

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC NHA TRANG  
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ  
ĐIỀU KHIỂN THIẾT BỊ SỬ DỤNG ĐIỆN THOẠI  
THÔNG QUA BLUETOOTH SỬ DỤNG KIT ARDUINO**

**Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh**

**Sinh viên thực hiện: Nguyễn Nhật Quỳnh**

**Mã số sinh viên: 61130967**

**KHÁNH HÒA - 2021**

**TRƯỜNG ĐẠI HỌC NHA TRANG**  
**KHOA CÔNG NGHỆ THÔNG TIN**  
**BỘ MÔN HỆ THỐNG THÔNG TIN**



**BÁO CÁO THỰC TẬP CƠ SỞ**  
**ĐIỀU KHIỂN THIẾT BỊ SỬ DỤNG ĐIỆN THOẠI**  
**THÔNG QUA BLUETOOTH SỬ DỤNG KIT ARDUINO**

Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh

Sinh viên thực hiện: Nguyễn Nhật Quỳnh

Mã số sinh viên: 61130967

Khánh Hòa, tháng 01/2021

**TRƯỜNG ĐẠI HỌC NHA TRANG**  
**Khoa: Công nghệ Thông tin**

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ BÁO CÁO THỰC TẬP CƠ SỞ**  
**Tên đề tài: ĐIỀU KHIỂN THIẾT BỊ SỬ DỤNG ĐIỆN THOẠI THÔNG QUA**  
**BLUETOOTH SỬ DỤNG KIT ARDUINO**

**Giảng viên hướng dẫn:** ThS. Đoàn Vũ Thịnh

**Sinh viên được hướng dẫn:** Nguyễn Nhật Quỳnh

**MSSV:** 61130967

**Khóa:** 60

**Ngành:** Công nghệ Thông tin

<b>Lần</b>	<b>Ngày</b>	<b>Nội dung</b>	<b>Nhận xét của GVHD</b>
1	7/12/2020	Nhận đề tài hướng dẫn và định hướng giải quyết vấn đề. Sau đó, tiến hành cài đặt phần mềm Proteus và tiến hành mô phỏng chức năng của các linh kiện, mô đun chức năng.	Sinh viên và GVHD trao đổi nội dung của đề tài. Sinh viên có nhiều phương án để triển khai, tuy nhiên cần kiểm thử các phương pháp trước khi đưa ra phương án cuối cùng thông qua phần mềm mô phỏng và linh kiện điện tử.
2	14/12/2020	Sinh viên trình bày ý tưởng và giải pháp của mình. Sinh viên mô phỏng phần mềm trên Proteus thể hiện chức năng hiện giờ, cài đặt bộ hẹn giờ.	Sinh viên có ý tưởng rõ ràng về nội dung thực hiện. Phần mô phỏng có kết quả tốt thể hiện được các yêu cầu đặt ra, tuy nhiên về mặt thực tế sản phẩm có thể gặp lỗi, đề nghị tiến hành kiểm thử trên mạch thực.
3	21/12/2020	Sinh viên trình bày kết quả mô phỏng và hoàn tất quá trình này. Về mặt phần cứng Sinh viên cần chuẩn bị thêm kiến thức cần thiết về điện tử, điện tử số.	Kết thúc quá trình mô phỏng sinh viên đã đáp ứng đầy đủ các yêu cầu đặt ra. Về giao diện trên điện thoại, sinh viên cần tìm hiểu về ứng dụng App Inventor. Về phần thiết kế mạch cần tìm hiểu Sprint Layout.
4	4/01/2021	Sinh viên nộp bản thảo của báo cáo thực tập lần thứ 1 và tiến hành chỉnh sửa.	Báo cáo chỉ trình bày chung chung chưa đi vào cụ thể phân tích các yêu cầu của bài toán, hình ảnh,

			bảng biểu chưa trình bày rõ ràng. Cần hiệu chỉnh theo yêu cầu của GVHD.
5	11/01/2021	Sinh viên nộp bản thảo lần 2 và trình diễn App Inventor. Mạch điều khiển, sinh viên không sử dụng phần thiết kế mạch do không đủ thời gian và dùng module relay sẵn có	Apps có giao diện khá đơn giản và cần chỉnh sửa thêm. Về mạch điện, cho phép sinh viên sử dụng module có sẵn trên thị trường để điều khiển nhưng yêu cầu hiểu nguyên lý. Chức năng điều khiển tốt.
6	18/1/2021	Sinh viên nộp báo cáo lần cuối sau khi đã hoàn thiện phần cứng.	Về phần báo cáo đã chỉnh sửa hoàn chỉnh. Phần mềm cơ bản đã hoạt động tốt. Cần khắc phục 1 số yêu cầu như giao diện, tính năng điều khiển, đóng gói sản phẩm.

**Nhận xét chung (sau khi sinh viên hoàn thành ĐA/KL):**

Sinh viên thực hiện tốt các yêu cầu của GVHD, trong quá trình thực hiện đề tài có sự liên hệ chặt chẽ với GV. Ngoài ra, cũng cần ghi nhận sự nỗ lực hết mình của các bạn, với lượng kiến thức còn hạn chế về điện tử và điện tử số nhưng cơ bản đề tài đã hoàn thành khối lượng công việc đã đặt ra trước đó.

Về nội dung báo cáo đã thỏa mãn các yêu cầu của đề tài như trong đề cương. Về phần mạch thực và kết quả thực hiện, đề tài có kết quả mô phỏng tốt và cũng thể hiện được kết quả đó trên sản phẩm thực tế.

Về hình thức của báo cáo và sản phẩm, báo cáo trình bày rõ ràng các mục tiêu, phương pháp, kết quả và thảo luận cho sản phẩm. Còn về sản phẩm chưa thực hiện việc ghép các mô đun vào mạch điện tổng thể (chưa thiết kế được mạch in) cho sản phẩm hoàn thiện.

Điểm hình thức: 8.3/10 Điểm nội dung: 8.3/10 **Điểm tổng kết: 8.3/10**

Đồng ý cho sinh viên: Được bảo vệ: ☒ Không được bảo vệ: ☐

Khánh Hòa, ngày 20 tháng 01 năm 2021

Cán bộ hướng dẫn

(Ký và ghi rõ họ tên)

## MỤC LỤC

DANH MỤC HÌNH .....	ii
DANH MỤC BẢNG .....	ii
TÓM TẮT.....	1
PHẦN 1. GIỚI THIỆU .....	2
1.1. App Inventor .....	2
1.2. Module Bluetooth (Module Bluetooth HC-05).....	4
1.3. Kit Arduino (Arduino UNO R3).....	5
1.4. Module Relay .....	7
1.5. Arduino IDE.....	7
1.7. IC lập trình được Atmega16u2 .....	9
PHẦN 2. PHƯƠNG PHÁP NGHIÊN CỨU .....	10
2.1. Mô phỏng phần mềm Proteus .....	10
2.2. Lập trình giao diện .....	12
2.3. Giao tiếp Arduino với HC-05 .....	16
2.4. Điều khiển thiết bị với Arduino .....	17
2.5. Điều khiển bằng điện thoại thông qua bluetooth HC-05 với rơ le.....	18
PHẦN 3. KẾT QUẢ ĐẠT ĐƯỢC .....	19
3.1. Giao diện phần mềm .....	19
3.2. Điều khiển thiết bị.....	20
KẾT LUẬN .....	20
PHỤ LỤC .....	21

## DANH MỤC HÌNH

Hình 1. 2 Sơ đồ chân và Module Bluetooth HC-05 .....	4
Hình 1. 3 Sơ đồ chân của Arduino Uno R3.....	5
Hình 1. 4. Module Relay (5V).....	7
Hình 1. 5. Giao diện sử dụng Arduino IDE.....	8
Hình 1. 6. Phần mềm mô phỏng mạch điện với Proteus .....	8
Hình 1. 7. Sơ đồ chân của IC Atmega16u2 .....	9
Hình 2. 1. Sơ đồ nguyên lý mạch điều khiển quạt thông qua Arduino .....	10
Hình 2.2. Sơ đồ tính toán giá trị điện trở RB .....	10
Hình 2.3. Giao diện thiết kế Project .....	12
Hình 2.4. Cửa sổ lập trình Code Block .....	13
Hình 2.5. Giao diện chính.....	13
Hình 2.6. Kết nối với thiết bị phát Bluetooth .....	14
Hình 2.7. Khởi chức năng để điều khiển thiết bị.....	14
Hình 2.8. Khởi chức năng để tắt thiết bị.....	14
Hình 2.10. Khởi chức năng để hiển thị thời gian hiện tại .....	15
Hình 2.11. Khởi chức năng thiết lập thời gian tắt .....	16
Hình 2.12. Khởi chức năng để so sánh thời gian hiện tại và thời gian thiết lập.....	16
Hình 2.13. Mã QR của ứng dụng .....	16
Hình 2.14. Khai báo dữ liệu .....	17
Hình 2.15. Đọc dữ liệu nhận được từ Bluetooth và cho vào biến data .....	17
Hình 2.16. Hàm bật tắt, điều khiển các chân.....	17
Hình 2.17. Hàm cài đặt giờ tắt.....	18
Hình 3.1. Giao diện phần mềm.....	19
Hình 3.2. Quạt khi chưa điều khiển.....	20
Hình 3.3. Trạng thái của quạt khi điều khiển .....	20

## DANH MỤC BẢNG

Bảng 1.1. Thông số của Arduino Uno R3 .....	6
Bảng 1.2. Thông số kỹ thuật của Module Relay .....	7

## TÓM TẮT

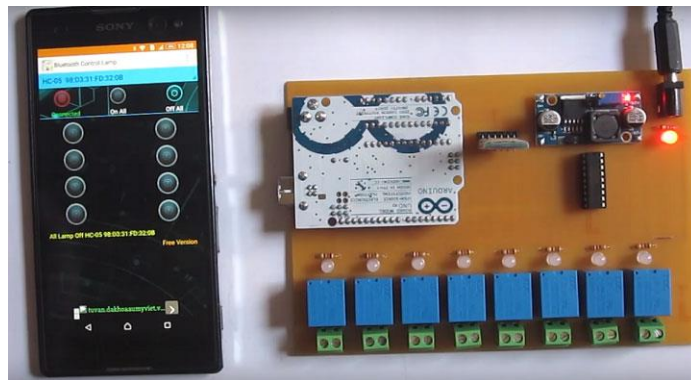
Trong thời đại 4.0 công nghệ ngày càng trở nên phát triển, mọi công việc đều trở nên dễ dàng nhờ vào các ứng dụng được thiết kế trên thiết bị di động. Để việc điều khiển các thiết bị trong nhà chẳng hạn như đèn, quạt, ... trở nên dễ dàng, thuận tiện hơn, ứng dụng thiết kế Apps sẽ thực hiện được điều đó. Ngoài ra, nó còn giúp tiết kiệm thời gian và vô cùng tiện lợi. Yêu cầu của bài toán là thiết kế, thi công, lắp đặt sản phẩm để thực hiện các chức năng đơn giản bao gồm: điều khiển quạt, hiển thị thời gian kết nối bluetooth thông qua App trên thiết bị di động Android. App có thể hoạt động sau khi tải về như các ứng dụng thông thường mà không cần phải kết nối mạng. Để thực hiện các công việc này cần sử dụng đến các thành phần, linh kiện điện tử như: module lập trình Arduino uno R3, Module Bluetooth HC-05, module relay, các thành phần hỗ trợ khác như dây cáp.

Kết quả thực hiện cho thấy sản phẩm có thể điều khiển quạt, cài đặt thời gian tắt cho quạt, kết nối bluetooth và hiển thị thời gian kết nối như yêu cầu đặt ra. Toàn bộ sản phẩm được đóng gói, mã nguồn được trình bày trong phụ lục và được upload theo địa chỉ: <https://github.com/thinhdoanvu/ThuctapCoSo2020/tree/main/NguyenNhatQuynh>

## PHẦN 1. GIỚI THIỆU

Ngày nay, điện thoại thông minh đã và đang trở thành một thiết bị vô cùng phổ biến trong cuộc sống con người. Việc sử dụng điện thoại thông minh để điều khiển và giám sát các thiết bị thông qua hệ điều hành Android hỗ trợ rất nhiều cho con người và trở thành một thiết bị tiện dụng, hiệu quả. Hệ điều hành Android là hệ điều hành di động phổ biến nhất hiện nay với thị phần hơn 50% trên thế giới và có mã nguồn mở dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Các thiết bị đang chạy bằng hệ điều hành Android như: Samsung, Sony, Oppo, HTC,...

**Hình 1.** là một ví dụ điển hình về hệ thống điều khiển thiết bị bằng Android thông qua sóng Bluetooth. Thành phần cơ bản bao gồm: điện thoại thông minh để hiển thị giao diện được thiết kế bởi App Inventor, mạch điều khiển gồm Arduino, Bluetooth HC-05.



*Hình 1 Điều khiển từ xa bằng module bluetooth HC-05 thông qua smart phone*

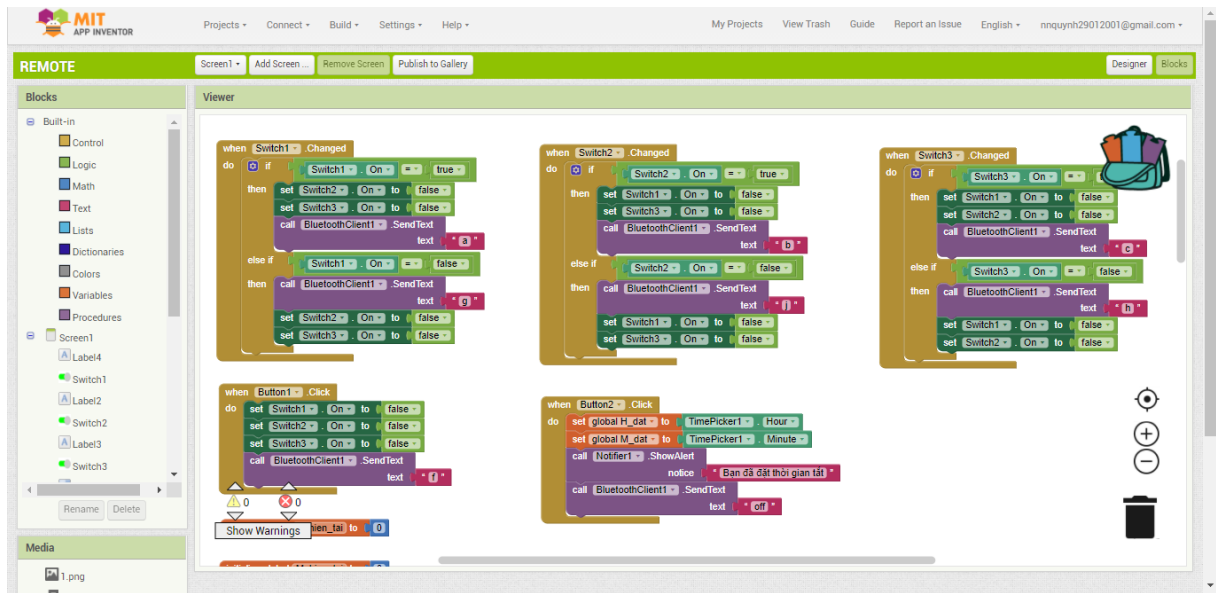
(Nguồn: [chotroi.vn](http://chotroi.vn))

### 1.1. App Inventor

Ngày 12/7/2010, Google chính thức giới thiệu công cụ lập trình trực quan App Inventor dùng để phát triển phần mềm ứng dụng trên hệ điều hành Android. Phiên bản mới nhất là MIT App Inventor 2. App Inventor thực chất là một ứng dụng web, chạy bởi trình duyệt trên máy tính cá nhân. Tuy nhiên, người dùng vẫn phải cài đặt một phần mềm Java mang tên App Inventor Extras, có nhiệm vụ điều khiển điện thoại Android (kết nối với máy tính thông qua cổng USB). Nhờ vậy, người dùng có thể nhanh chóng chuyển ứng dụng từ máy tính cá nhân qua điện thoại Android để chạy thử. Nhóm dự án



App Inventor tại Google dự định bổ sung những bộ mô phỏng điện thoại Android để có thể sử dụng App Inventor mà không cần kết nối thiết bị Android với máy tính.



Hình 1. 1. Giao diện lập trình Block của App Inventor

Những tính năng có trên MIT App Inventor:

(1) Cho phép xây dựng nhanh chóng những thành phần cơ bản (components) của một ứng dụng Android: Nút bấm, nút lựa chọn, chọn ngày giờ, ảnh, văn bản, thông báo, kéo trượt, trình duyệt web.

(2) Sử dụng nhiều tính năng trên điện thoại: Chụp ảnh, quay phim, chọn ảnh, bật video hoặc audio, thu âm, nhận diện giọng nói, chuyển lời thoại thành văn bản, dịch.

(3) Hỗ trợ xây dựng game bằng các components: Ball, Canvas, ImageSprite.

(4) Cảm biến: đo gia tốc (AccelerometerSensor), đọc mã vạch, tính giờ, con quay hồi chuyển (gyroscopeSensor), xác định địa điểm (locationSensor), NFC, đo tốc độ (pedometer), đo khoảng cách xa gần với vật thể (proximitySensor).

(5) Kết nối: Danh bạ, email, gọi điện, chia sẻ thông qua các ứng dụng mạng xã hội khác trên thiết bị, nhắn tin, sử dụng twitter qua API, bật ứng dụng khác, bluetooth, bật trình duyệt.

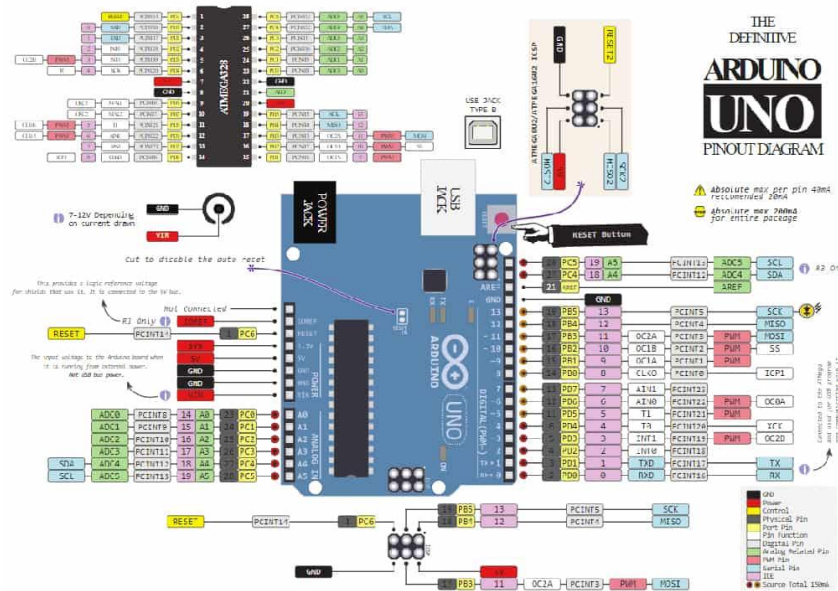
(6) Lưu trữ: đọc hoặc lưu tệp txt, csv, sử dụng FusionTablesControl, tạo cơ sở dữ liệu đơn giản trên điện thoại hoặc trên đám mây thông qua server tự tạo hoặc Firebase.

(7) Điều khiển robot thông qua LegoMindstorms; Và rất nhiều mở rộng do các nhà lập trình hoạt động riêng liên tục thêm vào như là: Mua bán trong ứng dụng, Floating button, Báo thức, cảm biến ánh sáng, kết nối dữ liệu SQLite.

- ✓ Bluetooth protocol: Bluetooth v2.0
- ✓ Tần số: 2.4GHz ISM band
- ✓ Tốc độ: 2.1Mbps/160kbps (Asynchronous), 1Mbps/1Mbps (Synchronous)
- ✓ Bảo mật: Authentication and encryption
- ✓ Giao tiếp: Bluetooth serial port
- ✓ Nguồn hoạt động: +3.3VDC 30mA ( Hỗ trợ IC 5.0V)
- ✓ Nhiệt độ làm việc: -20 ~ +75 Độ C
- ✓ Passkey: 1234 - Device Name: HC-05
- ✓ **Sơ đồ chân HC-05 gồm có:**
- ✓ KEY: Chân này để chọn chế độ hoạt động AT Mode hoặc Data Mode.
- ✓ VCC: chân này có thể cấp nguồn từ 3.6V đến 6V bên trong module đã có một IC nguồn chuyển về điện áp 3.3V và cấp cho IC BC417.
- ✓ GND: nối với chân nguồn GND .
- ✓ TXD và RXD đây là hai chân UART để giao tiếp module.

### 1.3. Kit Arduino (Arduino UNO R3)

Để đọc dữ liệu từ module bluetooth HC-05 và hiển thị đèn Led và quạt cần sử dụng Arduino UNO R3 ([http://electronoobs.com/eng\\_arduino\\_tut31\\_sch3.php](http://electronoobs.com/eng_arduino_tut31_sch3.php)). Arduino - một máy tính nhỏ để người dùng có thể lập trình và thực hiện các dự án điện tử mà không cần phải có các công cụ chuyên biệt để phục vụ việc nạp code.



Hình 1. 3 Sơ đồ chân của Arduino Uno R3

#### Cấu tạo:

- (1) USB: Cổng giao tiếp USB có 2 chức năng: cấp nguồn cho board mạch và truyền thông nối tiếp với máy tính để nạp chương trình hay giao tiếp nối tiếp.
- (2) Jack Power: Nguồn cấp cho board Arduino. Có 2 loại nguồn có thể sử dụng được là nguồn xoay chiều (6VAC) và nguồn một chiều (5VDC).
- (3) RESET: Đặt lại trạng thái ngay khi nạp chương trình
- (4) ISCP: Chân giao tiếp với USB, tín hiệu giao tiếp có thể giám sát từ đây.
- (5) AREF và GND: GND hay chân Mass hay chân đất dùng để cấp nguồn 0V cho các module khác có kết nối với Arduino. Trong khi chân AREF được dùng để phối hợp với các chân Analog Input (11) để điều chỉnh dải điện áp đầu vào cho ADC 10 bit.
- (6) DIGITAL Input/Output: 12 chân tín hiệu số được dùng làm đầu vào hoặc đầu ra tại mỗi thời điểm.
- (7) Serial Transmission: 2 chân RXD và TXD được dùng trong truyền thông nối tiếp. 2 chân này cũng có thể được cấu hình làm đầu vào/ra tín hiệu số.

(8) PWM: Chân băm xung có ký hiệu là ~ dùng để tạo ra chuỗi xung vuông trong điều khiển tốc độ động cơ hay hiển thị LED sáng dần hay tối dần.

(9) ICSP: Giao tiếp với Atmega328.

(10) IC lập trình ATMEGA16u2

(11) ANALOG Input: Tín hiệu chuyển đổi ADC được đưa vào các chân này. Trong trường hợp khác các chân này cũng có thể sử dụng để làm đầu vào/ra tín hiệu số.

(12) POWER: Cung cấp nguồn 5V, GND cho các thành phần mở rộng.

**Bảng 1.1. Thông số của Arduino Uno R3**

Vi điều khiển	ATmega328 họ 8bit
Điện áp hoạt động	5V DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	16 MHz
Dòng tiêu thụ	khoảng 30mA
Điện áp vào khuyến dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	14 (6 chân hardware PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 mA
Bộ nhớ flash	32 KB (ATmega328) với 0.5KB dùng bởi bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

Arduino UNO R3 có chức năng và bộ xử lý trung tâm của Arduino UNO R3 cũng gần giống với Arduino Mega 2560 với các chân điều khiển ít hơn. Arduino UNO R3 với vi xử lý trung tâm là Atmega328 có 14 chân I/O tín hiệu số, trong đó 6 chân có thể được sử dụng làm bộ điều chế độ rộng xung PWM, 6 ngõ vào tín hiệu tương tự, sử dụng thạch anh dao động 16MHz, kết nối USB, có ICSP Header.

#### 1.4. Module Relay

Module Relay là một loại linh kiện điện tử thụ động rất hay gặp trong các ứng dụng thực tế. Khi bạn gặp các vấn đề liên quan đến công suất và cần sự ổn định cao.

Module tập hợp các linh kiện điện tử (transistor, tụ điện, diot, cuộn cảm) được gắn kết trên một bo mạch và thực hiện một chức năng nhất định.

Rơ le là một công tắc điều khiển, nó dùng một dòng nhỏ để điều khiển một dòng lớn, vì vậy nó cũng được xem là một thiết bị bảo vệ.



Hình 1. 4. Module Relay (5V)

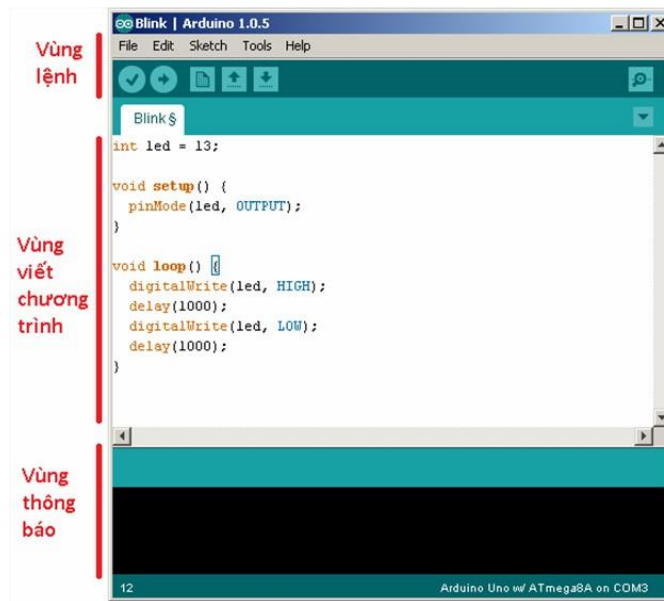
**Bảng 1.2. Thông số kỹ thuật của Module Relay**

Thông số	Giá trị
Điện áp tải tối đa	AC 250V-10A / DC 30V-10A
Điện áp điều khiển	5 VDC
Dòng kích Relay	5mA
Trạng thái kích	Mức thấp (0V)
Đường kính Lỗ ốc	3.1 mm
Kích thước	50 * 26 * 18.5 mm

#### 1.5. Arduino IDE

Arduino IDE được viết tắt (Arduino Integrated Development Environment) là một trình soạn thảo văn bản, giúp bạn viết code để nạp vào bo mạch Arduino.

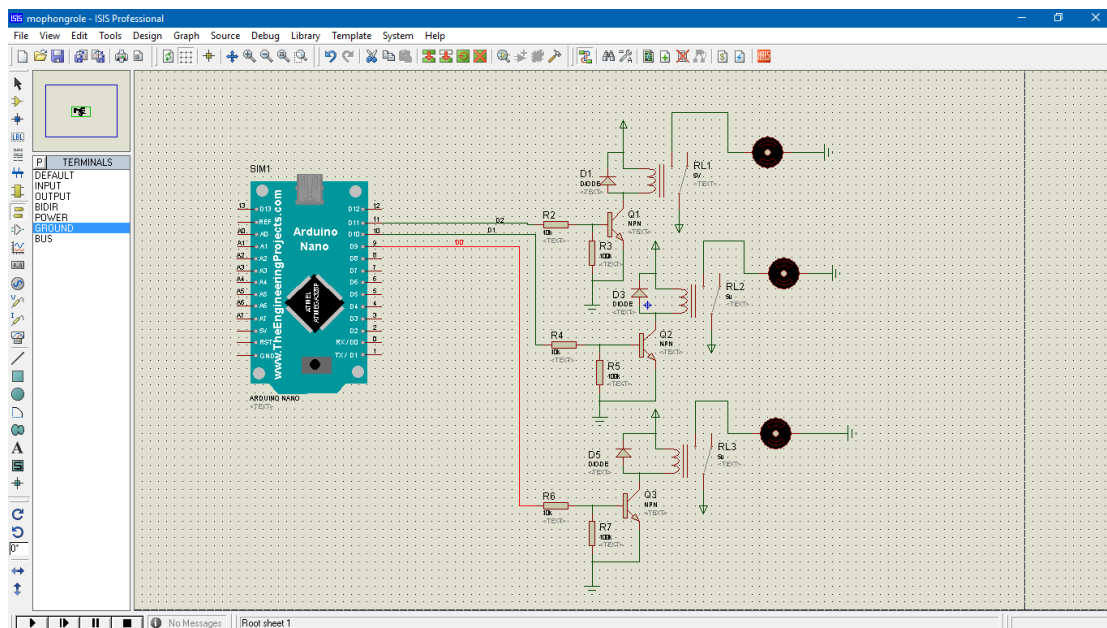
Arduino ide được viết bằng ngôn ngữ lập trình Java là ứng dụng đa nền tảng (cross-platform). Ngôn ngữ code cho các chương trình của arduino là bằng C hoặc C++ . Bản thân arduino ide đã được tích hợp một thư viện phần mềm thường gọi là "wiring", từ các chương trình "wiring" gốc sẽ giúp bạn thực hiện thao tác code dễ dàng hơn. Một chương trình chạy trong arduino được gọi là một sketch, chương trình được định dạng dưới dạng .ino .



Hình 1. 5. Giao diện sử dụng Arduino IDE

(Nguồn: <http://arduino.vn/reference/getting-started/step3a> )

## I.6. Phần mềm Proteus



Hình 1. 6. Phần mềm mô phỏng mạch điện với Proteus

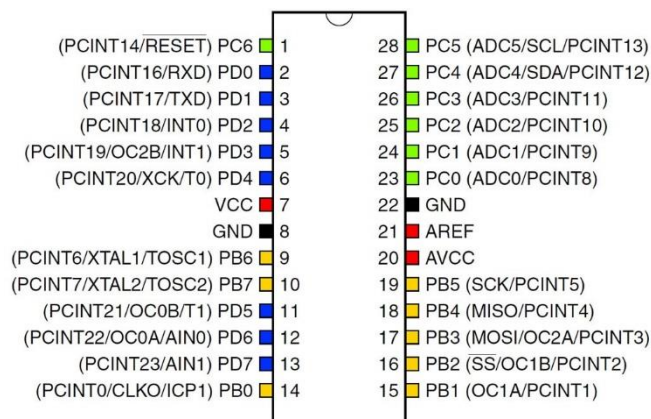
Proteus là phần mềm cho phép mô phỏng hoạt động của mạch điện tử bao gồm phần thiết kế mạch và viết chương trình điều khiển cho các họ vi điều khiển như MCS-51, PIC, AVR, ... Proteus là phần mềm mô phỏng mạch điện tử của Lanceter Electronics, mô phỏng cho hầu hết các linh kiện điện tử thông dụng, đặc biệt hỗ trợ cho cả các MCU như PIC, 8051, AVR, Motorola. Phần mềm bao gồm 2 chương trình: ISIS cho phép mô phỏng mạch và ARES dùng để vẽ mạch in. Proteus là công cụ mô phỏng cho các loại vi điều khiển khá tốt, nó hỗ trợ các dòng VĐK PIC, 8051, PIC, dsPIC,



AVR, HC11, MSP430, ARM7/LPC2000, giao tiếp I2C, SPI, CAN, USB, Ethernet. Proteus được sử dụng rất phổ biến cho việc mô phỏng các mạch điện tử. Quan trọng nhất là thanh thư viện nơi chứa các linh kiện và vùng làm việc dùng để vẽ mạch nguyên lý (Hình 1.6). Ngoài ra, các nút mô phỏng được sử dụng cho việc mô phỏng nguyên lý hoạt động (Hình 1.6 bên dưới cùng).

### 1.7. IC lập trình được Atmega16u2

Arduino R3 không thể hoạt động mà không cần đến bộ não của board mạch này. Atmega16u2 được sản xuất theo kiến trúc RISC (Reduce Instruction Set Computer) tiên tiến bao gồm: 125 tập lệnh đơn lệnh đơn chu kỳ; 32 thanh ghi 8 bit dùng chung; Tối đa 16 tập lệnh MIPS (Microprocessor without Interlocked Pipeline Stages) với tần số cho phép 16 MHz. Một bộ đếm 8 bit (Timer/Counters) và 1 bộ đếm 16 bit (Timer/Counter) dùng cho điều biến độ rộng xung PWM. Giao tiếp USART với chế độ SPI master và chế độ kiểm soát phản ứng (RTS/CTS). Bộ chuyển đổi tín hiệu ADC, công cụ thời gian thực và khả năng hoạt động ở mức điện áp thấp trong chế độ tiết kiệm năng lượng.



Hình 1. 7. Sơ đồ chân của IC Atmega16u2

#### Cấu tạo:

22 chân đa dụng vừa làm tín hiệu vào vừa làm tín hiệu ra tín hiệu số (PB0 - PB5 ~ chân 14 – chân 19, PC0 – PC6 ~ chân 23 – chân 28 và chân 1, PD0 – PD7 ~ chân 2 – chân 6 và chân 11 – chân 13).

2 chân nối với bộ dao động thạch anh tần số 16MHz (chân 8 và chân 9).

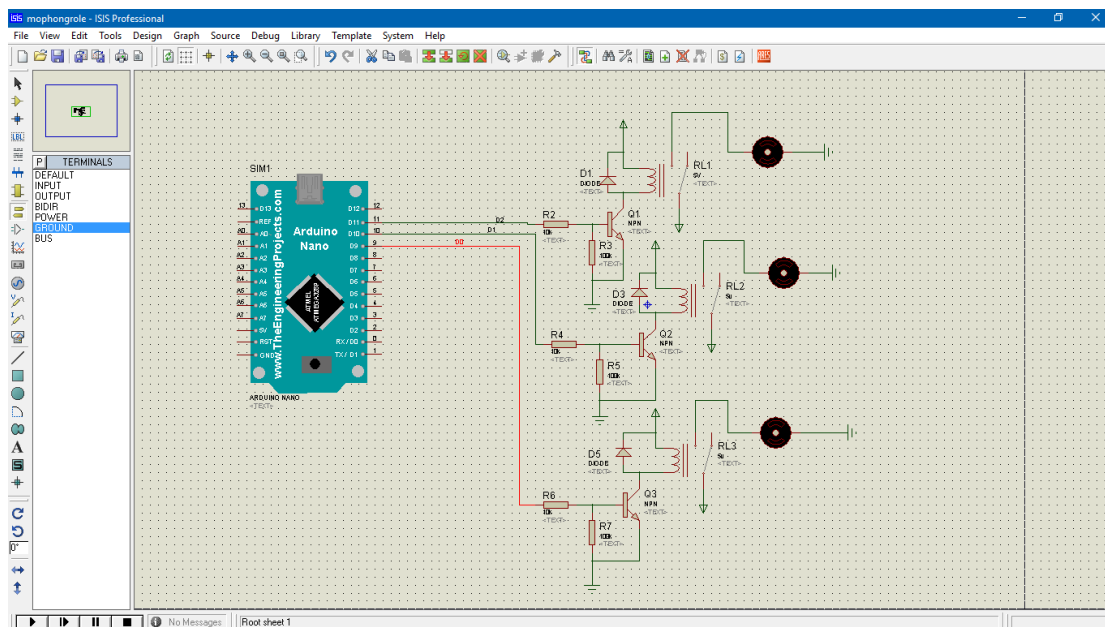
3 chân nguồn VCC và GND (chân 7, 8 và chân 22) cấp nguồn 5V và Mass.

6 chân ADC (Analog Digital Converter): bộ chuyển đổi ADC 10 bit (chân 23 – chân 28) phối hợp với chân AREF (chân 21) hiệu chỉnh giá trị điện áp đầu vào cho ADC.

## PHẦN 2. PHƯƠNG PHÁP NGHIÊN CỨU

### 2.1. Mô phỏng phần mềm Proteus

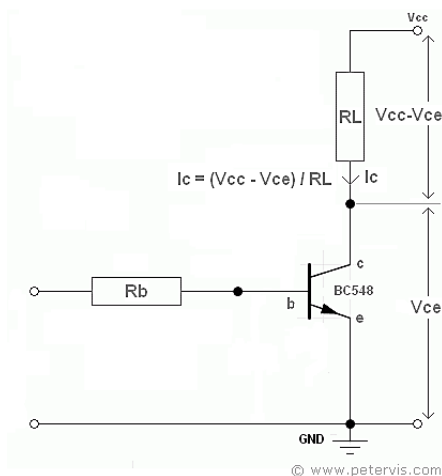
Để điều khiển các tải kết nối với các nút của quạt ta dùng các bit D0, D1, D2. Trong đó các bit D0, D1, D2 được nối với chân digital 8, 9, 10 của Arduino.



Hình 2. 1. Sơ đồ nguyên lý mạch điều khiển quạt thông qua Arduino

Nhìn vào sơ đồ ta thấy, chân số 9, 10 và 11 của Arduino được nối với 3 điện trở (R2, R4 và R6) được sử dụng để kích cho transistor loại NPN (Q1, Q2 và Q3). Các điện trở R3, R5, R7 là các điện trở hồi tiếp ngăn trường hợp transistor quá tải sẽ đưa dòng điện trở về đất. Các diode (D1, D2, D3) được dùng làm mạch bảo vệ trong trường hợp Relay quá dòng. Cuối cùng 3 motor được mô phỏng như 3 cuộn dây của quạt máy.

Để tính toán giá trị cho các R2, R4, R6 sử dụng công thức sau:



Hình 2.2. Sơ đồ tính toán giá trị điện trở RB



Giả sử, theo quy ước của nhà sản xuất NPN có thông số như sau:

- ✓  $V_{be} (ON) = 0.77 \text{ V}$
- ✓  $V_{ce} (SAT) = 0.25 \text{ V}$
- ✓  $h_{FE} = 110$

Trong đó,  $h_{FE}$  là hệ số khuếch đại dòng điện 1 chiều. Theo đó, hệ số khuếch đại dòng  $I_C$  được tính toán:  $I_C = h_{FE} \times I_B$

Từ hình 2.2, điện áp rơi trên tải  $R_L$  được tính theo công thức  $V_{cc} - V_{ce}$ . Với  $V_{cc}$  là điện áp nguồn cung cấp. Do đó, theo định luật Ohm ta có:

$$I_C = (V_{cc} - V_{ce}) / R_L$$

$I_C$  còn được gọi là dòng tiêu thụ cực đại (theo quy ước của nhà sản xuất,  $I_C$  (MAX)). Ví dụ với transistor BC548 thì dòng tiêu thụ cực đại là 500 mA.

Khi đó:

- ✓ Điện áp rơi trên cực B được tính bằng:  $V_i - V_{be}$
- ✓ Dòng  $I_B$  được tính theo công thức  $R = V / I$ .

$$\text{Từ đó: } R_b = (V_i - V_{be}) / (3 \times I_B)$$

**Ví dụ:**

Sử dụng rowle loại 5V với điện trở nội là  $69.4 \Omega$ , sử dụng transistor BC548.

Sử dụng tín hiệu đầu ra Arduino để điều khiển (trạng thái mức logic 1 tương đương với 3.3V). Từ đó,  $V_i = 3.3V$ . Transistor được cấp nguồn 5V hay  $V_{cc} = 5V$ . Hãy tính toán giá trị  $R_b$ ?

**Bước 1: Tính toán dòng  $I_C$**

$$I_C = (5 - 0.25) / 69.4$$

$$I_C = 0.068 \text{ A}$$

**Bước 2: Tính toán dòng  $I_B$**

$$h_{FE} = 110$$

$$I_B = 0.068 / 110$$

$$I_B = 0.00062 \text{ A}$$

**Bước 3: Tính toán giá trị  $R_b$**

$$R_b = (3.3 - 0.77) / (3 \times 0.00062)$$

$$R_b = 1360.21 \Omega$$

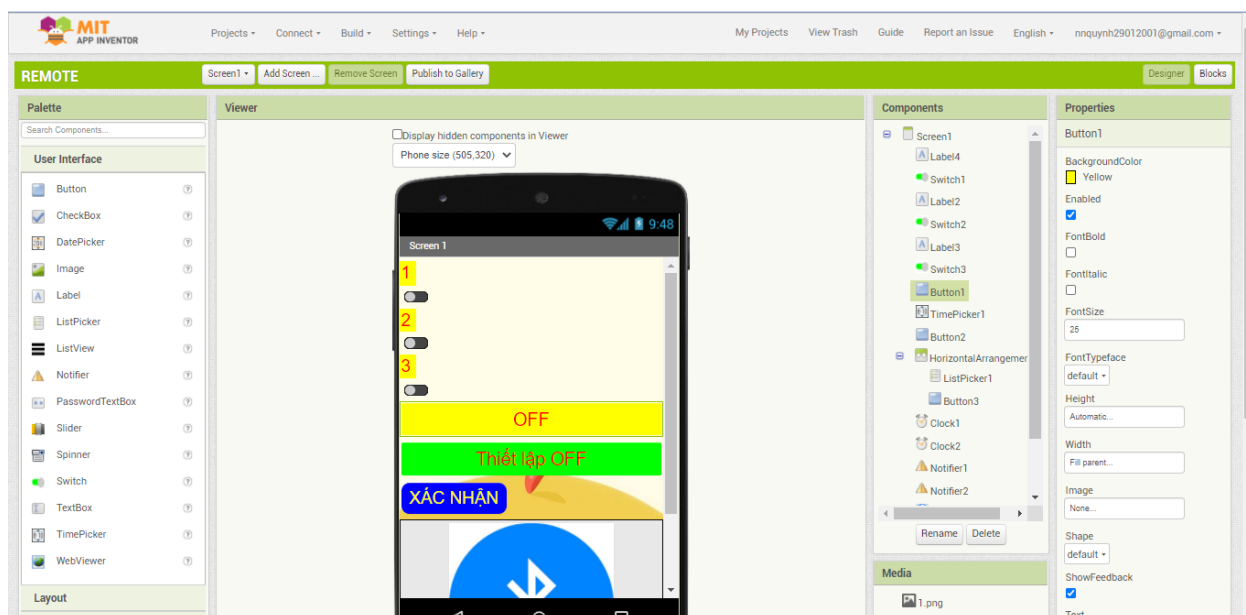
Trong thực tế, ta có thể sử dụng điện trở 1200  $\Omega$  thay vì tìm điện trở 1360.21  $\Omega$

Ở trạng thái bình thường các chân digital của Arduino bằng 0 thì các thiết bị tải không hoạt động. Khi chân digital 8, 9, 10 có giá trị bằng 1 thì D0, D1, D2 sẽ mang giá trị là 1, khi đó (1) có dòng điện chạy qua. Transistor lúc đó sẽ có một dòng điện khép kín đi từ chân số (2) sang chân (3) của Role (Relay). Bây giờ cuộn dây sẽ tạo thành một lực từ, lực từ này hút chân (5) từ chân số (6) về chân (4). Tạo ra vòng điện khép kín từ chân (4) nối với chân (5) và nối đất – motor sẽ quay. Tương tự với các bit khác khi điều khiển các tải khác.

## 2.2. Lập trình giao diện

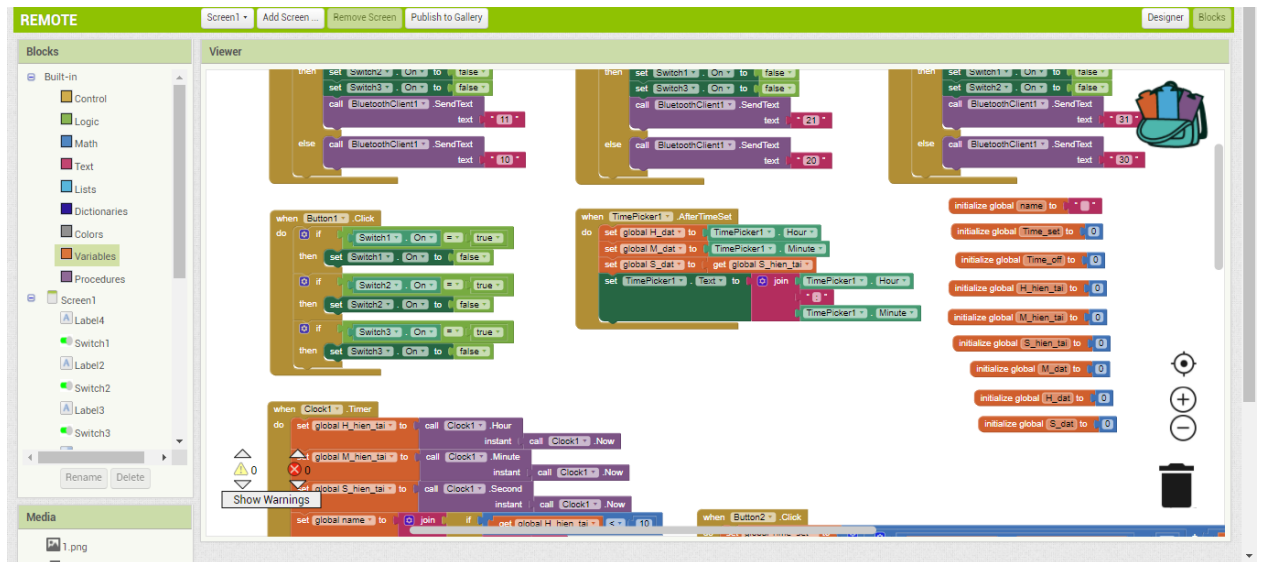
Truy cập vào địa chỉ <http://ai2.appinventor.mit.edu>, đăng ký tài khoản để tạo dự án (project) cho việc xây dựng một ứng dụng (apps). Có thể sử dụng địa chỉ Email làm tài khoản đăng nhập, khởi tạo project, thiết kế giao diện, lập trình block.

**Hình 2.3** thể hiện giao diện cửa sổ thiết kế ứng dụng MIT APP INVENTOR. Bên trái là các control bao gồm: User Interface, Media, Sensor, Social (Hình 2.3 trái) chỉ cần click và drag đối tượng cần thiết kế vào khu vực thiết kế (Hình 2.3 giữa). Khu vực bên phải dùng để thiết lập các thuộc tính cho đối tượng được thiết kế (Hình 2.3 phải).



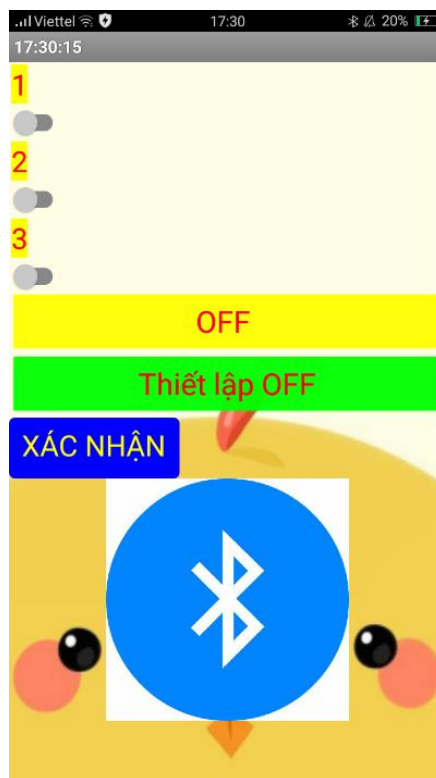
Hình 2.3. Giao diện thiết kế Project

Block - khối lệnh bao gồm Control, Logic, Math, Text,...và các khối lệnh ứng với mỗi button. Để thực hiện một lệnh nào đó, thực hiện kéo thả các Blocks này sang khung Viewer và kết nối chúng theo chức năng mong muốn.



Hình 2.4. Cửa sổ lập trình Code Block

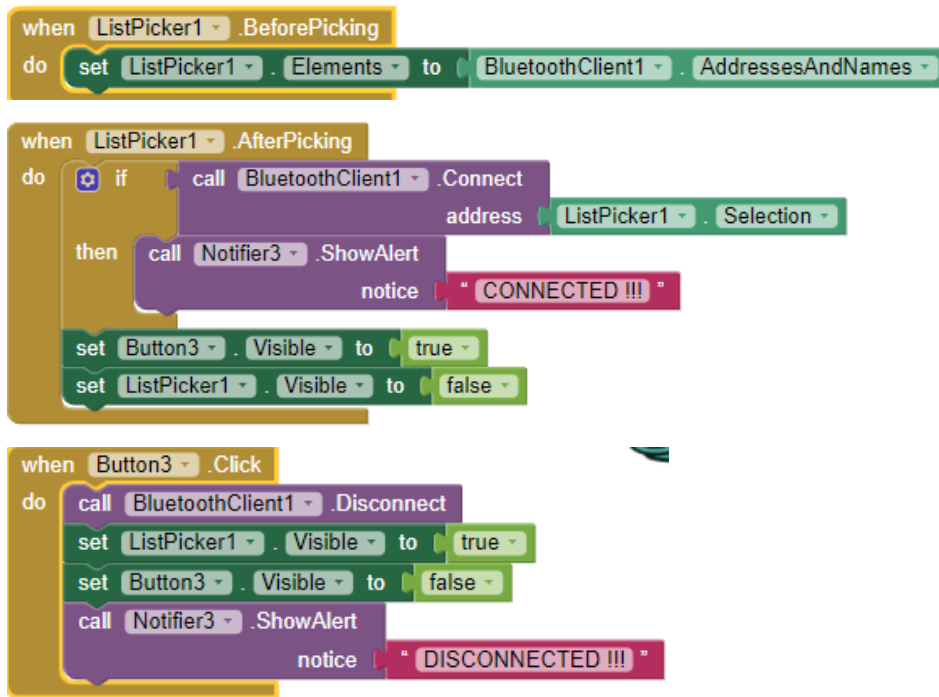
**Giao diện chính:** Gồm các chức năng: các nút điều khiển quạt (1,2,3); nút tắt các nút điều khiển quạt; hẹn thời gian tắt và cài đặt bluetooth.



Hình 2.5. Giao diện chính

### **Bước 1:** Kết nối Bluetooth.

Hình 2.6 thể hiện các khối chức năng để cài đặt Bluetooth. Khi nhấn vào Bluetooth, màn hình hiện thị danh sách các thiết bị bluetooth đang bật xung quanh nó, gồm có địa chỉ và tên thiết bị của các Bluetooth.

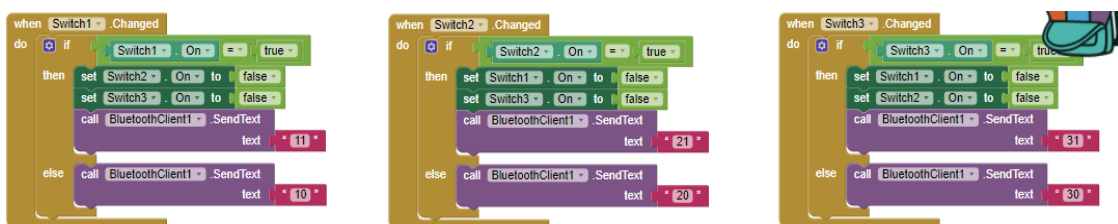


Hình 2.6. Kết nối với thiết bị phát Bluetooth

Chọn trong danh sách tên thiết bị mình muốn kết nối, khi được kết nối, sẽ hiện thị thông báo CONNECTED !!! và nút Button 3 hiện lên, Bluetooth ẩn đi.

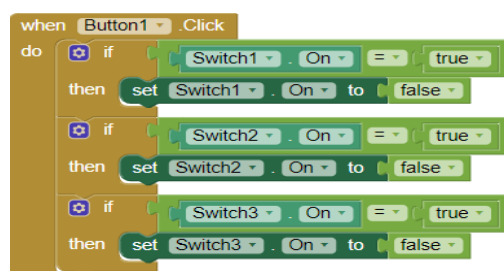
Khi không muốn kết nối nữa thì nhấn vào nút Button 3, khi Button 3 được nhấn khi đó nó sẽ ngắt kết nối Bluetooth, thông báo hiện lên DISCONNECTED !!! và Button 3 ẩn đi và nút Bluetooth lại hiện lên.

**Bước 2:** Bật 1 trong 3 nút điều khiển. Nếu bật nút 1, thì các nút còn lại sẽ tắt, sau đó gán giá trị tín hiệu cho nó, và các nút khác cũng vậy.



Hình 2.7. Khởi chức năng để điều khiển thiết bị

Nếu muốn tắt cả thì nhấn Button 1 (nút OFF), nút nào đang bật thì sẽ tắt đi.



Hình 2.8. Khởi chức năng để tắt thiết bị

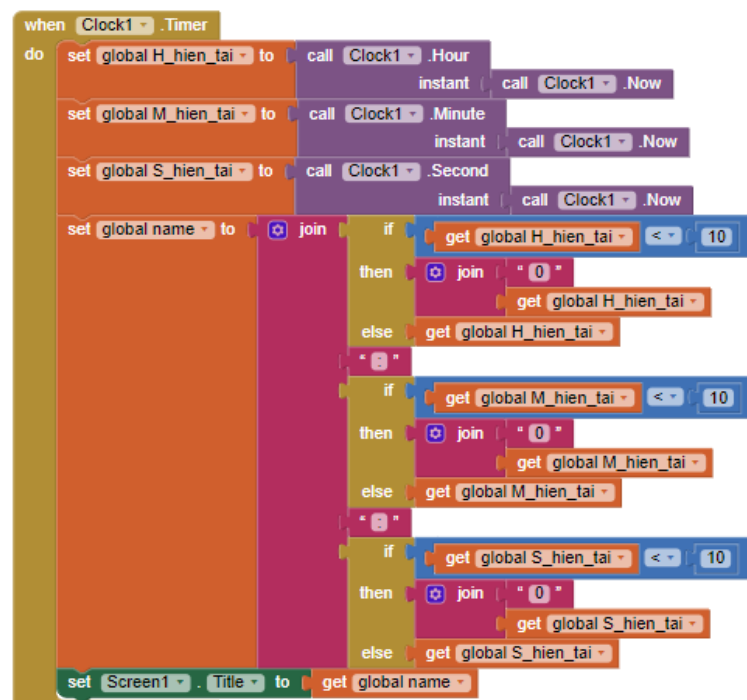
**Bước 3:** Thiết lập thời gian tắt cho thiết bị.

Khai báo giá trị thời gian hiện tại và thời gian ta đặt bằng 0.



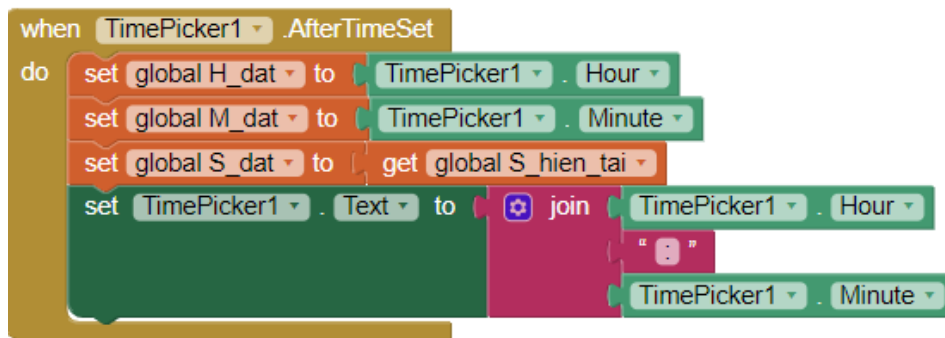
Hình 2.9. Khởi chức năng khai báo giá trị

### Hiện thị thời gian hiện tại



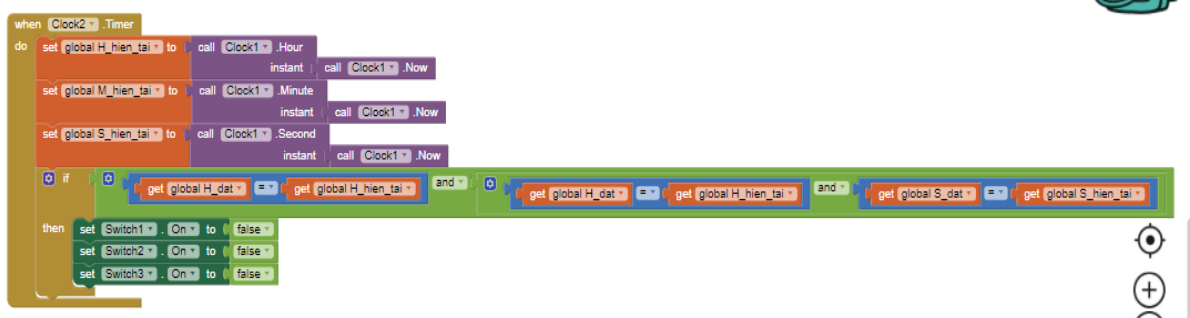
Hình 2.10. Khởi chức năng để hiển thị thời gian hiện tại

## Thiết lập thời gian muốn tắt gồm có giờ, phút, giây



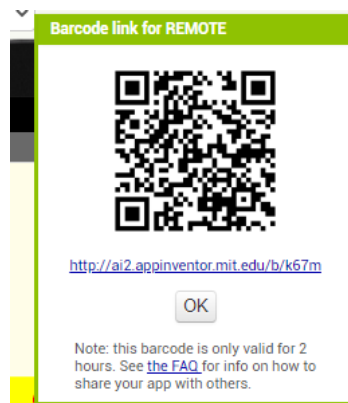
Hình 2.11. Khối chức năng thiết lập thời gian tắt

So sánh thời gian hiện tại và thời gian ta đã thiết lập, nếu thời gian hiện tại và thời gian thiết lập có cùng giờ, cùng phút và cùng giây thì các nút đang bật sẽ tự động tắt.



Hình 2.12. Khối chức năng để so sánh thời gian hiện tại và thời gian thiết lập

**Đóng gói ứng dụng thành file \*.apk:** Sau khi trải qua giai đoạn thiết kế, ứng dụng được đóng gói với tập tin apk. Theo các bước: Build → App (provide QR code cho file apk) → tìm đến đường dẫn chứa tập tin apk trên máy tính.



Hình 2.13. Mã QR của ứng dụng

## 2.3. Giao tiếp Arduino với HC-05

Để giao tiếp Arduino chân TX của Bluetooth HC-05 được nối với chân RX của Arduino, chân này có nhiệm vụ phát tín hiệu khi Bluetooth HC-05 được tín hiệu của

ứng dụng trên App Inventor đã được kết nối với Bluetooth, khi đó chân TX sẽ phát tín hiệu truyền đến chân RX của Arduino, Arduino sẽ nhận tín hiệu đó và xử lý. Và ngược lại, khi chân TX của Arduino phát tín hiệu vào chân RX của Bluetooth thì Bluetooth sẽ nhận tín hiệu và xử lý trên ứng dụng đã được kết nối.

## 2.4. Điều khiển thiết bị với Arduino

```
#include<string.h>
int nut1=8,nut2=9,nut3=10; //Khai báo chân cần hoạt động
int d=0;
int val1;
String data; //Cho tín hiệu truyền là kiểu chuỗi
void setup() {
    Serial.begin(9600); //Tốc độ truyền tín hiệu của bluetooth là 9600
    //thiết lập các chân đầu ra
    for (int i=8;i<=10;i++) pinMode(i,OUTPUT);
}
```

Hình 2.14. Khai báo dữ liệu

```
void loop() {
    if(Serial.available()>0)
    {
        data = Serial.readString();
        Serial.println("ket noi thanh cong");
        Serial.println(data);
        if(data[0]!='@') ON_OFF();
        if(data[0]=='@') Time_off();
    }
}
```

Hình 2.15. Đọc dữ liệu nhận được từ Bluetooth và cho vào biến data

```
,
void Time_off()//Hàm hẹn giờ tắt
{
    d=0;
    Serial.println(data); //Hiển thị dữ liệu đã đọc để kiểm tra
    data.remove(0,2); // xóa 2 ký tự đầu của dữ liệu nhận được
    String myStr="";
    myStr = String(d); //biến kiểu dữ liệu của d từ số nguyên sang chuỗi
    Serial.println(data); //Hiển thị dữ liệu đã đọc được sau khi xóa 2 ký tự đầu.
    while (myStr!=data) //Thực hiện vòng lặp đến khi biến d bằng dữ liệu đọc được sau cùng
    {
        d++; //Biến d đếm thời gian (phút) đến khi bằng dữ liệu được nhận
        myStr=String(d);
        delay(59000); //Tạm dừng 1 phút
    }
    {
        digitalWrite(8,LOW); //Khi biến d bằng dữ liệu nhận được thì tắt hết các chân đang hoạt động.
        digitalWrite(9,LOW);
        digitalWrite(10,LOW);
    }
}
```

Hình 2.16. Hàm bật tắt, điều khiển các chân

```

void ON_OFF()
{
    for (int i=0;i<=data.length();i=i+2)
    {
        if(data[i]=='1') //Nếu dữ liệu ở vị trí đầu là 1 thì xử lý chân 8
            if (data[i+1]=='1') digitalWrite(8,HIGH); //Nếu dữ liệu ở vị trí tiếp theo là 1 thì bật sáng
            else digitalWrite(8,LOW) //Nếu không phải là 1 thì tắt đi
        if(data[i]=='2')
            if (data[i+1]=='1') digitalWrite(9,HIGH);
            else digitalWrite(9,LOW);
        if(data[i]=='3') //Các Chân còn lại tương tự
            if (data[i+1]=='1') digitalWrite(10,HIGH);
            else digitalWrite(10,LOW);
    }
}

```

*Hình 2.17. Hàm cài đặt giờ tắt*

## 2.5. Điều khiển bằng điện thoại thông qua bluetooth HC-05 với rơ le

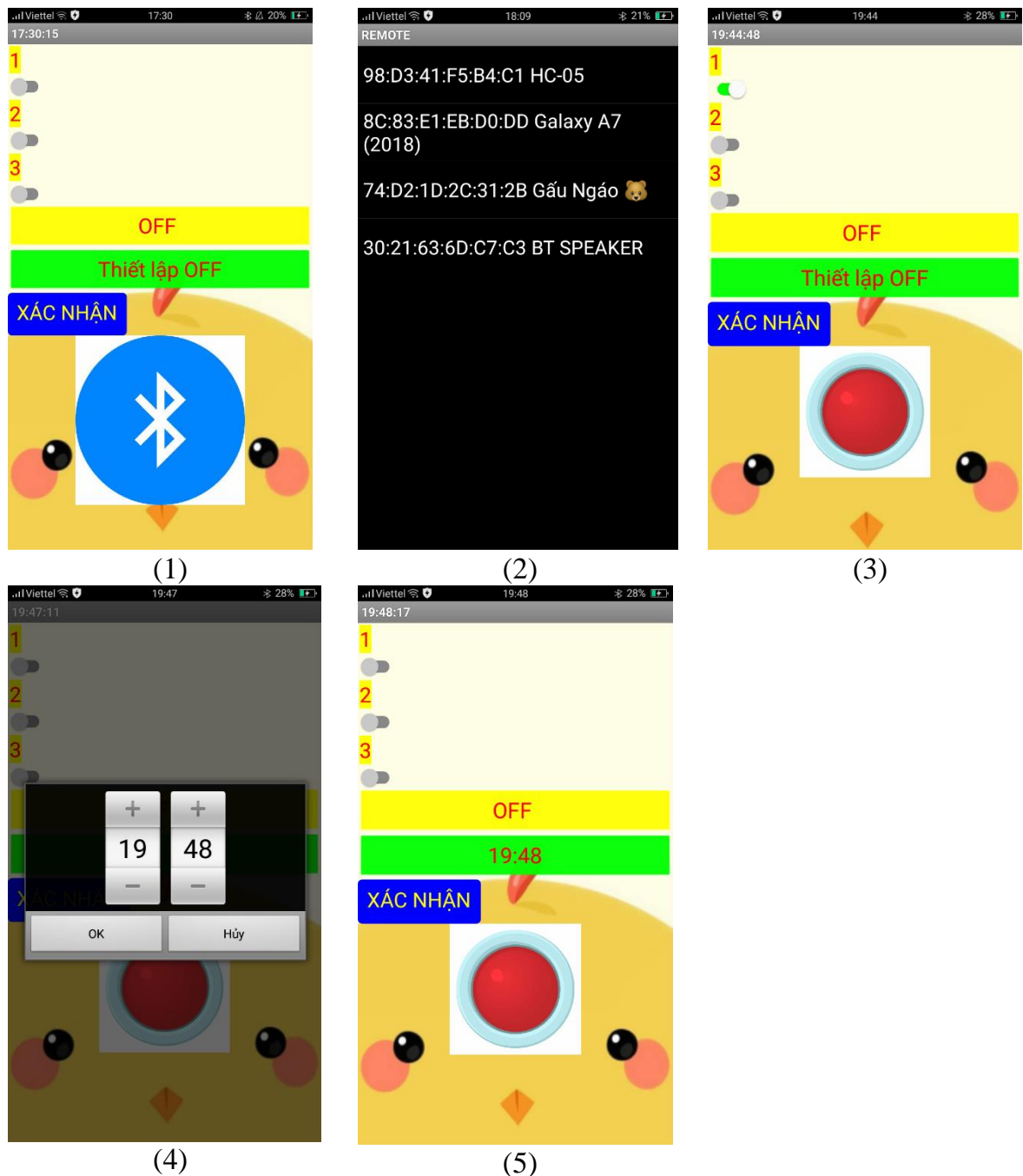
Để điều khiển rơ le thông qua bluetooth HC-05, thực hiện theo trình tự các bước:

- (1) Kết nối cổng nguồn 5V trên Arduino vào cổng nguồn 5V trên mạch rơ le.
- (2) Kết nối thiết bị quạt vào cổng đầu ra trên board mạch rơ le.
- (3) Kết nối nguồn điện 220V vào cổng nguồn 220V trên board mạch rơ le.
- (4) Mở ứng dụng App Inventor và điều khiển.



## PHẦN 3. KẾT QUẢ ĐẠT ĐƯỢC

### 3.1. Giao diện phần mềm



Hình 3.1. Giao diện phần mềm

Theo thứ tự từ hình (1) đến hình (5) của hình 3.1 là giao diện được thiết kế được mô tả như sau:

- (1): Giao diện chính khi mở app lên
- (2): Danh sách Bluetooth hiện ra khi nhấn vào hình Bluetooth
- (3): Sau khi kết nối Bluetooth sẽ hiện nút đỏ để ngắt kết nối, và ta có thể chọn nút ta muốn bật.

(4): Khi ta nhấn vào thiết lập OFF sẽ hiện lên thời gian và phút, chọn xong ta nút nhấn xác nhận.

(5): Màn hình khi thời gian hiện tại chạy đến thời gian ta đặt sẽ tự động tắt đi.

### 3.2. Điều khiển thiết bị



Hình 3.2. Quạt khi chưa điều khiển



Hình 3.3. Trạng thái của quạt khi điều khiển

Thiết bị được kết nối với module điều khiển và được điều khiển thông qua ứng dụng được thiết kế. Khi kết nối Bluetooth và nhấn nút số 1 trên app tương ứng với quạt điện sẽ được bật, nếu nhấn số 2 thì quạt sẽ chạy ở chế độ 2 và nút số 1 tắt đi. Ngoài ra, thời gian ta đặt sẽ hiện lên trên app, và thời gian hiện tại sẽ có trong app để tiện theo dõi. 2 module Relay có tác dụng đóng/mở thiết bị, còn Arduino có tác dụng truyền tín hiệu từ Bluetooth sang, điều khiển module Relay đóng ngắt thiết bị.

## KẾT LUẬN

Sản phẩm hoàn thiện đã thể hiện các tiêu chí mà đề tài thực tập cơ sở yêu cầu. Việc điều khiển thiết bị được thực hiện cho kết quả tốt tuy có độ trễ nhất định. Nguyên nhân có thể đến từ module HC-05 giao tiếp chưa tốt với Arduino. Tuy nhiên, số lượng thiết bị chỉ được thêm vào ngay khi thiết kế ứng dụng, sản phẩm chưa cho phép thêm số lượng thiết bị ngay bên trong ứng dụng.

## PHỤ LỤC

```
#include<string.h>

int nut1=8,nut2=9,nut3=10; //Khai báo chân cần hoạt động
int d=0; int val1;
String data; //Cho tín hiệu truyền là kiểu chuỗi
void setup() {
    Serial.begin(9600); //Tốc độ truyền tín hiệu của bluetooth là 9600
    //thiết lập các chân đầu ra
    for (int i=8;i<=10;i++) pinMode(i,OUTPUT);
}
void loop() {
    if(Serial.available(>0)           //Nếu nhận được tín hiệu từ bluetooth
    {
        data = Serial.readString();    //Đọc dữ liệu nhận được vào biến data
        Serial.println("ket noi thanh cong"); //Hiển thị dòng thông báo để biết được đã
        nhận dữ liệu
        Serial.println(data);           //Hiển thị dữ liệu đã đọc
        if(data[0]!='@') ON_OFF();      //Nếu vị trí đầu của dữ liệu không phải ký tự
                                         '@' thì thực hiện hàm bật tắt
        if(data[0]=='@') Time_off();     //Nếu vị trí đầu của dữ liệu là ký tự '@' thì thực
                                         hiện hàm hẹn giờ
    }
}
// hàm điều khiển bật tắt
void ON_OFF()
{
    for (int i=0;i<=data.length();i=i+2)
    {
        if(data[i]=='1')    //Nếu dữ liệu ở vị trí đầu là 1 thì xử lý chân 8
            if (data[i+1]=='1') digitalWrite(8,HIGH); //Nếu dữ liệu ở vị trí tiếp theo là
                                                         1 thì bật sáng
            else digitalWrite(8,LOW)    //Nếu không phải là 1 thì tắt đi
    }
}
```

```

    if(data[i]=='2')
        if (data[i+1]=='1') digitalWrite(9,HIGH);
        else digitalWrite(9,LOW);
    if(data[i]=='3')                //Các Chân còn lại tương tự
        if (data[i+1]=='1') digitalWrite(10,HIGH);
        else digitalWrite(10,LOW);
    }
}
}

//Hàm hẹn giờ tắt
void Time_off()
{
    d=0;
    Serial.println(data);           //Hiển thị dữ liệu đã đọc để kiểm tra
    data.remove(0,2);               // xóa 2 kí tự đầu của dữ liệu nhận được
    String myStr="";
    myStr = String(d);              //biến kiểu dữ liệu của d từ số nguyên sang chuỗi
    Serial.println(data);           //Hiển thị dữ liệu đã đọc được sau khi xóa 2 ký tự đầu.
    while (myStr!=data)             //Thực hiện vòng lặp đến khi biến d bằng dữ liệu đọc
                                    //được sau cùng
    {
        d++; //Biến d đếm thời gian (phút) đến khi bằng dữ liệu được nhận
        myStr=String(d);
        delay(59000);               //Tạm dừng 1 phút
    }
    {
        digitalWrite(8,LOW);        //Khi biến d bằng dữ liệu nhận được thì tắt hết các chân
                                    //đang hoạt động.

        digitalWrite(9,LOW);
        digitalWrite(10,LOW);
    }
}

```