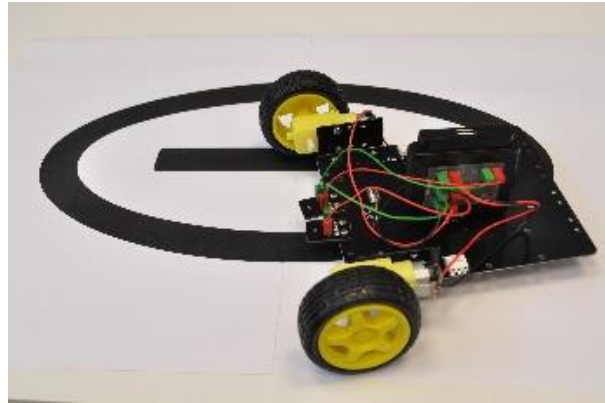


# **HƯỚNG DẪN SỬ DỤNG**

**ROBOT LẮP RÁP  
EISENSE-MAKE**

# EISENSE-MAKE ROBOT LẮP RÁP



# MỤC LỤC

|  |           |
|--|-----------|
| <b>CHƯƠNG 1 – ĐIỀU KHIỂN VÀ ROBOT .....</b>                            | <b>1</b>  |
| <b>1.1. Robot.....</b>   | <b>1</b>  |
| <b>1.2. Hệ thống điều khiển.....</b>                                   | <b>1</b>  |
| <b>1.3. Khối điều khiển (Brain unit) .....</b>                         | <b>2</b>  |
| <b>1.4. Phần mềm RobocklySense .....</b>                               | <b>2</b>  |
| <b>1.5. EIROBO-206.....</b>  | <b>3</b>  |
| <b>1.6. Chuẩn bị dây kết nối cho các thí nghiệm .....</b>              | <b>4</b>  |
| <b>1.7. Cài đặt phần mềm RobocklySense.....</b>                        | <b>4</b>  |
| <b>1.8. Bắt đầu chương trình RobocklySense .....</b>                   | <b>8</b>  |
| <b>THÍ NGHIỆM 1.1 – CHẾ ĐỘ TRỰC TIẾP (DIRECT).....</b>                 | <b>10</b> |
| 1.1.1. Kiểm tra động cơ .....  | 11        |
| 1.1.2. Kiểm tra Còi (Buzzer) .....                                     | 12        |
| 1.1.3. Kiểm tra đèn LED .....  | 13        |
| 1.1.4. Kiểm tra đèn (Lamp) .....                                       | 13        |
| 1.1.5. Kiểm tra cảm biến ánh sáng (Light).....                         | 14        |
| 1.1.6. Kiểm tra công tắc gạt (Switch).....                             | 14        |
| 1.1.7. Kiểm tra công tắc nhấn nhả (Tact).....                          | 14        |
| <b>THÍ NGHIỆM 1.2 – CÁC CHƯƠNG TRÌNH ĐẦU TIÊN .....</b>                | <b>16</b> |
| 1.2.1. Chương trình đầu tiên - Đèn LED nhấp nháy .....                 | 18        |
| 1.2.2. Tải xuống chương trình .....                                    | 21        |
| 1.2.3. Đèn LED nhấp nháy .....   | 21        |
| 1.2.4. Bài tập thử thách – Đèn LED nhấp nháy.....                      | 22        |
| <b>THÍ NGHIỆM 1.3 – CHƯƠNG TRÌNH CON .....</b>                         | <b>23</b> |
| 1.3.1. Chương trình chính và chương trình con .....                    | 24        |
| 1.3.2. Vòng lặp .....  | 26        |
| 1.3.3. Bài tập thử thách – Đèn giao thông.....                         | 28        |
| <b>THÍ NGHIỆM 1.4 – CÔNG TẮC .....</b>                                 | <b>29</b> |
| 1.4.1. Ô nhớ và biến .....   | 29        |
| 1.4.2. Lệnh Wait until.....  | 32        |
| 1.4.3. Thay đổi tốc độ.....  | 34        |
| 1.4.4. Bài tập thử thách - Động cơ được điều khiển bằng nút nhấn ..... | 35        |
| <b>THÍ NGHIỆM 1.5 – ĐIỀU KIỆN VÀ QUYẾT ĐỊNH.....</b>                   | <b>36</b> |

|  |           |
|--|-----------|
| 1.5.1. Lệnh If (Nếu).....  | 36        |
| 1.5.2. Các định nghĩa.....   | 38        |
| 1.5.3. Hàm điều kiện OR (Hoặc).....                                      | 38        |
| 1.5.4. Hàm điều kiện AND.....  | 39        |
| 1.5.5. Bài tập thử thách - Động cơ được điều khiển bởi hai nút nhấn..... | 40        |
| <b>THÍ NGHIỆM 1.6 – CÔNG QUANG.....</b>                                  | <b>41</b> |
| 1.6.1. Thiết bị Brain.....   | 41        |
| 1.6.2. Động cơ servo.....  | 41        |
| 1.6.3. Công quang.....   | 42        |
| 1.6.4. Chương trình công quang.....                                      | 44        |
| 1.6.5. Bài tập thử thách - Công quang với còi báo động.....              | 45        |
| <b>THÍ NGHIỆM 1.7 – XE GẮN ĐỘNG CƠ.....</b>                              | <b>46</b> |
| 1.7.1. Bài tập thử thách – Xe tự hành.....                               | 47        |
| <b>THÍ NGHIỆM 1.8 – MÁY TỰ ĐỘNG.....</b>                                 | <b>48</b> |
| 1.8.1. Bài tập thử thách - Máy giặt.....                                 | 49        |

# CHƯƠNG 1 – ĐIỀU KHIỂN VÀ ROBOT

## 1.1. Robot

Thế giới ngày nay là một thế giới của các hệ thống máy tính lập trình nhúng. Chúng ta tìm thấy chúng trong các hệ thống truyền thông, đồng hồ, điện thoại, điều khiển từ xa, ô tô và nhiều hơn nữa. Vài năm trước, chúng ta không hiểu được các thuật ngữ như “wearable computing” hoặc “internet of things”.

Mỗi ngày một sản phẩm mới hoặc ứng dụng mới xuất hiện và nhiều tháng sau, chúng ta không thể nhận ra chúng ta đã sống như thế nào nếu không có sự xuất hiện của chúng.

Thành phần các robot của hệ thống máy tính nhúng là hệ thống thực hiện các hoạt động độc lập như tìm kiếm, thao tác, nhận dạng, kích hoạt, bảo vệ, v.v.

Nhiều hệ thống kết hợp một loại trí tuệ nhân tạo nhất định trong vận hành và giao tiếp giữa các máy.

Mỗi hệ thống robot bao gồm một bộ điều khiển cho phép nó hoạt động theo các chương trình khác nhau. Nhà phát triển robot viết các chương trình này trên máy tính và chuyển tiếp chúng đến bộ điều khiển.

Hệ thống robot bao gồm bộ điều khiển, các cơ cấu chấp hành, bánh xe, bánh răng, động cơ, cảm biến, v.v.

Xây dựng hệ thống robot có một thách thức đòi hỏi sự tiếp thu kiến thức trong các lĩnh vực công nghệ khác nhau (điện tử, máy tính, cơ khí, điện, v.v.)

Có rất nhiều loại robot khác nhau như robot cánh tay, robot di động, robot đi bộ và nhiều loại khác.

Robot EISENSE là một dòng robot kết hợp với bộ nhớ lưu trữ để nghiên cứu, lập trình và chế tạo robot với nhiều ứng dụng khác nhau.

EISENSE là một robot tự hành cho phép chúng ta lập trình nhiều ứng dụng và chức năng như chuyển động trên một đường thẳng, di chuyển dọc theo tường, theo dõi, AGV (xe dẫn hướng tự động), xe tự hành, xe bảo vệ tự hành, taxi tự hành, giám sát môi trường, v.v. Tất cả các ứng dụng sẽ được mô tả dưới dạng các bài tập trong tài liệu này.

## 1.2. Hệ thống điều khiển

Robot là một hệ thống điều khiển máy tính.

"Hệ thống điều khiển" có thể được định nghĩa là một nhóm các bộ phận được vận hành cùng nhau để kiểm soát các biến khác nhau, quản lý hoạt động của hệ thống.

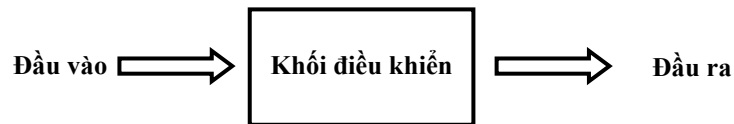
### Ví dụ:

- Hệ thống điều hòa không khí kiểm soát nhiệt độ trong phòng.
- Hệ thống điều khiển nhà kính kiểm soát nhiệt độ, độ ẩm, ánh sáng và tưới nước.
- Hệ thống điều khiển tốc độ duy trì tốc độ động cơ ổn định mặc dù tải trên động cơ thay đổi.

Một hệ thống điều khiển ánh sáng có thể duy trì mức ánh sáng ổn định, bất kể lượng ánh sáng mặt trời thay đổi. Hệ thống điều khiển BẬT hoặc TẮT đèn theo nhu cầu ánh sáng.

Ba khối cơ bản nằm trong mọi hệ thống điều khiển máy tính:

1. **Khối đầu vào** – bộ phận đọc các tín hiệu cảm biến của hệ thống như nhiệt độ, ánh sáng, khoảng cách, công tắc cảm ứng, v.v. và gửi thông tin đến bộ điều khiển.
2. **Khối điều khiển** – là "Bộ não" của hệ thống, khối điều khiển chứa chương trình trong bộ nhớ của nó và thực hiện các lệnh chương trình, xử lý dữ liệu nhận được.
3. **Khối đầu ra** – bộ phận vận hành các cơ cấu chấp hành của hệ thống như động cơ, đèn, máy bơm, quạt hoạt động dựa theo dữ liệu từ các đầu vào và "quyết định" của chương trình.



**Hình 2. 1**

Khối điều khiển được kết nối với máy tính để lập trình và tải chương trình từ máy tính vào bộ nhớ flash của bộ điều khiển.

Ngắt kết nối bộ điều khiển khỏi máy tính và kết nối nguồn điện hoặc pin với nó, tạo ra một hệ thống hoạt động độc lập.

### **1.3. Khối điều khiển (Brain unit)**

Một số thiết bị đầu vào có thể có "Bộ não" riêng của nó. Các cảm biến là thiết bị có chức năng đó. Chúng gửi đến bộ điều khiển theo yêu cầu các dữ liệu được xử lý như: nhiệt độ (°C hoặc °F), cường độ ánh sáng tính bằng Lux, khoảng cách tính bằng mét, v.v.

Các thiết bị đầu ra cũng có thể là các khối "brain". Ví dụ, bộ điều khiển tốc độ và hướng động cơ, cường độ đèn, góc động cơ servo, v.v.

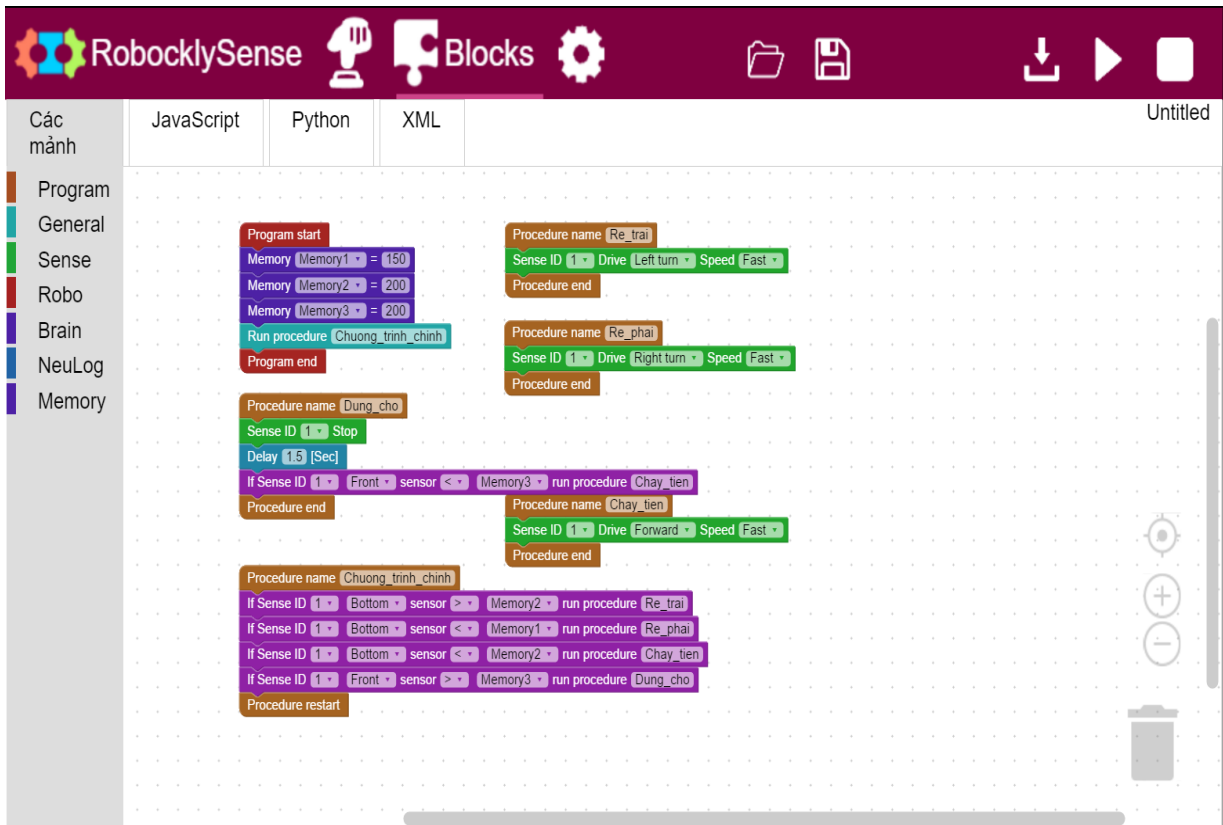
Các khối "brain" này được kết nối trong một chuỗi với bộ điều khiển chính, bộ điều khiển giao tiếp với chúng thông qua các tệp tin.

### **1.4. Phần mềm RobocklySense**

RobocklySense là một trình soạn thảo lập trình dạng khối trực quan. Nó sử dụng các khối lệnh liên kết với nhau để xây dựng một chương trình thay vì phải viết từng câu lệnh.

RobocklySense sử dụng các khối lệnh đặc biệt cho robot EISENSE-MAKE đọc đầu vào, vận hành đầu ra của nó và đọc bất kỳ các cảm biến nào.

Robockly Sense rất thân thiện, dễ lập trình và chạy các chương trình robot.



Hình 2. 2

### 1.5. EIROBO-206

Trong tài liệu này, chúng ta sẽ sử dụng EIROBO-206 làm bộ điều khiển trung tâm cho các thiết bị và robot.



Hình 2. 3

Bộ điều khiển có ba đầu vào (S1-S3) để kết nối với các công tắc và cảm biến, ba đầu ra (M1-M3) để kết nối với đèn và động cơ, chỉ thị LED, nút nhấn và giắc cắm USB mini để truyền dữ liệu từ máy tính sang bộ điều khiển EIROBO-206.

Một chương trình có thể được tải xuống bộ nhớ flash của bộ điều khiển và có thể chạy được bằng cách sử dụng nút nhấn trên bảng điều khiển của nó.

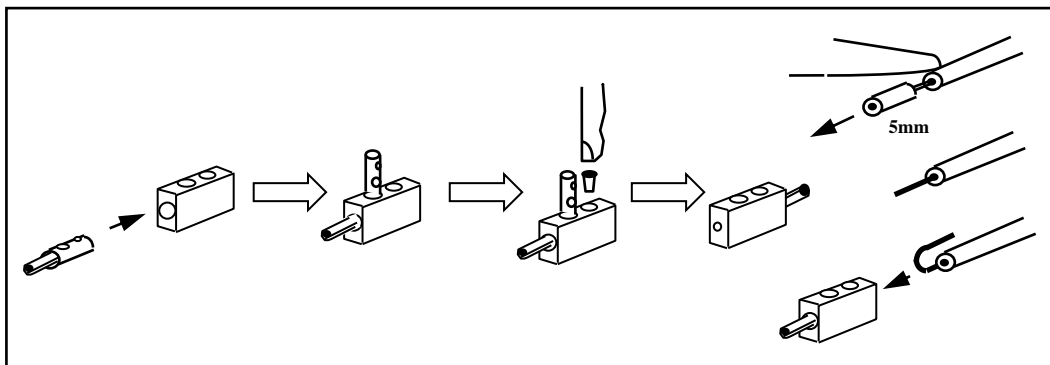
Có thể kết nối mô-đun pin (như là EIBAT-200 hoặc pin sạc dự phòng) với EIROBO-206 thông qua ổ cắm USB để vận hành bộ điều khiển một cách độc lập mà không cần máy tính. Chạy và dừng chương trình được thực hiện bằng cách nhấn nút màu cam trên bộ điều khiển.

## 1.6. Chuẩn bị dây kết nối cho các thí nghiệm

Để thực hiện các bài tập mô tả trong các thí nghiệm sau, chúng ta cần dây nối có phích cắm (dây và phích cắm có thể được tìm thấy trong hộp dụng cụ).

Chuẩn bị dây như sau:

- Chuẩn bị 6 cặp dây, mỗi dây dài 12cm.
- Loại bỏ một phần ngắn (5mm) của lớp cách điện bằng nhựa ở đầu dây.
- Uốn các đầu của dây lại, sao cho chúng phẳng so với lớp cách điện (xem hình 2.4).
- Chèn đầu dây đã chuẩn bị (với các sợi gấp lại trên lớp cách điện) vào đầu phích cắm (xem hình 2.4).
- Vặn chặt ốc vít. Không siết quá chặt để tránh làm đứt dây.
- Phích cắm bao gồm ba phần: đầu cắm kim loại, chất cách điện bằng nhựa và vít. Nếu chúng được tách ra, hãy lắp phích cắm qua lỗ lớn nhất vào chất cách điện. Lưu ý hướng lắp vào.
- Sử dụng một phích cắm bổ sung để ổn định phích cắm kim loại trong vỏ cách điện của nó.
- Vặn vít một vòng vào ren của nó (không vặn chặt hết cỡ).



Hình 2. 4

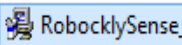
## 1.7. Cài đặt phần mềm RobocklySense

**Phần mềm và trình điều khiển phải được cài đặt trước khi kết nối bất kỳ mô-đun nào với máy tính.**

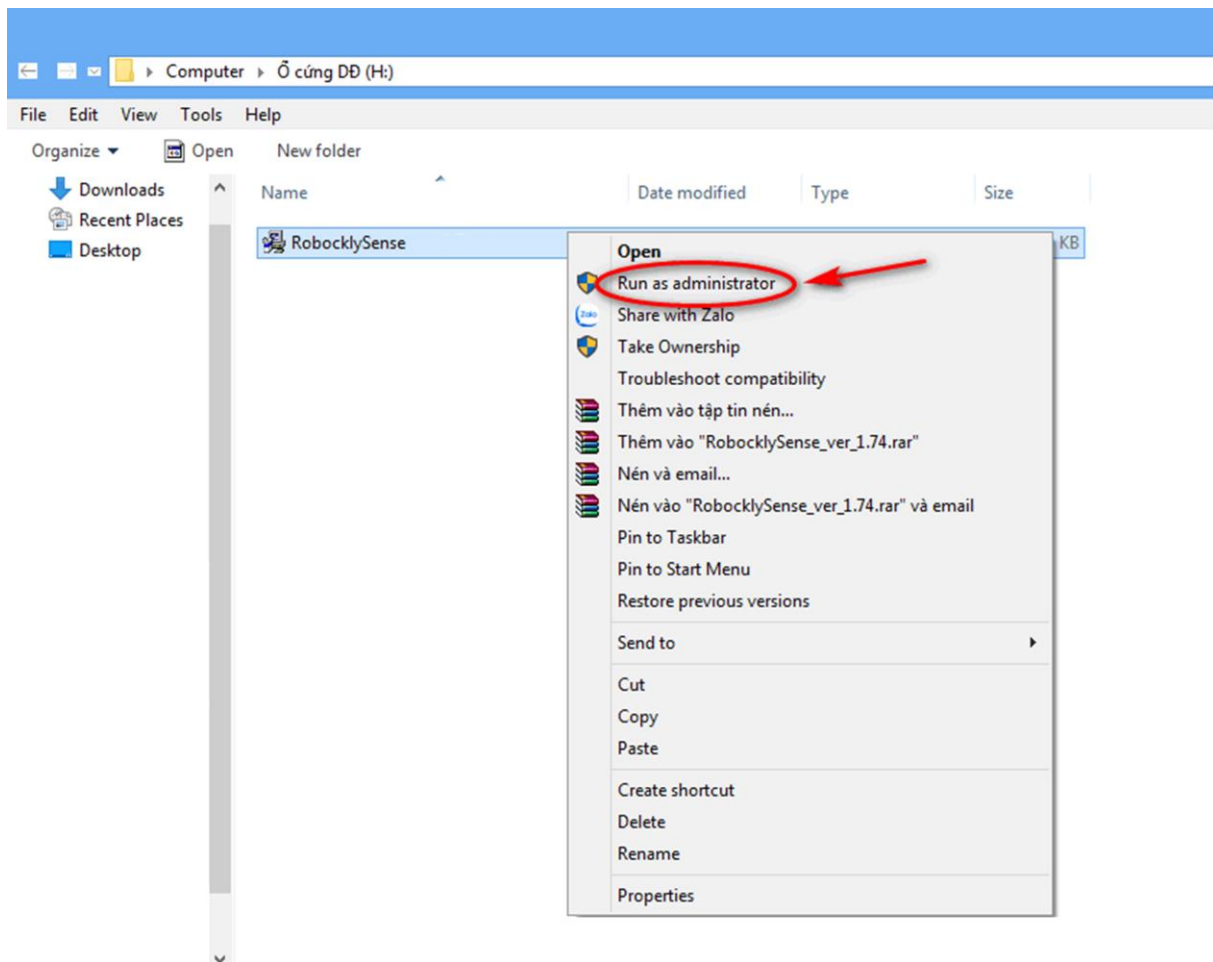
Làm theo hướng dẫn trên màn hình. Quá trình cài đặt rất đơn giản và các trình điều khiển cần thiết được cài đặt một cách tự động.

Quá trình cài đặt bao gồm hai phần: cài đặt phần mềm RobocklySense và cài đặt trình điều khiển USB.

### Cách cài đặt phần mềm RobocklySense.

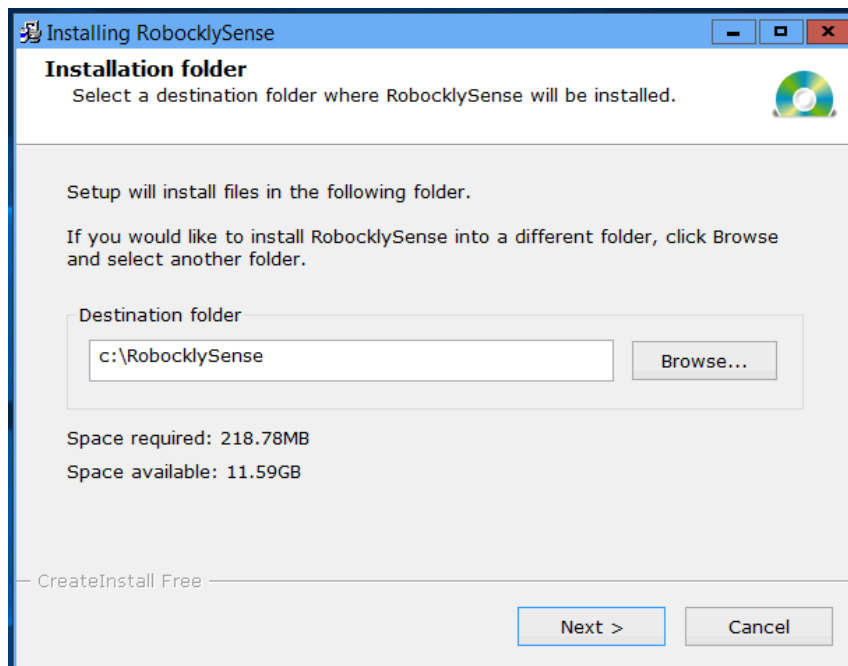
- Kích chuột phải vào biểu tượng  . Kích Run administrator.





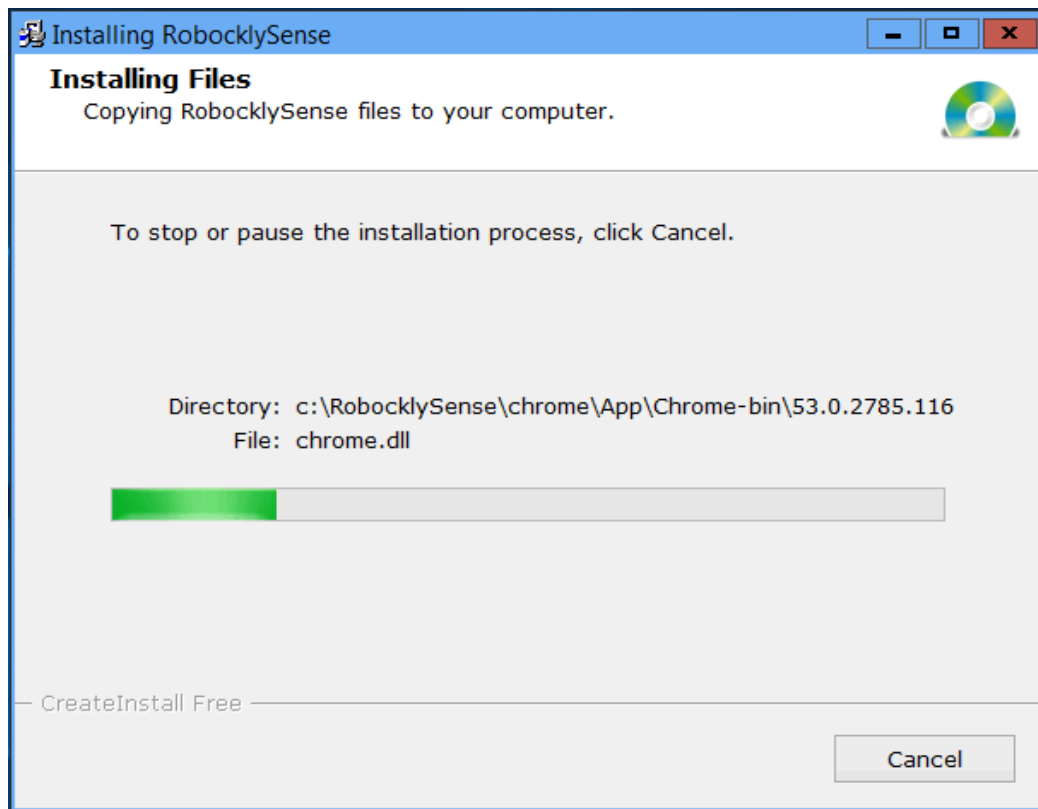
Hình 2. 5

2. Màn hình xuất hiện.



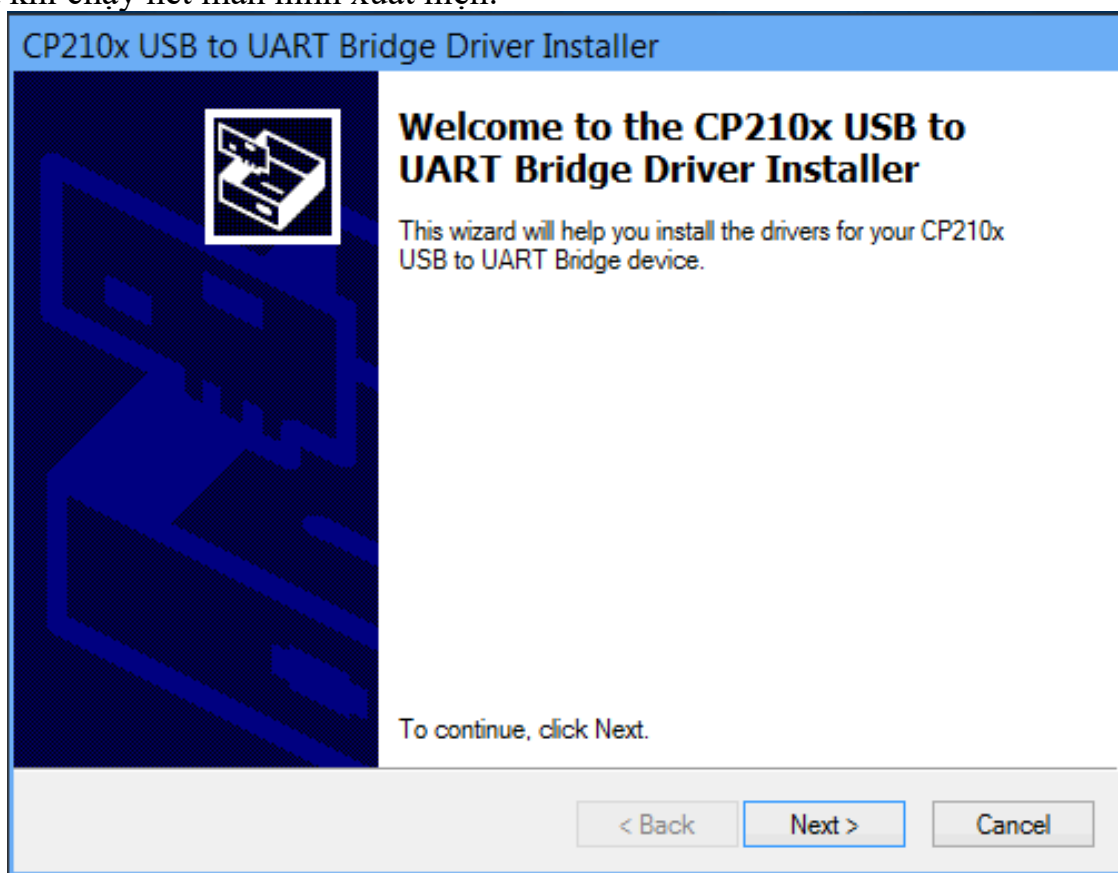
Hình 2. 6

3. Nhấn **Next** quá trình cài đặt bắt đầu



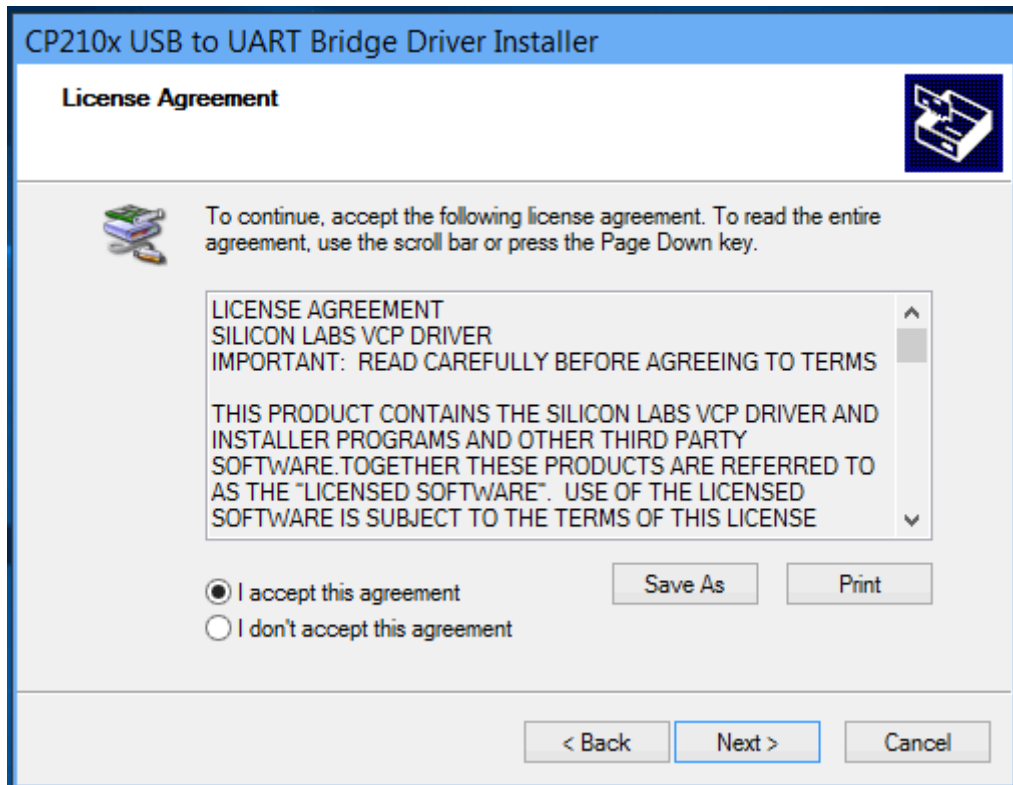
Hình 2. 7

Sau khi chạy hết màn hình xuất hiện.



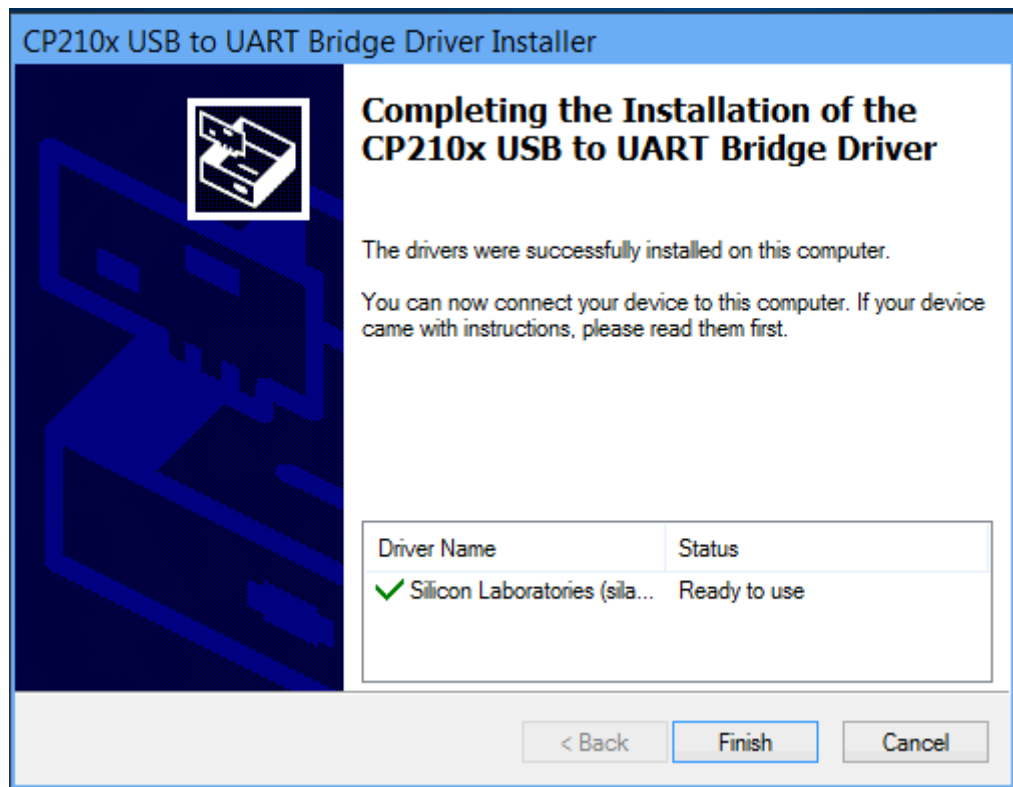
Hình 2. 8

4. Nhấn **Next** để cài đặt trình điều khiển USB.



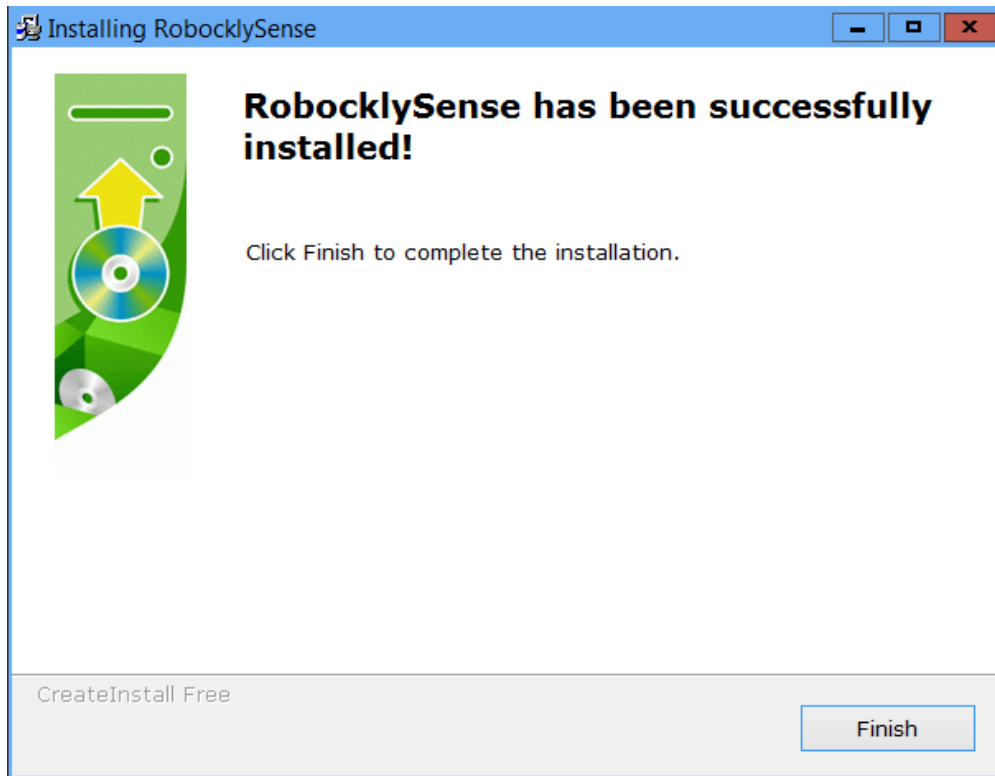
Hình 2. 9

5. Nhấn **Next**



Hình 2. 10

6. Nhấn **Finish** để kết thúc cài đặt trình điều khiển USB. màn hình xuất hiện



**Hình 2. 11**

## 7. Nhấn **Finish** để hoàn thành cài đặt phần mềm


Sau khi quá trình cài đặt hoàn tất, Phần mềm RobocklySenses đã sẵn sàng để sử dụng.

Biểu tượng phím tắt RobocklySense  sẽ xuất hiện trên màn hình máy tính .

### **Ghi chú:**

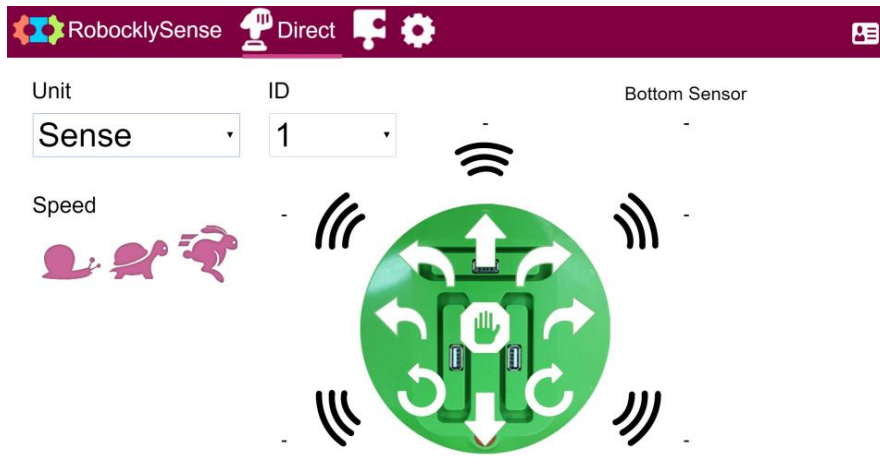
Việc nâng cấp phần mềm có thể được thực hiện bất cứ lúc nào. Cài đặt phần mềm nâng cấp chỉ cần thay thế các tệp có liên quan, vì vậy không cần gỡ cài đặt phần mềm trước khi nâng cấp.

## 1.8. Bắt đầu chương trình RobocklySense

Bạn có thể tìm thấy biểu tượng chương trình RobocklySense  trên màn hình máy tính.

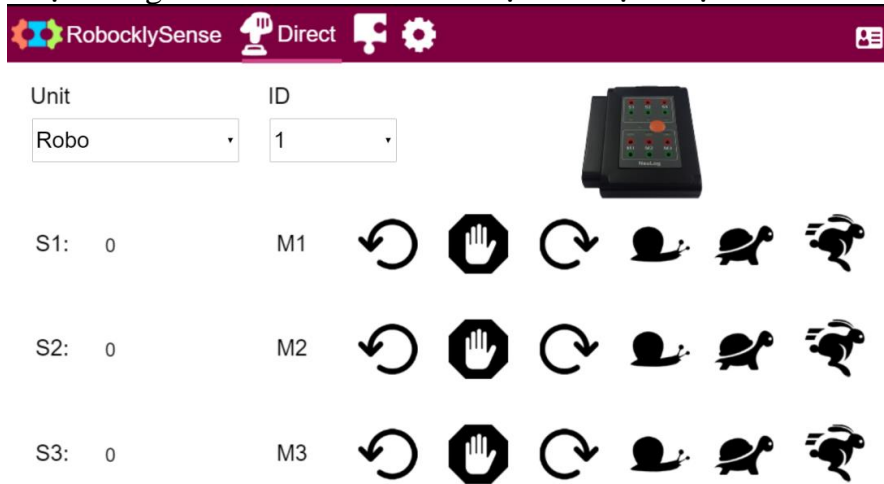
Nhấp đúp vào biểu tượng RobocklySense  để chạy phần mềm RobocklySense.

Chương trình được mở trong trình duyệt và màn hình sau xuất hiện:



**Hình 2. 12**

Màn hình này là chế độ Trực tiếp (**Direct**) cho robot EISENSE-MAKE. Thay đổi lựa chọn trong ô **Unit** thành **Robo** và bạn sẽ nhận được màn hình sau:



**Hình 2. 13**

Màn hình này sẽ được giải thích trong thí nghiệm 1.1.

Đóng chương trình được thực hiện trong hai bước:

1. Đóng cửa sổ trình duyệt.

2. Nhấp vào biểu tượng RobocklySense  ở phía dưới và đóng cửa sổ đã mở.



**Hình 2. 14**

## THÍ NGHIỆM 1.1 – CHẾ ĐỘ TRỰC TIẾP (DIRECT)

### Mục tiêu:

- Tìm hiểu chi tiết các bộ phận của EISENSE-MAKE.
- Cách kết nối các bộ phận với EIROBO-206.
- Cách vận hành các bộ phận ở chế độ Trực tiếp (**Direct**).

### Thiết bị cần thiết:

- Máy tính
- Phần mềm RobocklySense
- Bộ linh kiện lắp ráp EISENSE-MAKE

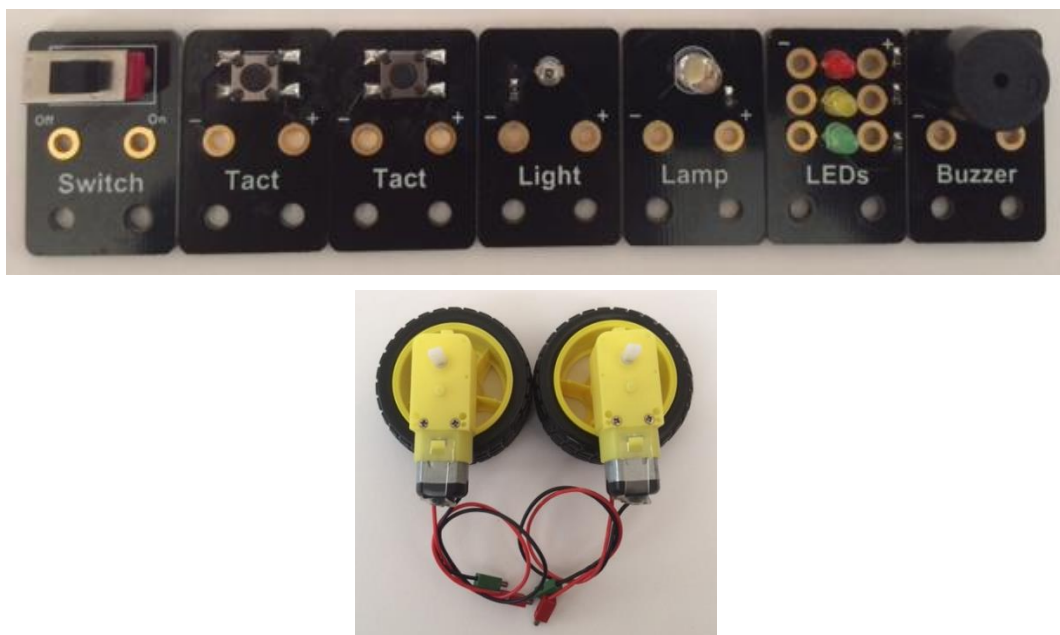
### Thảo luận:

Phần mềm RobocklySense cho phép vận hành trực tiếp bộ điều khiển EIROBO-206.

Chúng ta sẽ kết nối các mô-đun với EIROBO-206 và tìm hiểu cách vận hành trực tiếp mà không cần lập trình.

### Trình tự thực hiện:

1. EISENSE-MAKE bao gồm các bộ phận sau:

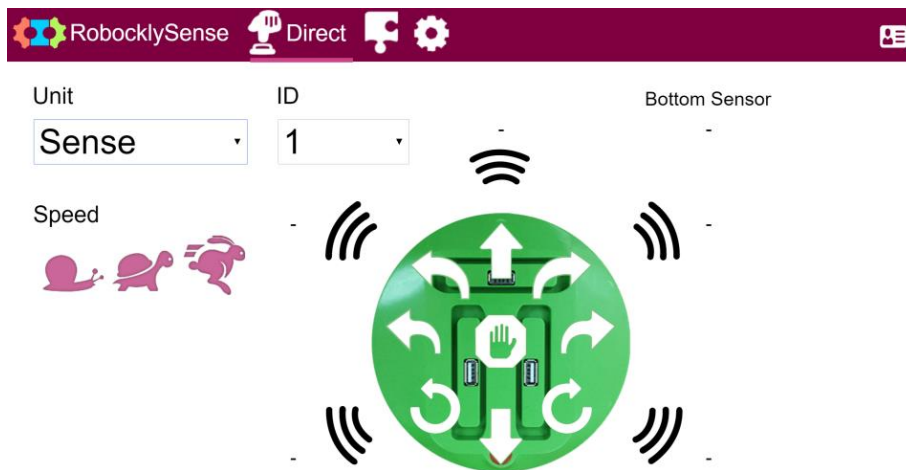


**Hình 2. 15**

Mỗi bộ phận được kết nối với bộ điều khiển EIROBO-206 bằng hai dây điện có phích cắm.

2. Kết nối bộ điều khiển EIROBO-206 với máy tính bằng cáp USB.
3. Chạy phần mềm RobocklySense.

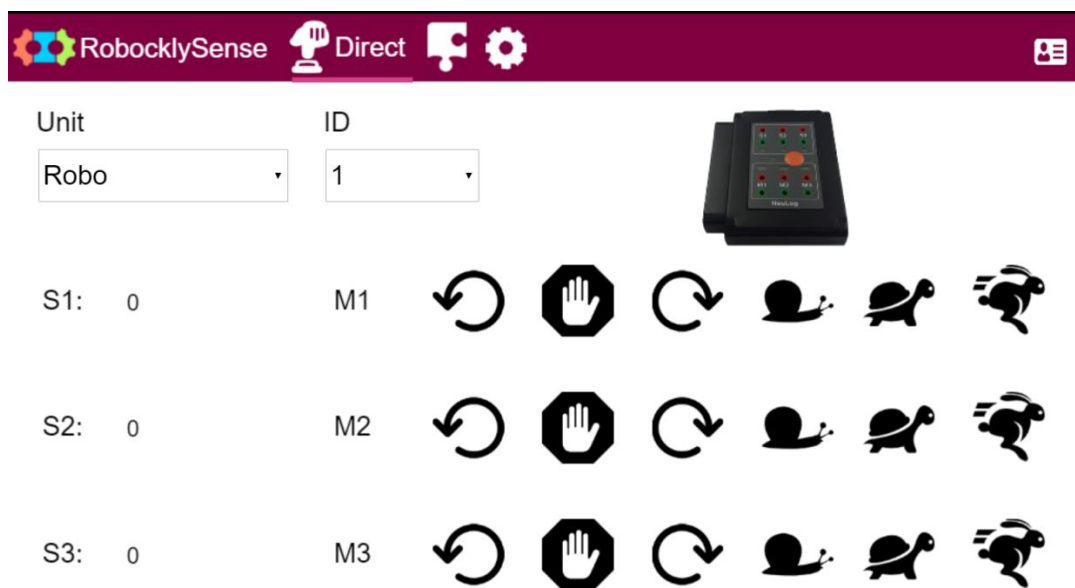
Chương trình được mở trong trình duyệt và màn hình sau xuất hiện:



Hình 2. 16

Màn hình này là chế độ Trực tiếp (**Direct**) cho robot EISENSE-MAKE.

4. Thay đổi lựa chọn trong ô **Unit** thành **Robo** và bạn sẽ nhận được màn hình sau:




Hình 2. 17


Chế độ Trực tiếp (**Direct**) cho phép bạn kiểm tra hệ thống và kết nối trước khi lập trình và chạy chương trình. Giai đoạn này giúp hạn chế rất nhiều lỗi trong quá trình phát triển.

Màn hình chế độ Trực tiếp (**Direct**) được thay đổi theo tùy chọn từng loại thiết bị. Mỗi thiết bị có số ID mặc định riêng. Người dùng có thể thay đổi số ID mô-đun. Trong tài liệu này, chúng ta sẽ sử dụng số ID mặc định của các thiết bị là “1”.


### 1.1.1. Kiểm tra động cơ



1. Kết nối một trong các động cơ với các cực M1.

2. Nhấp vào biểu tượng **Quay thuận**  trên dòng **M1**.  
Động cơ phải quay theo chiều kim đồng hồ.

3. Nhấp vào biểu tượng **Dừng**  và kiểm tra xem động cơ có dừng lại không.

4. Nếu động cơ quay ngược chiều kim đồng hồ, hãy thay đổi vị trí các dây được kết nối với động cơ và kiểm tra vòng quay động cơ.

5. Nhấp vào biểu tượng **Quay ngược** .  
Động cơ phải xoay ngược chiều kim đồng hồ.

6. Đổi hướng quay của động cơ bằng các lệnh quay ngược  và quay thuận .

### Lưu ý:

**Thay đổi hướng quay đột ngột mà không dừng động cơ sẽ làm giảm tuổi thọ của động cơ. Nên dừng động cơ trước khi thay đổi hướng quay của nó.**

7. Thay đổi tốc độ quay của động cơ bằng các lệnh tốc độ:



**Chậm**




**Vừa**



**Nhanh**


**Hình 2. 18**

8. Nhấp vào biểu tượng **Dừng**  và kiểm tra xem động cơ có dừng lại không.

### **1.1.2. Kiểm tra Còi (Buzzer)**

1. Kết nối còi với công ra M1 cùng với động cơ.


Chúng ta sử dụng các nút điều khiển động cơ để điều khiển còi.

2. Nhấp vào biểu tượng **Quay thuận**  trên dòng M1.  
Còi sẽ phải kêu.

Nếu còi không kêu, hãy kiểm tra các dây kết nối.


Để còi hoạt động được thì dòng điện qua còi phải đi theo một chiều nhất định.

3. Nhấp vào biểu tượng **Dừng**  và kiểm tra xem còi có tắt không.

4. Nhấp vào biểu tượng **Quay ngược** .  
Còi sẽ **không** kêu.

Chiều của dòng điện đầu ra ảnh hưởng đến còi.



5. Nhấp vào điều khiển **Dừng** .

### 1.1.3. Kiểm tra đèn LED

Bộ đèn LED bao gồm 3 đèn LED (đỏ, vàng và xanh).

1. Kết nối đèn LED màu đỏ với đầu ra M1.
2. Kết nối đèn LED màu vàng với đầu ra M2.
3. Kết nối đèn LED màu xanh với đầu ra M3.


Chúng ta sẽ sử dụng các nút điều khiển động cơ để điều khiển đèn LED.

4. Nhấp vào biểu tượng **Quay thuận**  của M1.

Đèn LED màu đỏ sẽ sáng.

Nếu đèn không sáng, hãy thay đổi đầu dây kết nối.


Vì để làm sáng đèn LED, dòng điện qua đèn LED phải chạy theo một chiều nhất định.

5. Nhấp vào biểu tượng **Dừng**  và kiểm tra xem đèn LED đỏ có tắt không.

6. Nhấp vào biểu tượng **Quay ngược** .


Đèn LED sẽ **không** bật.

Lưu ý: Chiều của dòng điện đầu ra ảnh hưởng đến đèn LED.

7. Nhấp vào biểu tượng **Dừng** .


8. Nhấp vào biểu tượng **Quay thuận**  của M2.

Đèn LED màu vàng sẽ sáng.

9. Nhấp vào biểu tượng **Dừng**  và kiểm tra xem đèn LED vàng có tắt không.

10. Nhấp vào biểu tượng **Quay thuận**  của M3.

Đèn LED màu xanh sẽ sáng.

11. Nhấp vào biểu tượng **Dừng**  và kiểm tra xem đèn LED xanh có tắt không.

12. Ngắt kết nối đèn LED.

### 1.1.4. Kiểm tra đèn (Lamp)

1. Kết nối đèn (Lamp) với các cổng M1.

Chúng ta sẽ sử dụng các nút điều khiển động cơ để điều khiển đèn.

2. Nhấp vào biểu tượng **Quay thuận**  của M1.

Đèn (Lamp) sẽ sáng.

Nếu đèn không sáng, hãy thay đổi kết nối của dây.


Đèn (Lamp) cũng là một loại LED. Để bật đèn sáng, dòng điện phải chạy theo một chiều nhất định.

3. Nhấp vào biểu tượng **Dừng**  và kiểm tra xem đèn có tắt không.

4. Nhấp vào biểu tượng **Quay ngược** .

Đèn sẽ **không** sáng.

Lưu ý: Chiều của dòng điện đầu ra ảnh hưởng đến đèn (Lamp).

5. Nhấp vào biểu tượng **Dừng** .

6. Không ngắt kết nối đèn.

#### **1.1.5. Kiểm tra cảm biến ánh sáng (Light)**

1. Kiểm tra xem đèn (Lamp) có tắt không.

2. Kết nối các cổng của cảm biến ánh sáng (**Light**) với các cổng **S1**.

Một giá trị sẽ xuất hiện trong trường (ô) **S1**.

Cảm biến đã đo cường độ ánh sáng môi trường xung quanh.

3. Nếu giá trị là 0, hãy kiểm tra xem dây cảm đã được kết nối đúng chưa.

4. Bật đèn (Lamp) và đặt nó gần cảm biến ánh sáng.

Giá trị đo mới phải cao hơn nhiều so với giá trị trước.

5. Chặn ánh sáng của đèn (Lamp).

Giá trị đo được sẽ bị thay đổi.

6. Nhấp vào biểu tượng **Dừng**  để tắt đèn (Lamp).

7. Ngắt kết nối đèn (Lamp) và cảm biến ánh sáng (Light).

#### **1.1.6. Kiểm tra công tắc gạt (Switch)**

1. Kết nối công tắc với cổng **S1**.

2. Gạt công tắc sang trái (vị trí Off).

Giá trị đọc được của S1 phải là 0.

3. Gạt công tắc sang phải (vị trí On).

Giá trị đọc được của S1 phải là 1000 (giá trị tối đa).

Khi công tắc ở bên phải (On), cổng đầu vào được kết nối với nguồn điện 5V và điện áp này được chuyển đổi thành số 1000.

#### **1.1.7. Kiểm tra công tắc nhấn nhả (Tact)**

1. Kết nối một trong các Công tắc nhấn nhả (Tact) với các cổng **S1**.

Giá trị đọc được của S1 phải là 0.

2. Nhấn công tắc và giữ nó.

Giá trị đọc được của S1 phải là 1000 (giá trị tối đa).

Khi công tắc được nhấn, cổng đầu vào được kết nối với nguồn điện 5V và điện áp này được chuyển đổi thành số 1000.

3. Đóng phần mềm RobocklySense.

## THÍ NGHIỆM 1.2 – CÁC CHƯƠNG TRÌNH ĐẦU TIÊN

### Mục tiêu:

- Sử dụng các khối lệnh để xây dựng một thủ tục (chương trình con).
- Tải chương trình xuống bộ điều khiển và chạy nó.

### Thiết bị cần thiết:

- Máy tính
- Phần mềm RobocklySense
- Robot lắp ráp EISENSE-MAKE

### Thảo luận:

Một chương trình máy tính bao gồm các chuỗi lệnh theo tập lệnh ngôn ngữ lập trình. Có nhiều ngôn ngữ lập trình khác nhau với các tập lệnh khác nhau và các kiểu lập trình khác nhau.

Chúng ta phải cho máy tính biết lệnh nào là lệnh đầu tiên trong chuỗi. Máy tính sẽ thực hiện lệnh này và sẽ tiếp tục thực hiện một lệnh phía sau trong chuỗi.

Chương trình có thể bao gồm các câu lệnh thực thi chuyển đến một lệnh khác phía sau.

Chương trình có thể bao gồm các câu lệnh thực thi chuyển đến các chuỗi khác theo điều kiện hoặc không có điều kiện. Các thí nghiệm sau đây mô tả các tùy chọn này.

Có nhiều loại ngôn ngữ lập trình. Mỗi loại có cú pháp riêng và bộ câu lệnh riêng.

RobocklySense là một trình soạn thảo lập trình dạng khối trực quan. RobocklySense sử dụng các khối liên kết với nhau để xây dựng một chương trình giúp người học tập trung vào giải quyết vấn đề thay vì viết đoạn mã của một ngôn ngữ lập trình nhất định.

Robockly rất thân thiện, rất dễ lập trình và chạy các chương trình robot. Nó rất mạnh mẽ cho các chương trình robot và là phần mềm tốt nhất để bắt đầu học tập.

### Trình tự thực hiện:

1. Kết nối mô-đun đèn LED với bo mạch chính như trong hình dưới đây:

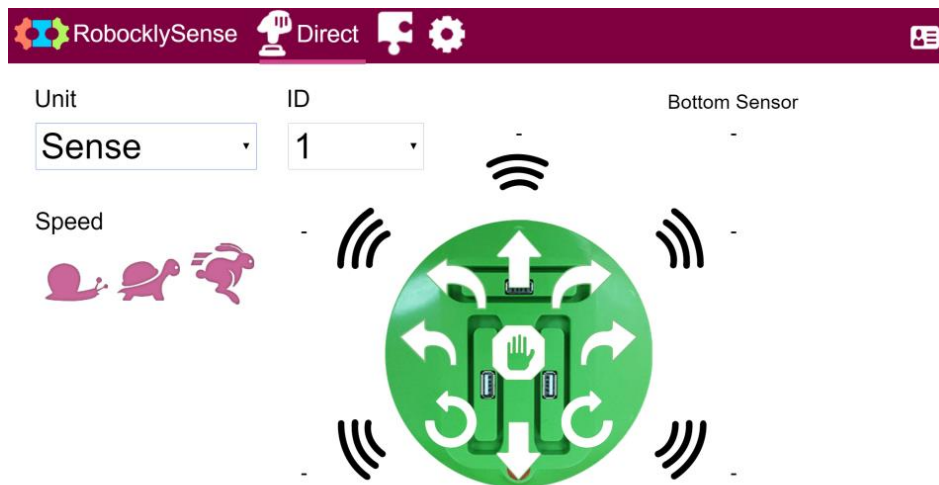


Hình 2. 19

2. Lắp EIROBO-206 vào bo mạch chính.

3. Kết nối EIROBO-206 với máy tính bằng cáp USB.
4. Kết nối đèn LED màu đỏ với cổng M1.
5. Kết nối đèn LED màu vàng với cổng M2.
6. Kết nối đèn LED màu xanh với cổng M3.
7. Chạy phần mềm RobocklySense.

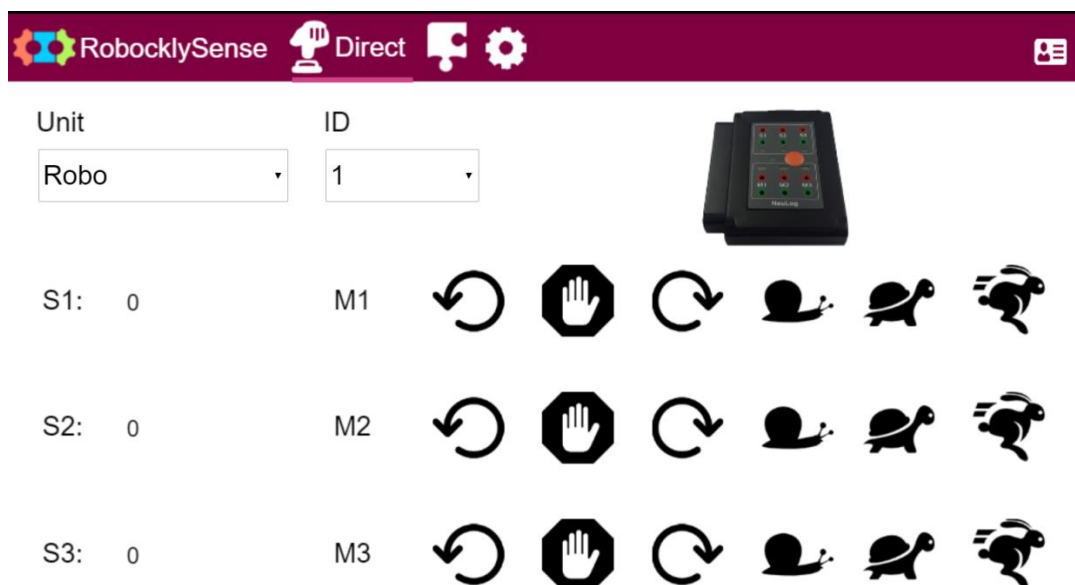
Chương trình được mở trong trình duyệt và màn hình sau xuất hiện



**Hình 2. 20**

Màn hình này là chế độ Trực tiếp (**Direct**) cho robot EISENSE-MAKE.

8. Thay đổi lựa chọn trong ô **Unit** thành **Robo** và bạn sẽ nhận được màn hình sau:



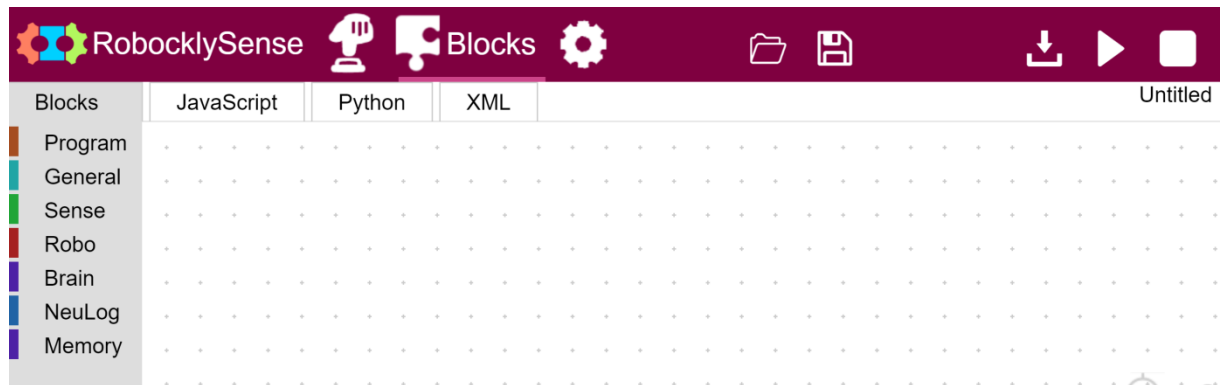
**Hình 2. 21**

Chế độ Trực tiếp (**Direct**) cho phép bạn kiểm tra hệ thống và kết nối trước khi lập trình và chạy chương trình. Giai đoạn này giúp hạn chế rất nhiều lỗi trong quá trình phát triển.

9. Kiểm tra ba đèn LED như được mô tả trong thí nghiệm 1.1.

### 1.2.1. Chương trình đầu tiên - Đèn LED nhấp nháy

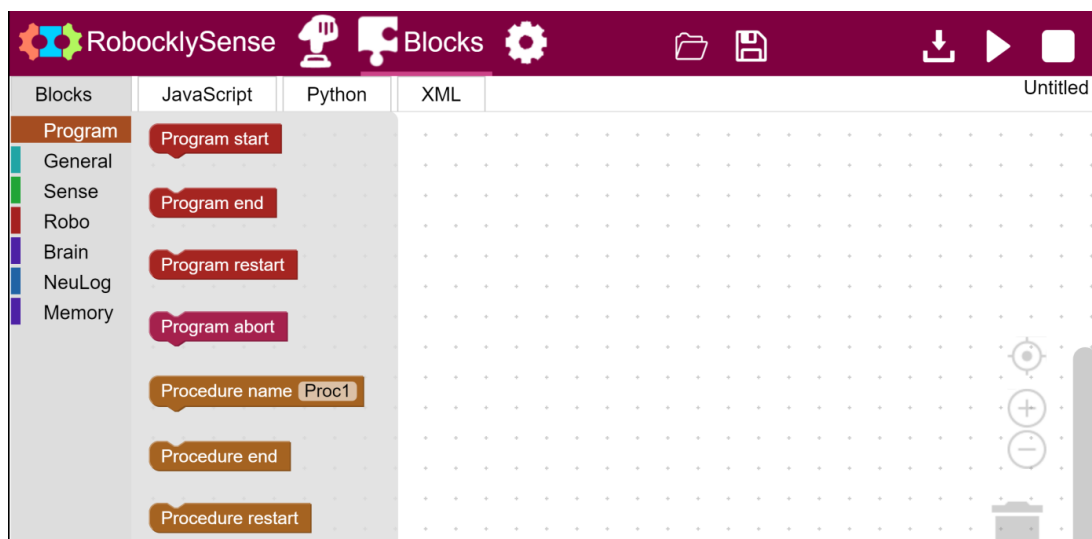
1. Nhấp chọn biểu tượng **Block** để chuyển sang giao diện lập trình.



Hình 2. 22

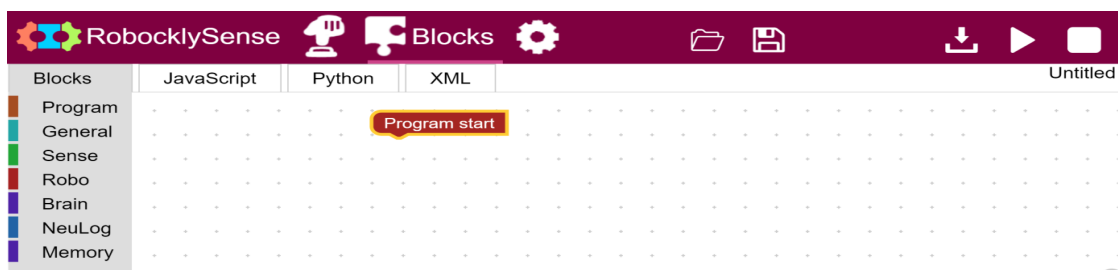
Một chương trình lập trình bao gồm các chuỗi câu lệnh.

2. Nhấp vào **Program** và danh sách câu lệnh chương trình sẽ xuất hiện:



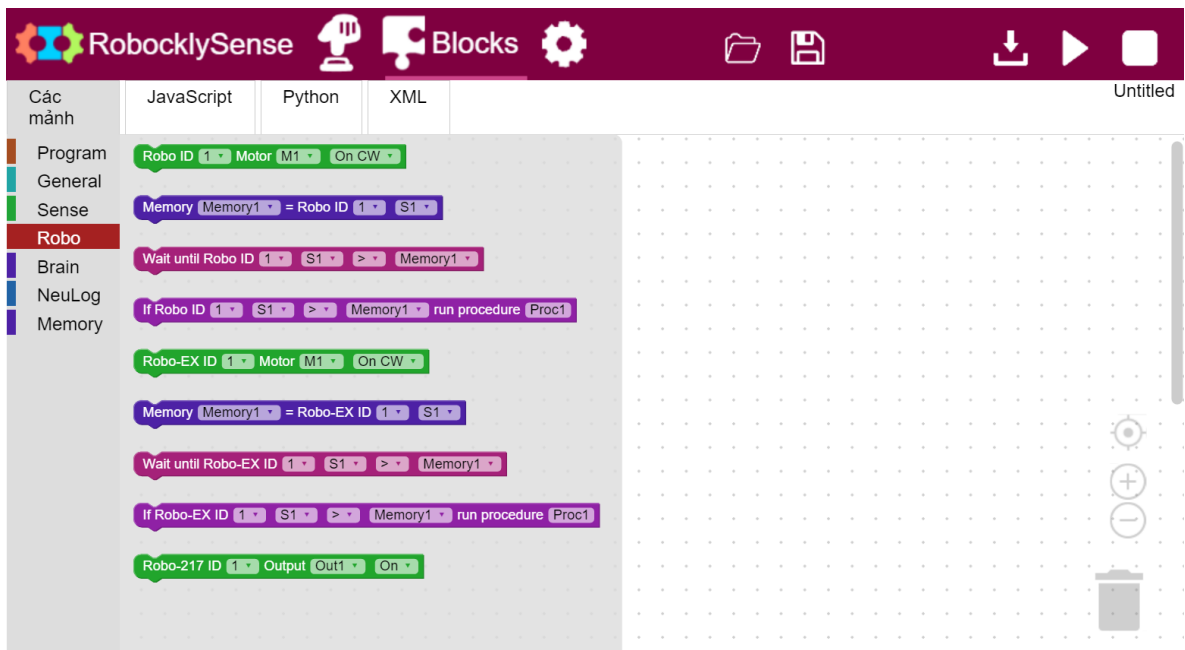
Hình 2. 23

3. Nhấn giữ khối lệnh **Program start** trong **Program** và kéo nó sang phải.



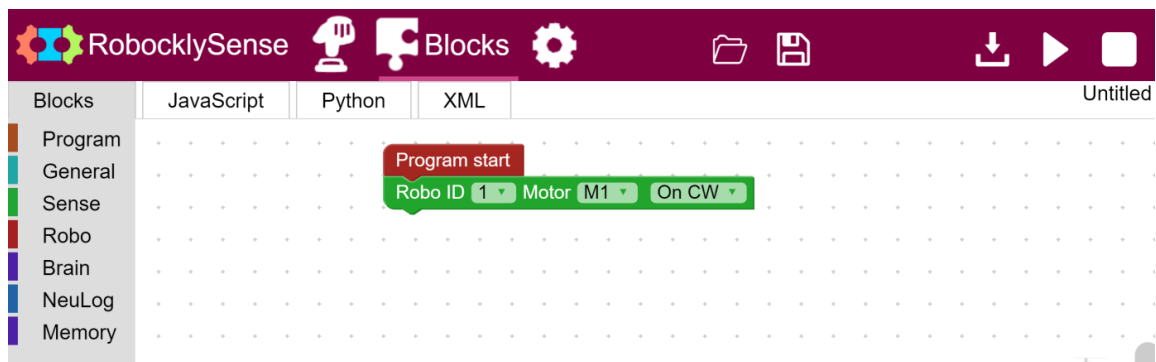
Hình 2. 24

4. Nhấp vào **Robo** và danh sách câu lệnh của Robo sẽ xuất hiện:



Hình 2. 25

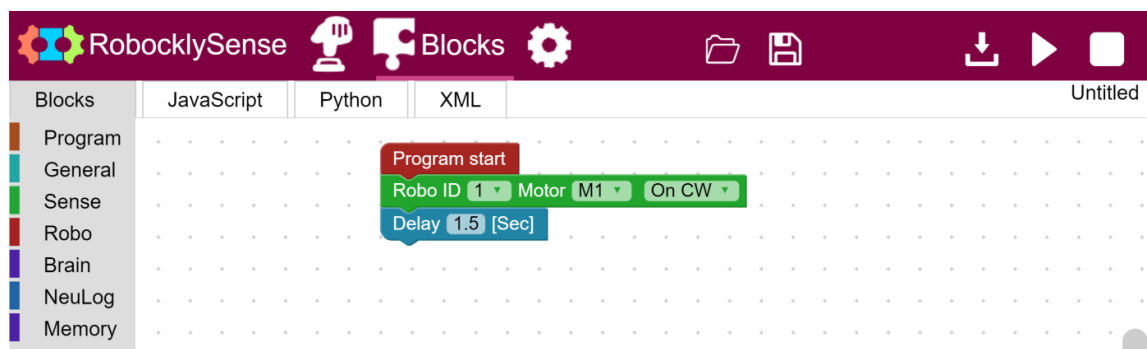
5. Nhấp vào khối câu lệnh **Robo Motor** và kéo nó vào bên dưới khối lệnh **Program start**. Chúng sẽ được gắn vào với nhau.



Hình 2. 26

6. Nhấp vào **General** và chọn khối lệnh **Delay**.

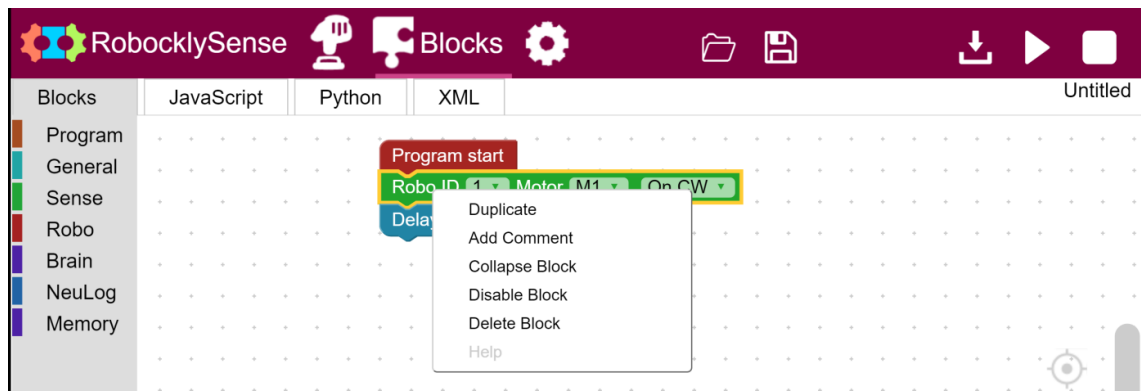
Kéo nó vào bên dưới khối lệnh **Robo Motor**.



Hình 2. 27

7. Thay đổi giá trị độ trễ thành 5 giây.

8. Nhấp chuột phải vào khối lệnh **Robo Motor**.

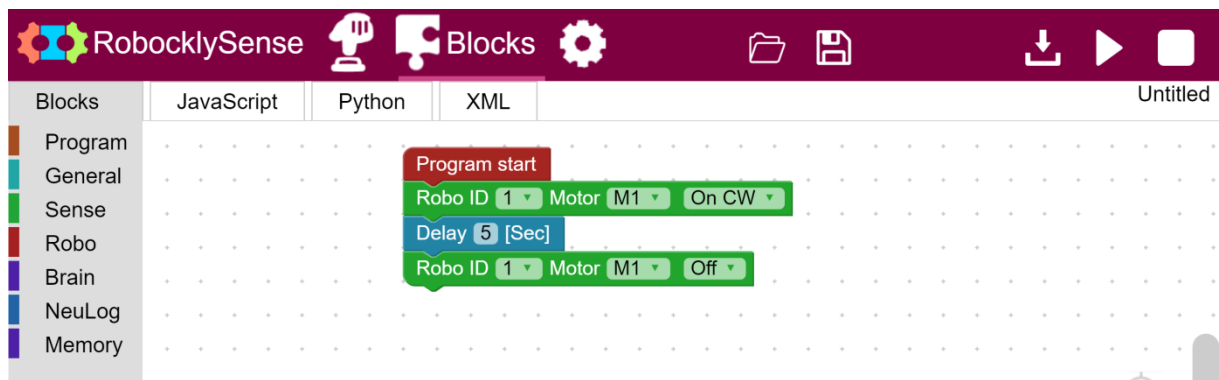


Hình 2. 28

9. Chọn **Duplicate** (Nhân đôi) và khối câu lệnh **Robo Motor** mới sẽ xuất hiện. Kéo nó vào sau khối lệnh **Delay**.

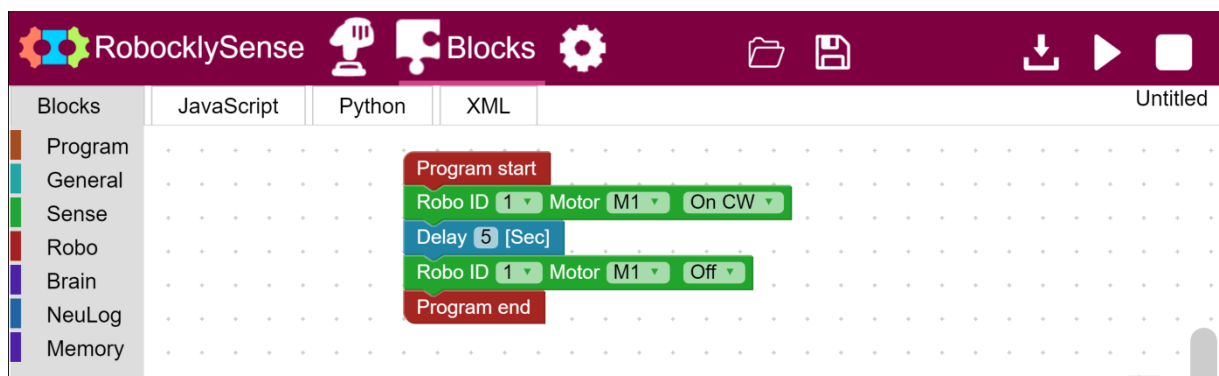
10. Thay đổi chỉ thị **On CW** (Quay thuận) của khối lệnh **Robo Motor** mới thành **Off** (Dừng).

11. Kiểm tra xem chương trình của bạn có giống như chương trình sau không:




Hình 2. 29

12. Nhấp vào **Program** và chọn khối lệnh **Program end**. Kéo nó vào sau khối lệnh **Robo Motor**.



Hình 2. 30

13. Nhấp vào biểu tượng **Lưu**  và lưu chương trình dưới tên **Den\_LED\_1**.



### 1.2.2. Tải xuống chương trình

1. EIROBO-206 cũng là một máy tính có bộ nhớ flash.

Chúng ta có thể tải chương trình vào bộ nhớ của nó.

2. Nhấp vào biểu tượng  để tải xuống chương trình.

Bước này sẽ truyền chương trình máy tính vào bộ nhớ flash của EIROBO-206 và thay thế chương trình trước, nếu đã lưu trong đó.

3. Nhấp vào biểu tượng  để chạy chương trình.

Đèn màu đỏ sẽ sáng trong 5 giây và sau đó sẽ tắt.

4. Để chạy chương trình lưu trong bộ điều khiển, chúng ta cũng có thể sử dụng nút ấn **Run/Stop** màu cam nằm trên bộ điều khiển.

Nhấn nút **Run/Stop** màu cam trên EIROBO-206 và bạn sẽ thấy đèn trên bộ điều khiển nhấp nháy.

Bạn có thể đóng phần mềm RobocklySense và chạy chương trình bằng nút nhấn này.

Trong trường hợp này, máy tính chỉ được sử dụng với vai trò cung cấp nguồn điện.

Thay thế máy tính bằng mô-đun pin sẽ tạo ra một bộ điều khiển robot độc lập.

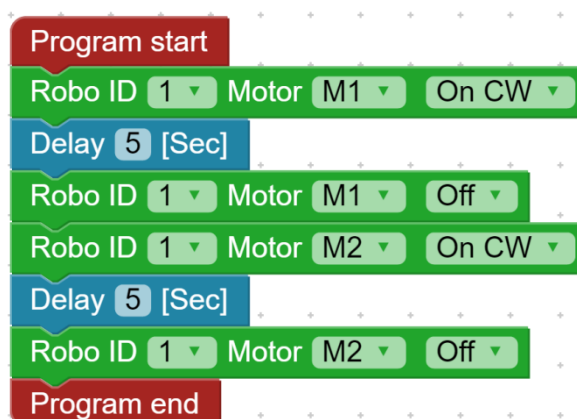
#### Những lưu ý quan trọng:

Kết nối lại EIROBO-206 với máy tính nên được thực hiện khi nó đang không chạy (đèn Led trên bộ điều khiển không nhấp nháy). Bạn có thể sử dụng pin dự phòng để kết nối với ổ cắm của EIROBO-206.


### 1.2.3. Đèn LED nhấp nháy

1. Nhân đôi (**Duplicate**) khối lệnh **Delay** và hai khối lệnh **Robo Motor**.

2. Kéo các khối lệnh và xây dựng chương trình như sau:



Hình 2. 31

3. Nhấp vào biểu tượng  và lưu chương trình dưới tên **Den\_LED\_2**.

4. Nhấp vào biểu tượng  để tải xuống chương trình.

5. Nhấp vào biểu tượng  để chạy chương trình.

Đèn màu đỏ sẽ bật trong 5 giây, đèn đỏ tắt và đèn vàng bật trong 5 giây và sau đó tắt.

#### **1.2.4. Bài tập thử thách – Đèn LED nhấp nháy**

Nhiệm vụ 1: Nâng cấp chương trình Den\_LED\_2 để cả ba đèn LED sẽ được bật cùng nhau trong 3 giây và tắt cùng nhau.

Tải chương trình xuống bộ điều khiển và chạy nó.

Chạy chương trình từ máy tính và kiểm tra hoạt động của đèn.

Nhiệm vụ 2: Thay đổi chương trình để đèn LED sẽ bật theo trình tự từng cái một.

Tải chương trình xuống bộ điều khiển và chạy nó.

Chạy chương trình từ máy tính và kiểm tra hoạt động của nó.

## THÍ NGHIỆM 1.3 – CHƯƠNG TRÌNH CON

### Mục tiêu:

- Sử dụng các chương trình con trong một chương trình.
- Xây dựng và triển khai đèn tín hiệu giao thông.

### Thiết bị cần thiết:

- Máy tính
- Phần mềm RobocklySense
- Robot lắp ráp EISENSE-MAKE

### Thảo luận:

Một chương trình máy tính bao gồm các chuỗi lệnh.

Thay vì dùng một chuỗi câu lệnh, chúng ta có thể chia chương trình thành các chương trình con, đó là các chuỗi câu lệnh và có gắn tên cho từng chuỗi.

Trong một chương trình, có một chương trình chính và các chương trình con. Bằng cách này, khi chúng ta chạy chương trình, máy tính phải biết bắt đầu từ đâu.

Trong thí nghiệm này, chúng ta sẽ tạo ra một chương trình điều khiển đèn nhấp nháy khi nhấn công tắc. Chúng ta sẽ sử dụng một chương trình con nhấp nháy đèn.

Vào cuối chương trình, chúng ta cần thêm một lệnh, lệnh này sẽ đưa chương trình trở về trạng thái bắt đầu (khi chúng ta muốn chương trình được thực thi lặp đi lặp lại), hoặc dừng chương trình và quay trở lại chế độ chờ.


### Trình tự thực hiện:

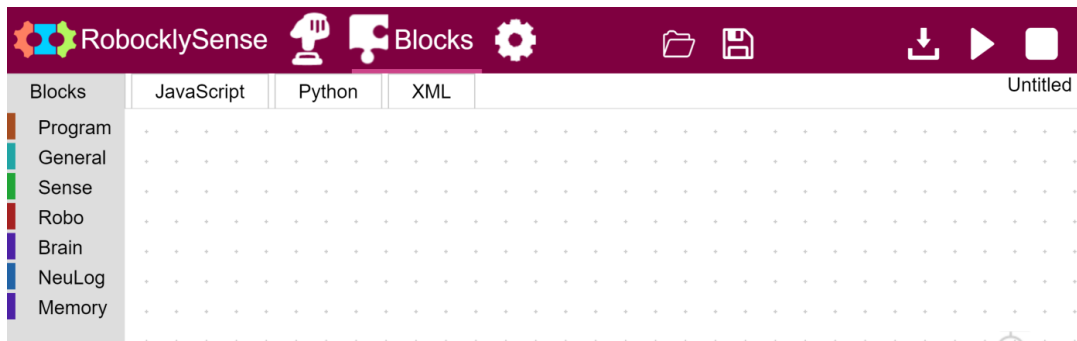
1. Kết nối mô-đun đèn LED với mạch chính như trong hình dưới đây:



**Hình 2. 32**

2. Lắp EIROBO-206 trên mạch chính.
3. Kết nối EIROBO-206 với máy tính bằng cáp USB.
4. Kết nối đèn LED màu đỏ với cổng M1.
5. Kết nối đèn LED màu vàng với cổng M2.

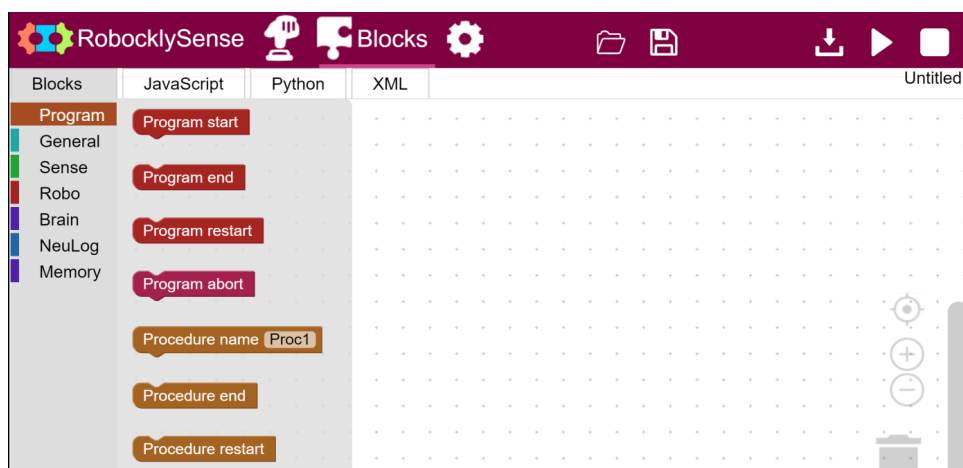
6. Kết nối đèn LED màu xanh với cổng M3.
7. Chạy phần mềm RobocklySense.
8. Kiểm tra ba đèn LED như được mô tả trong thí nghiệm 1.1.
9. Nhấp chọn biểu tượng **Block**  để chuyển sang giao diện lập trình.



Hình 2. 33

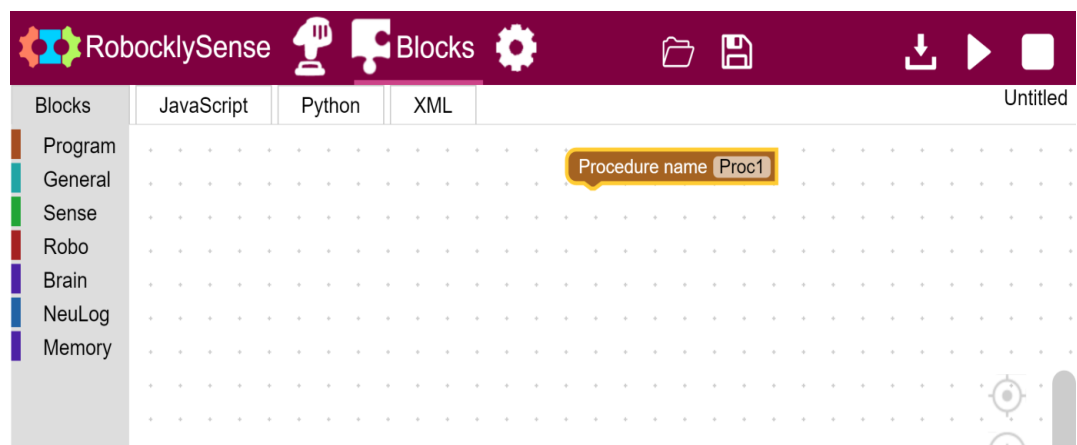
### 1.3.1. Chương trình chính và chương trình con

- 1 Nhấp vào **Program** và danh sách các lệnh chương trình sẽ xuất hiện:



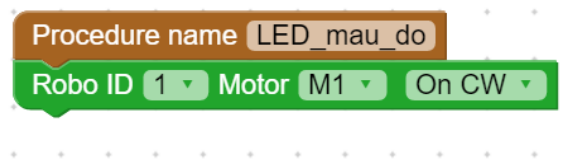
Hình 2. 34

2. Nhấp vào khối lệnh “**Procedure name [Proc1]**” và kéo nó sang phải.



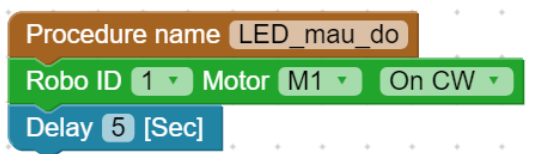
Hình 2. 35

3. Khởi lệnh **Procedure name** có một trường để đặt tên cho chương trình con. Viết tên **LED\_mau\_do** trong trường này.
4. Kéo thêm lệnh **Robo Motor** vào bên dưới khối lệnh **Procedure Name**.



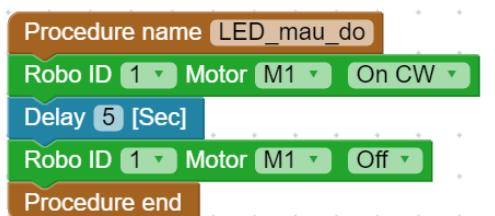
Hình 2. 36

5. Nhấp vào **General** và chọn khối lệnh **Delay**. Kéo nó vào dưới khối lệnh **Robo Motor** và thay đổi giá trị thành 5 giây.



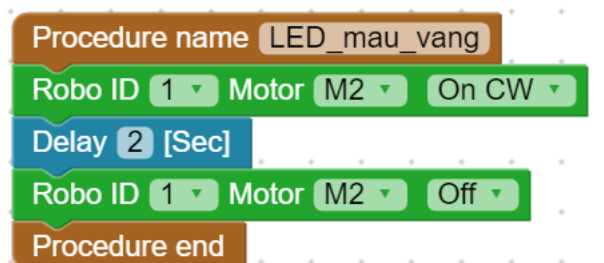
Hình 2. 37

6. Nhân đôi khối lệnh **Robo Motor** và đính kèm nó vào sau khối lệnh **Delay**.
7. Thay đổi chỉ thị của khối lệnh **Robo Motor** mới thành **OFF**.
8. Nhấp vào **Program** và chọn khối lệnh **Procedure end**. Kéo nó vào dưới khối lệnh **Robo Motor**, dưới cùng của đoạn chương trình.



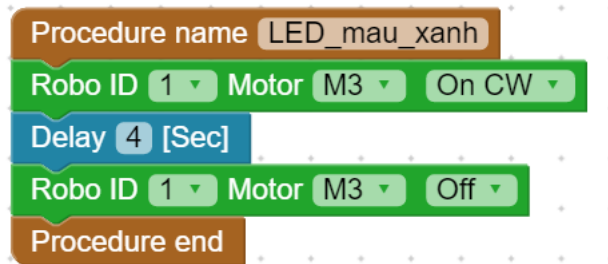
Hình 2. 38

9. Lặp lại các bước như trên và tạo trương trình con **LED\_mau\_vang** với độ trễ 2 giây.



Hình 2. 39

10. Lặp lại các bước trên và tạo chương trình con **LED\_mau\_xanh** với độ trễ 4 giây.



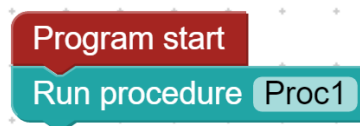
Hình 2. 40

11. Bây giờ chúng ta sẽ xây dựng chương trình chính gọi các chương trình con đó.

Nhấp vào **Program** và danh sách các lệnh chương trình xuất hiện.

12. Nhấp vào khối **Program start** và kéo nó sang phải.

13. Nhấp vào **General** và kéo thêm khối lệnh “**Run procedure Proc1**” vào sau khối lệnh **Program start**.



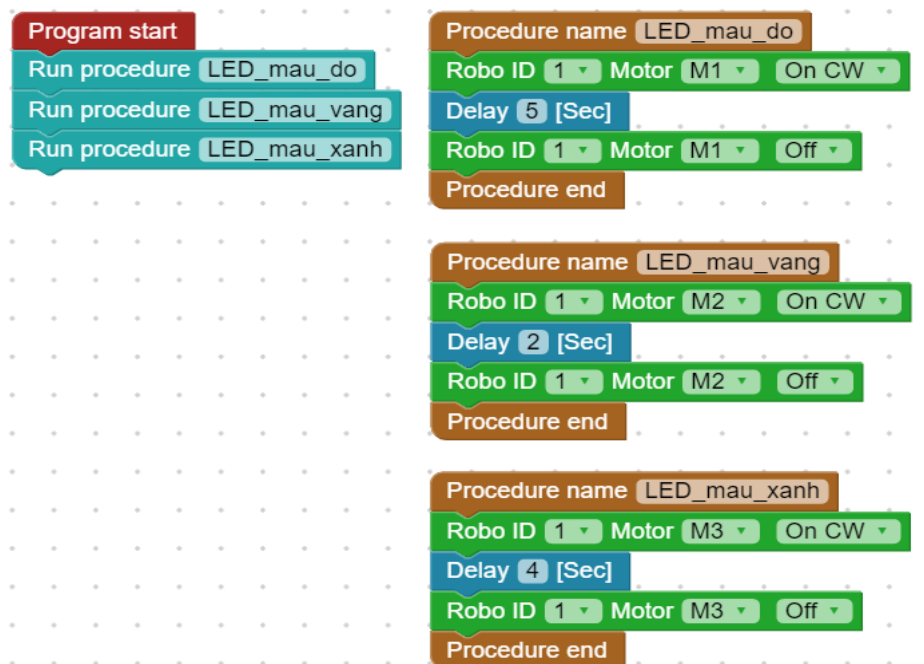
Hình 2. 41

14. Thay đổi tên **Proc1** thành **LED\_mau\_do**.

15. Tạo khối lệnh **Run procedure LED\_mau\_vang**.

16. Tạo khối lệnh **Run procedure LED\_mau\_xanh**.

17. Kiểm tra xem chương trình của bạn có như trường trình sau không:

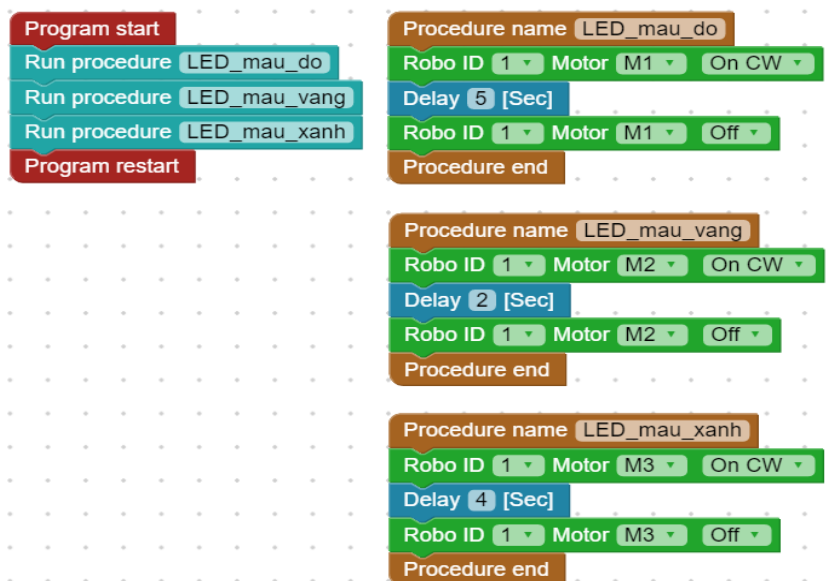


Hình 2. 42




### 1.3.2. Vòng lặp

Hầu hết các chương trình điều khiển và robot là các chương trình chạy trong vòng lặp.

1. Thêm khối lệnh **Program restart** dưới khối lệnh **Run procedure**.
2. Kiểm tra xem chương trình của bạn có như trương trình sau không:




Hình 2. 43

3. Nhấp vào biểu tượng  và lưu chương trình với tên là **Den\_giao\_thong\_1**.
4. Nhấp vào biểu tượng  để tải xuống chương trình.
5. Nhấp vào biểu tượng  để chạy chương trình.


Đèn LED màu đỏ trên bộ điều khiển EIROBO-206 sẽ nhấp nháy cho biết chương trình của nó đang chạy.

Ba đèn LED sẽ sáng giống một đèn giao thông trong vòng lặp vô hạn.

Trong khi chạy, dòng menu được thay đổi thành dòng sau với biểu tượng **Dừng**  ở bên phải.



Hình 2. 44

6. Dừng chương trình bằng cách bấm vào biểu tượng **Dừng** .
  7. Để chạy chương trình trong bộ nhớ EIROBO-206, chúng ta cũng có thể sử dụng nút bấm Run/Stop màu cam nằm trên EIROBO-206.
- Nhấn vào nút Run/Stop màu cam trên bộ điều khiển và ba đèn LED sẽ sáng giống một đèn giao thông trong vòng lặp vô hạn.
8. Để dừng hệ thống đang chạy, hãy nhấn nút nhấn Run/Stop màu cam trên EIROBO-206. Đèn LED màu đỏ trên bộ điều khiển sẽ ngừng nhấp nháy.

Bạn có thể dùng mô-đun pin EIBAT-202 để cắm trực tiếp vào EIROBO-206 hoặc vào khe cắm của bo mạch chủ.

Khi kết nối pin như vậy với EIROBO-206 và ngắt kết nối khỏi máy tính, nó sẽ trở thành một robot chạy độc lập theo chương trình được lưu trong bộ nhớ flash của nó.



Kết nối mô-đun pin với khe cắm của bo mạch chủ và ngắt kết nối EIROBO-206 khỏi máy tính.

Dòng menu được thay đổi thành như sau:



Hình 2. 45

9. Nhấn vào nút Run/Stop màu cam trên bộ điều khiển và ba đèn LED sẽ sáng giống một đèn giao thông trong vòng lặp vô hạn.

10. Để dừng hệ thống đang chạy, hãy nhấn nút Run/Stop màu cam trên EIROBO-206. Đèn LED màu đỏ trên bộ điều khiển sẽ ngừng nhấp nháy.

11. Kết nối lại EIROBO-206 với máy tính và đợi cho đến khi dòng menu được thay đổi trở lại như sau.



Hình 2. 46

### 1.3.3. Bài tập thử thách – Đèn giao thông

Nhiệm vụ 1: Tăng thời gian trễ của đèn LED màu đỏ và giảm thời gian trễ của đèn LED màu xanh lá cây.

Lưu ý rằng với các chương trình con, bạn không phải tìm kiếm lệnh Delay có liên quan trong chương trình chính.

Tải xuống chương trình, chạy và kiểm tra nó.

Nhiệm vụ 2: Thay đổi chương trình **Den\_giao\_thong\_1** để đèn LED màu xanh lá cây nhấp nháy hai lần (bật một giây và tắt một giây) trước khi tắt.

Tải xuống chương trình, chạy và kiểm tra nó.



## THÍ NGHIỆM 1.4 – CÔNG TÁC

### Mục tiêu:

- Phản hồi của công tác

### Thiết bị cần thiết:

- Máy tính
- Phần mềm RobocklySense
- Robot lắp ráp EISENSE-MAKE

### Thảo luận:

Một hệ thống vòng kín là một hệ thống điều khiển, tương tác với các cảm biến và công tắc.

Hệ thống chiếu sáng tự động bao gồm cảm biến ánh sáng là một ví dụ cho hệ thống vòng kín. Hệ thống điều khiển sẽ sáng đèn khi trời tối và tắt đèn khi có ánh sáng. Hệ thống này tự động điều chỉnh theo thời gian mùa hè (khi trời tối muộn) và mùa đông (khi trời tối sớm).

Tất nhiên, chúng ta cần chú ý rằng việc bật đèn không ảnh hưởng đến cảm biến ánh sáng.

Chương trình của hệ thống vòng kín chứa các khối lệnh như:

**'Wait – until', 'If – then'.**

Các chương trình trong thí nghiệm này sử dụng khối lệnh **“Wait – until”**.

#### 1.4.1. Ô nhớ và biến

Trong lệnh tạo trễ (**Delay**), chúng ta viết một số xác định độ dài của thời gian trễ tính bằng giây.

Chúng ta gọi số đó là **hằng số**, nó là một phần của lệnh. Khi muốn thay đổi số này, chúng ta phải tìm kiếm một lệnh có liên quan.

Trong lập trình, chúng ta ít khi sử dụng hằng số mà thay vào đó là các biến số.

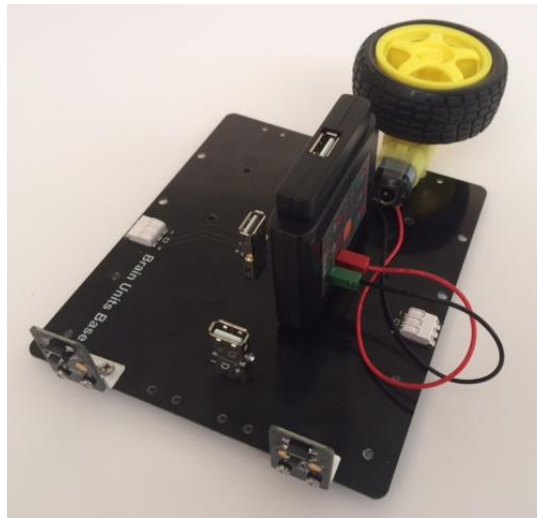
Một **biến** là một ô nhớ có tên, trong câu lệnh chúng ta chỉ ra tên của biến.

Chúng ta có thể đặt giá trị biến ở một vị trí nhất định của chương trình, điều này giúp chúng ta tiết kiệm việc tìm kiếm các khối lệnh.

Trong **RobocklySense** sử dụng các ô nhớ (từ Memory1 đến Memory19) làm biến.

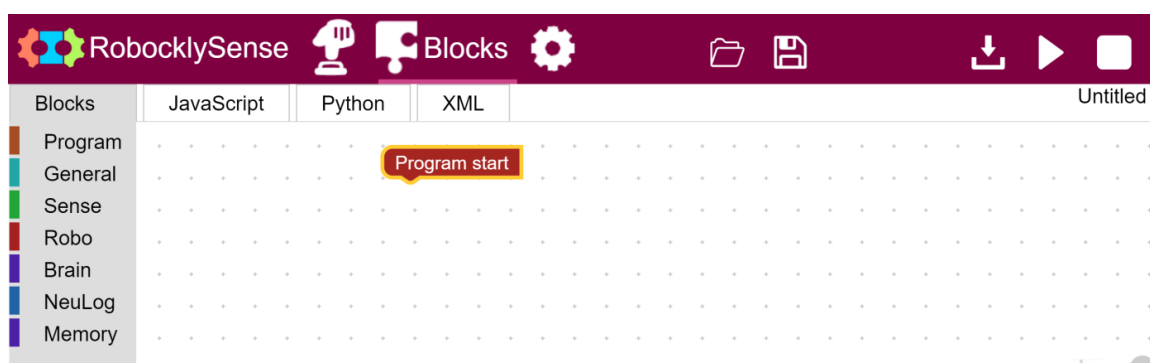
### Trình tự thực hiện:

1. Kết nối 2 mô-đun nút nhấn (**Tact**) và Động cơ DC với bo mạch như trong hình dưới đây:



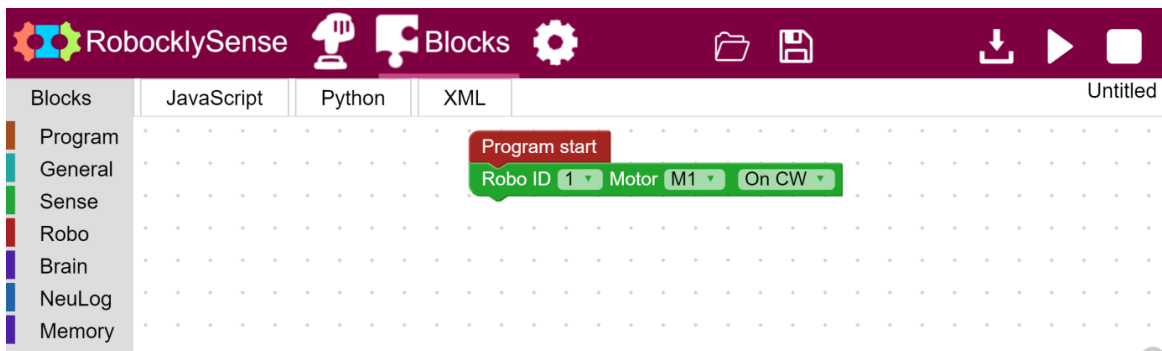
**Hình 2. 47**

2. Lắp EIROBO-206 lên trên bo mạch chủ.
3. Kết nối EIROBO-206 với máy tính bằng cáp USB.
4. Kết nối động cơ với cổng M1.
5. Kết nối một nút nhấn với cổng S1
6. Kết nối nút nhấn còn lại với cổng S2
7. Chạy phần mềm RobocklySense.
8. Kiểm tra động cơ như được mô tả trong thí nghiệm 1.1.
9. Kiểm tra các nút nhấn như được mô tả trong thí nghiệm 1.1
10. Nhấp chọn biểu tượng **Block** để chuyển sang giao diện lập trình.
11. Nhấp vào **Program**, chọn khối lệnh **Program start** và kéo nó sang phải.



**Hình 2. 48**

12. Nhấp vào **Robo** và kéo thêm khối lệnh **Robo Motor** ghép vào bên dưới khối lệnh **Program start**.




Hình 2. 49

13. Nhấp vào **General** và thêm khối lệnh **Delay**.
14. Thay đổi giá trị độ trễ thành 3 giây.
15. Nhân đôi khối lệnh **Robo Motor**. Đính kèm nó vào sau khối lệnh **Delay**.
16. Thay đổi chỉ thị của khối lệnh **Robo Motor** mới thành **Off** (Dừng).
17. Nhấp vào **Program** và chọn khối lệnh **Program end**. Kéo nó vào sau khối lệnh **Robo Motor** cuối cùng.



Hình 2. 50

18. Nhấp vào biểu tượng  và lưu chương trình với tên là **Dong\_co\_1**.

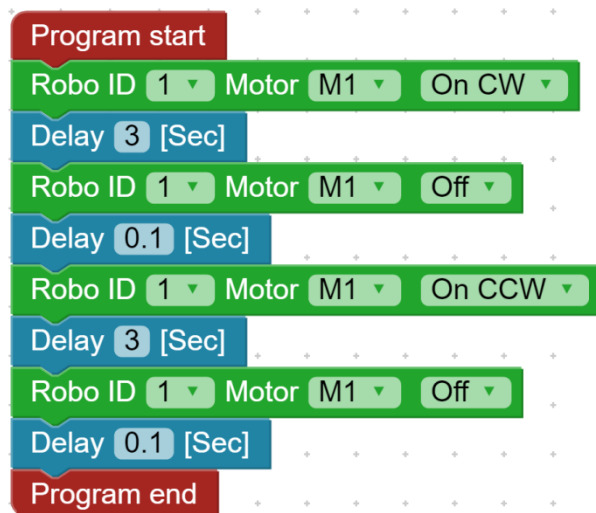
19. Nhấp vào biểu tượng  để tải xuống chương trình.

20. Nhấp vào biểu tượng  để chạy chương trình.

Động cơ sẽ quay trong 3 giây và sau đó dừng lại.

21. Trước khi thay đổi hướng quay của động cơ, tốt hơn là nên dừng nó lại trong một thời gian ngắn và sau đó bật nó quay theo hướng khác.

Thay đổi chương trình thành như sau:



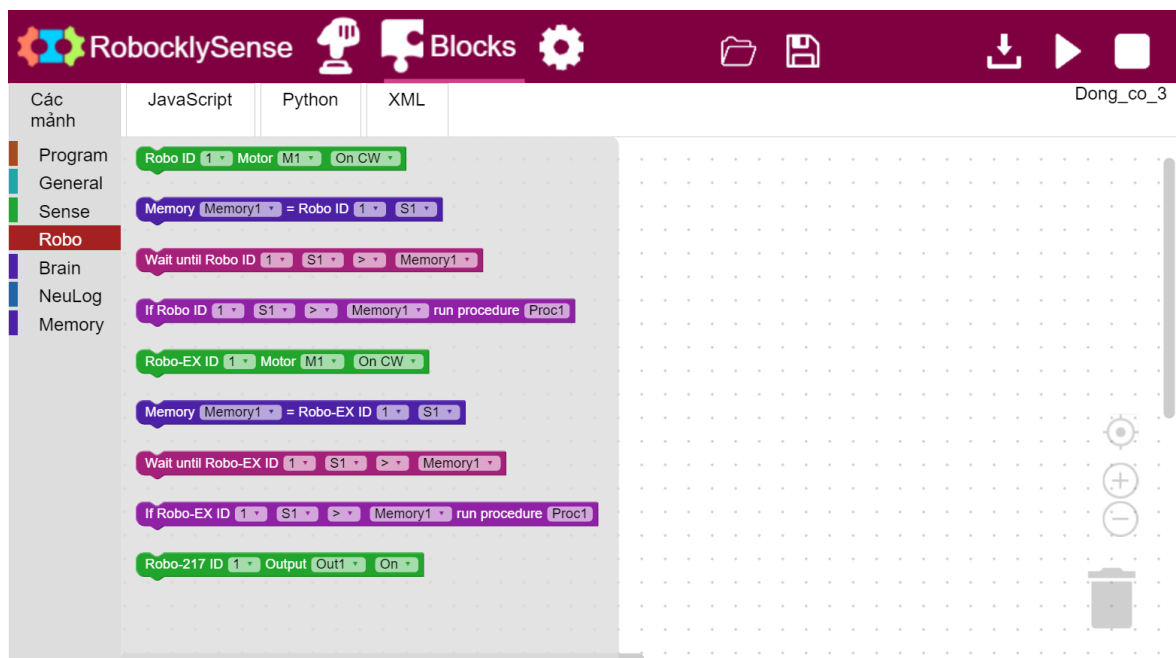
Hình 2. 51

22. Tải xuống, chạy và kiểm tra hoạt động của chương trình.
23. Thay thế khối lệnh **Program end** bằng khối lệnh **Program restart**.
24. Tải xuống, chạy và kiểm tra hoạt động của chương trình.
25. Dừng chương trình đang chạy.

#### 1.4.2. Lệnh Wait until

Chúng ta sẽ dùng lệnh **Wait until** bắt đầu chương trình.

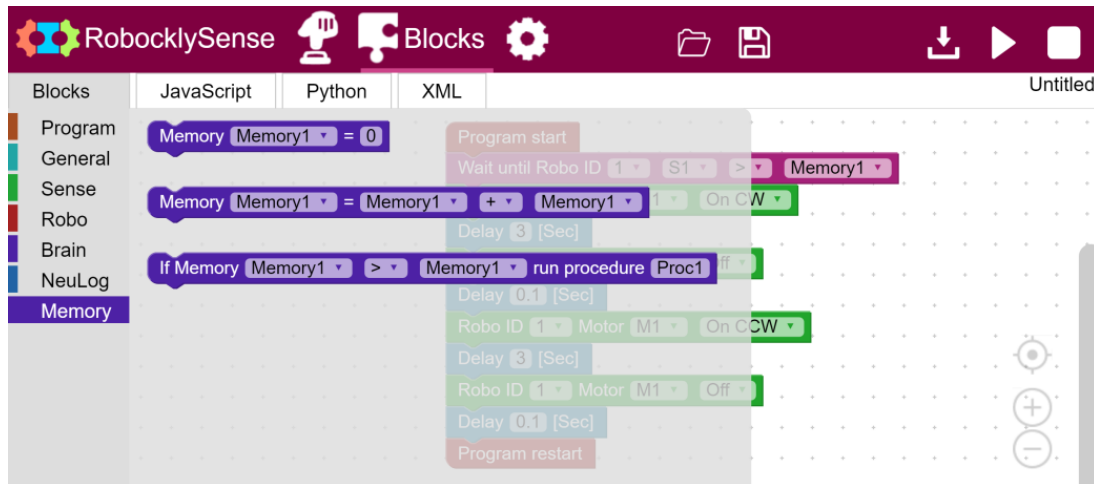
1. Nhấp vào **Robo** và một danh sách các câu lệnh của Robo sẽ xuất hiện.



Hình 2. 52

2. Nhấp vào khối lệnh **“Wait until Robo ID [1] [S1] [>] [Memory1]”** và kéo nó trở thành lệnh đầu tiên trong chương trình.

3. Nhấp vào **Memory** và danh sách các lệnh đầu vào xuất hiện:

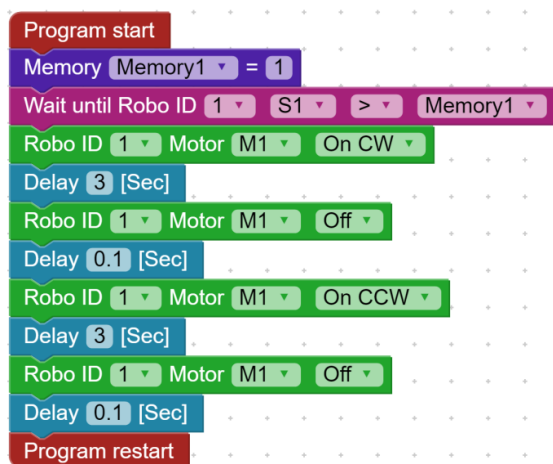


Hình 2. 53

4. Nhấp vào khối lệnh “**Memory [Memory1] = [0]**” và kéo nó để trở thành câu lệnh đầu tiên trong chương trình.

5. Thay đổi giá trị “0” trong câu lệnh thành “1”.

6. Kiểm tra xem trương trình có giống như sau không:



Hình 2. 54

7. Lưu chương trình dưới tên **Nut\_nhan\_1**.

8. Tải xuống và chạy chương trình.

Sẽ không có điều gì xảy ra với động cơ.

9. Bấm vào nút nhấn (Tact).

Động cơ sẽ quay và thay đổi hướng quay sau khi bạn bấm nút nhấn.

**Lưu ý:**

Như được mô tả trong thí nghiệm 1, khi bấm nút nhấn, giá trị đọc trong S1 là 1000.

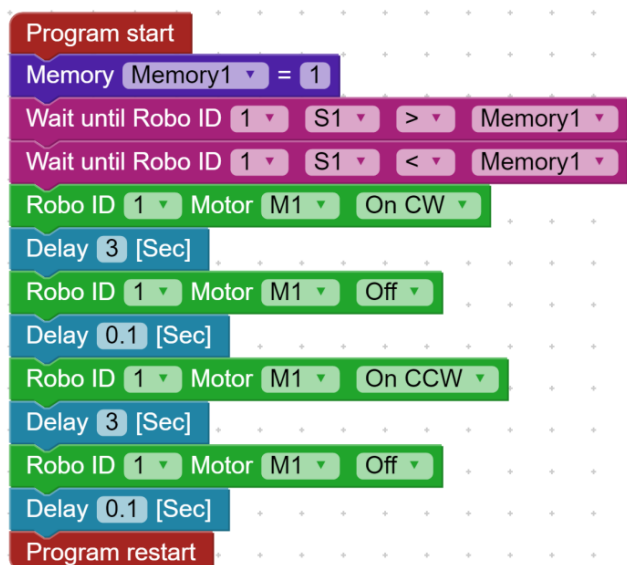
10. Nhả nút nhấn.

Động cơ sẽ dừng lại sau khi kết thúc 1 chu kỳ.

Chương trình chờ tới khi nút nhấn được bấm lần nữa.

11. Dừng chương trình đang chạy.

12. Thay đổi chương trình để mỗi lần bấm và nhả nút nhấn sẽ tạo ra một chu kỳ của động cơ.



Hình 2. 55

13. Lưu chương trình.

14. Tải xuống và chạy chương trình.

Động cơ sẽ thực hiện chu kỳ của nó một lần sau khi bấm và nhả nút nhấn.

Mỗi lần bấm và nhả sẽ tạo ra một chu kỳ quay của động cơ.

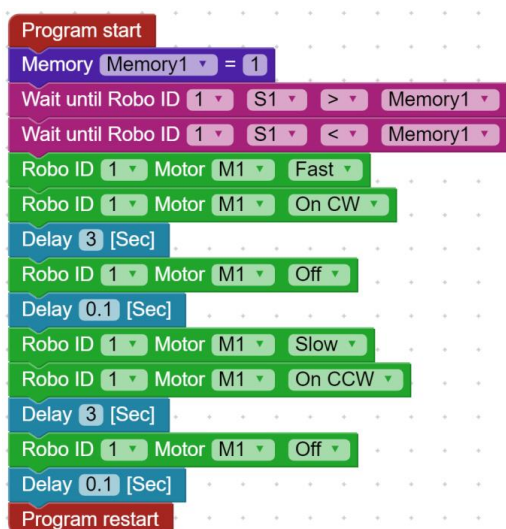
Kiểm tra điều này.

### 1.4.3. Thay đổi tốc độ

Lệnh **Robo Motor** cũng cho phép chúng ta thay đổi tốc độ động cơ.

Sau khi thay đổi, tốc độ sẽ được giữ thiết lập cho đến lần thay đổi tiếp theo.

1. Thay đổi chương trình để mỗi lần bấm và nhả nút nhấn sẽ tạo ra một chu kỳ của động cơ. Tốc độ động cơ nhanh khi quay thuận (CW) và chậm khi quay ngược (CCW).



Hình 2. 56

2. Lưu trương trình

### 3. Tải xuống và chạy chương trình.

Động cơ sẽ thực hiện chu kỳ quay của nó 1 lần chỉ sau khi bấm và nhả nút nhấn với các tốc độ quay thuận và quay ngược khác nhau.

Mỗi lần bấm và nhả sẽ tạo ra một chu kỳ quay của động cơ.

Kiểm tra điều này.

#### **1.4.4. Bài tập thử thách - Động cơ được điều khiển bằng nút nhấn**

Nhiệm vụ 1: Tạo một chương trình bật động cơ sau khi bấm và nhả nút nhấn và tắt nó sau khi bấm và nhả một nút nhấn khác.

Chương trình chạy trong một vòng lặp vô hạn.

Nhiệm vụ 2: Tạo một chương trình bật động cơ sau bấm và nhả ba lần nút nhấn và tắt nó sau một lần bấm và nhả nút nhấn khác.

Chương trình chạy trong một vòng lặp vô tận.

## THÍ NGHIỆM 1.5 – ĐIỀU KIỆN VÀ QUYẾT ĐỊNH

### Mục tiêu:

- Lệnh điều kiện IF (Nếu).
- Hàm OR (Hoặc).
- Hàm AND (Và).

### Thiết bị cần thiết:

- Máy tính
- Phần mềm RobocklySense
- Robot lắp ráp EISENSE-MAKE

### Thảo luận:

#### 1.5.1. Lệnh If (Nếu)

Lệnh điều kiện là lệnh thực hiện có điều kiện.

Lệnh được thực hiện khi điều kiện được viết trong lệnh tồn tại.

Lệnh có điều kiện trong phần mềm RobocklySense được gọi một chương trình con.

Lệnh **If** là lệnh điều kiện chính. Nó bao gồm một điều kiện và chương trình con nào đó để hoạt động khi điều kiện tồn tại.

Điều kiện được kiểm tra khi lệnh điều kiện được thực hiện. Nếu điều kiện không tồn tại, lệnh tiếp theo phía sau sẽ được thực hiện.

Chương trình sau đây là một ví dụ về cách sử dụng lệnh **If**.

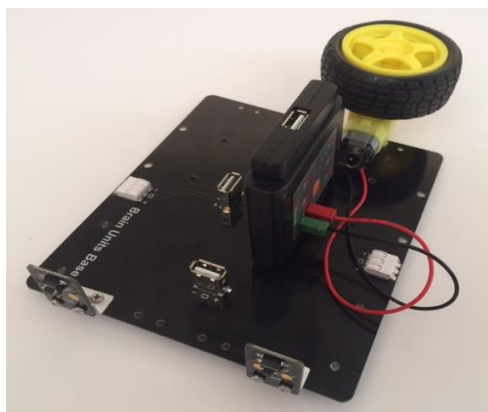


Hình 2. 57

Chúng ta có thể tạo ra các điều kiện phức tạp, được gọi là AND (Và) và OR (Hoặc), sẽ được giải thích chi tiết trong quá trình thí nghiệm.

### Trình tự thực hiện:

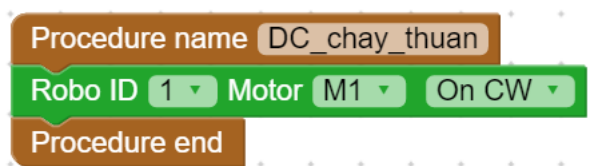
1. Kết nối 2 mô-đun nút nhấn (**Tact**) và **Động cơ** với bo mạch chính như trong hình dưới đây:



Hình 2. 58

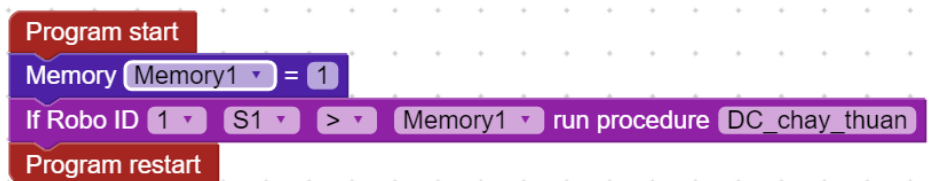


2. Lắp EIROBO-206 lên trên bo mạch chủ.
3. Kết nối EIROBO-206 với máy tính bằng cáp USB.
4. Kết nối động cơ với cổng M1.
5. Kết nối một nút nhấn với cổng S1
6. Kết nối nút nhấn còn lại với cổng S2
7. Chạy phần mềm RobocklySense.
8. Kiểm tra động cơ như được mô tả trong thí nghiệm 1.1.
9. Kiểm tra các nút nhấn như được mô tả trong thí nghiệm 1.1
10. Tạo chương trình con **DC\_chay\_thuan** với một khối lệnh duy nhất:




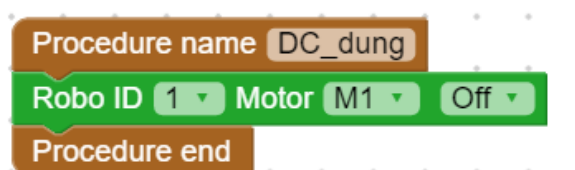
**Hình 2. 59**

11. Tạo chương trình chính với các khối lệnh sau:



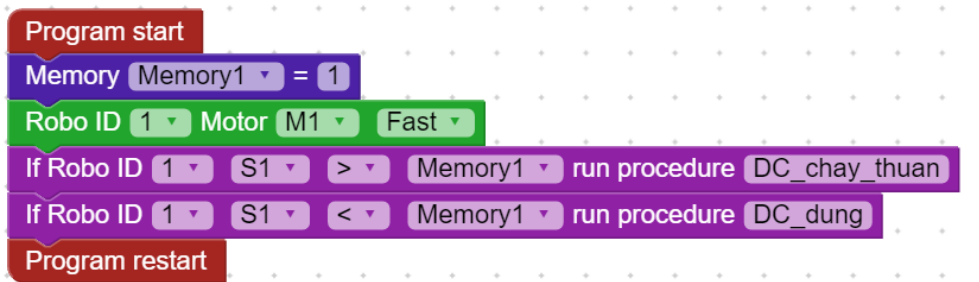
**Hình 2. 60**

12. Lưu chương trình này dưới tên **Dong\_co\_DC\_1**.
13. Tải xuống và chạy chương trình.  
Sẽ không có điều gì xảy ra.
14. Bấm nút nhấn.  
Động cơ sẽ bắt đầu quay và không dừng lại.
15. Nhấn vào biểu tượng  để dừng chương trình và động cơ.  
Để dừng động cơ, chúng ta cần đưa thêm vào khối lệnh Dừng (**Off**).
16. Tạo chương trình con **DC\_dung** với một khối lệnh duy nhất:



**Hình 2. 61**

17. Thay đổi chương trình thành như sau:



Hình 2. 62

18. Tải xuống và chạy chương trình.

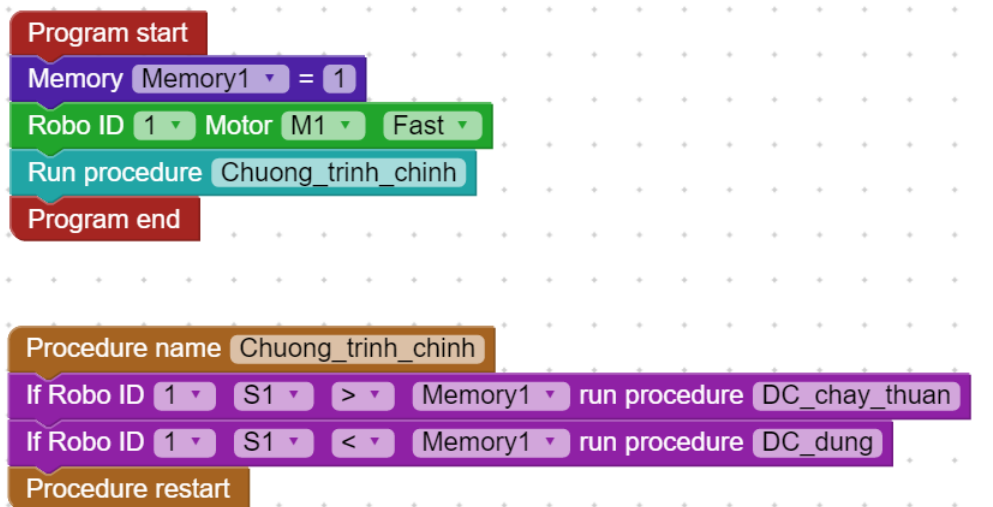
Lần này, bấm nút nhấn sẽ bật động cơ và nhả nút nhấn sẽ dừng động cơ.

### 1.5.2. Các định nghĩa

Định nghĩa của biến nhớ **Memory1** được thực hiện trong mọi vòng lặp của chương trình mặc dù chúng ta chỉ cần định nghĩa nó một lần. Nó sẽ làm chậm đi thời gian chu kỳ của chương trình.

Để làm điều đó chỉ khi chúng ta tạo ra một chương trình con **Chương\_trình\_chinh** mà không có các định nghĩa chạy trong vòng lặp vô hạn. Các định nghĩa được thực hiện trong chương trình và chương trình con **Chương\_trình\_chinh** sẽ được gọi sau đó.

1. Thay đổi chương trình thành chương trình con sau và đặt tên là **Chương\_trình\_chinh**:



Hình 2. 63

2. Tải xuống và chạy chương trình.

Bấm nút nhấn sẽ bật động cơ và nhả nút nhấn sẽ dừng động cơ.

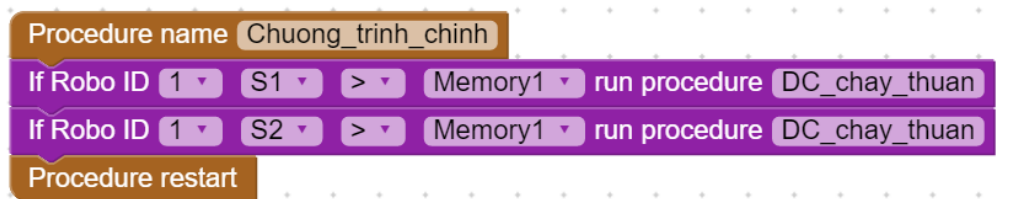
Biến nhớ **Memory1** chỉ được định nghĩa một lần trong chương trình.

### 1.5.3. Hàm điều kiện OR (Hoặc)



Chúng ta muốn vận hành động cơ bằng hai nút nhấn theo hàm điều kiện OR (Hoặc).

Trong điều kiện OR, động cơ được bật khi bấm nút nhấn thứ nhất **hoặc** nút nhấn thứ hai được bấm **hoặc** bấm cả hai nút nhấn.

1. Thay đổi chương trình con **Chương\_trình\_chính** thành như sau:



**Hình 2. 64**

2. Tải xuống và chạy chương trình.
3. Bấm một trong hai nút nhấn và kiểm tra xem động cơ có bật không.
4. Nhấn vào biểu tượng  để dừng chương trình đang chạy. Động cơ sẽ dừng lại.
5. Chạy lại chương trình.
6. Bấm nút nhấn còn lại và kiểm tra xem động cơ có bật không.
7. Nhấn vào biểu tượng  để dừng chương trình đang chạy.

#### 1.5.4. Hàm điều kiện AND

Để dừng động cơ, chúng ta cần điều kiện AND (Và).

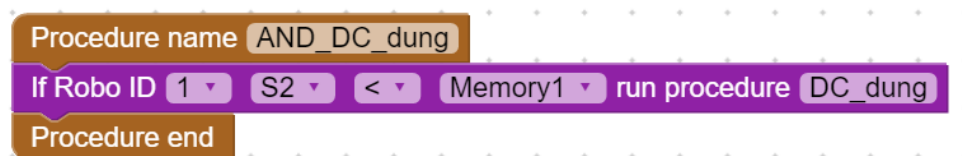
Động cơ nên dừng khi nhấn hai nút nhấn.

Trong một số phần mềm lập trình, điều kiện **AND** có thể được thêm vào lệnh **IF**.

RobocklySense không có tùy chọn này.

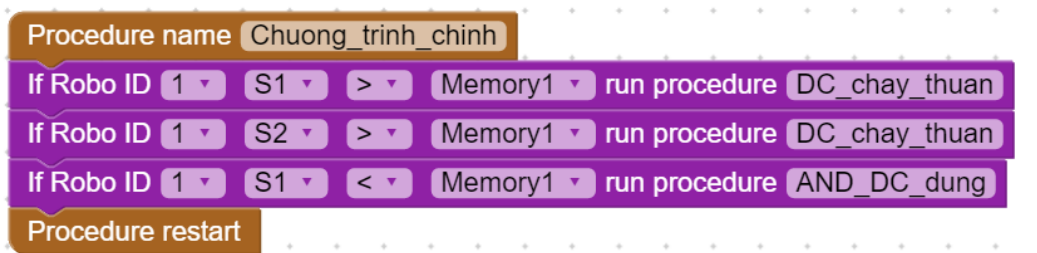
Điều kiện AND có thể đạt được bằng cách tạo một chương trình con khác.

1. Tạo chương trình con **AND\_DC\_dung** với một khối lệnh duy nhất:



**Hình 2. 65**

2. Thay đổi chương trình chính thành như sau:



**Hình 2. 66**

3. Phân tích chương trình và các chương trình con.
4. Tải xuống và chạy chương trình.
5. Nhấn các nút nhấn và kiểm tra xem động cơ chỉ dừng khi nhả hai nút nhấn.

#### **1.5.5. Bài tập thử thách - Động cơ được điều khiển bởi hai nút nhấn**

Nhiệm vụ 1: Xây dựng chương trình Bật động cơ khi bấm và nhả nút nhấn và Dừng khi bấm và nhả nút nhấn còn lại.

## THÍ NGHIỆM 1.6 – CÔNG QUANG

### Mục tiêu:

- Khối “Brain” hoạt động như thế nào?
- Động cơ Servo hoạt động như thế nào?
- Cảm biến ánh sáng.
- Bộ ghép quang.

### Thiết bị cần thiết:

- Máy tính
- Phần mềm RobocklySense
- Robot lắp ráp EISENSE-MAKE

### Thảo luận:

#### 1.6.1. Thiết bị Brain

Khối “Brain” là một linh kiện hoặc thiết bị có bộ điều khiển riêng đã được lập trình sẵn để kiểm soát hoạt động. Ví dụ, một thiết bị điều khiển tốc độ và hướng của động cơ, cường độ đèn, góc động cơ servo, đọc cảm biến, v.v.

Ba dây kết nối khối “Brain” với bộ điều khiển hệ thống. Thiết bị Brain có hai đầu ra, mỗi đầu ra có ba dây.

Các khối “Brain” được kết nối trong một chuỗi với bộ phận điều khiển chính, bộ điều khiển này giao tiếp với chúng thông qua các tệp tin.

Mọi khối “Brain” đều có một mã số ID. Mọi thông điệp từ đơn vị điều khiển đều bắt đầu bằng số ID tương ứng. Chỉ khối “Brain” có số ID này mới nhận được thông điệp và thực thi nó.

Việc xây dựng hệ thống như này là cách các hệ thống hiện đại được xây dựng và có những ưu điểm vượt trội.

Nó tạo ra một hệ thống với ít dây nối hơn. Các dây dẫn đi từ mô-đun này sang mô-đun khác chứ không phải từ tất cả các mô-đun nối đến thiết bị điều khiển.

Loại hệ thống này có thể dễ dàng thay đổi và mở rộng và không phụ thuộc vào số lượng đầu vào và đầu ra của đơn vị điều khiển.

#### 1.6.2. Động cơ servo

Động cơ servo là một động cơ có phản hồi. Phản hồi các trạng thái có thể là điện áp theo tốc độ động cơ, xung điện theo vòng quay và hướng trục động cơ, v.v.

Động cơ servo của bộ EISENSE-MAKE có bánh răng truyền động và chiết áp. Chiết áp là các biến trở tạo ra điện áp thay đổi theo góc trục động cơ servo.

Bộ điều khiển động cơ servo nhận được góc cần thiết của trục. Nó quay động cơ theo chiều kim đồng hồ (CW) hoặc ngược chiều kim đồng hồ (CCW) cho đến khi điện áp của chiết áp phù hợp với góc này.

Nó kiểm tra góc trục mọi lúc. Nếu có thay đổi cơ học, bộ điều khiển sẽ BẬT động cơ để đưa trục quay về đúng vị trí.

Bộ thiết bị này chứa hai động cơ servo - một có ID #1 và một có ID #2.

### 1.6.3. Cổng quang

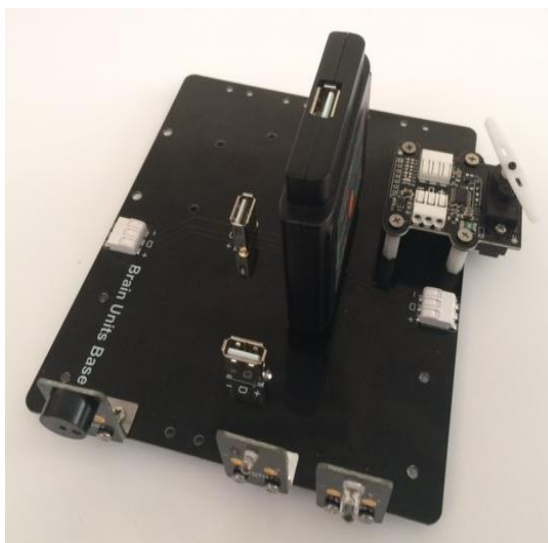
Cổng quang bao gồm nguồn sáng (thường là đèn LED) và cảm biến ánh sáng. Cảm biến ánh sáng tạo ra điện áp theo cường độ ánh sáng trên đó.

Đèn và cảm biến ánh sáng của cổng quang có thể đặt đối nhau. Điện áp cảm biến ánh sáng sẽ cao khi đèn không bị chặn. Khi ánh sáng bị chặn, điện áp cảm biến ánh sáng sẽ giảm đáng kể.

Một lựa chọn khác của cổng quang là đặt đèn và cảm biến ánh sáng cạnh nhau.

#### Trình tự thực hiện:

1. Kết nối Đèn (Lamp), cảm biến ánh sáng (Light) và Động cơ servo với bo mạch chính như trong hình dưới đây:



Hình 2. 67

Sử dụng động cơ servo với ID #1.

#### Những lưu ý quan trọng:

Bo mạch chính có hai đầu nối 3 chân ở hai bên. Các đầu nối này sử dụng các dây nối có hai đầu tuốt sẵn lớp vỏ cách điện trong bộ thiết bị.

Mỗi chân của đầu nối có hai tấm kim loại bên trong. Nhấn vào phần nhựa phía trên bằng tuốc nơ vít dẹt sẽ mở các tấm kim loại này.

Nhấn miếng nhựa phía trên của chân đầu nối và giữ nó trong khi luồn dây điện vào đó.

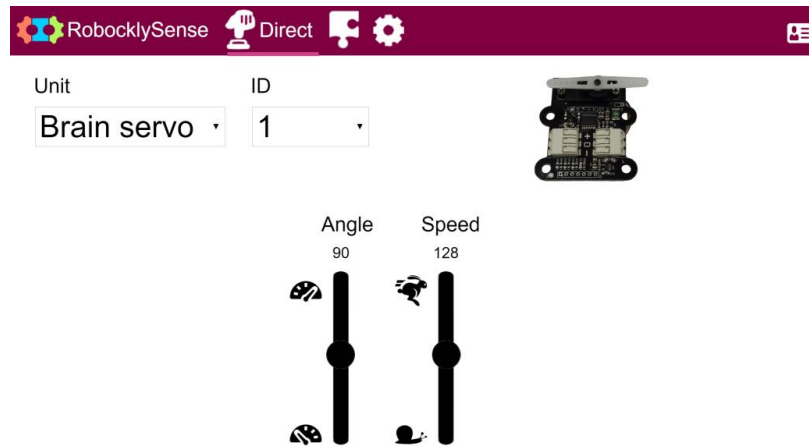
Hãy chú ý đến màu sắc của dây nối.

2. Lắp mô đun EIROBO-206 lên trên bo mạch chủ.
3. Kết nối EIROBO-206 với máy tính bằng cáp USB.
4. Kết nối động cơ Servo với cổng 3 chân trên bo mạch chính.
5. Kết nối Còi (Buzzer) với cổng M3 trên bộ điều khiển.
6. Kết nối Đèn (Lamp) với cổng M1 trên bộ điều khiển.
7. Kết nối cảm biến ánh sáng (Light) với cổng S1 trên bộ điều khiển.

8. Chạy phần mềm RobocklySense.

9. Kiểm tra đèn (Lamp) và cảm biến ánh sáng (Light) như được mô tả trong thí nghiệm 1.1.

10. Chọn **Brain Servo** trong ô **Unit** của màn hình Trực tiếp (**Direct**):



**Hình 2. 68**

Màn hình này cho phép chúng ta cài đặt góc động cơ và tốc độ di chuyển của nó.

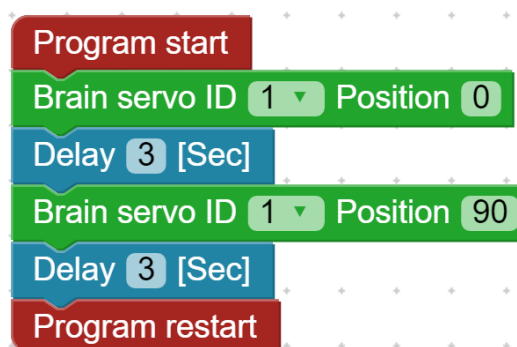
11. Tìm số ID của động cơ Servo. Số ID mặc định là 1.

12. Thiết lập số ID của thiết bị đã cài đặt.

13. Kiểm tra động cơ Servo bằng cách thay đổi thanh trượt của góc (Angle) và tốc độ (Speed).

14. Nhấp chọn biểu tượng **Block**  để chuyển sang giao diện lập trình.

15. Xây dựng chương trình như sau, di chuyển động cơ từ góc  $0^\circ$  sang góc  $90^\circ$  trong một vòng lặp vô tận.



**Hình 2. 69**

16. Lưu chương trình dưới tên **Servo\_1**.


17. Tải xuống và chạy chương trình.

Trục động cơ servo di chuyển giữa hai góc.

18. Dừng chương trình lại.

#### 1.6.4. Chương trình công quang

1. Chuyển đến chế độ Trực tiếp (**Direct**).

2. Nhấn vào biểu tượng  trên hàng M1 để bật đèn (Lamp).

3. Đọc giá trị của cảm biến ánh sáng trong ô S1 và ghi lại nó.

Chúng ta sẽ gọi giá trị này là **giá trị Mở**.

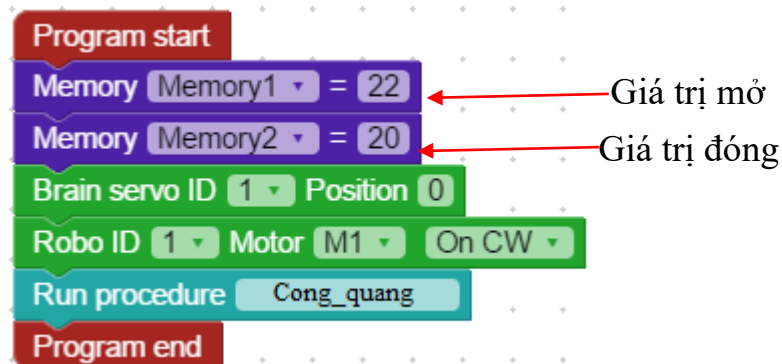
4. Đưa tay của bạn đến gần công quang và cảm biến ánh sáng.

5. Giữ tay bạn gần công quang và cảm biến ánh sáng, đọc giá trị của cảm biến ánh sáng trong ô S1 và ghi lại nó.

Chúng ta sẽ gọi giá trị này là **giá trị Đóng**.

6. Tạo một chương trình chính như sau:

- Xác định **Memory1** là **giá trị Mở**.
- Xác định **Memory2** là **giá trị Đóng**.
- Di chuyển trục động cơ servo đến  $0^\circ$ .
- Bật đèn (Lamp).
- Chạy chương trình con **Cong\_quang**

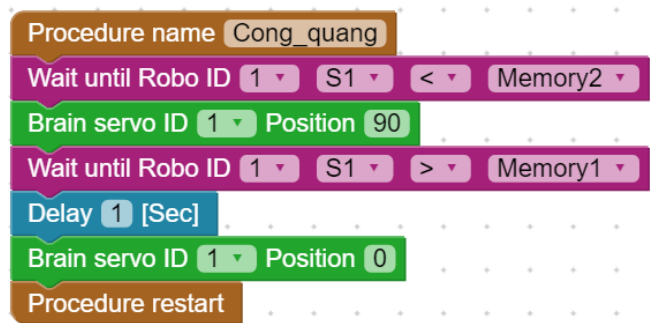


Hình 2. 70

7. Tạo chương trình con **Cong\_quang** như sau:

- Chờ đến khi giá trị cảm biến ánh sáng (Light) nhỏ hơn **giá trị Đóng**.
- Di chuyển trục động cơ servo đến góc  $90^\circ$ .
- Chờ đến khi giá trị cảm biến ánh sáng (Light) lớn hơn **giá trị Mở**.
- Di chuyển trục động cơ servo đến  $0^\circ$ .
- Chèn độ trễ (Delay).
- Lặp lại chương trình con.





**Hình 2. 71**

8. Lưu chương trình dưới tên **Cong\_quang\_1**.

9. Tải xuống và chạy chương trình.

Trục của động cơ servo sẽ di chuyển về  $0^\circ$  và đèn (Lamp) sẽ bật.

10. Đưa tay đến gần cổng quang và cảm biến ánh sáng.

Trục động cơ servo sẽ di chuyển đến  $90^\circ$ .

11. Bỏ tay ra khỏi cổng quang và cảm biến ánh sáng.

Trục động cơ sẽ di chuyển đến  $0^\circ$ .

12. Lặp lại các bước 10 và 11 và thay đổi các giá trị thiết lập của biến **Memory1** và **Memory2** cho đến khi chương trình hoạt động tốt.

13. Dừng chương trình lại.

### **1.6.5. Bài tập thử thách - Cổng quang với còi báo động**

Nhiệm vụ 1: Cải thiện chương trình để trước khi đóng cổng quang, nó sẽ bật còi (buzzer) trong 1 giây.

## THÍ NGHIỆM 1.7 – XE GẮN ĐỘNG CƠ

### Mục tiêu:

- Xây dựng một chiếc xe dừng lại trước một bức tường.
- Xây dựng một xe di chuyển tiến lùi.

### Thiết bị cần thiết:

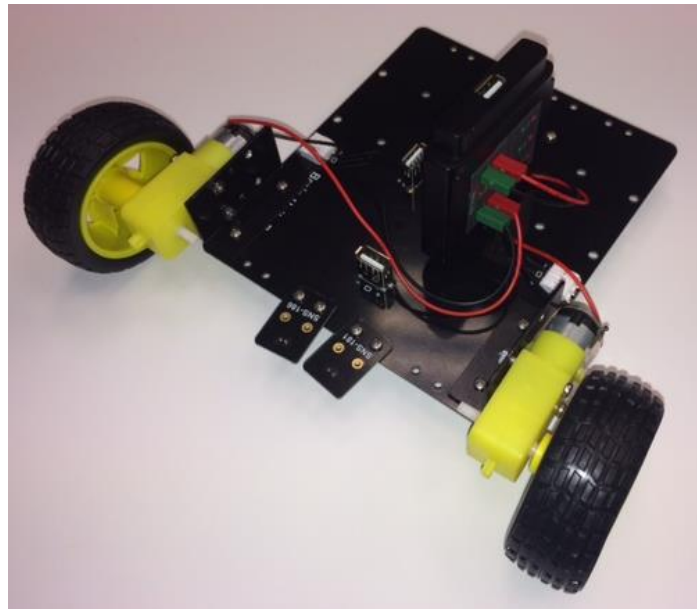
- Máy tính
- Phần mềm RobocklySense
- Robot lắp ráp EISENSE-MAKE

### Thảo luận:

Trong thí nghiệm này, chúng ta sẽ xây dựng một chiếc xe, di chuyển tiến lùi với khoảng thời gian nhất định.

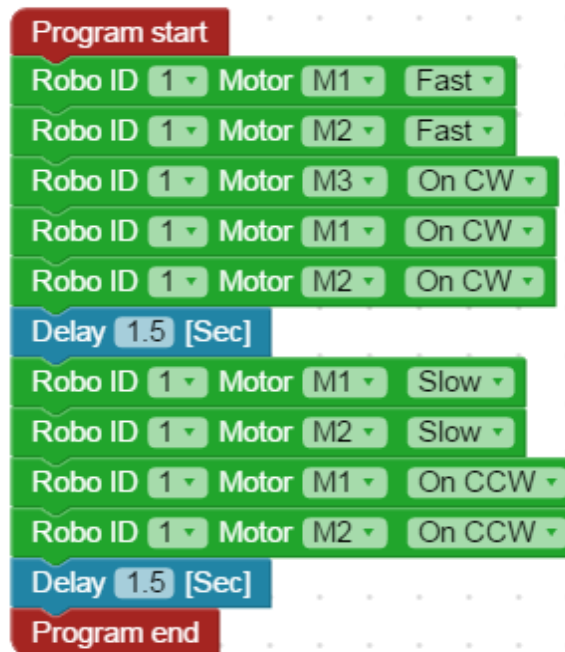
### Trình tự thực hiện:

1. Kết nối các bộ phận để được một chiếc xe như trong hình dưới đây:



**Hình 2. 72**

2. Lắp EIROBO-206 lên trên bo mạch chủ.
3. Kết nối EIROBO-206 với máy tính bằng cáp USB.
4. Chạy phần mềm RobocklySense.
5. Kết nối các bộ phận của xe với EIROBO-206 và với các thiết bị đầu cuối.
6. Kiểm tra từng bộ phận như được mô tả trong thí nghiệm 1.
7. Tạo chương trình và chương trình di chuyển xe đi tiến với tốc độ cao và lùi với tốc độ thấp một khoảng thời gian nhất định như sau:



**Hình 2. 73**

9. Tải xuống, chạy và kiểm tra chương trình.

**1.7.1. Bài tập thử thách – Xe tự hành**

Nhiệm vụ 1: Nâng cấp chương trình để xe dừng chờ 2 giây trước khi thay đổi hướng.

Nhiệm vụ 2: Thay đổi chương trình để xe chạy tiến về phía trước chậm và chạy lùi nhanh.

## THÍ NGHIỆM 1.8 – MÁY TỰ ĐỘNG

### Mục tiêu:

- Xây dựng một cỗ máy được điều khiển bởi một cảm biến ánh sáng.
- Kiểm soát và chỉ huy một hệ thống phụ theo thời gian.
- Điều khiển hệ thống tự động

### Thiết bị cần thiết:

- Máy tính
- Phần mềm RobocklySense
- Robot lắp ráp EISENSE-MAKE

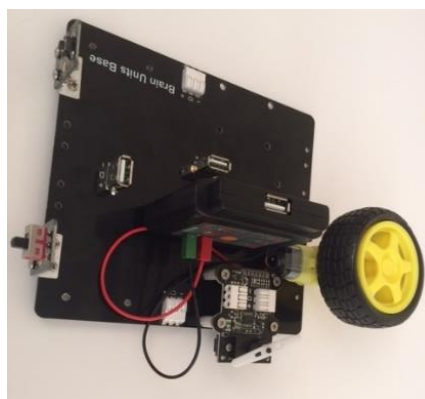
### Thảo luận:

Trong thí nghiệm này, chúng ta sẽ xây dựng một hệ thống tự động. Hệ thống này mô phỏng máy giặt.

- Động cơ DC mô phỏng động cơ máy giặt.
- Động cơ servo mô phỏng vòi nước với ba vị trí:
  - \* Cấp nước sạch ( $0^\circ$ )
  - \* Đóng vòi ( $90^\circ$ )
  - \* Xả nước bẩn ( $180^\circ$ )
- Công tắc gạt (Switch) mô phỏng nút ON-OFF (Bật/Tắt).
- Nút nhấn (Tact) mô phỏng chốt cửa.
- Đèn (Lamp) mô phỏng gia nhiệt nước.

### Trình tự thực hiện:

1. Kết nối các bộ phận với bo mạch chủ như sau.

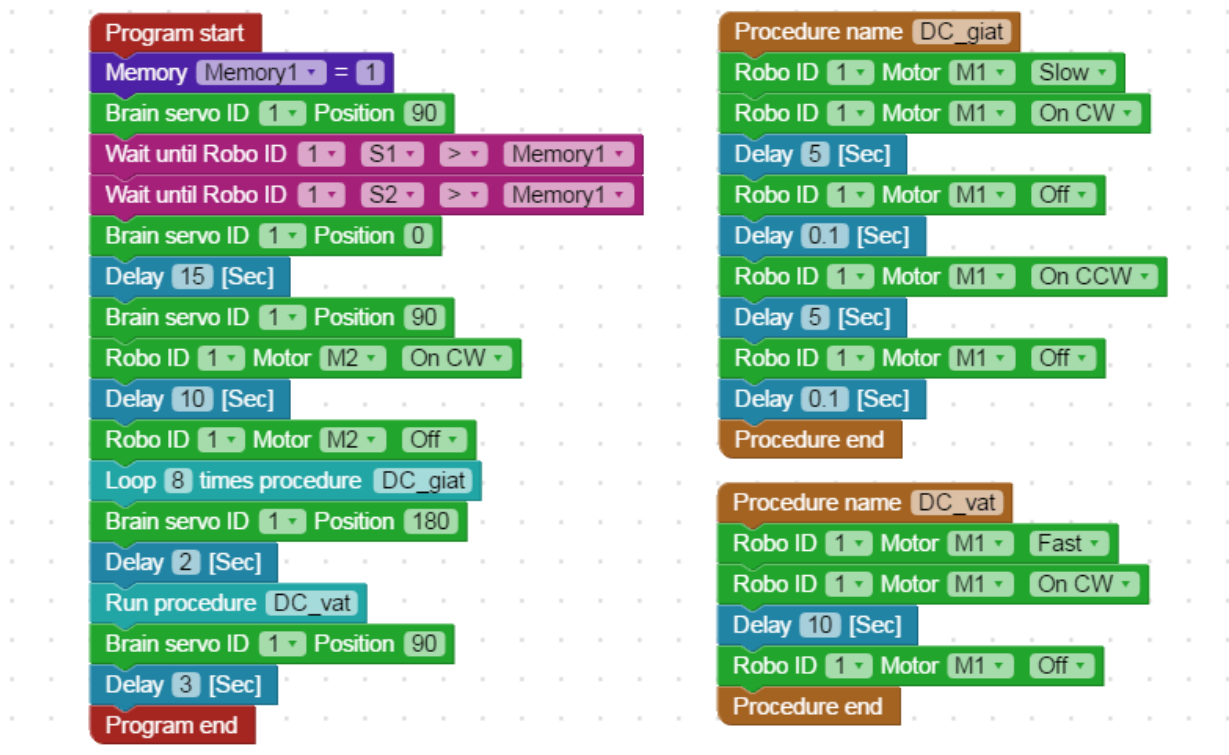


**Hình 2. 74**

2. Lắp EIROBO-206 lên trên bo mạch chủ.
3. Kết nối EIROBO-206 với máy tính bằng cáp USB.
4. Chạy phần mềm RobocklySense.
5. Kết nối các bộ phận với EIROBO-206 và với các thiết bị đầu cuối.
6. Kiểm tra từng bộ phận như được mô tả trong các thí nghiệm 1.1 và 1.6

7. Xây dựng chương trình và các bước thực hiện như sau:

- Đóng vòi nước- Động cơ servo quay 90<sup>0</sup>
- Chờ tới khi công tắc BẬT – công tắc gạt ON-OFF
- Kiểm tra xem cửa khoang giặt đã đóng chưa – “nút nhấn tact” mô phỏng cửa khoang máy giặt
- Mở vòi nước sạch trong 15 giây – Động cơ servo quay về góc 0<sup>0</sup>.
- Đóng vòi nước – Động cơ servo quay góc 90<sup>0</sup>.
- Bật gia nhiệt nước trong 10 giây – Bật đèn Lamp mô phỏng gia nhiệt nước.
- Tắt gia nhiệt nước – Tắt đèn Lamp mô phỏng gia nhiệt nước.
- Bật động cơ quay thuận (CW) và quay ngược (CCW) 8 lần ở tốc độ thấp trong 5 giây cho mỗi hướng quay – mô phỏng quá trình giặt.
- Mở vòi nước để xả nước bẩn ra ngoài. – Động cơ servo quay góc 180<sup>0</sup>.
- Bật động cơ quay thuận (CW) ở tốc độ cao trong 10 giây – mô phỏng quá trình vắt.
- Dừng động cơ lại.
- Đóng vòi nước – Động cơ servo quay về góc 90<sup>0</sup>.



Hình 2. 75

8. Tải xuống, chạy và kiểm tra chương trình.

### 1.8.1. Bài tập thử thách - Máy giặt

Nhiệm vụ 1: Nâng cấp chương trình để nó dừng động cơ khi công tắc chính TẮT.

Liên hệ hỗ trợ kỹ thuật:

**CTCP ĐIỆN TỬ CHUYÊN DỤNG HANEL**

Địa chỉ: Tầng 11 toà nhà Diamond Flower, số 48 Lê Văn Lương, Thanh Xuân, Hà Nội

Hotline: 0942195862