MỤC LỤC

1. **Giới thiệu** 2

1.1 App Invertor 2

1.2 Kit Arduino (Adruino UNO R3) 3

1.3 Module Bluetooth ( Module Bluetooth HC-05 ) 5

1.4 Rơ le đóng ngắt (Relay) 7

1.5 Arduino IDE 8

1.6 Phần mềm thiết kế mạch và mô phỏng (Proteus 7.10SP2) 9

1.7 Phần mềm Sprint Layout Viewer 9

2. **Phương pháp nghiên cứu** 11

1.8 Thi công thiết kế mạch in với Sprint Layout 11

1.9 Mô phòng phần mềm Proteus 11

1.10 Lập trình giao diện 14

1.10.1 Giao diện thiết kế với App Inventor 14

1.10.2 Giao diện lập trình (Blocks) với App Inventor. 14

1.11 Giao tiếp Arduino với HC-05 18

1.12 Điều khiển thiết bị với Arduino 18

3. **Kết quả** 20

4. **Kết luận** 24

5. **Phụ lục** 25

Danh mục hình ảnh

[*Hình 1.1 . Điều khiển đèn huỳnh quang bằng điện thoại thông minh thông qua module HC-05* 1](#_Toc30140738)

[*Hình 1.1. Giao diện thiết kế trên MIT App Inventor 2* 2](#_Toc30140739)

[*Hình 1.2. Module Arduino UNO R3* 4](#_Toc30140740)

[*Hình 1.3. Sơ đồ chân Bluetooth HC05* 6](#_Toc30140741)

[*Hình 1.4. Rơ-le 5V 5 chân* 7](#_Toc30140742)

[*Hình 1.5. Giao diện của phần mềm viết code cho bo mạch Arduino* 8](#_Toc30140743)

[*Hình 1.6. Phần mềm thiết kế mạch và mô phỏng* 9](#_Toc30140744)

[*Hình 1.7. Phần mềm Sprint Layout Viewer* 9](#_Toc30140745)

[*Hình 2.1. Quy trình làm mạch in* 11](#_Toc30140746)

[*Hình 2.2. Sơ đồ nguyên lý điều khiển các thiết bị thông qua Arduino trong Proteus* 12](#_Toc30140747)

[*Hình 2.3. Khi tải kết nối với quạt không có dòng điện chạy qua* 12](#_Toc30140748)

[*Hình 2.4. Khi tải kết nối với quạt có dòng điện chạy qua* 13](#_Toc30140749)

[*Hình 2.5. Khi tải kết nối với đèn hoạt động* 13](#_Toc30140750)

[*Hình 2.6. Giao diện phần mềm App Inventor* 14](#_Toc30140751)

[*Hình 2.7. Giao diện lập trình Blocks* 15](#_Toc30140752)

[*Hình 2.8. Tải ứng dụng* 17](#_Toc30140753)

[*Hình 2.9. Barcode của ứng dụng* 18](#_Toc30140754)

[*Hình 2.10. Thiết lập tín hiệu được gửi đi từ App Inventor* 18](#_Toc30140755)

[*Hình 2.11. Khai báo các chân vào ra của thiết bị* 19](#_Toc30140756)

[*Hình 2.12. Hàm điều khiển đèn phòng khách* 19](#_Toc30140757)

[*Hình 2.13. Hàm điều khiển chính* 19](#_Toc30140758)

[*Hình 3.1. Linh kiện rơ le, tụ điện, biến trở, transistor, đi-ốt được hàn cố định* 20](#_Toc30140759)

[*Hình 3.2. Mạch Arduino kết nối với board mạch rơ le* 20](#_Toc30140760)

[*Hình 3.3. Giao diện điều khiển các thiết bị* 21](#_Toc30140761)

[*Hình 3.4. Giao diện đăng nhập* 21](#_Toc30140762)

[*Hình 3.5. Hiển thị các địa chỉ Bluetooth* 22](#_Toc30140763)

[*Hình 3.6. Giao diện điều khiển thiết bị đèn led* 22](#_Toc30140764)

[*Hình 3.7. Kết quả thi công hoàn chỉnh* 23](#_Toc30140765)

Tóm Tắt

Ngày nay, xã hội phát triển mạnh mẽ, kỹ thuật ngày càng hiện đại nên nhu cầu về trao đổi thông tin giải trí, nhu cầu về điều khiển các thiết bị từ xa, ngày càng cao. Và những hệ thống dây cáp phức tạp lại không thể đáp ứng nhu cầu này, nhất là ở những khu vực chật hẹp, những nơi xa xôi, trên các phương tiện vận chuyển. Vì vậy công nghệ không dây đã ra đời và phát triển mạnh mẽ, tạo rất nhiều thuận lợi cho con người trong đời sống hằng ngày.

Trong những năm gần đây công nghệ truyền nhận dữ liệu không dây đang có những bước phát triển mạnh mẽ, góp công lớn trong việc phát triển các hệ thống điều khiển, giám sát từ xa, đặc biệt là các hệ thống thông minh. Hiện nay, có khá nhiều công nghệ không truyền nhận dữ liệu không dây như RF, Wifi, Bluetooth, NFC, ... Trong đó, Bluetooth là một trong những công nghệ được phát triển từ lâu và luôn được cải tiến để nâng cao tốc độ cũng như khả năng bảo mật.

Trên thị trường Việt Nam hiện nay chưa có nhiều sản phẩm điều khiển thiết bị không dây, đa số những sản phẩm hiện có đều là nhập khẩu từ nước ngoài với giá thành cao. Việc nghiên cứu và thiết kế một bộ sản phẩm điều khiển thiết bị không dây có một ý nghĩa lớn, giúp tăng thêm sự lựa chọn cho người sử dụng, sản phẩm được sản xuất trong nước nên giá thành rẻ và góp phần phát triển các hệ thống điều khiển thông minh.



*Hình 1.1 . Điều khiển đèn huỳnh quang bằng điện thoại thông minh thông qua module HC-05*

(Nguồn: <https://tinhte.vn> )

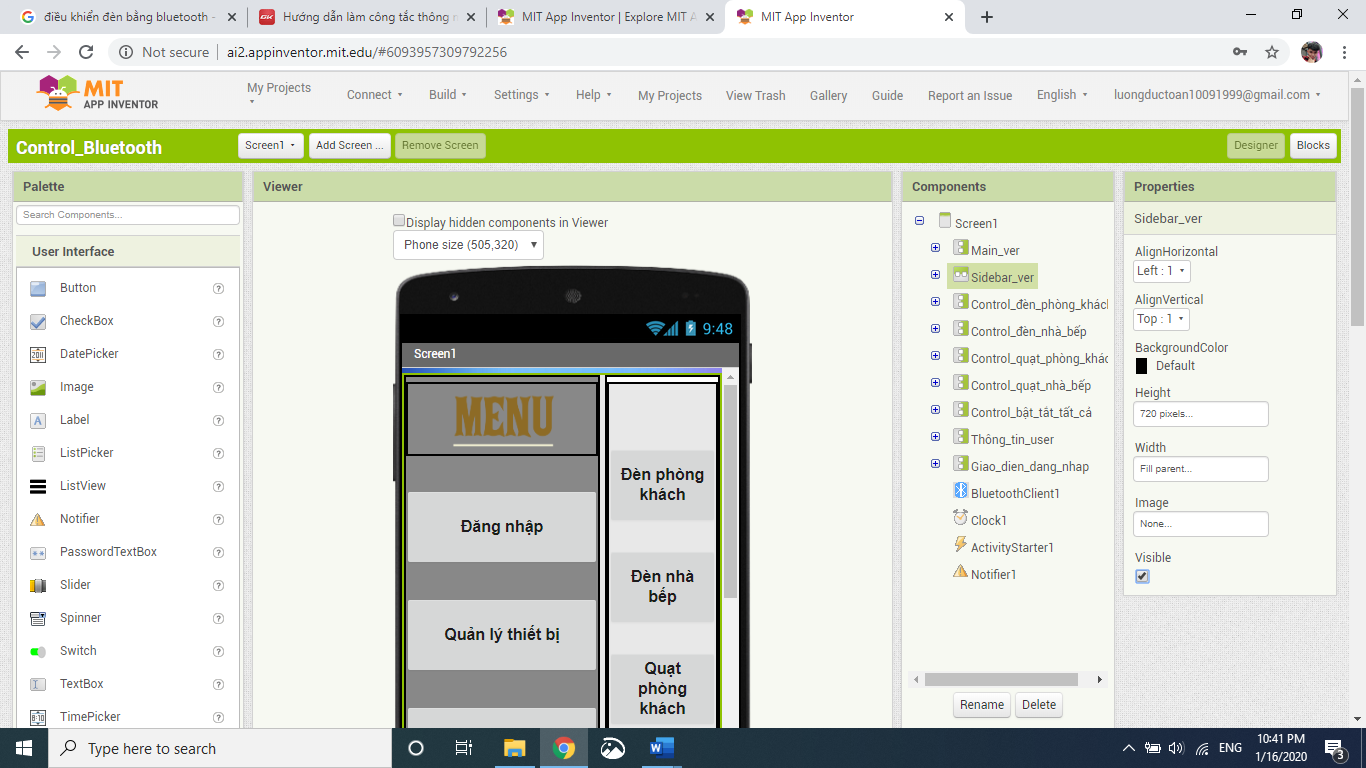
# GIỚI THIỆU

## App Invertor

MIT App Inventor dành cho Android là một ứng dụng web nguồn mở ban đầu được cung cấp bởi Google và hiện tại được duy trì bởi Viện Công nghệ Massachusetts (MIT). Nền tảng cho phép nhà lập trình tạo ra các ứng dụng phần mềm cho hệ điều hành Android (OS). Bằng cách sử dụng giao diện đồ họa, nền tảng cho phép người dùng kéo và thả các khối mã (blocks) để tạo ra các ứng dụng có thể chạy trên thiết bị Android. Nền tảng App Inventor được đưa ra thông qua yêu cầu vào ngày 12 tháng 7 năm 2010 và được phát hành công khai vào ngày 15 tháng 12 năm 2010. Nhóm App Inventor được dẫn dắt bởi Hal Abelson và Mark Friedman.

Những tính năng có trên MIT App Inventor là:

1. 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.
2. 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.
3. 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).
4. Hỗ trợ xây dựng game bằng các components: Ball, Canvas, ImageSprite.
5. 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.



*Hình 1.1. Giao diện thiết kế trên MIT App Inventor 2*

**Palette:** Chứa các thành phần có thể đặt lên trên Screen như: Button, Label, Image, Listview, Video player, đến các thanh phần chức năng không nhìn thấy trên Screen như: BLE extension, Notifier, các sensors, …

**User interface:** Là một thành phần cơ bản để xây dựng nên một ứng dụng.

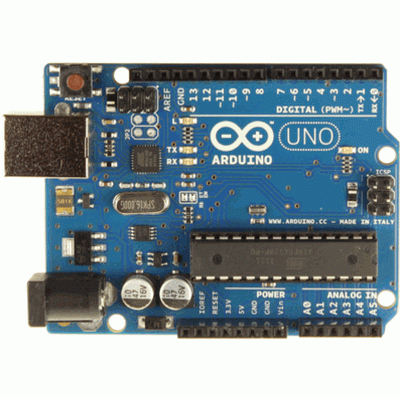
**Layout:** Là thành phần dùng để tạo hoặc bố trí sắp xếp các bố cục theo chiều ngang hoặc chiều dọc.

**Media:** Là thành phần dùng để chụp ảnh, ghi âm, tạo video, nhận giọng nói và chuyển giọng nói thành văn bản.

**Drawing and Animation:**Cung cấp các thành phần để tạo bản vẽ và ảnh động.

**Sensor:**Cho phép ứng dụng truy cập vào gia tốc kế của thiết bị để phát hiện các chuyển động, cảm biến vị trí.

## Kit Arduino (Adruino Uno R3)

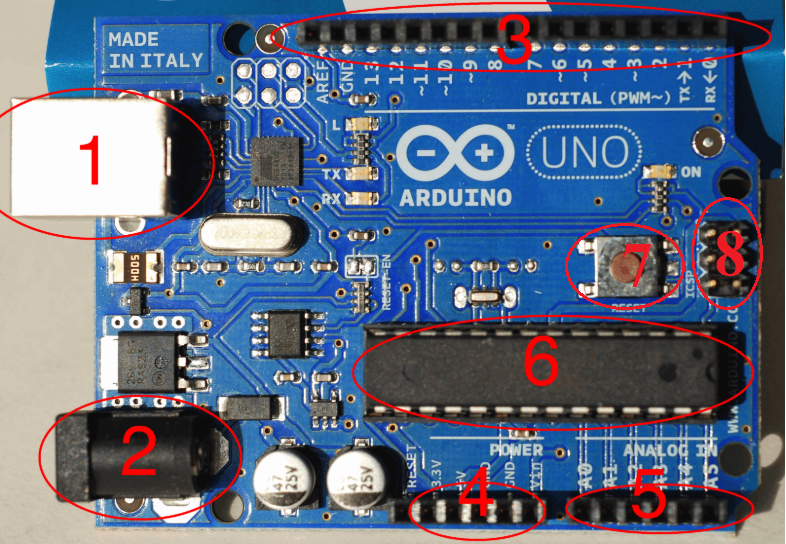


*Hình 1.2. Module Arduino UNO R3*

Arduino giống như 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. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau. Đi cùng với nó là một môi trường phát triển tích hợp (IDE) chạy trên các máy tính cá nhân thông thường và cho phép người dùng viết các chương trình cho Aduino bằng ngôn ngữ C hoặc C++.

**Bảng 1.1: Thông số kỹ thuật 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 khuyên 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) |



*Hình 1.3.Cấu tạo phần cứng của Arduino Uno R3*

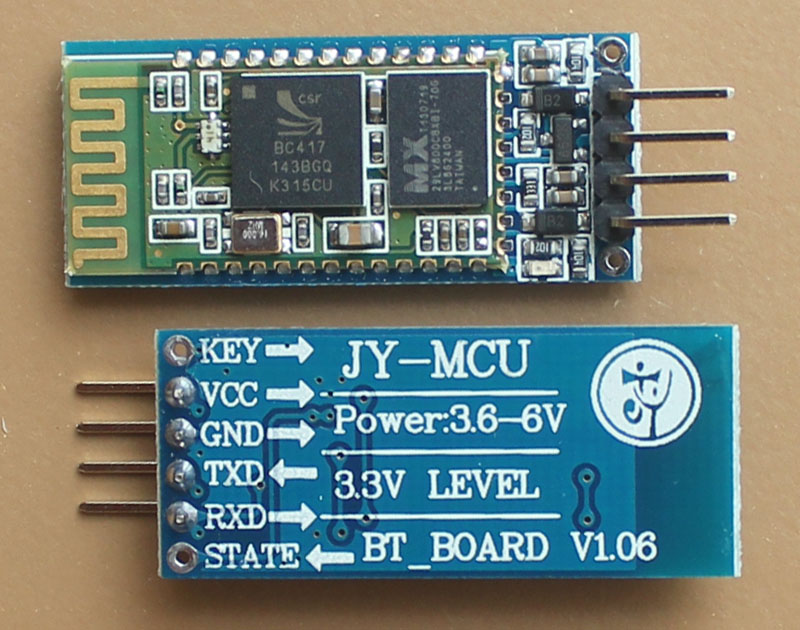
**Cấu tạo phần cứng của Arduino Uno R3:**

1. USB: Cổng giao tiếp USB có 2 chức năng: cấp nguồn cho board mạch và truyền thông nối tiếp với máy tính trong việc nạp chương trình hay giao tiếp nối tiếp.
2. Jack Power: Nguồn cấp cho board mạch Arduino. Có 2 loại nguồn có thể sử dụng được là nguồn xoay chiều (AC) tối đa 6V và nguồn một chiều (DC) tối đa 5V.
3. Hàng Header: đánh số từ 0 đến 12 là hàng digital pin, nhận vào hoặc xuất ra các tín hiệu số. Ngoài ra có một pin đất (GND) và pin điện áp tham chiếu (AREF).
4. Hàng header thứ hai: chủ yếu liên quan đến điện áp đất, nguồn.
5. Hàng header thứ ba: các chân để nhận vào hoặc xuất ra các tín hiệu analog. Ví dụ như đọc thông tin của các thiết bị cảm biến.
6. Vi điều khiển AVR: đây là bộ xử lý trung tâm của toàn bo mạch. Với mỗi mẫu Arduino khác nhau thì con chip này khác nhau. Ở con Arduino Uno này thì sử dụng ATMega328.
7. RESET: Đặt lại trạng thái ngay khi nạp chương trình.
8. ICSP: Giao tiếp với Atmega328.

## Modue Bluetooth (Module bluetooth hc-05)

Bluetooth là chuẩn truyền thông không dây để trao đổi dữ liệu ở khoảng cách ngắn.Chuẩn truyền thông này sử dụng sóng radio ngắn(UHF radio) trong dải tần số ISM (2.4 tới 2.485 GHz). Khoảng cách truyền của module này vào khoảng 10m.

**Sơ đồ chân HC-05 gồm có:**



*Hình 1.4. Sơ đồ chân Bluetooth HC05*

( Nguồn : <http://arduinolearning.com/wp-content/uploads/2016/05/hc-06.jpg> )

* **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, RND**: đây là hai chân UART để giao tiếp module hoạt động ở mức logic 3.3V
* **STATE**: các bạn chỉ cần thả nổi và không cần quan tâm đến chân này.

**Thông số kỹ thuật:**

* Điện thế hoạt động của UART 3.3 - 5V.
* Dòng điện khi hoạt động: khi Pairing 30 mA, sau khi pairing hoạt động truyền nhận bình thường 8 mA.
* Baudrate UART có thể chọn được: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
* Kích thước của module chính: 28 mm x 15 mm x 2.35 mm.
* Dải tần sóng hoạt động: 2.4GHz.
* Bluetooth protocol: Bluetooth Specification v2.0+EDRo.
* Kích thước: 26.9mm x 13mm x 2.2 mm.

**Cơ chế hoạt động:**

Tự động kết nối; Đáp ứng theo lệnh: tập lệnh AT để giao tiếp với module HC05 nhận 1 trong 3 chức năng: Master, Slave, Loopback (tập lệnh AT). Giao tiếp với module bằng giao tiếp nối tiếp không đồng bộ qua 2 đường RX và TX, vì vậy các bạn có thể sử dụng PC với chuẩn RS232 hoặc các dòng vi điều khiển để giao tiếp.

## Rơ le đóng ngắt (relay)

Rơ le (Relay) là 1 công tắc chuyển đổi hoạt động bằng điện. Nói là một công tắc vì rơ le có 2 trạng thái là ON và OFF. Rơ le ở trạng thái ON hay OFF phụ thuộc vào có dòng điện chạy qua rơ le hay không.

Relay 5 chân HLS8L-5VDC là loại linh kiện đóng ngắt điện cơ đơn giản. Nó gồm 2 phần chính là cuộn hút và các tiếp điểm.

Cấu tạo của relay được mô tả trong Hình 1.5.



*Hình 1.5. Rơ-le 5V 5 chân*

(Nguồn: <https://sptshop.vn/san-pham/relay-12v-5-chan-dc-ro-le-5-chan-gan-den-coi/> )

**Thông số kỹ thuật:**

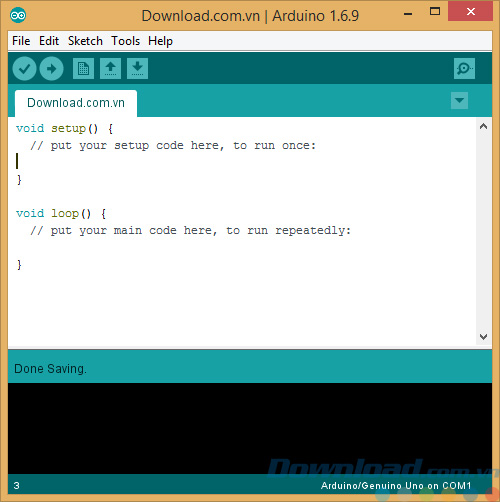
* Điện áp điều khiển: 5V.
* Dòng điện cực đại: 10A.
* Thời gian tác động: 10ms.
* Thời gian nhả hãm: 5ms.
* Nhiệt độ hoạt động: -45oC ~ 75oC.

**Ứng dụng của Rơ le:**

* Nhìn chung, công dụng của rơ-le là "dùng một năng lượng nhỏ để đóng cắt nguồn năng lượng lớn hơn".
* Rờ-le được dùng khá thông dụng trong các ứng dụng điều khiển động cơ và chiếu sáng.
* Khi cần đóng cắt nguồn năng lượng lớn, rơ-le thường được ghép nối tiếp. Nghĩa là một rơ-le nhỏ điều khiển một rơ-le lớn hơn, và rơ-le lớn sẽ điều khiển nguồn công suất.

## Arduino IDE

Arduino IDE là môi trường phát triển tích hợp mã nguồn mở, cho phép người dùng dễ dàng viết code và tải nó lên bo mạch. Môi trường phát triển được viết bằng Java dựa trên ngôn ngữ lập trình xử lý và phần mềm mã nguồn mở khác. Phần mềm này có thể được sử dụng với bất kỳ bo mạch Arduino nào.



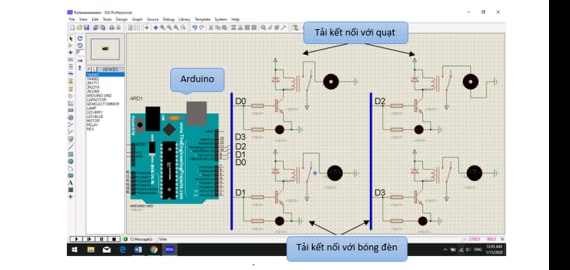
*Hình 1.6. Giao diện của phần mềm viết code cho bo mạch Arduino*

( Nguồn: <https://download.com.vn/arduino/download> )

Arduino IDE là một môi trường phát triển tích hợp đa nền tảng, làm việc cùng với một bộ điều khiển Arduino để viết, biên dịch và tải code lên bo mạch. Phần mềm này cung cấp sự hỗ trợ cho một loạt các bo mạch Arduino như Arduino Uno, Nano, Mega, Esplora, Ethernet, Fio, Pro hay Pro Mini cũng như LilyPad Arduino.

Ngôn ngữ phổ quát cho Arduino C và C++, do đó phần mềm phù hợp cho những lập trình viên đã quen thuộc với cả 2 ngôn ngữ này. Các tính năng như làm nổi bật cú pháp, thụt đầu dòng tự động, ... làm cho nó trở thành một sự thay thế hiện đại cho các IDE khác.

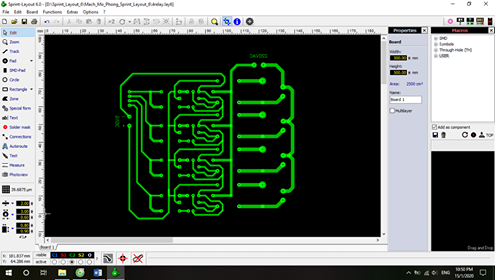
## Phần mềm thiết kế mạch và mô phỏng (PROTEUS 7.10SP2)



*Hình 1.7. Phần mềm thiết kế mạch và mô phỏng*

Trước khi thi công mạch sản phẩm thực tế cần trải qua quá trình mô phỏng mạch để kiểm chứng thuật toán cũng như hoạt động logic của linh kiện điện tử. Proteus là bộ công cụ chuyên về mô phỏng mạch điện tử chạy trên môi trường Windows, là sản phẩm của công ty Labcenter Electronics - một công ty sản xuất phần mền CAD của Anh. Sử dụng ISIS Schematic Capture để thiết kế nguyên lý và mô phỏng nguyên lý hoạt động của đề tài .Quan trọng nhất là thanh thư viện nơi chứa các linh kiện và vùng làm việc dùng để vẽ mạch nguyên lý .Ngoài ra, các nút được sử dụng cho việc mô phỏng nguyên lý hoạt động.

## Phần mềm Sprint Layout Viewer



*Hình 1.7. Phần mềm Sprint Layout Viewer*

( Nguồn: <http://dientusangtao123.blogspot.com/2016/05/phan-mem-sprint-layout-60-full.html> )

Sprint-Layout-Viewer là ứng dụng cho phép bạn xem và in các bản thiết kế PCB được tạo bằng chương trình Sprint-Layout, có định dạng LAY, và kiểm tra các lớp và các thành phần được nhúng vào bản thiết kế này. Sprint-Layout-Viewer chỉ có một vài tùy chọn cài đặt cơ bản, vì vậy đối tượng sử dụng ứng dụng cũng đa dạng, không chỉ giới hạn trong những người có chuyên môn nhất định.

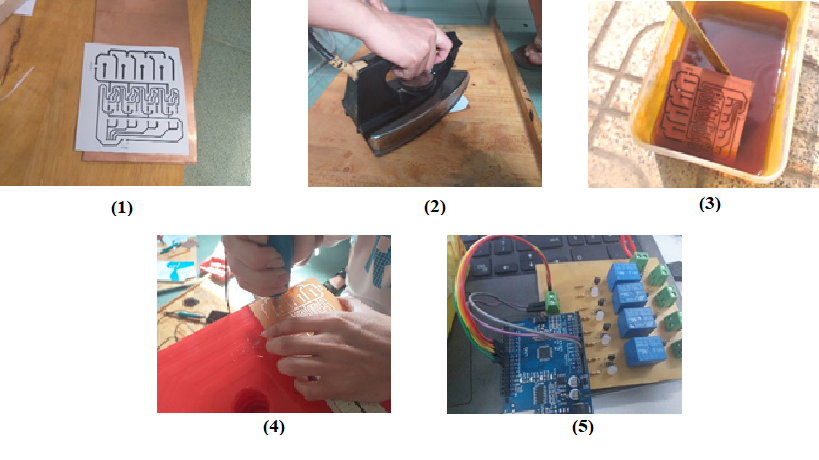
Sprint-Layout-Viewer được thiết kế với giao diện GUI đơn giản, gồm một cửa sổ duy nhất, nơi bạn chỉ có thể chọn file muốn mở qua trình duyệt file bởi phương pháp kéo thả file không được hỗ trợ.

# Phương pháp nghiên cứu



## Thi công thiết kế mạch in với Sprint Layout

**Các bước thiết kế mạch in:**

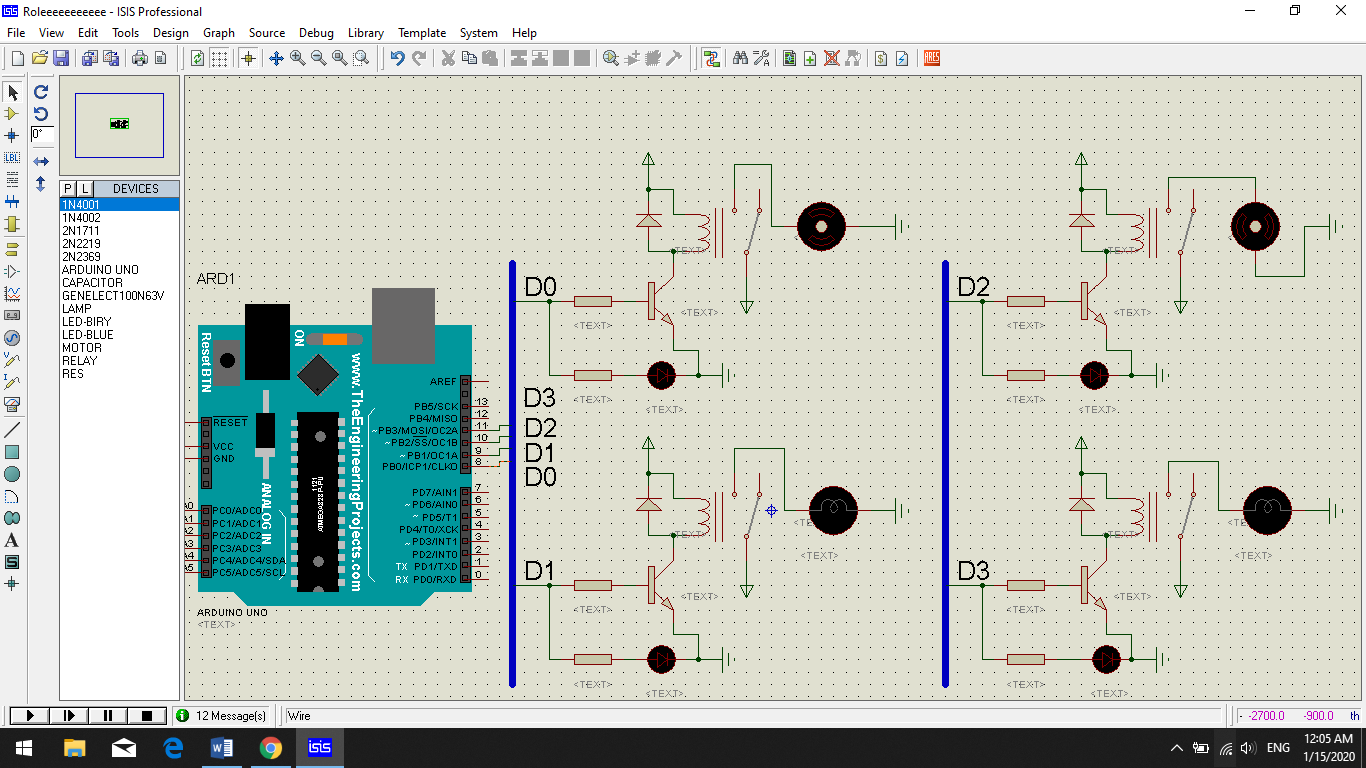


*Hình 2.1. Quy trình làm mạch in*

* In mạch (1): Mạch in được in trên giấy decal.
* Ủi mạch (2): Giấy decal được phủ lên mạch đồng và được gia nhiệt trong một thời gian nhất định tầm khoảng 5 đến 10 phút.
* Ăn mòn (3): Lớp đồng sau khi được phủ bởi mực máy in từ giấy decal sẽ được bảo vệ chống lại sự ăn mòn của dung dịch Fe2O3 được hòa tan trong nước. Trong quá trình ăn mòn để rút ngắn thời gian này chúng ta cần gia nhiệt cũng như tốc độ truyền động của board mạch trong dung dịch này.
* Khoan mạch (4): Các vị trí linh kiện được cố định bởi các kích thước của mũi khoan (thường chọn mũi khoan có kích thước 1mm)
* Lắp linh kiện (5): Các linh kiện được cố định trên từng vị trí tương ứng bởi thiếc hàn. Trong quá trình này vừa lắp kết hợp với kiểm tra sự thông mạch.

## Mô phỏng phần mềm Proteus.

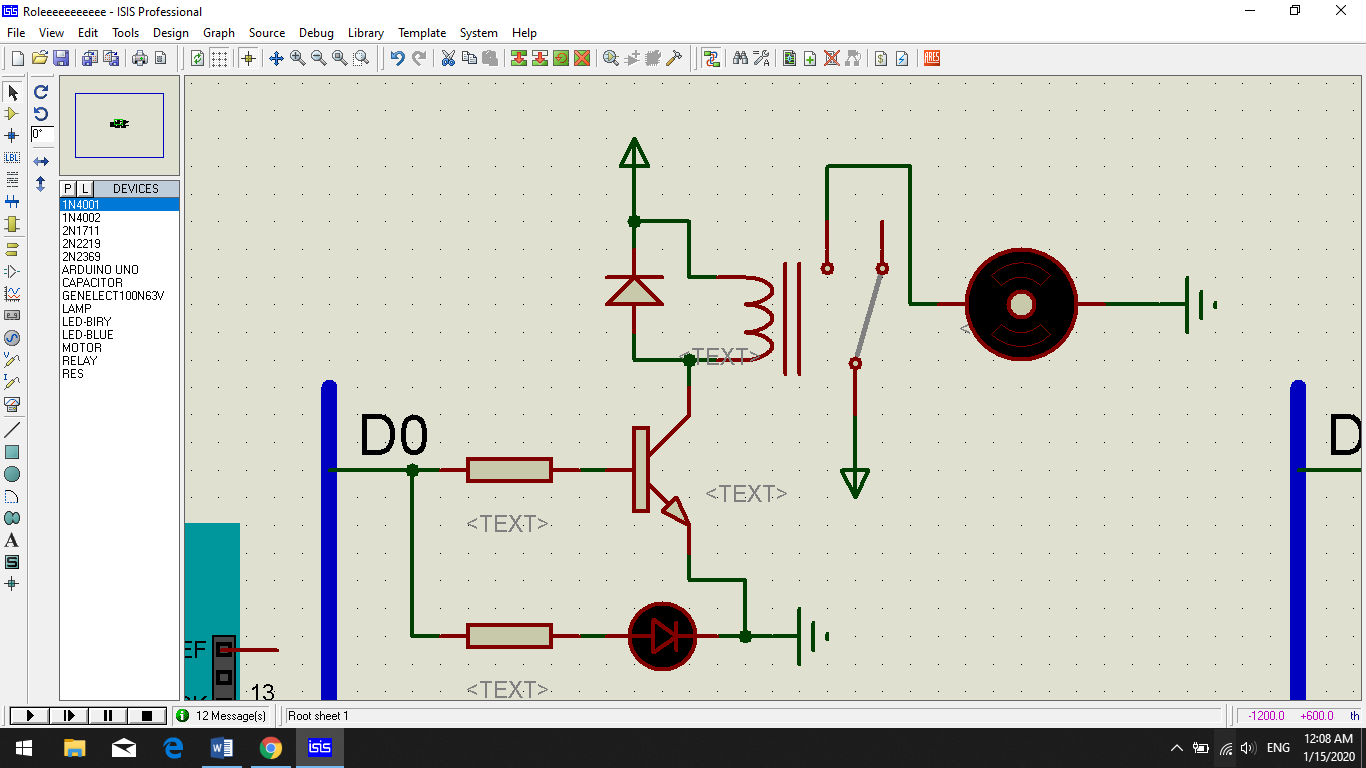
**Nguyên lý hoạt động:**

1. 

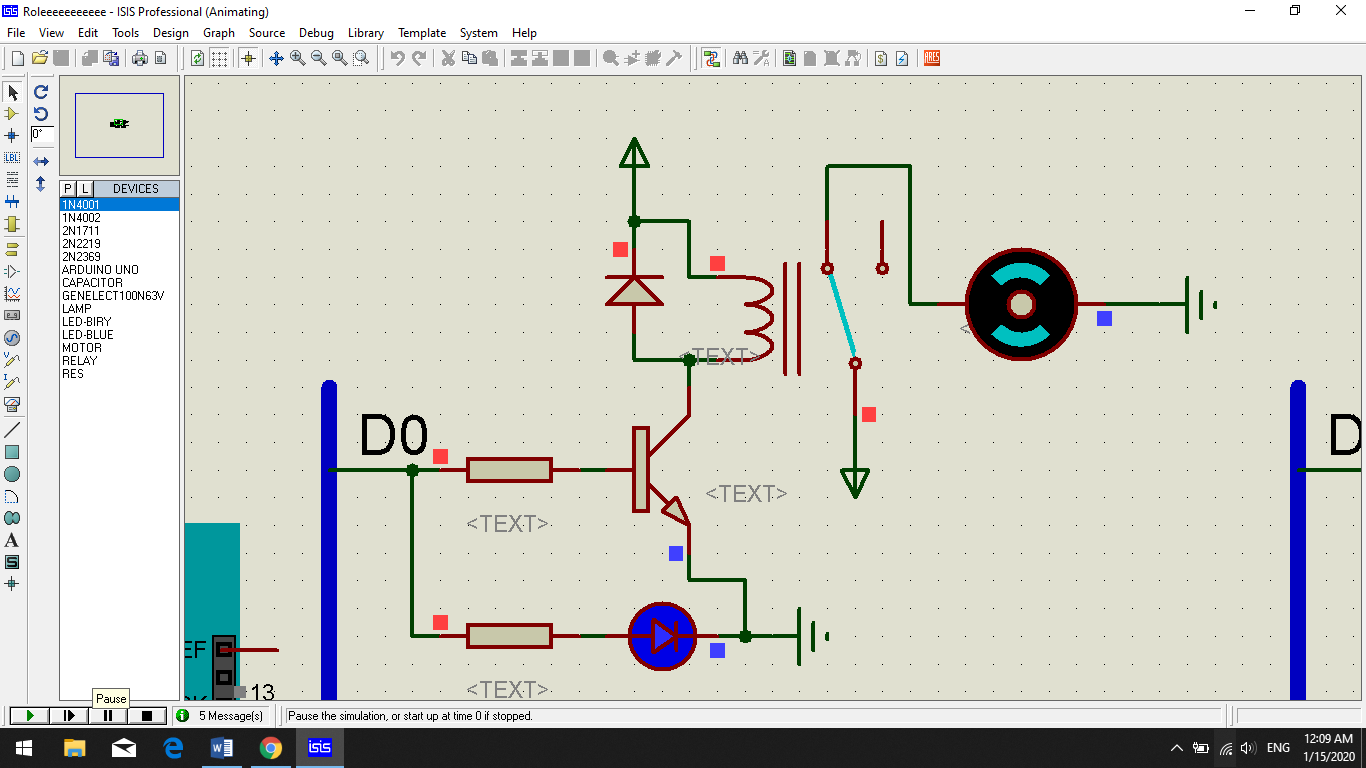
*Hình 2.2. Sơ đồ nguyên lý điều khiển các thiết bị thông qua Arduino trong Proteus*

Tổng quan về sơ đồ nguyên lý:

* Để điều khiển các tải kết nối với quạt ta dùng các bit D0, D2
* Để điều khiển các tải kết nối với đèn ta dùng các bit D1, D3
* Trong đó các chân D0, D1, D2, D3 được nối với các chân digital 8, 9, 10, 11 của Arduino.



*Hình 2.3. Khi tải kết nối với quạt không có dòng điện chạy qua*

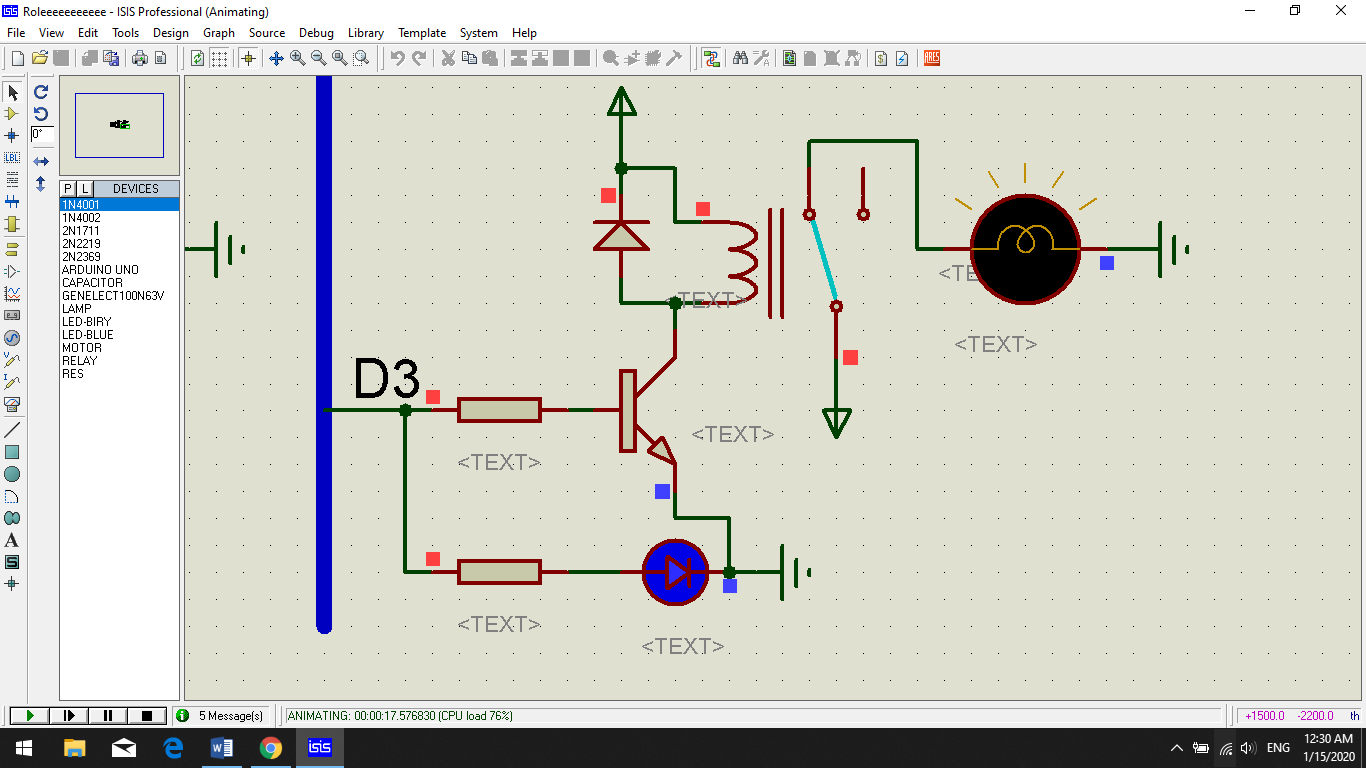


*Hình 2.4. Khi tải kết nối với quạt có dòng điện chạy qua*

Ở trạng thái bình thường các chân digital của Arduino bằng 0 thì các thiết bị điện tử không hoạt động.

Khi chân digital 8 có giá trị bằng 1 thì D0 sẽ mang giá trị là 1, làm cho đén báo sang báo hiệu có dòng điện chạy qua. Transistor được mắc song song với đèn báo sẽ dẫn, lúc đó sẽ có một dòng khép kín đi từ chân số 1 sang chân số 2 của Role (Relay). Bây giờ cuộn dây sẽ tạo thành một lực từ, lực từ này hút chân số 4 từ chân số 5 về chân số 3. Tạo ra vòng điện khép kín từ chân số 3 nối với chân số 4 và nối đất. bây giờ quạt sẽ quay.

Tương tự với tải kết nối với quạt có bit D2 và quạt có bit là D1, D3.

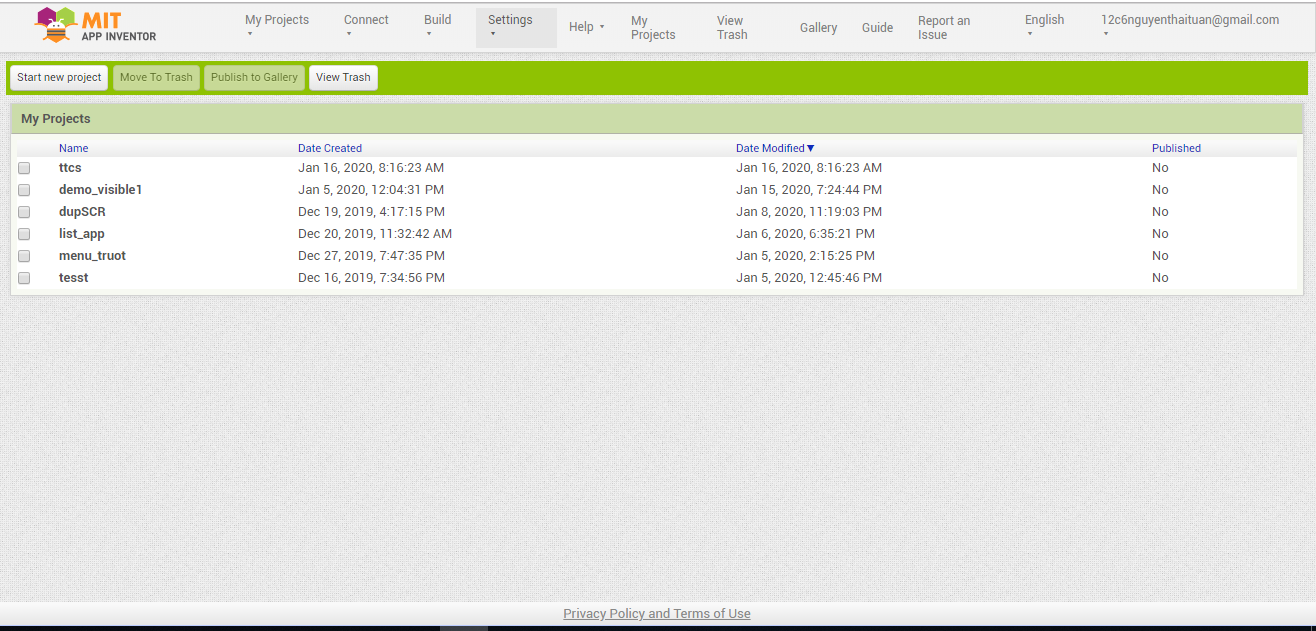


*Hình 2.5. Khi tải kết nối với đèn hoạt động*

## Lập trình giao diện

### Giao diện thiết kế với App Inventor

Để sử dụng được App Inventor, truy cập vào địa chỉ <http://ai2.appinventor.mit.edu>, sau đó tiến hành đăng nhập bằng tài khoản Google của bạn để mở trang quản lí các project.

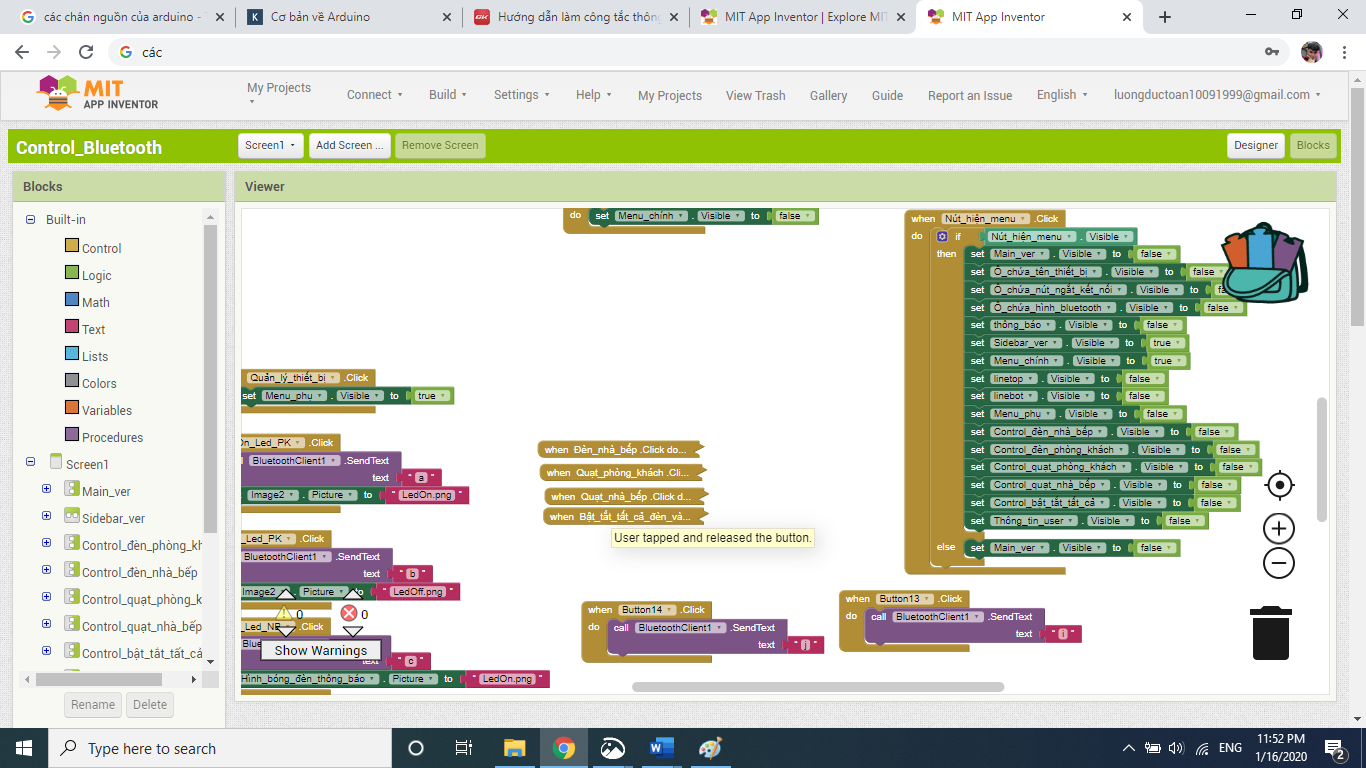


*Hình 2.6. Giao diện phần mềm App Inventor*

Bên trái là các control User Interface, Media, Sensor, Social,… để sử dụng, chỉ cần kéo thả mục Viewer. Ở giữa là mục viewer dùng để mô phỏng màn hình ứng dụng. Bên phải là cửa sổ quản lý các component, media và property cho từng control. Các đối tượng điều khiển bằng cách kéo thả vào từng layout. Để xây dựng chức năng kết nối bluetooth, chọn công cụ Control Bluetooth Client trong mục Connectivity. Để cài đặt thời gian truy cập chi tiết, chọn Clock trong mục Sensors.

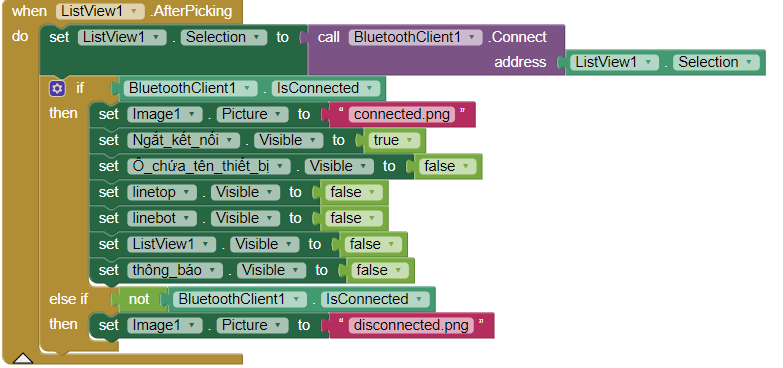
### Giao diện lập trình (Blocks) với App Inventor.

Blocks là lập trình bằng cách kéo thả các khối lệnh cho các đối tượng điều khiển. Hình 2.3 là giao diện khối lập trình cho ứng dụng điều khiển thiết bị thông qua bluetooth. Bên trái là các khối lệnh: Control, Logic, Math, Text. Khung bên phải là khung Viewer dùng để lập trình và sắp xếp các khối lệnh.



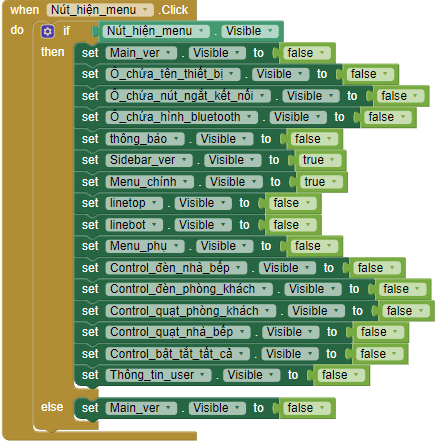
*Hình 2.7. Giao diện lập trình Blocks*

Để kết nối thiết bị bluetooth, thực hiện kéo thả các block.



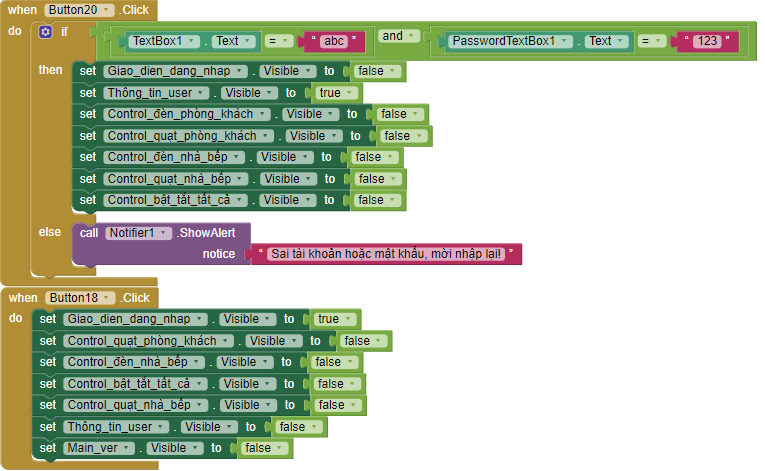
*Hình 2.8. Khối chức năng kết nối bluetooth*

Để mở giao menu ta thực hiện các khối lệnh:



*Hình 2.9. Khối chức năng mở Menu*

Để mở giao diện đăng nhập, thực hiện các khối lệnh:



*Hình 2.10. Khối chức năng đăng nhập và kiểm tra tài khoản*

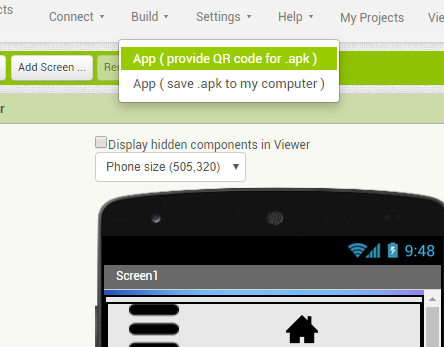
Để điều khiển thiết bị quạt, lần lượt kéo thả các block như hình:



*Hình 2.11. Khối chức năng điều khiển thiết bị*

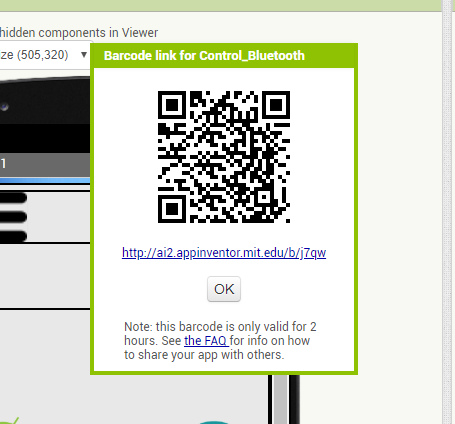
### Đóng gói ứng dụng thành file \*.apk.

Để sử dụng ứng dụng vừa tạo, vào build và chọn App (provide QR code for. apk) hoặc chọn App (save .apk to my computer).



*Hình 2.12. Tải ứng dụng*

Sau khi biên dịch xong sẽ xuất hiện một mã QR code, sử dụng phần mềm quét QR code trên điện thoại để tải file apk về và cài đặt.



*Hình 2.13. Barcode của ứng dụng*

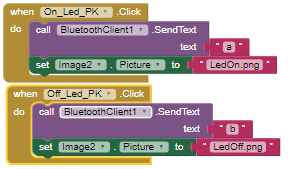
## Giao tiếp Arduino với HC-05

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.

## Điều khiển thiết bị với Arduino

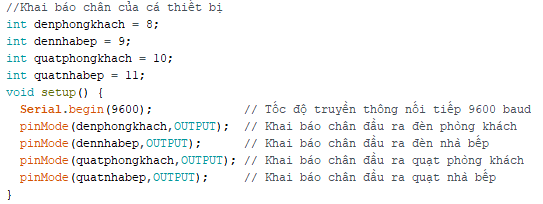
Đầu tiên, để gửi tín hiệu từ ứng dụng của App Inventor, ta thiết lập block từ App inventor với các tín hiệu dạng text:



*Hình 2.14. Thiết lập tín hiệu được gửi đi từ App Inventor*

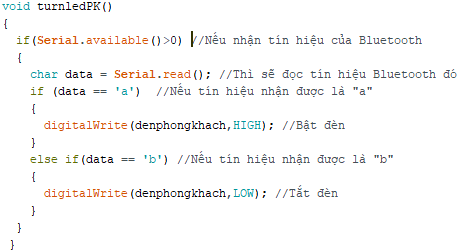
Trên Arduino IDE, thiết lập code để nhận tín hiệu từ App Inventor:

**Bước 1:** Khai báo các chân đầu vào và đầu ra của thiết bị như hình 2.11



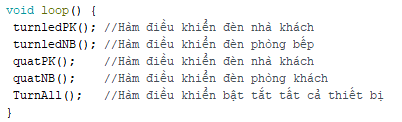
*Hình 2.15. Khai báo các chân vào ra của thiết bị*

**Bước 2:** Thiết lập hàm điều khiển từng thiết bị như hình 2.12

****

*Hình 2.16. Hàm điều khiển đèn phòng khách*

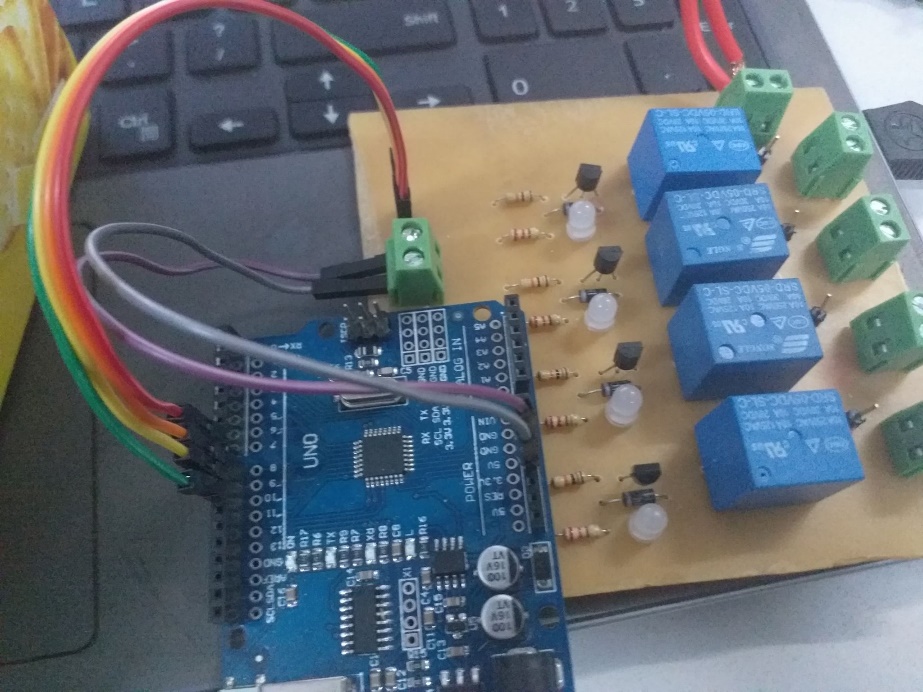
**Bước 3:** Thiết lập hàm điều khiển chính như hình 2.13.



*Hình 2.17. Hàm điều khiển chính*

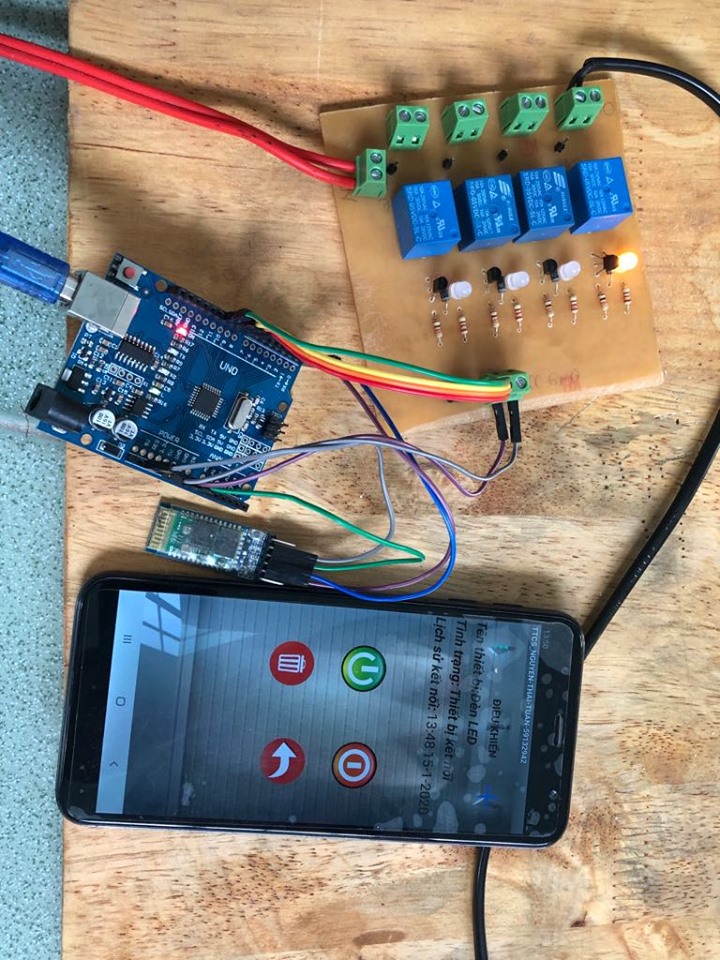
# Kết quả

Kết quả thiết kế gia công mạch in hoàn thành sẽ được mạch điện với các rơ le, tụ điện, biến trở, transistor, đi-ốt được hàn cố định. Trong đó, các rơ le đóng vai trò đóng ngắt dòng điện đi qua.



*Hình 3.1. Linh kiện rơ le, tụ điện, biến trở, transistor, đi-ốt được hàn cố định*

Sau khi tiến hành nối dây cho các thành phần, cho kết quả như hình



*Hình 3.2. Mạch Arduino kết nối với board mạch rơ le*

**App Inventor điều khiển:**

Khi mở ứng dụng sẽ xuất hiện giao diện như hình 3.3 gồm có các chức năng chính: Đăng nhập, Quản lý thiết bị, Cài đặt Bluetooth, Quay lại.



*Hình 3.3. Giao diện điều khiển các thiết bị*



*Hình 3.4. Giao diện đăng nhập*

Để ứng dụng kết nối bluetooth, click “Cài đặt Bluetooth” và chọn bluetooth cần kết nối (hình 3.3) và hiển thị các địa chỉ bluetooth như hình 3.5).



*Hình 3.5. Hiển thị các địa chỉ Bluetooth*

Để hiển thị giao diện điều khiển từng thiết bị, nhấn vào thiết bị cần chọn trong mục danh sách thiết bị (hình 3.3) và chọn thiết bị như hình 3.6.



*Hình 3.6. Giao diện điều khiển thiết bị đèn led*

Kết quả thi công hoàn chỉnh.



*Hình 3.7. Kết quả thi công hoàn chỉnh*

# Kết luận

Trên cơ sở thiết kế phần cứng và phần mềm khi thực hiện đồ án đã đạt được một số kết quả sau:

* Về phần cứng: Tìm hiểu về Arduino Uno R3 và các linh kiện cũng như cách thiết kế và làm mạch thủ công.
* Đi sâu tìm hiểu một số thuật toán điều khiển. - Nhận dạng và mô phỏng đối tượng điều khiển.

Trong gian đoạn làm đồ án, chúng em đã rất cố gắng nghiên cứu, thiết kế để có được những kết quả đã nêu trên. Tuy nhiên do thời gian và kiến thức của chúng em có hạn, mặt khác một số điều kiện về thiết bị không cho phép nên trong đồ án còn những hạn chế và thiếu sót.

# Phụ lục

Code trên Arduino IDE:

//Khai báo chân của cá thiết bị

int denphongkhach = 8;

int dennhabep = 9;

int quatphongkhach = 10;

int quatnhabep = 11;

void setup() {

Serial.begin(9600); // Tốc độ truyền thông nối tiếp 9600 baud

pinMode(denphongkhach,OUTPUT); // Khai báo chân đầu ra đèn phòng khách

pinMode(dennhabep,OUTPUT); // Khai báo chân đầu ra đèn nhà bếp

pinMode(quatphongkhach,OUTPUT); // Khai báo chân đầu ra quạt phòng khách

pinMode(quatnhabep,OUTPUT); // Khai báo chân đầu ra quạt nhà bếp

}

void loop() {

TurnLedPK(); //Hàm điều khiển đèn nhà khách

TurnLedNB(); //Hàm điều khiển đèn phòng bếp

TurnFanPK(); //Hàm điều khiển đèn nhà khách

TurnFanNB(); //Hàm điều khiển đèn phòng khách

TurnAll(); //Hàm điều khiển bật tắt tất cả thiết bị

}

void TurnLedPK()

{

if(Serial.available()>0) //Nếu nhận tín hiệu của Bluetooth

{

char data = Serial.read(); //Thì sẽ đọc tín hiệu Bluetooth đó

if (data == 'a') //Nếu tín hiệu nhận được là "a"

{

digitalWrite(denphongkhach,HIGH); //Bật đèn

}

else if(data == 'b') //Nếu tín hiệu nhận được là "b"

{

digitalWrite(denphongkhach,LOW); //Tắt đèn

}

}

}

void TurnLedNB()

{

if(Serial.available()>0)//Nếu nhận tín hiệu của Bluetooth

{

char data = Serial.read();//Thì sẽ đọc tín hiệu Bluetooth đó

if (data == 'c') //Nếu tín hiệu nhận được là "c"

{

digitalWrite(dennhabep,HIGH);//Bật đèn

}

else if(data == 'd') //Nếu tín hiệu nhận được là "d"

{

digitalWrite(dennhabep,LOW);// Tắt đèn

}

}

}

void TurnFanPK()

{

if(Serial.available()>0)//Nếu nhận tín hiệu của Bluetooth

{

char data = Serial.read();//Thì sẽ đọc tín hiệu Bluetooth đó

if (data == 'e') //Nếu tín hiệu nhận được là "e"

{

digitalWrite(quatphongkhach,HIGH);//Bật quạt

}

else if(data == 'f') //Nếu tín hiệu nhận được là "f"

{

digitalWrite(quatphongkhach,LOW);//Tắt quạt

}

}

}

void TurnFanNB()

{

if(Serial.available()>0)//Nếu nhận tín hiệu của Bluetooth

{

char data = Serial.read();//Thì sẽ đọc tín hiệu Bluetooth đó

if (data == 'g') //Nếu tín hiệu nhận được là "g"

{

digitalWrite(quatnhabep,HIGH);//Bật quạt

}

else if(data == 'h') //Nếu tín hiệu nhận được là "h"

{

digitalWrite(quatnhabep,LOW);//Tắt quạt

}

}

}

void TurnAll()

{

if(Serial.available()>0)//Nếu nhận tín hiệu của Bluetooth

{

char data = Serial.read();//Thì sẽ đọc tín hiệu Bluetooth đó

if (data == 'i') //Nếu tín hiệu nhận được là "i"

{

digitalWrite(denphongkhach,HIGH);

digitalWrite(dennhabep,HIGH);

digitalWrite(quatphongkhach,HIGH);

digitalWrite(quatnhabep,HIGH);

}

else if(data == 'j') //Nếu tín hiệu nhận được là "j"

{

digitalWrite(denphongkhach,LOW);

digitalWrite(dennhabep,LOW);

digitalWrite(quatphongkhach,LOW);

digitalWrite(quatnhabep,LOW);

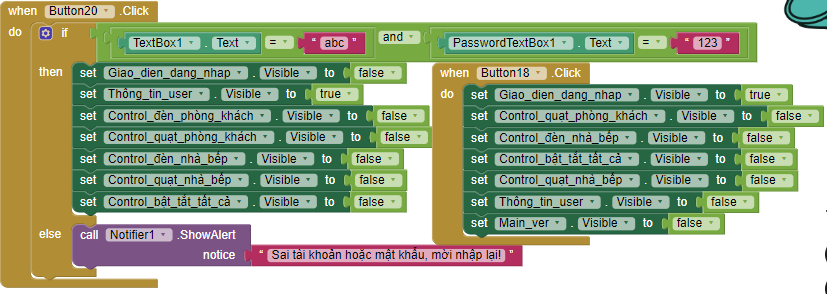
}

}

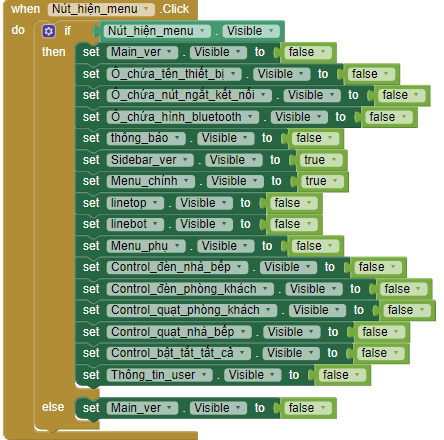
}

Code block trên MIT App Invnetor:

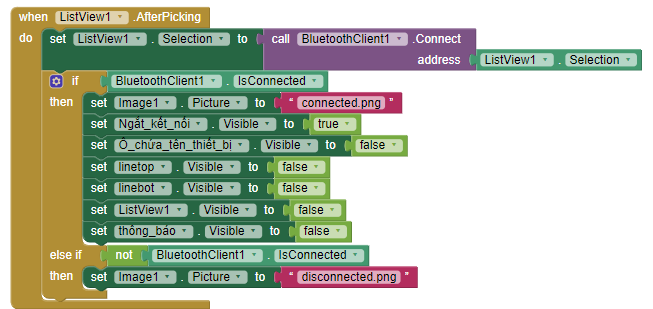
Code block đăng nhập:



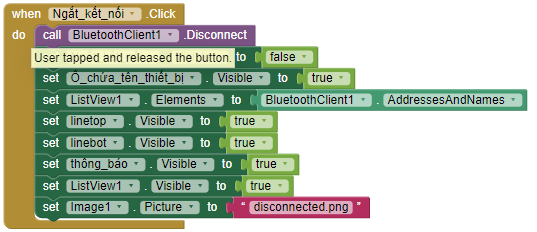
Code block mở các chức năng chính:



Code block kết nối bluetooth:



Code block ngắt kết nối bluetooth:



Code block thoát ứng dụng:



Code block điều khiển thiết bị quạt đèn :

