

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN ĐIỆN TỬ VIỄN THÔNG



BÁO CÁO

MÔN: HỆ THỐNG NHÚNG

Đề tài: Xây dựng hệ thống phát hiện và cảnh báo khi có xâm nhập

Sinh viên thực hiện:

Họ và tên:	MSSV	Lớp
1. Đoàn Duy Kiên	20111715	ĐTT04-K56
2. Phạm Văn Lâm	20111746	ĐTT02-K56
3. Nguyễn Văn Thuận	20112293	ĐTT04-K56

GVHS: TS. Ngô Vũ Đức

Hà Nội, 5/2015

Mục Lục

LỜI NÓI ĐẦU.....	3
1. Tổng quan hệ thống.....	4
1.1. Sơ đồ tổng quan hệ thống.....	4
1.2. Mô tả hoạt động hệ thống.....	5
1.3. Yêu cầu phi chức năng.....	5
1.4. Chỉ tiêu kỹ thuật.....	6
2. Thiết kế hệ thống.....	7
2.1. Trung tâm xử lý-Điện thoại android.....	7
2.1.1. Input.....	7
2.1.2. Output.....	7
2.1.3. Lưu đồ thuật toán cơ bản.....	7
2.1.4. Triển khai.....	8
2.2. Khối vi điều khiển, cảm biến và camera.....	18
2.2.1. Vi điều khiển.....	18
2.2.2. Cảm biến.....	19
2.2.3. Camera.....	20
3. Kết quả.....	21

LỜI NÓI ĐẦU

Ngày nay, khi đời sống xã hội ngày càng phát triển thì việc bảo vệ ngôi nhà của mình ngày càng được coi trọng. Tuy nhiên, không phải lúc nào mọi người cũng có thể ở nhà để trông coi mọi thứ được. Do đó việc xây dựng một hệ thống cảnh báo giúp chủ nhà có thể đi ra ngoài mà vẫn quan sát được tất cả mọi thứ trong nhà là hoàn toàn cần thiết.

Vì vậy, nhóm đã quyết định chọn đề tài "Xây dựng hệ thống phát hiện và cảnh báo khi có xâm nhập". Nhiệm vụ chính của đề tài bao gồm xây dựng một hệ thống nhận diện khi có người vào trong căn phòng, gửi bản tin cảnh báo đến chủ nhà thông qua tin nhắn SMS, tiến hành ghi lại hình ảnh hiện trạng của căn phòng, tiến hành triển khai phương pháp streaming video giúp chủ nhân có thể xem trực tuyến căn phòng hiện tại. Thông qua hệ thống, chủ nhà có thể nắm bắt được tình hình của ngôi nhà 24/24 và có được bằng chứng để có thể tìm kiếm người đã thực hiện hành vi xâm nhập căn phòng sau này.

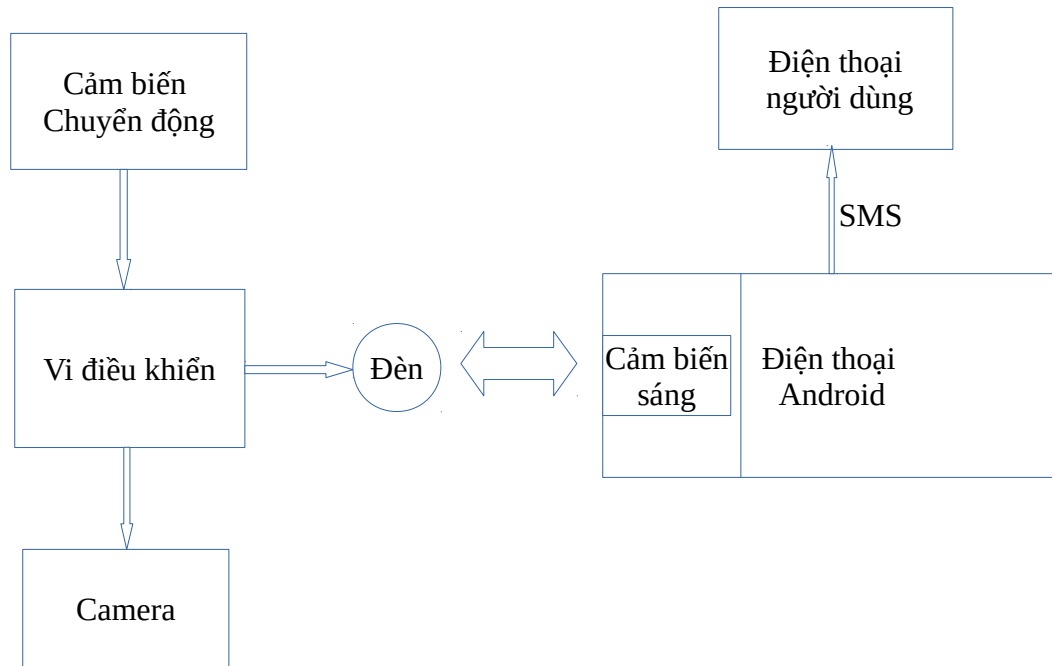
Với đề tài này chúng em có thể được nghiên cứu rất nhiều kiến thức hay và cực kì thiết yếu như kĩ năng lập trình C/C++ cho vi điều khiển và lập trình android. Đặc biệt chúng em đã ứng dụng được các kỹ thuật chuyên môn, cách tiếp cận thực hiện đề tài và những bài học về công việc cũng như những hướng phát triển cho tương lai từ thầy giáo TS. Ngô Vũ Đức. Em xin chân thành cảm ơn sự giúp đỡ và chỉ bảo từ thầy giáo.

Bản báo cáo này sẽ trình bày :

- + Tổng quan về hệ thống
- + Chi tiết về phần mềm android
- + Chi tiết về phần cứng - vi điều khiển.

1. Tổng quan hệ thống

1.1. Sơ đồ tổng quan hệ thống



+ Trong đó:

- Điện thoại android là trung tâm xử lý
- Điện thoại người dùng là bất kỳ loại điện thoại nào để gửi nhận SMS
- Vi điều khiển có kết nối với cảm biến chuyển động và camera giám sát
- Vi điều khiển còn kết nối với đèn, bật tắt đèn để thông báo cho điện thoại android biết thông qua cảm biến sáng trên điện thoại

1.2. Mô tả hoạt động hệ thống

*) **Vi điều khiển:** có kết nối với cảm biến chuyển động và camera.

+ Khi xác định có chuyển động (người, vật), vi điều khiển sẽ bật camera để ghi lại hiện trường.

+ Đồng thời, vi điều khiển gửi tín hiệu thông báo đến cho Điện thoại trung tâm thông qua việc bật đèn báo.

*) **Điện thoại Android** làm trung tâm xử lý: Khi nhận tín hiệu thông báo từ vi điều khiển thông qua cảm biến sáng, sẽ gửi thông báo SMS đến cho điện thoại người dùng hoặc phát tín hiệu chuông cảnh báo (tùy chọn)

*) **Điện thoại người dùng:** nhận thông báo từ điện thoại trung tâm.

+Người dùng có thể xem video về hiện trường trực tiếp trên web, hoặc qua phần mềm được cung cấp bởi nhà sản xuất.

1.3. Yêu cầu phi chức năng

+ Hệ thống chạy ổn định

+ Hệ thống chạy tiết kiệm năng lượng

+ Mạch vi điều khiển đẹp, ổn định.

+ Phần mềm android nhẹ, giao diện đẹp

1.4. Chỉ tiêu kỹ thuật

	Thành phần	Thông số	Ghi chú
Cảm biến chuyển động	Khoảng cách	0.01-10m	Có thể thay đổi được tùy thuộc vào vị trí đặt cảm biến so với nơi cần phát hiện chuyển động
	Điện áp hoạt động	12-24V	
Camera	Tốc độ khởi động	30s	
	Điện áp hoạt động	12V	Có thể dùng chung nguồn với cảm biến
	Chuẩn nén hình ảnh	H264	
Vi điều khiển	Điện áp	3.3V	
	Dòng điện	15 mA	
Phần mềm android	Điểm mốc độ rọi	300 lux	Điểm mà khi độ sáng vượt qua nó thì điện thoại android sẽ xác định có kẻ xâm nhập
	Tốc độ phản ứng khi đèn bật sáng	1s	
	Gửi tin nhắn cho người dùng	2s	Thời gian mà người dùng nhận được tin nhắn còn phụ thuộc tốc độ mạng di động
	Dung lượng phần mềm	Nhỏ hơn 4MB	
	Thời gian phát cảnh báo	Tối đa 1 phút 30 giây	Hoặc có thể dừng lại khi người dùng tắt trên điện thoại android.

2. Thiết kế hệ thống

2.1. Trung tâm xử lý-Điện thoại android

+ Ta sẽ lập trình, cài đặt 1 phần mềm trên điện thoại này để xử lý những yêu cầu đưa ra

2.1.1. Input

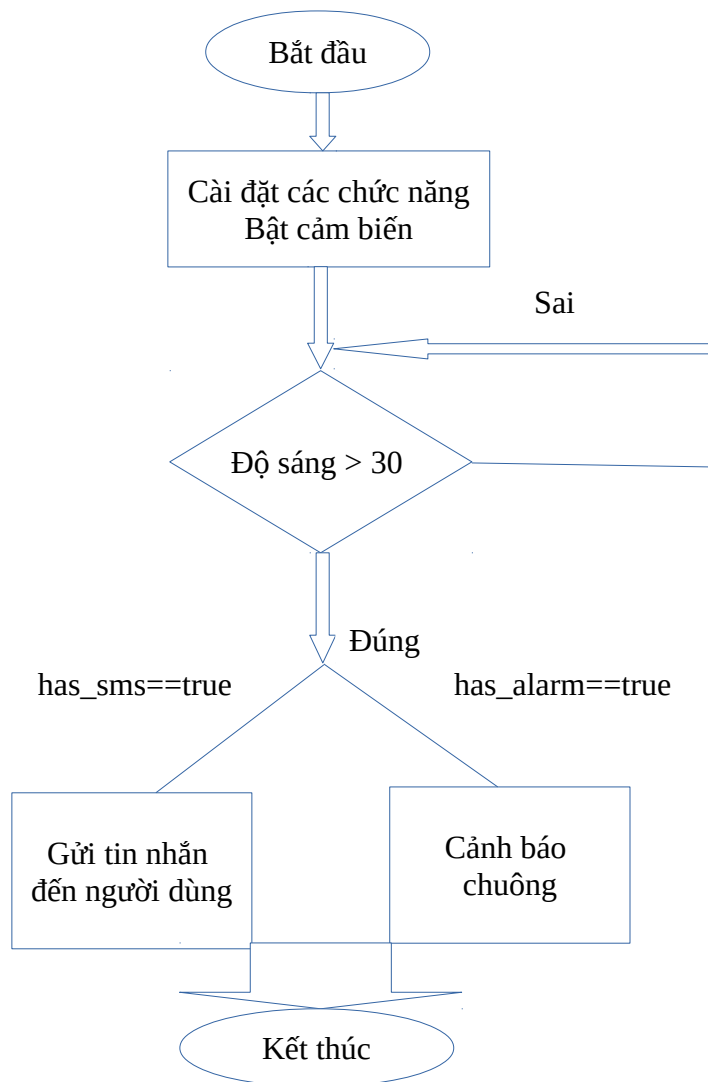
+ Độ sáng môi trường lấy từ cảm biến sáng

2.1.2. Output

+ Gửi tin nhắn SMS với nội dung cảnh báo tới người dùng

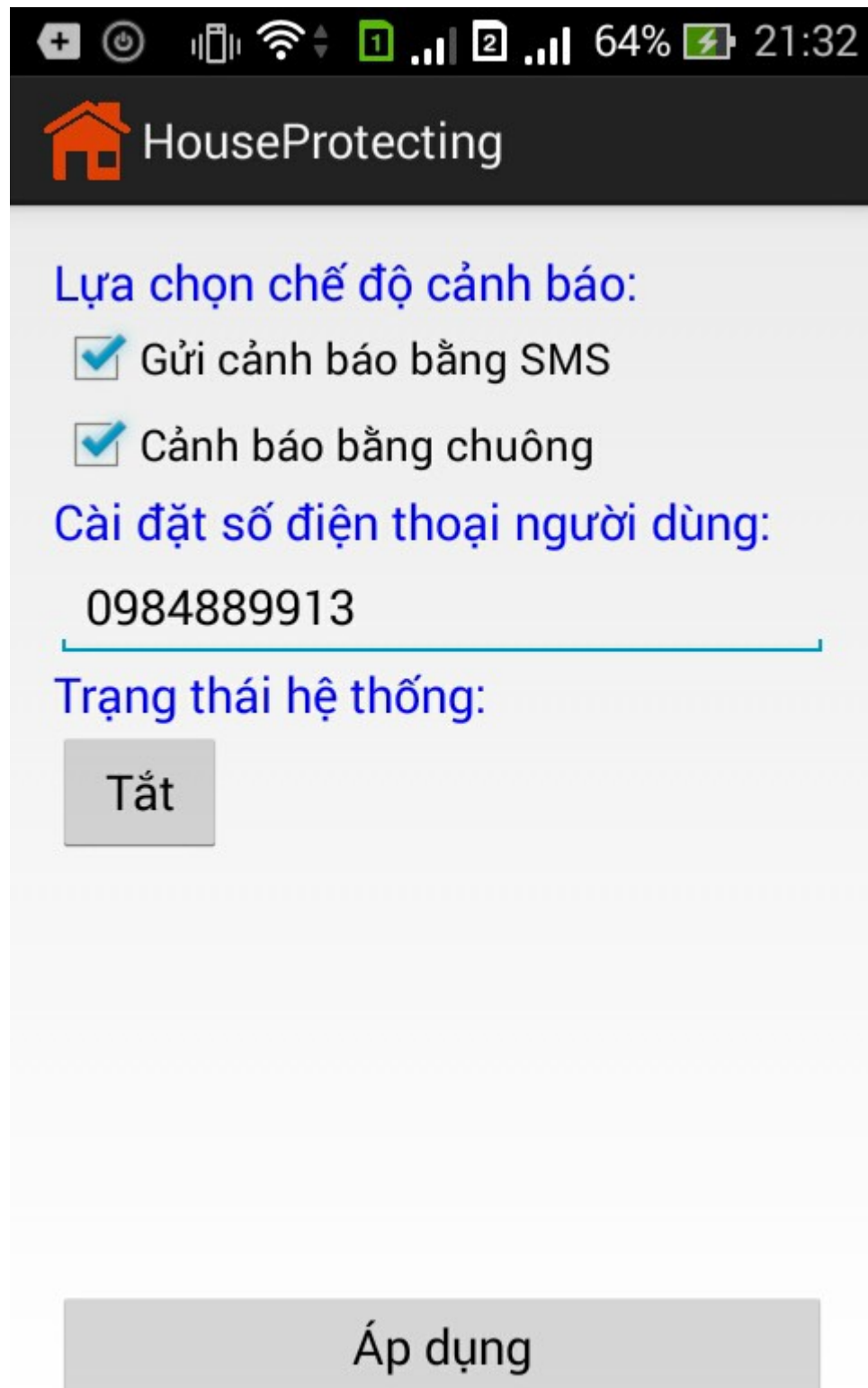
+ Phát ra âm thanh cảnh báo độ dài tối đa là 2 phút

2.1.3. Lưu đồ thuật toán cơ bản



2.1.4. Triển khai

I. Giao diện



The screenshot shows the 'HouseProtecting' app interface. At the top is an Android status bar with icons for signal, Wi-Fi, battery (64%), and time (21:32). Below the status bar is a dark header with an orange house icon and the text 'HouseProtecting'. The main content area has a light gray background and contains the following elements:

- Lựa chọn chế độ cảnh báo:** (Select alarm mode:)
 - ☒ Gửi cảnh báo bằng SMS (Send alarm by SMS)
 - ☒ Cảnh báo bằng chuông (Alarm by bell)
- Cài đặt số điện thoại người dùng:** (Set user phone number:)
 - 0984889913
- Trạng thái hệ thống:** (System status:)
 - Tắt (Off)
- Áp dụng** (Apply)

II. Giải thích code

*) Các biến khai báo:

```
// Khai báo
Button btn_switch, btn_apply;
CheckBox send_sms, chbx_alarm;
MediaPlayer mediaPlayer;
EditText adminNumber;
SharedPreferences sharedPreferences;
public static final String MyPREFERENCES = "MyPrefs" ;

String phoneNumber = "";
String msg = "Cảnh báo có người xâm nhập!";
boolean has_send_sms = true;
boolean has_alarm = true;

int countSMS=0;
int start = 0;
```

+ Button:

- **btn_switch**: chính là nút bật tắt hệ thống: khi btn_switch là "Tắt" thì hệ thống đang tắt và không hoạt động. Không có cảnh báo được thực hiện. Khi btn_switch là "Bật" thì hệ thống sẽ bật và sẽ có cảnh báo khi có tùy chọn cảnh báo
- **btn_apply** : chính là nút "áp dụng" các lựa chọn cài đặt cho hệ thống: Sau khi cài đặt bất kỳ một cái gì thì người sử dụng phải nhấn vào nút "áp dụng" thì cài đặt mới có tác dụng

+ Checkbox:

- **send_sms**: checkbox dùng để xác nhận việc có gửi tin nhắn hay không khi có cảnh báo. Nếu ô này được tích nghĩa là sẽ có tin nhắn được gửi khi cảnh báo xảy ra.
- **chbx_alarm**: checkbox dùng để xác nhận việc có thông báo chuông hay không khi có cảnh báo xảy ra. Nếu ô này được tích thì khi có cảnh báo, điện thoại sẽ phát

ra chuông báo động.

+ **MediaPlayer mediaPlayer**: dùng để quản lý việc phát file mp3 là file nhạc cảnh báo

+ **EditText adminNumber**: dùng để nhập vào số điện thoại người sử dụng, chính là người nhận tin nhắn SMS cảnh báo.

+ **SharedPreferences sharedPreferences**: dùng để lưu tùy chọn cho hệ thống, có tác dụng là nó sẽ không mất đi khi ứng dụng tắt. Vì vậy, các tùy chọn sẽ không thay đổi sau các lần bật tắt ứng dụng.

+ **public static final String MyPREFERENCES = "MyPrefs"** : là tên sharedPreferences

+ **String phoneNumber = ""**: dùng để lưu số điện thoại người nhận

+ **boolean has_send_sms = true**: dùng để lưu trạng thái là có muốn gửi tin nhắn SMS hay không

+ **boolean has_alarm = true**: dùng để lưu trạng thái là có muốn cảnh báo chuông hay không

+ **int countSMS=0**: Đếm số lượng tin nhắn đã gửi, thực tế là mỗi lần có cảnh báo ta chỉ cần gửi một tin nhắn đến người sử dụng

+ **int start = 0**: Dùng để kiểm tra xem hệ thống đang trong trạng thái bật hay tắt. Nếu start=0 nghĩa là hệ thống đang tắt, còn lại thì hệ thống đang bật.

*) **Hàm Oncreate()**:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    send_sms = (CheckBox) findViewById(R.id.send_sms);
    chbx_alarm = (CheckBox) findViewById(R.id.alarm);
}
```

```

        adminNumber = (EditText) findViewById(R.id.admin_Number);

        btn_apply = (Button) findViewById(R.id.btn_apply);
        btn_apply.setOnClickListener(apply_action);
        btn_switch = (Button) findViewById(R.id.btn_switch);
        btn_switch.setOnClickListener(action_switch);

        mediaPlayer =
MediaPlayer.create(getApplicationContext(),R.raw.alarm_sound);

        SensorManager
            mySensorManager =
(SensorManager)getSystemService(SENSOR_SERVICE);

        Sensor LightSensor =
mySensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
        mySensorManager.registerListener(
            LightSensorListener,
            LightSensor,
            SensorManager.SENSOR_DELAY_NORMAL);
        getOption();
        setOption();
    }

```

+ Kết nối file java với file xml mô tả giao diện:

```

send_sms = (CheckBox) findViewById(R.id.send_sms);
chbx_alarm = (CheckBox) findViewById(R.id.alarm);
adminNumber = (EditText) findViewById(R.id.admin_Number);

btn_apply = (Button) findViewById(R.id.btn_apply);
btn_apply.setOnClickListener(apply_action);
btn_switch = (Button) findViewById(R.id.btn_switch);
btn_switch.setOnClickListener(action_switch);

```

+ Khai báo mediaplayer để quản lý file âm thanh cảnh báo:

```

        mediaPlayer =
MediaPlayer.create(getApplicationContext(),R.raw.alarm_sound);

```

+ Khai báo quản lý cảm biến ánh sáng:

```

    SensorManager mySensorManager
=(SensorManager)getSystemService(SENSOR_SERVICE);

    Sensor LightSensor = mySensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
    mySensorManager.registerListener(
        LightSensorListener,
        LightSensor,
        SensorManager.SENSOR_DELAY_NORMAL);

```

+ Lấy thông tin cài đặt từ trước trong sharedPreferences:

```

    public void getOption() {

        sharedPreferences = getSharedPreferences(MyPREFERENCES,
Context.MODE_PRIVATE);

        if (sharedPreferences.contains("has_send_sms")){
            has_send_sms = sharedPreferences.getBoolean("has_send_sms",false);
        }
        if (sharedPreferences.contains("has_alarm")){
            has_alarm = sharedPreferences.getBoolean("has_alarm",false);
        }
        if (sharedPreferences.contains("admin_number")){
            phoneNumber =
sharedPreferences.getString("admin_number","0984889913");
        }
    }
}

```

+ Cài đặt thông tin cấu hình để hiển thị lên giao diện:

```

    public void setOption() {

        sharedPreferences =
getSharedPreferences(MyPREFERENCES,Context.MODE_PRIVATE);

        if (sharedPreferences.contains("has_send_sms")){
            send_sms.setChecked(has_send_sms);
        }
    }
}

```

```

    }
    if (sharedpreferences.contains("has_alarm")){
        chbx_alarm.setChecked(has_alarm);
    }
    if (sharedpreferences.contains("admin_number")){
        adminNumber.setText(phoneNumber);
    }
}
}

```

***) Quản lý các sự kiện:**

+ Khi nhấn nút "Áp dụng":

```

OnClickListener apply_action = new OnClickListener() {
    @Override
    public void onClick(View v) {
        callAlertDialog();
    }
};

public void callAlertDialog() {
    AlertDialog.Builder alerDialogBuilder = new
AlertDialog.Builder(MainActivity.this);
    alerDialogBuilder.setTitle("Xác nhận");
    alerDialogBuilder.setMessage("Bạn đã chắc chắn chưa?");
    alerDialogBuilder.setIcon(R.drawable.ic_launcher);

    alerDialogBuilder.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Editor editor = sharedpreferences.edit();
            editor.putBoolean("has_send_sms", send_sms.isChecked());
            editor.putBoolean("has_alarm", chbx_alarm.isChecked());
        }
    });
}

```

```

        editor.putString("admin_number", adminNumber.getText().toString());
        editor.commit();

        getOption();
        Toast.makeText(getApplicationContext()
            , "Chế độ cài đặt:\n" +
            "Gửi cảnh báo bằng sms:" + String.valueOf(has_send_sms)
            + "\n" +
            "Gửi cảnh báo bằng chuông:" + String.valueOf(has_alarm)
            + "\n" +
            "Số điện thoại người dùng là:" + phoneNumber + "\n"
            , Toast.LENGTH_LONG).show();
    }

    });
    alertDialogBuilder.setNegativeButton("Bỏ qua", new
    DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            setOption();
            dialog.cancel();
        }
    });
    alertDialogBuilder.show();
}

```

Mô tả:

- Khi nhấn vào nút áp dụng: Một cảnh báo sẽ hiện lên hỏi xem ta đã chắc chắn hay chưa
- Nếu chọn OK thì: Toàn bộ thông tin cài đặt sẽ được cập nhật và sharedPreferences
- Nếu chọn Bỏ qua thì: sẽ không có gì thay đổi trong thông tin cài đặt

+ Khi nhấn nút bật tắt hệ thống:

```
OnClickListener action_switch = new OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if(start == 1) {  
            start = 0;  
            btn_switch.setText("Tắt");  
            countSMS=0;  
            if(mediaPlayer.isPlaying()) mediaPlayer.stop();  
        }  
        else if(start == 0) {  
            start = 1;  
            btn_switch.setText("Bật");  
        }  
    }  
};
```

Mô tả:

- Khi start == 1 thì nghĩa là hệ thống đang bật. Nếu ta nhấn vào thì hệ thống sẽ tắt, start =0,countSMS=0(tức ta reset lại) nếu có nhạc cảnh báo đang được bật thì nó sẽ tắt.
- Khi start == 0 thì nghĩa là hệ thống đang tắt. Nếu ta nhấn vào thì hệ thống sẽ chuyển sang trạng thái bật.

+ Quản lý sự thay đổi của cảm biến:

```
private final SensorEventListener LightSensorListener = new SensorEventListener()  
{  
    @Override  
    public void onAccuracyChanged(Sensor sensor, int accuracy) {  
    }  
    @Override  
    public void onSensorChanged(SensorEvent event) {
```

```

if(event.sensor.getType() == Sensor.TYPE_LIGHT){
    if(event.values[0]>30) {
        getOption();
        if(start==1) {
            countSMS++;
            if(countSMS==1) {
                if(has_alarm==true && has_send_sms==false) {
                    mediaPlayer.start();
                }
                else if(has_alarm==false && has_send_sms==true) {
                    try {
                        SmsManager smsManager = SmsManager.getDefault();
                        smsManager.sendTextMessage(phoneNumber, null,
msg,null,null);

                        Toast.makeText(
                            getApplicationContext(),
                            "MessageSent",
                            Toast.LENGTH_LONG).show();
                    } catch (Exception e) {
                        Toast.makeText(
                            getApplicationContext(),
                            e.getMessage().toString(),
                            Toast.LENGTH_LONG).show();
                        e.printStackTrace();
                    }
                }
            } else if(has_alarm==true && has_send_sms==true) {
                mediaPlayer.start();
                try {
                    SmsManager smsManager = SmsManager.getDefault();
                    smsManager.sendTextMessage(phoneNumber, null,
msg,null,null);

                    Toast.makeText(
                        getApplicationContext(),
                        "Message Sent",
                        Toast.LENGTH_LONG).show();
                } catch (Exception e) {
                    Toast.makeText(
                        getApplicationContext(),

```



```

        e.getMessage().toString(),
        Toast.LENGTH_LONG).show();
        e.printStackTrace()
    }
}
}
}
}
}
};

```

Mô tả: Khi có sự thay đổi của cảm biến sáng:

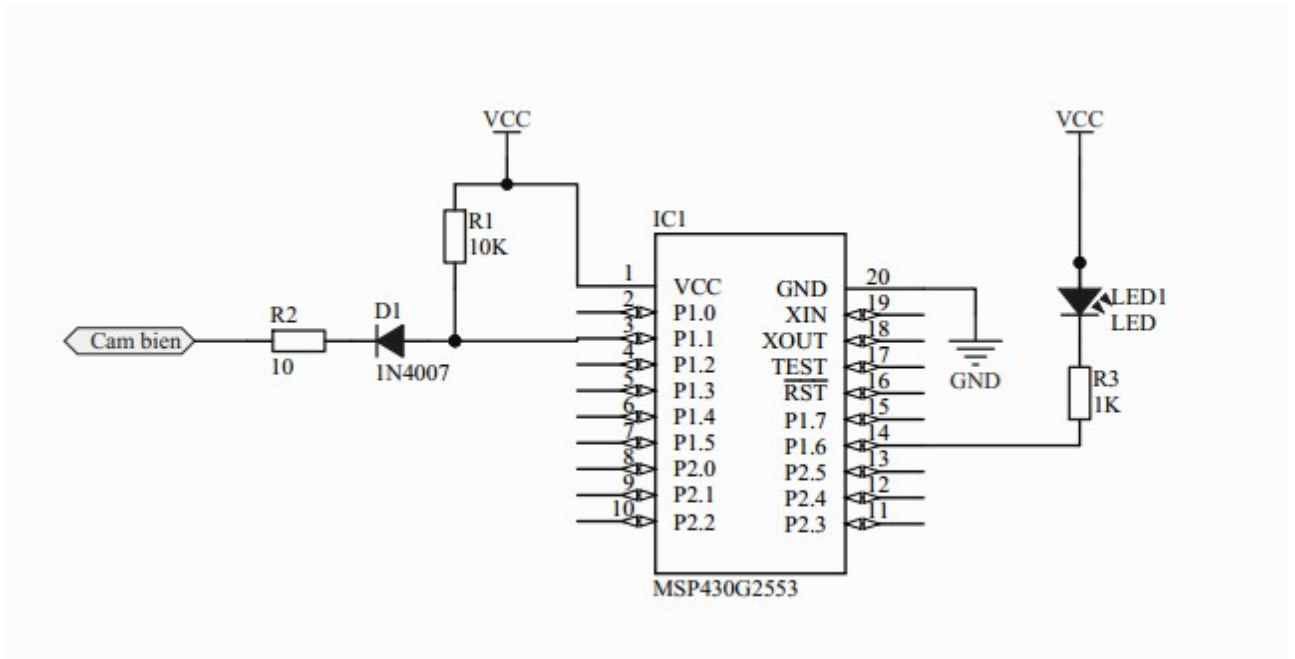
- Đại lượng mô tả giá trị độ sáng đo được là: `event.values[0]`
- Ta sẽ so sánh giá trị trên với 30. Khi nó lớn hơn 30 thì ta bắt đầu cảnh báo
- Ta kiểm tra biến `start` để xem hệ thống đang được bật hay tắt. Khi `start = 1` nghĩa là hệ thống được bật và ta sẽ cảnh báo
- Lúc này biến `countSMS++` để đảm bảo ta chỉ gửi tin nhắn SMS một lần
- Sau đó ta kiểm tra xem các trường hợp của việc: có gửi tin nhắn hay không, có cảnh báo chuông hay không. Ta có 4 trường hợp xảy ra.
- Nếu có cảnh báo chuông thì ta sẽ cho `mediaPlayer.start()`; để bật nhạc cảnh báo
- Nếu có cảnh báo qua SMS thì ta sẽ gửi tin nhắn thông qua `SmsManager` với số điện thoại là `phoneNumber` được lấy từ `sharePreferences` và nội dung là: "Cảnh báo có người xâm nhập"

2.2. Khối vi điều khiển, cảm biến và camera

2.2.1. Vi điều khiển

+ Có thể sử dụng bất kỳ một loại vi điều khiển nào. Ở đây nhóm dùng Kit MSP430G2553

+ Sơ đồ mạch nguyên lý là:



+ Ở đây:

- VCC là điện áp dương nguồn: giá trị 3.3 V
- Đầu vào nối với cảm biến chuyển động qua trở và diode như hình vẽ
- Đầu ra là đèn LED dùng để điều khiển, giúp cảm biến ánh sáng của điện thoại android có thể nhận biết được.

+ Code nạp cho vi điều khiển:

```
#include <msp430.h>

/*
 * main.c
 */
void main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
    BCSCTL1 = CALBC1_1MHZ;      //2 thanh ghi cài đặt tần số hoạt động
    DCOCTL = CALDCO_1MHZ;
    P1DIR |= BIT6;
    P1DIR &= ~BIT1;
    while(1){

        if(P1IN & BIT1){        // Neu nhan duoc tin hieu tu cam bien den led se sang
            P1OUT |= BIT6;
        }
        else P1OUT &= ~BIT6;    // Neu khong nhan duoc, deo led tat

    }
}
```

2.2.2. Cảm biến

+ Cảm biến dùng là loại cảm biến tiệm cận



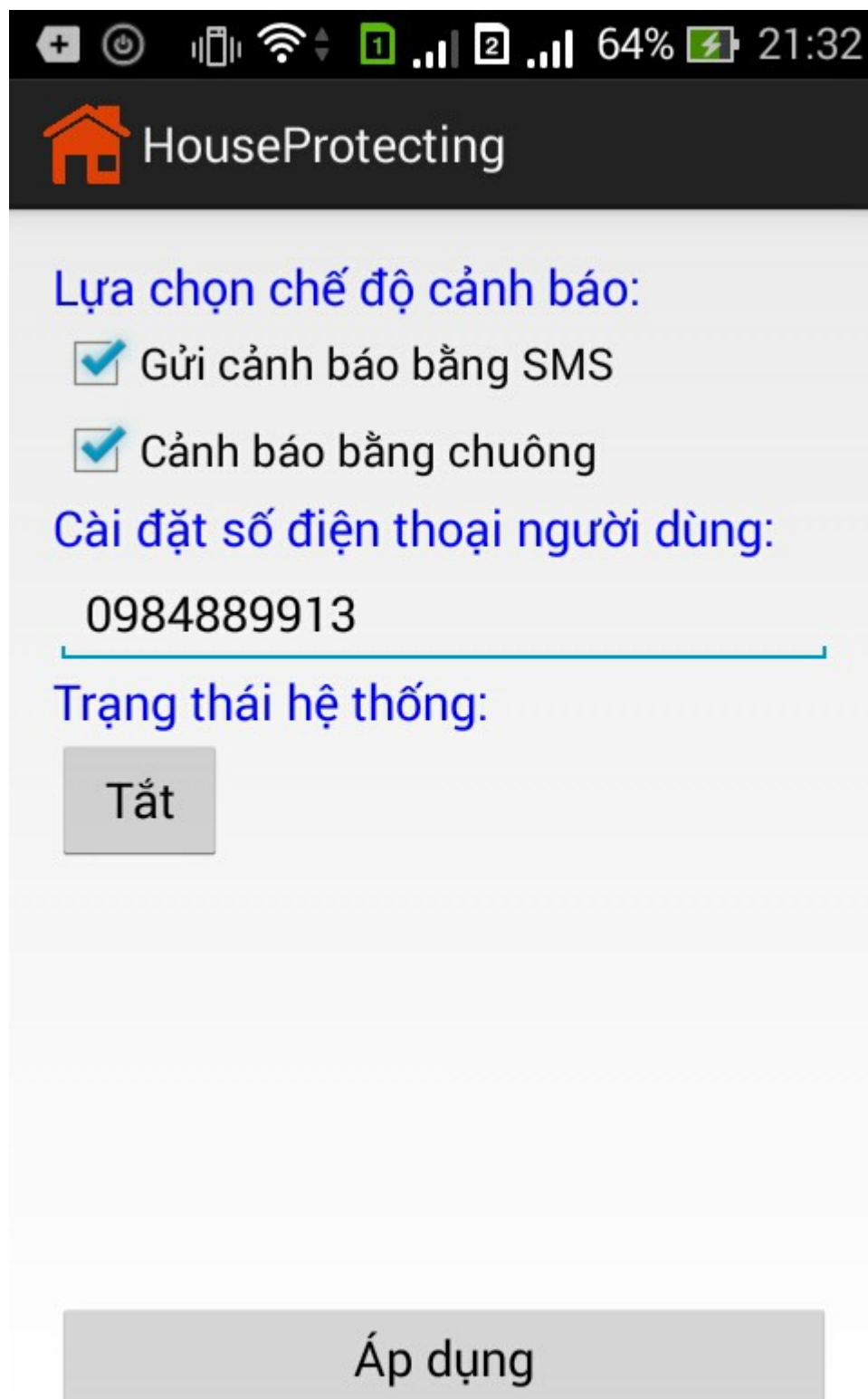
2.2.3. Camera

+ Camera loại: Vivoteck



3. Kết quả

* Giao diện phần mềm:



The screenshot shows the 'HouseProtecting' application interface on a mobile device. At the top is a status bar with icons for cellular signal, Wi-Fi, battery (64%), and time (21:32). Below the status bar is a dark header with an orange house icon and the text 'HouseProtecting'. The main content area has a light gray background and contains the following elements: a blue heading 'Lựa chọn chế độ cảnh báo:', two checked checkboxes for 'Gửi cảnh báo bằng SMS' and 'Cảnh báo bằng chuông', a blue heading 'Cài đặt số điện thoại người dùng:', a text input field containing '0984889913', a blue heading 'Trạng thái hệ thống:', a gray button labeled 'Tắt', and a large gray button at the bottom labeled 'Áp dụng'.

HouseProtecting

Lựa chọn chế độ cảnh báo:

- ☒ Gửi cảnh báo bằng SMS
- ☒ Cảnh báo bằng chuông

Cài đặt số điện thoại người dùng:

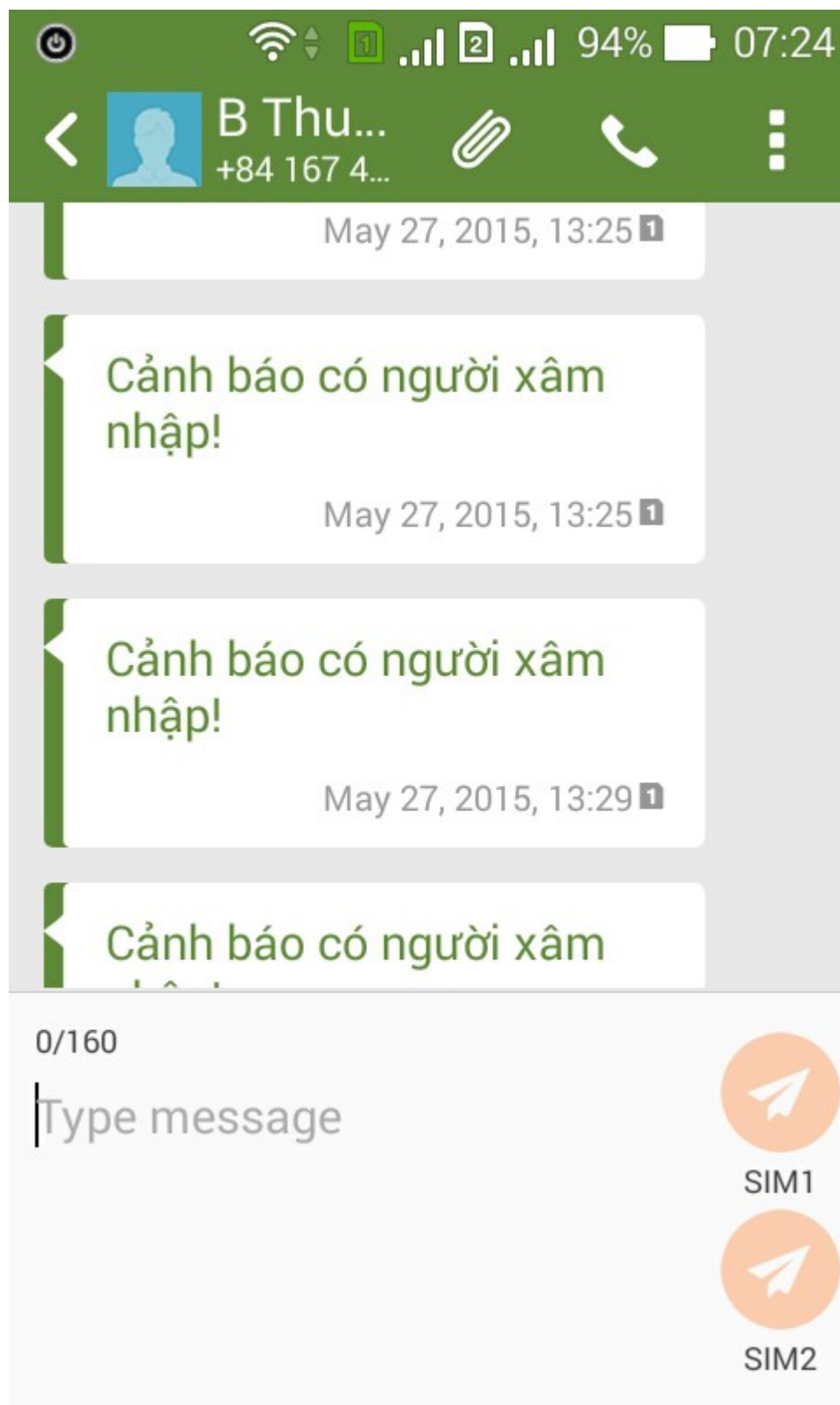
0984889913

Trạng thái hệ thống:

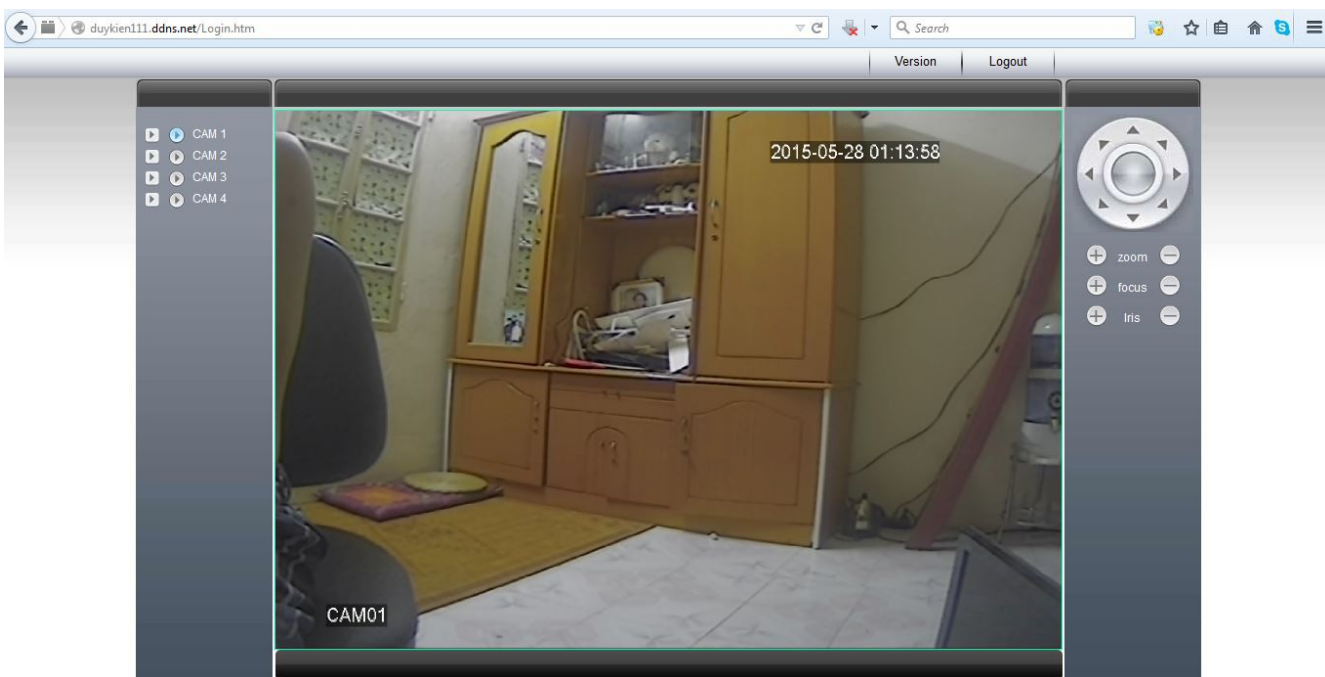
Tắt

Áp dụng

* Tin nhắn nhận được ở điện thoại người dùng:



* Video xem từ máy tính:



* Video xem từ điện thoại:

