     | Revision |
| A                                                                              |            |          |
| Date: 08-Aug-15<br>File: E:\Chaitanya\9.0\CL_SmartControl\Sch\SmartControl.sch | Sheet of 1 |          |

- Emitting block design



The sphere of  
emitted block



cross section of  
one platform



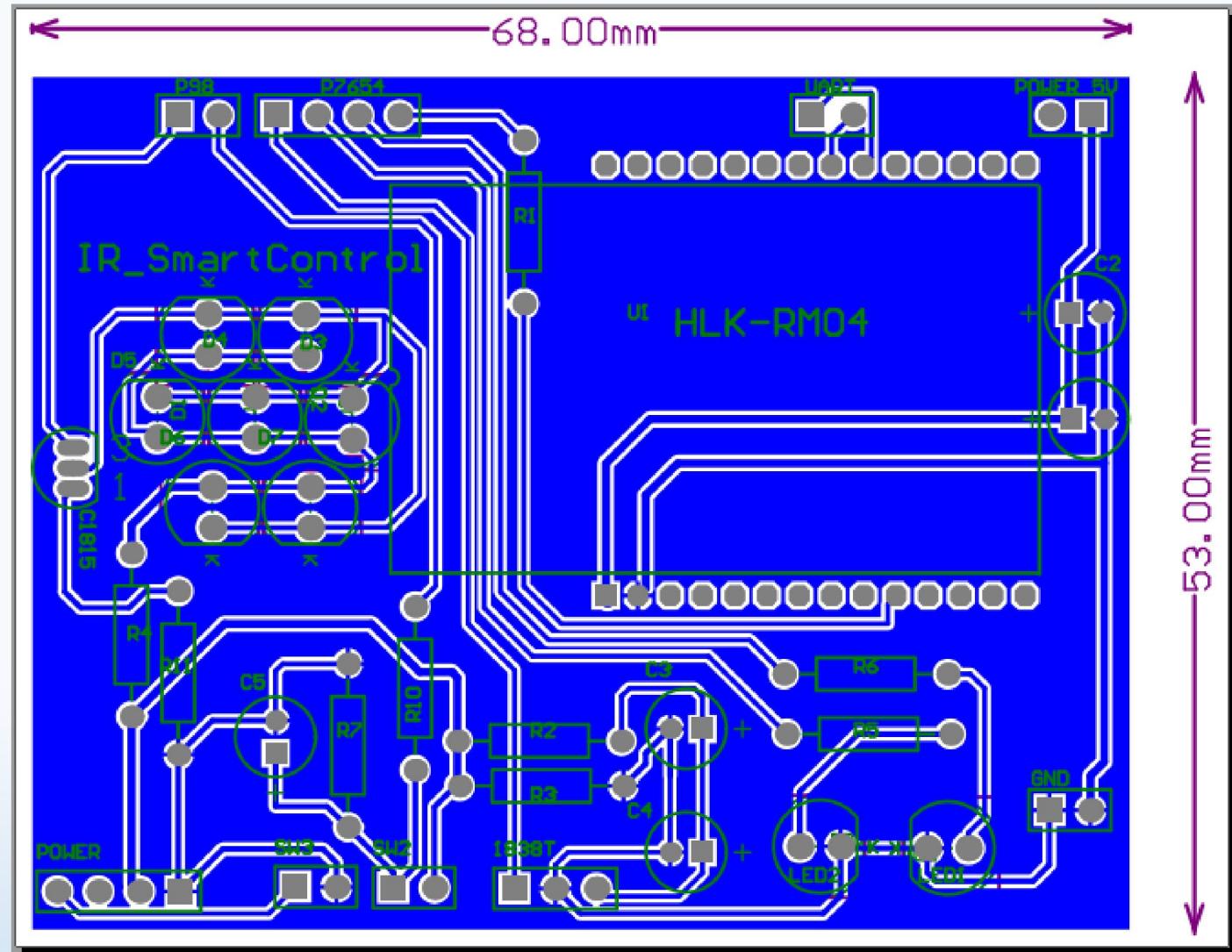
# Hardware design

- IR led : 1.6V – 20mA
- Output current of emitted pin : 5V – 40mA-> need transistor to amplify
- Power pin current : 5V-200mA

We have 7 IR LEDs in parallel

$$\Rightarrow 7 \times 20\text{mA} = 140 \text{ mA}$$

$$\text{Using resistor} = (5 - 1.6)/140\text{mA} = 22 \text{ Ohm}$$





# Infrared Ray Smart Control

## Demo our product



# Infrared Ray Smart Control

## Q&A

# Thanks For Watching

