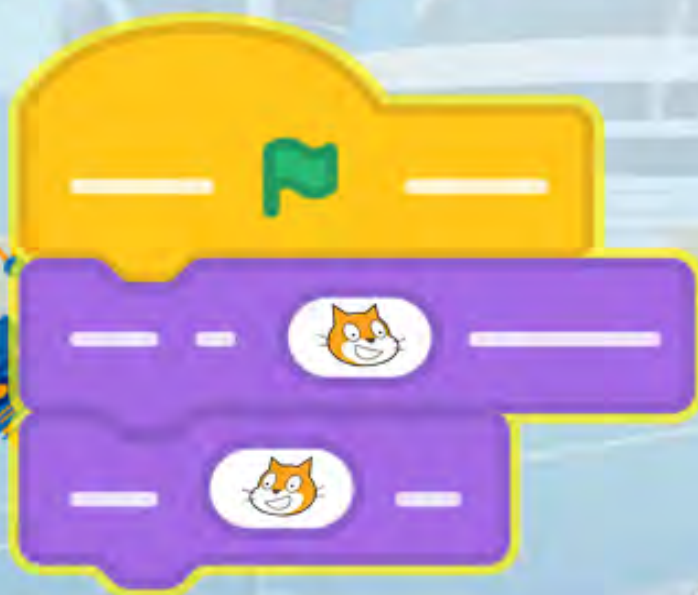




THE DARIU FOUNDATION

# Trí tuệ Nhân tạo và Robotics

## SCRATCH 3.0



Nguyễn Văn Hạnh - Lê Trọng Nhân  
Lê Phương Nam - Băng Ngọc Bảo Tâm



# LỜI MỞ ĐẦU

Những ngày đầu năm 2018, khi hợp tác với quỹ Dariu(The Dariu Foundation) trong dự án "Phổ Cập Kỹ Năng Số - Digital Literacy Initiative", lập trình kéo thả bằng ngôn ngữ Scratch đã bắt đầu trở thành một ngôn ngữ phổ cập cho các em học sinh. Không lâu sau đó, chúng ta có thêm ScratchJR dành cho lứa tuổi cấp 1 và Scratch 3.0, một phiên bản nâng cấp với đồ họa đẹp hơn dành cho học sinh cấp 2.

Với sự phát triển rất nhanh của khoa học và công nghệ, đặc biệt là lĩnh vực Trí tuệ nhân tạo và Robotics, ngôn ngữ lập trình cũng phát triển theo để cung cấp thêm những câu lệnh mới, làm phong phú thêm trải nghiệm của người học. Không nằm ngoài mục đích đó, Scratch 3.0 phiên bản trực tuyến đã được ra đời. Một điều vô cùng thú vị, là mã nguồn cho phần mềm Scratch 3.0 Online được phổ biến để rất nhiều tổ chức hay nhà phát triển định nghĩa cho phần mềm của mình những tính năng mới.

Trong giáo trình này, chúng tôi chọn phần mềm từ MIT (Massachusetts Institute of Technology), học viện vô cùng nổi tiếng và là nguồn cội của ngôn ngữ lập trình Scratch. Chúng tôi rất hân hạnh gửi đến bạn đọc những công nghệ mới nhất về Trí tuệ nhân tạo, Robotics hay công nghệ nhận dạng giọng nói, vốn rất phổ biến trong các thiết bị thông minh ngày nay. Dựa vào những kiến thức nền tảng này, chúng tôi vô cùng tham vọng rằng bạn đọc có thể tạo ra những dự án phức tạp, như Trợ lý ảo hay Robot tự hành.

Lời cuối cùng, tôi xin cảm ơn The Dariu Foundation đã cho chúng tôi cơ hội để đồng hành trong nhiều dự án liên quan đến lập trình, để tài liệu này có cơ hội đến với người đọc. Tôi cũng xin chân thành cảm ơn tất cả bạn bè, đối tác đã đồng hành cùng tôi trong suốt một chặng đường dài trong dự án "Phổ Cập Kỹ Năng Số - Digital Literacy Initiative".



---

# Mục lục

---

<b>I</b>	<b>Tìm hiểu Scratch 3.0 Online</b>	<b>9</b>
<b>Chương 1.</b>	<b>Scratch 3.0 Online</b>	<b>11</b>
1	Giới thiệu . . . . .	12
2	Các thao tác cơ bản . . . . .	12
2.1	Chỉnh ngôn ngữ sang Tiếng Việt . . . . .	12
2.2	Tạo mới/ Lưu/ Mở dự án . . . . .	13
3	Các khối lệnh cơ bản . . . . .	13
4	Các khối lệnh nâng cao . . . . .	14
5	Câu hỏi ôn tập . . . . .	16
<b>Chương 2.</b>	<b>Đồ họa và Âm nhạc</b>	<b>17</b>
1	Giới thiệu . . . . .	18
2	Đồ họa - Bút vẽ . . . . .	18
3	Chương trình vẽ hoa văn . . . . .	20
4	Âm nhạc . . . . .	21
4.1	Hiện thực tiếng trống . . . . .	22
4.2	Hiện thực tiếng piano . . . . .	23
5	Câu hỏi ôn tập . . . . .	25
<b>Chương 3.</b>	<b>Học máy với Google - AI Video</b>	<b>27</b>
1	Giới thiệu . . . . .	28
2	Bắt đầu với Học Máy Google . . . . .	28
3	Huấn luyện dữ liệu sai . . . . .	31
4	Xuất kết quả huấn luyện . . . . .	32
5	Lưu dự án vào Google drive . . . . .	33
6	Mở dự án từ Google drive . . . . .	34
7	Câu hỏi ôn tập . . . . .	35
<b>Chương 4.</b>	<b>Tích hợp nhận dạng hình ảnh</b>	<b>37</b>
1	Giới thiệu . . . . .	38
2	Công cụ học máy với Google . . . . .	38
3	Nhận dạng hình ảnh . . . . .	39
4	Cải thiện độ chính xác . . . . .	39
5	Câu hỏi ôn tập . . . . .	41

<b>Chương 5. Học máy với Google - AI Audio</b>	<b>43</b>
1 Giới thiệu . . . . .	44
2 Huấn luyện hệ thống âm thanh . . . . .	44
3 Xuất kết quả huấn luyện . . . . .	47
4 Lưu dự án vào Google drive . . . . .	49
5 Mở dự án từ Google drive . . . . .	49
6 Câu hỏi ôn tập . . . . .	50
<b>Chương 6. Tích hợp nhận dạng âm thanh</b>	<b>51</b>
1 Giới thiệu . . . . .	52
2 Nhận dạng âm thanh . . . . .	52
3 Text Classification . . . . .	53
4 Câu hỏi ôn tập . . . . .	55
<b>Chương 7. Kết nối với Microbit</b>	<b>57</b>
1 Giới thiệu . . . . .	58
2 Phần mềm chạy nền - Firmware . . . . .	58
3 Kết nối Scratch 3.0 và Microbit . . . . .	59
4 Hiển thị trên Microbit . . . . .	60
5 Nút nhấn trên Microbit . . . . .	61
6 Câu hỏi ôn tập . . . . .	62
<b>Chương 8. Tùy chỉnh firmware cho Microbit</b>	<b>63</b>
1 Giới thiệu . . . . .	64
2 Môi trường lập trình MakeCode . . . . .	64
3 Thêm sự kiện hành vi . . . . .	65
4 Hiện thực sự kiện trên Scratch 3.0 . . . . .	66
5 Câu hỏi ôn tập . . . . .	68
<b>Chương 9. Điều khiển Robot Microbit</b>	<b>69</b>
1 Giới thiệu . . . . .	70
2 Nạp firmware cho Robot . . . . .	70
3 Điều khiển Robot di chuyển . . . . .	71
4 Đọc dữ liệu từ cảm biến . . . . .	72
4.1 Cảm biến khoảng cách . . . . .	72
4.2 Cảm biến đường kẻ . . . . .	72
5 Scratch 3.0 trên di động . . . . .	73
6 Câu hỏi ôn tập . . . . .	74
<b>Chương 10. Hiện thực AGV Robot</b>	<b>75</b>
1 Giới thiệu . . . . .	76
2 Thay đổi firmware cho Robot . . . . .	76
3 Di chuyển theo đường kẻ . . . . .	78
4 Điều khiển cánh tay . . . . .	78
5 Câu hỏi ôn tập . . . . .	80

<b>II</b>	<b>Dự án với Scratch 3.0 Online</b>	<b>83</b>
<b>Chương 11.</b>	<b>Trợ lý ảo với Text Classification</b>	<b>85</b>
1	Giới thiệu . . . . .	86
2	Nhập diện giọng nói . . . . .	86
3	Trí thông minh nhân tạo . . . . .	87
4	Phân loại văn bản . . . . .	88
5	Câu hỏi ôn tập . . . . .	91
<b>Chương 12.</b>	<b>Thị giác Máy tính với Video AI</b>	<b>93</b>
1	Giới thiệu . . . . .	94
2	Dự án điểm danh . . . . .	94
3	Dự án phân loại rau quả . . . . .	96
4	Robot tự hành . . . . .	96
5	Triển khai với Mini PC . . . . .	96
5.1	Máy tính NUC . . . . .	97
5.2	Máy tính nhúng LettePanda . . . . .	97
5.3	Điện thoại di động . . . . .	98
6	Câu hỏi ôn tập . . . . .	99





# **Phần I**

## **Tìm hiểu Scratch 3.0 Online**



# CHƯƠNG 1

---

SCRATCH



## Scratch 3.0 Online

---



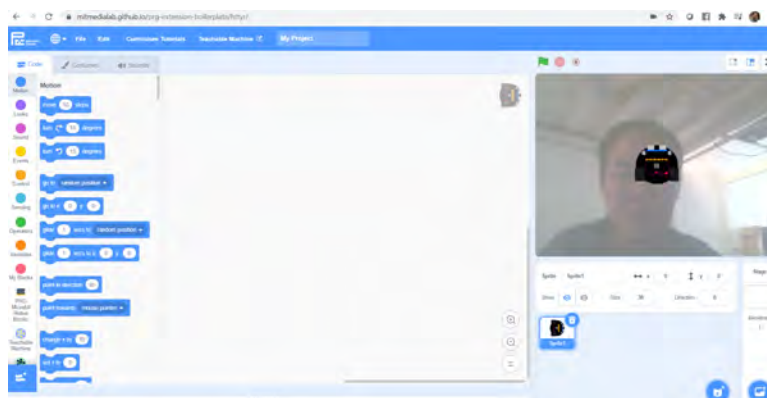
# 1 Giới thiệu

Với mục tiêu là đưa kiến thức lập trình trên máy tính sớm tới học sinh lứa tuổi cấp 2, Scratch đã trở thành ngôn ngữ hết sức gần gũi trong giáo dục tin học hiện tại. Dựa trên nền tảng là ngôn ngữ hình khối (Blocky), rất nhiều các nền tảng lập trình tương tự đã được phát triển, có thể kể ra như MakeCode cho MicroBit hay CodeBlock cho các nền tảng Arduino. Bên cạnh đó, bản thân môi trường lập trình Scratch truyền thống của MIT đã được nâng cấp từ 2.0 lên 3.0.

Trong thời gian gần đây, mã nguồn mở cho ngôn ngữ Scratch đã được công bố bởi chính MIT, để cộng đồng có thể tích hợp thêm nhiều tính năng hiện đại và cập nhật theo thời đại, như là Trí tuệ nhân tạo và Robotics. Trong giáo trình này, chúng tôi gọi đây là phiên bản **Scratch 3.0 Online**. Việc lập trình hiện tại đã trở nên vô cùng tiện dụng với tính năng lập trình trực tuyến: không cần cài đặt, và quan trọng hơn, việc học lập trình có thể dùng cho điện thoại thông minh hoặc máy tính bảng. Để bắt đầu lập trình, chúng ta sẽ vào đường dẫn sau đây:

<https://mitmedialab.github.io/prg-extension-boilerplate/http/>

Giao diện lập trình trực tuyến sẽ hiện ra như hình bên dưới. Điều đầu tiên mà chúng ta cần lưu ý là trang web này yêu cầu **quyền truy cập vào webcam và micro của máy tính**. Bạn hãy nhớ đồng ý để có thể sử dụng 2 tính năng này trong tương lai. Trong trường hợp quên đồng ý, hãy tắt và vào lại trang web này.



Hình 1.1: Giao diện lập trình Scratch 3.0 Online

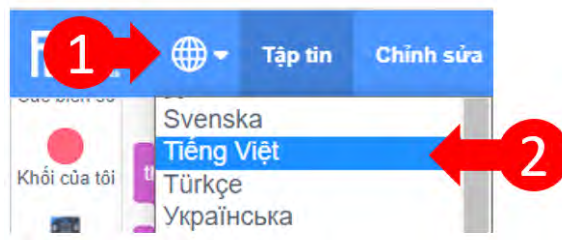
Chúng ta có thể thấy rằng, hình nền sân khấu đang là hình trực tiếp từ webcam của máy tính. Không chỉ như vậy, chương trình đang thu âm trực tiếp từ micro của máy tính. Đây là những chức năng mới trên môi trường lập trình trực tuyến so với phiên bản Scratch 3.0 trên máy tính.

## 2 Các thao tác cơ bản

### 2.1 Chuyển ngôn ngữ sang Tiếng Việt

Đây gần như là thao tác mà chúng ta sẽ làm đầu tiên để tiện cho việc tương tác. Mặc dù việc dịch từ tiếng Anh sang tiếng Việt trên Scratch 3.0 Online vẫn chưa

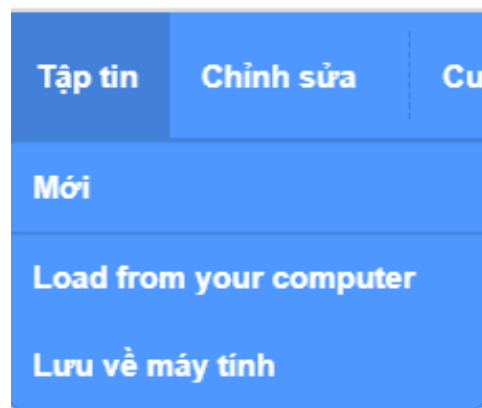
chuẩn xác hoàn toàn, chúng ta vẫn rất cần tính năng hỗ trợ ngôn ngữ tiếng Việt. Các thao tác được minh họa như hình bên dưới: chọn vào biểu tượng **quả địa cầu** và chọn tiếp **Tiếng Việt**.



Hình 1.2: Chuyển ngôn ngữ sang Tiếng Việt

## 2.2 Tạo mới/ Lưu/ Mở dự án

Bạn đọc cũng có thể mở lại những dự án cũ đã làm trên phiên bản Scratch 3.0 Desktop. Phần mềm Scratch 3.0 Online có độ tương thích tương đối ổn và có thể tự chuyển đổi. Chức năng này nằm trong mục **Tập tin**, như hình bên dưới:



Hình 1.3: Các chức năng trong mục Tập tin: Tạo mới - Mở - Lưu dự án

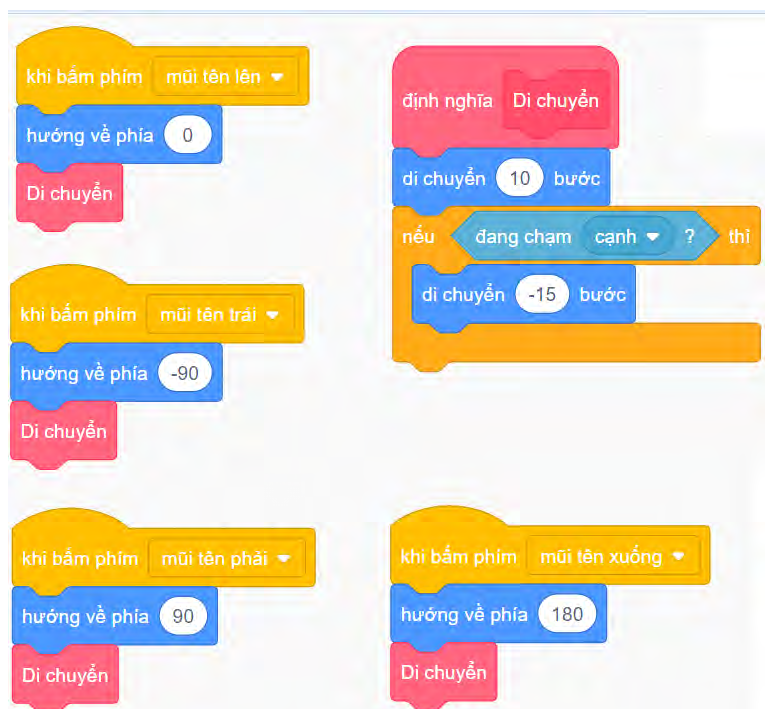
Ngay trong phần này, chúng ta có thể thấy một lỗi dịch thuật trên Scratch 3.0, **Load from your computer** là chức năng để mở lại dự án cũ (định dạng file là **.sb3**).

## 3 Các khối lệnh cơ bản

Các khối lệnh đầu tiên, từ **Chuyển động** cho tới **Các biến số** gần như khá đồng bộ giữa nhiều phiên bản khác nhau của môi trường lập trình Scratch. Tuy nhiên, thay vì là nhân vật chú mèo, các câu lệnh của chúng ta sẽ tạo ra hiệu ứng lên một nhân vật Robot trên màn hình sân khấu.

Bạn đọc hay tự mình hiện thực lại các dự án cũ đã làm, nhưng trên môi trường Scratch 3.0 Online. Bạn đọc cũng có thể mở lại dự án cũ và bổ sung thêm các câu lệnh không tương thích trong phiên bản này. Các thao tác như kéo thả, ráp nối,

thay đổi thông số của một câu lệnh là tương tự với các phiên bản cũ hơn. Chúng tôi xin đưa ra một chương trình nhỏ để minh họa cho việc điều khiển sự di chuyển của nhân vật bằng 4 phím mũi tên, như minh họa ở hình bên dưới:



Hình 1.4: Chương trình điều khiển nhân vật di chuyển

Để tạo ra một **chương trình con**, hay khối mới có tên **Di chuyển**, bạn vào **Khởi của tôi** và làm tương tự như trên Scratch 2.0. Hãy lưu ý vào màu của các khối lệnh khi tham khảo chương trình ở trên.

Có một lưu ý quan trọng là các câu lệnh liên quan tới **Bút vẽ không còn được hỗ trợ mặc định trên Scratch 3.0 Online**. Bài hướng dẫn tiếp theo sẽ trình bày cách sử dụng lại các câu lệnh liên quan tới Bút vẽ.

## 4 Các khối lệnh nâng cao

Phiên bản hiện tại của MIT hỗ trợ 3 khối lệnh mới. Tuy nhiên trong bài này, chúng tôi chỉ giới thiệu tổng quan chức năng của nó mà không đi vào chi tiết, như sau:

- PRG Microbit Robot Blocks: Các câu lệnh trong nhóm này dùng để kết nối với một Robot sử dụng mạch Microbit. Tuy nhiên, bạn cũng có thể dùng nó để điều khiển một mạch Microbit trong 1 dự án của mình chứ không nhất thiết phải có một Robot. Trong trường hợp này, bạn cần phải hiệu chỉnh lại **phần dẻo**, hay còn gọi là firmware cho mạch Microbit, sẽ được trình bày ở Bài 8.
- Cảm biến video: Scratch 3.0 Online nhận dạng webcam như là một dữ liệu đầu vào quan trọng để thực thi các giải thuật nhận dạng sử dụng trí tuệ nhân tạo. Để có thể sử dụng được các câu lệnh nằm trong nhóm này, bạn cần phải huấn luyện hệ thống bằng công cụ Teachable Machine của Google, sẽ được

trình bày trong Bài 4. Không những thế, âm thanh tích hợp trong webcam cũng có thể được huấn luyện để nhận dạng.

- Text Classification: Ban đầu, nó có thể khá giống với nhận dạng giọng nói bằng công cụ Teachable Machine, nhưng đây mới là phiên bản Trí tuệ nhân tạo thực thụ mà Google hỗ trợ. Phần mềm sẽ ghi âm từ micro của máy tính và nhận dạng bằng hệ thống của Google. Đây chính là công nghệ trong tìm kiếm giọng nói rất phổ biến trên các thiết bị di động hay tivi thông minh hiện tại. Tuy nhiên, tiếng Việt chưa hỗ trợ ở phiên bản hiện tại.

## 5 Câu hỏi ôn tập

1. Những khác biệt của phiên bản Scratch 3.0 Online so với phiên bản Scratch 2.0 là gì?
  - A. Có thêm hình ảnh từ webcam
  - B. Có thêm các khối lệnh liên quan đến AI
  - C. Có thêm các khối lệnh liên quan đến Robot
  - D. Tất cả đều đúng
2. Các khối nào sau đây của Scratch 3.0 Online là giống với Scratch 2.0?
  - A. Di chuyển
  - B. PRG Microbit Robot Blocks
  - C. Cảm biến video
  - D. Text Classification
3. Khối nào sau đây có trên Scratch 2.0 nhưng không có trên mặc định của Scratch 3.0 Online?
  - A. Di chuyển
  - B. Ngoại hình
  - C. Âm thanh
  - D. Bút vẽ
4. Để tạo một khối mới, chúng ta sẽ vào nhóm lệnh nào?
  - A. Di chuyển
  - B. Ngoại hình
  - C. Sự kiện
  - D. Khối của tôi
5. Câu lệnh khi bấm phím mũi tên lên nằm trong nhóm lệnh nào?
  - A. Di chuyển
  - B. Ngoại hình
  - C. Sự kiện
  - D. Khối của tôi
6. Câu lệnh kiểm tra đang chạm cạnh thuộc nhóm lệnh nào?
  - A. Cảm biến
  - B. Ngoại hình
  - C. Sự kiện
  - D. Khối của tôi
7. Câu lệnh hướng về phía thuộc nhóm lệnh nào?
  - A. Di chuyển
  - B. Ngoại hình
  - C. Sự kiện
  - D. Khối của tôi

Đáp án

1. D   2. A   3. D   4. D   5. C   6. A   7. A



# CHƯƠNG 2

---

SCRATCH



## Đồ họa và Âm nhạc

---



# 1 Giới thiệu

Một trong những cải tiến của công cụ lập trình Scratch 3.0 là các câu lệnh mới liên quan đến Đồ họa và Âm nhạc. Các câu lệnh liên quan đến 2 phần này đã thực sự cải tiến và tách thành phần mở rộng, thay vì tích hợp sẵn trong giao diện lập trình như Scratch 2.0. Hai nhóm lệnh lập trình đặc biệt phù hợp cho lứa tuổi nhỏ để phát triển sự sáng tạo đã được hỗ trợ rất sinh động trong Scratch 3.0 Online. Nếu như phần đồ họa khá tương tự với Scratch2.0, phần Âm nhạc đã được cải tiến rất nhiều. Không chỉ với các âm thanh nhạc bình thường, Scratch 3.0 hỗ trợ tiếng trống, tiếng guitar cùng nhiều loại nhạc cụ khác.

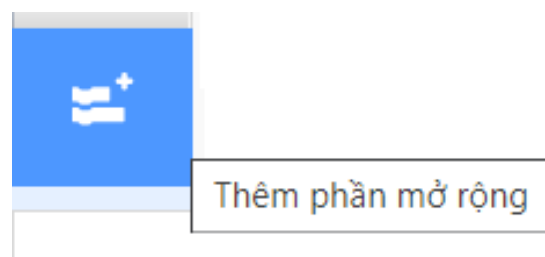
Trong bài hướng dẫn này, chúng tôi sẽ tập trung vào việc hiện thực chương trình vẽ hình hoa văn, một dự án đã quen thuộc với bạn đọc và từng là một phong trào trong dự án "Phổ Cập Kỹ Năng Số - Digital Literacy Initiative" do Dariu phát động. Dự án này sẽ sử dụng hầu hết các câu lệnh quan trọng trong nhóm lệnh Bút vẽ. Tiếp theo, hướng dẫn sẽ trình bày những bước cơ bản để bạn có thể phối tiếng nhạc và tiếng trống trong một bài hát. Ở đây chúng tôi chọn bài **Bụi phấn** để hướng dẫn. Các mục tiêu chính của bài hướng dẫn này như sau:

- Sử dụng được khối lệnh Bút vẽ
- Hiện thực được dự án vẽ hoa văn
- Sử dụng được khối lệnh Âm nhạc
- Hiện thực được chương trình phối nhạc tiếng trống và piano

## 2 Đồ họa - Bút vẽ

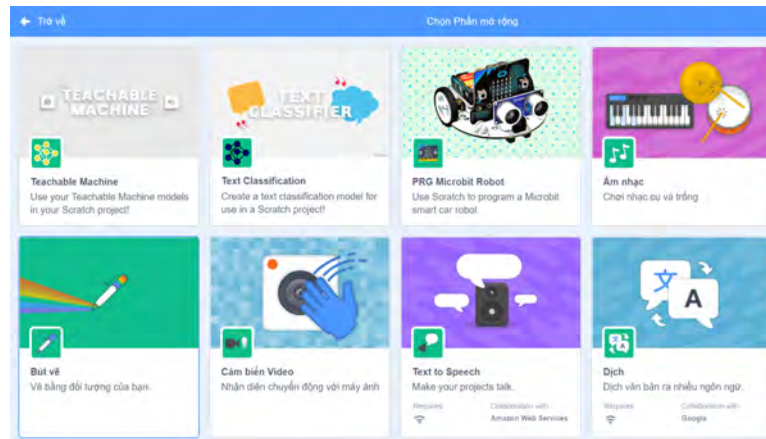
Khác với môi trường lập trình Scratch 2.0, chức năng này có sẵn trong các nhóm lệnh. Từ phiên bản 3.0, bút vẽ được xem như phần mở rộng mà bạn cần phải thêm vào môi trường lập trình. Các bước cần thiết để sử dụng chức năng này được hướng dẫn ở bên dưới.

**Bước 1:** Kéo chuột tới ô cuối cùng trong nhóm lệnh, để chọn vào chức thêm **Thêm phần mở rộng**, như minh họa ở hình bên dưới.



Hình 2.1: Thêm khối lệnh bút vẽ vào chương trình

**Bước 2:** Ở giao diện mới hiện ra, hãy tìm tới khối **Bút vẽ** và chọn nó. Trong trường hợp muốn thoát giao diện này, nhấn vào nút **Back** ở góc trên bên trái.



Hình 2.2: Chọn khối lệnh Bút vẽ

**Bước 3:** Khối lệnh Bút vẽ sẽ được thêm vào, với các câu lệnh mới như trình bày ở hình bên dưới.



Hình 2.3: Các câu lệnh mới được thêm vào

### 3 Chương trình vẽ hoa văn

Chúng ta sẽ hiện thực lại chương trình này trên môi trường Scratch 3.0 là có thể thông thạo các câu lệnh liên quan tới Bút vẽ. Các bước cơ bản để hiện thực chương trình này được trình bày như sau.

**Bước 1:** Tạo một biến số để lưu một bước vẽ. Chúng ta sẽ vào khối lệnh **Các biến số** và chọn **Tạo một biến**, như hướng dẫn bên dưới.



Hình 2.4: Các bước để tạo một biến số

Một giao diện sẽ hiện lên cho chúng ta đặt tên biến số. Ở đây, chúng tôi tạo mới một biến **i**, và nhấn vào nút **OK**. Sau khi biến **i** được tạo ra, các câu lệnh liên quan tới nó như gán giá trị ban đầu hoặc thay đổi giá trị của nó, chúng ta sử dụng như trên phần mềm Scratch 2.0.

**Bước 2:** Hiện thực chương trình vẽ hoa văn như gợi ý ở hình bên dưới.



Hình 2.5: Chương trình vẽ hoa văn

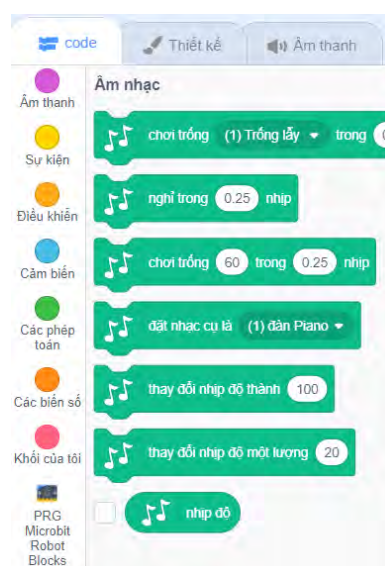
Bạn đọc hãy thử thay đổi số lần lặp cũng như góc xoay để trải nghiệm kết quả của chương trình. Bạn đọc cũng có thể thay đổi màu của hoa văn trong câu lệnh **chọn bút màu**. Mỗi lần lặp, nếu số bước thay đổi 1 lượng nhỏ, hoa văn sẽ mịn hơn, ngược lại, hoa văn sẽ thưa hơn. Với chương trình của chúng tôi, kết quả sẽ như sau:



Hình 2.6: Kết quả chương trình hoa văn

## 4 Âm nhạc

Tương tự như việc thêm các câu lệnh trong nhóm Bút vẽ, chúng ta cũng làm tương tự như hướng dẫn ở Hình 2.1 và Hình 2.2 để thêm khối mở rộng. Lần này, chúng ta sẽ chọn vào khối **Âm nhạc**. Khi thêm khối này thành công, chúng ta sẽ có các câu lệnh riêng cho phần Âm nhạc, như sau:



Hình 2.7: Các câu lệnh liên quan tới Âm nhạc



Chúng ta sẽ sử dụng các câu lệnh này để hiện thực một đoạn đầu trong bài Bụi phần, với các nốt nhạc được minh họa như hình bên dưới:

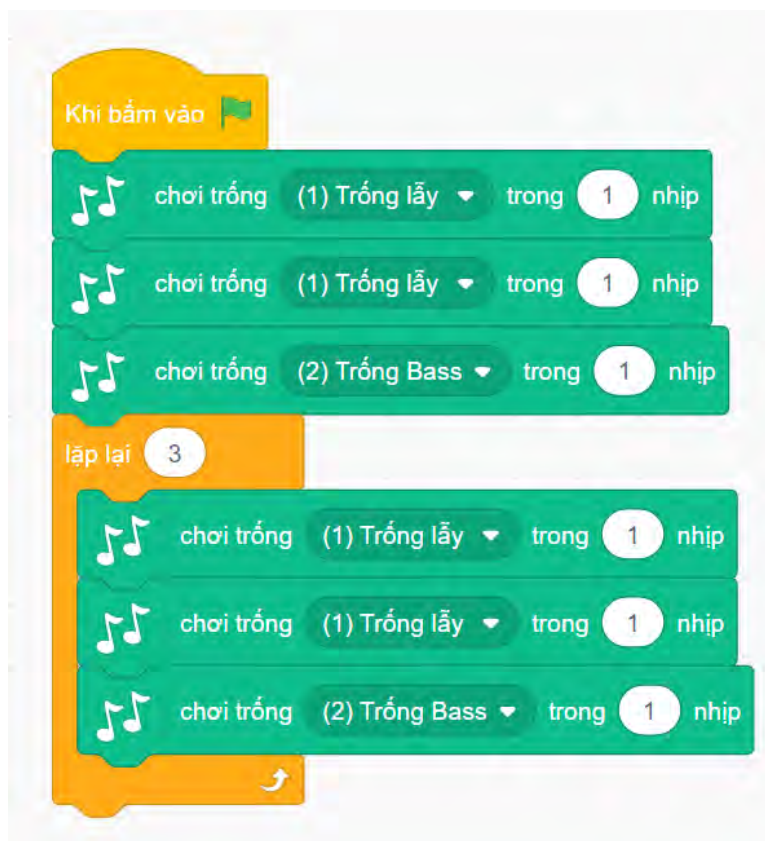


Hình 2.8: Các nốt nhạc của bài Bụi phần

## 4.1 Hiện thực tiếng trống

Trước tiên, chúng ta sẽ hiện thực tiếng trống cho bài hát. Hãy lưu ý rằng, đây là một bài nhạc có nhịp là  $3/4$ . Tức là trong 1 khuôn nhạc F (từ nốt ứng với chữ **thầy** cho đến hết chữ **viết**) có tất cả 3 nhịp tất cả, **2 nhịp cho trống lấy và 1 nhịp cho trống bass**. Ở đoạn nhạc trên, chúng ta có tất cả 4 khuôn nhạc (các nốt ứng với các từ **thầy, bảng, phần** và **rơi**).

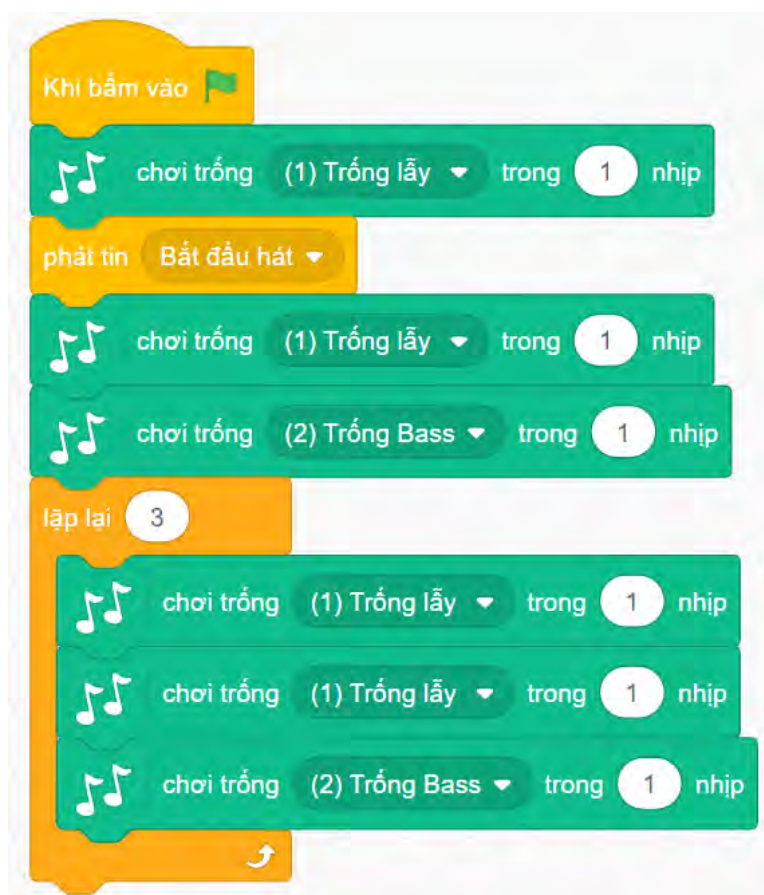
Như vậy, chúng ta sẽ có 12 nhịp trống. Một chương trình gợi ý cho việc tạo ra 12 nhịp trống như sau:



Hình 2.9: Các nốt nhạc của bài Bụi phần

Sở dĩ, chúng tôi tách 3 nhịp đầu tiên ra, vì nó sẽ dùng để đồng bộ với một nhạc cụ khác để hòa âm vào nhịp trống. Từ sau đó, nhịp trống mới lặp đi lặp lại, tùy vào độ dài của bài hát mà bạn đọc có thể tăng số lần lặp lên.

Hãy để ý lại bản nhạc ở Hình 2.8. Nguyên lý để vào cho đúng nhạc, là **nốt ở đầu mỗi khuôn nhạc phải trùng với tiếng trống bass**. Như vậy, có thể thấy rằng, sau 1 nhịp trống lấy đầu tiên, chúng ta có thể báo hiệu cho nhạc cụ piano bắt đầu hòa nhạc. Chúng ta hiện thực điều này bằng kỹ thuật gửi tin nhắn, vốn đã rất quen thuộc với bạn đọc qua Scratch 2.0. Chương trình gợi ý cho chúng ta như sau:



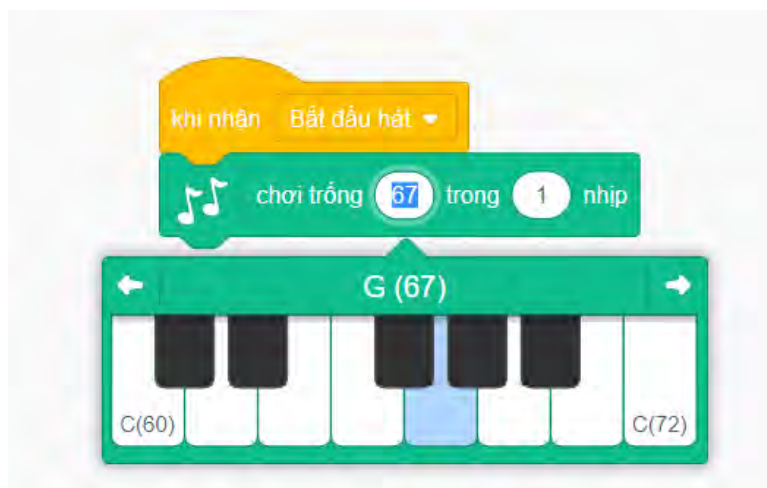
Hình 2.10: Thông báo cho nhạc cụ piano bắt đầu hòa âm từ nhịp thứ 2

Để đặt tin cho 1 thông báo, bạn nhấn vào tùy chọn của câu lệnh **phát tin**, và chọn vào **Thông báo mới**.

## 4.2 Hiện thực tiếng piano

Việc hiện thực tiếng piano the nốt của bài hát cần phải lưu ý như sau: nốt đen là một nhịp trống, nốt trắng là 2 nhịp trống và nốt kép là nửa nhịp trống. Tiếng piano sẽ bắt đầu nốt đầu tiên khi nhận được thông báo **Bắt đầu hát**. Câu lệnh chính mà chúng ta sẽ sử dụng để phát một nốt nhạc, là **câu lệnh thứ 3 trong nhóm Âm nhạc**, được minh họa như hình bên dưới:

Do quá trình chuyển đổi sang tiếng Việt, làm cho câu lệnh phát một nốt nhạc giống hệt với câu lệnh tiếng trống. Tuy nhiên khi nhấn vào tùy chọn của nó, chúng ta sẽ



Hình 2.11: Phát một nốt nhạc trên Scratch 3.0

có hình ảnh của một cây đàn. Chúng ta bắt đầu nốt đầu tiên của bài nhạc, là nốt Sol (cao độ là số 67).

Việc hiện thực cả đoạn nhạc cần sự kiên nhẫn và một chút kiến thức về nốt nhạc. Cuối cùng là bạn đọc cần phải rất lưu ý về nhịp của từng nốt, nếu không tiếng piano sẽ không khớp với tiếng trống, mà chúng ta hay gọi là trật nhịp!!! Chương trình gợi ý cho bạn đọc như sau:



Hình 2.12: Phát tiếng đàn piano cho bài Bụi phấn



## 5 Câu hỏi ôn tập

1. Công cụ để vẽ hình trên Scratch 3.0 Online có tên là gì?  
A. Bút vẽ  
B. Hội họa  
C. Vẽ hình  
D. Không hỗ trợ
2. Câu lệnh nào sau đây là điểm khác biệt lớn giữa Scratch 3.0 Online và Scratch 2.0 trong nhóm lệnh Bút vẽ?  
A. Xóa tất cả  
B. Bắt đầu vẽ  
C. Chọn màu vẽ  
D. Tất cả đều sai
3. Trong chương trình vẽ hoa văn, mẫu chốt để có một hình hoa văn khác là gì?  
A. Số vòng lặp  
B. Góc xoay sau mỗi lần lặp  
C. Độ dài cạnh tăng lên sau mỗi lần lặp  
D. Tất cả các yếu tố trên
4. Công cụ phát nhạc trên Scratch 3.0 Online có tên là gì?  
A. Âm thanh  
B. Âm nhạc  
C. Nhạc cụ  
D. Tất cả đều sai
5. Công cụ Âm nhạc trên Scratch 3.0 Online hỗ trợ những âm thanh nào sau đây?  
A. Tiếng trống  
B. Tiếng piano  
C. Tiếng guitar  
D. Tất cả đều đúng
6. Công cụ Âm nhạc có thể hỗ trợ phát cùng lúc bao nhiêu nhạc cụ khác nhau?  
A. 1  
B. 2  
C. 3  
D. 4
7. Kỹ thuật để đồng bộ việc phát nhạc giữa 2 nhạc cụ khác nhau, chẳng hạn như trống và piano là gì?  
A. Câu lệnh đợi  
B. Câu lệnh phát tin nhắn  
C. Câu lệnh âm thanh  
D. Tất cả đều sai

Đáp án

1. A   2. C   3. D   4. B   5. D   6. B   7. B



SCRATCH



## CHƯƠNG 3

---

### Học máy với Google - AI Video

---



# 1 Giới thiệu

Trong thời đại bùng nổ của cuộc cách mạng công nghệ 4.0, chắc hẳn chúng ta đã nghe nói nhiều về khái niệm Trí tuệ nhân tạo, hay còn gọi là AI (Artificial Intelligence). Tuy nhiên để có thể hiểu nó một cách cơ bản và trực quan không phải là một điều dễ dàng. Vì vậy, từ năm 2017 Google đã xây dựng một công cụ trực tuyến, có tên gọi là Teachable Machine, để mang lại cho mọi người một cái nhìn tổng quan nhất về trí tuệ nhân tạo. Chỉ cần rất ít thao tác, ở mọi lứa tuổi, chúng ta có thể cảm nhận được trí tuệ nhân tạo gồm có những khái niệm cơ bản nào, nó có thể làm được gì.

Teachable Machine ban đầu chỉ là một trang web trực tuyến, với mục đích chính là cho người dùng trải nghiệm với các sản phẩm của trí tuệ nhân tạo, như nhận diện hình ảnh hay nhận diện âm thanh. Tuy nhiên hiện nay, với phiên bản 2.0, Teachable Machine đã trở thành một công cụ trong việc thiết kế một ứng dụng trí tuệ nhân tạo cho chúng ta. Công cụ này đã hỗ trợ nhiều nền tảng lập trình khác nhau, như Scratch (ngôn ngữ cho học sinh) và Python, ngôn ngữ lập trình rất mạnh mẽ cho các ứng dụng thời đại mới. Ngoài ra, rất nhiều các nền tảng AI khác đang dựa trên lõi trung tâm là Google Teachable Machine.

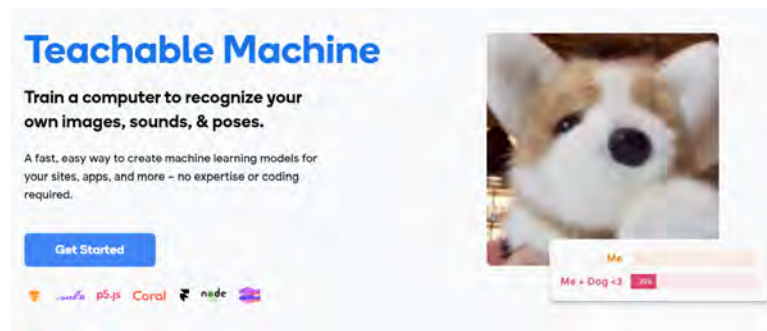
Trong bài hướng dẫn này, chúng tôi sẽ giới thiệu các thao tác cơ bản trên Teachable Machine, trước khi chúng ta bắt đầu việc lập trình kết nối giữa Python và Teachable Machine. Chúng tôi cũng tập trung vào hướng dẫn công cụ đầu tiên, là nhận diện bằng hình ảnh (Image Classifier). Các công cụ khác sẽ được trình bày trong những bài tiếp theo. Các mục tiêu chính trong bài đầu tiên được liệt kê như sau:

- Hiểu được các thành phần cơ bản của AI
- Sử dụng được công cụ Teachable Machine của Google
- Trải nghiệm Học Máy với Google trên trang web
- Lưu dự án AI trên công cụ Google Drive

Trong phần còn lại của hướng dẫn, chúng tôi sẽ xài thuật ngữ việt hóa cho Teachable Machine là "Học Máy Google" cho thuận tiện với bạn đọc.

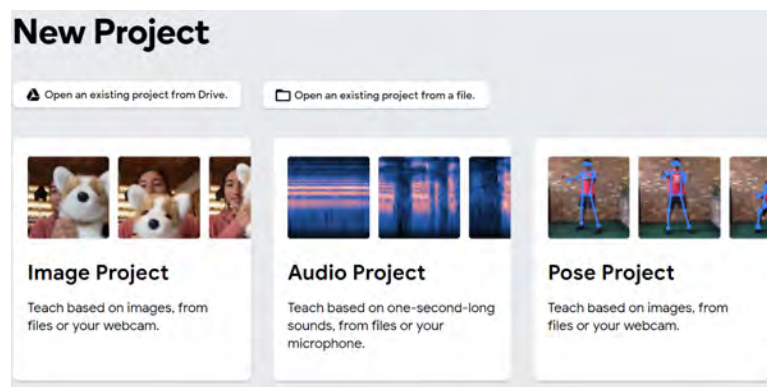
## 2 Bắt đầu với Học Máy Google

Học Máy Google đã được xây dựng thành một công cụ trực tuyến. Chúng ta bắt đầu bằng việc vào đường dẫn <https://teachablemachine.withgoogle.com/>, giao diện sau đây sẽ hiện ra:



Hình 3.1: Trang web online của Học máy Google

Tại trang web này, chúng ta có thể xem trước một số hình ảnh cũng như video giới thiệu của Google về công nghệ Học máy. Tiếp theo, nhấn vào nút **Get Started** để bắt đầu, giao diện của công cụ học máy với Google được hiện ra như sau:



Hình 3.2: Công cụ Học máy với Google - Image Project

Chúng ta chọn tiếp vào Image Project, công cụ để phân nhận diện hình ảnh. Hai công cụ còn lại (Nhận dạng âm thanh và Nhận dạng dáng người) sẽ được trình bày sau. Giao diện sau đây sẽ được hiện ra:

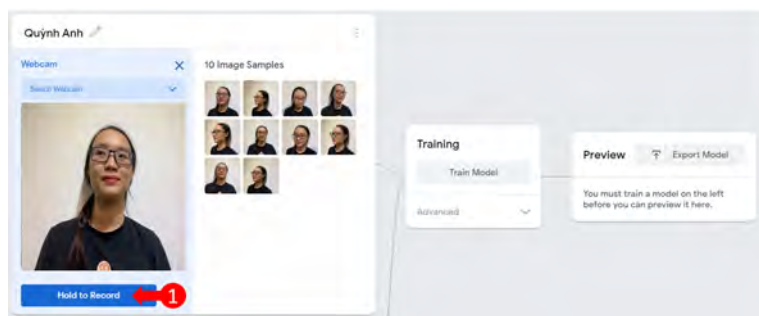


Hình 3.3: Giao diện để huấn luyện hình ảnh

Để mô hình AI có thể hoạt động, chúng ta phải **huấn luyện** (train) nó trước. Bạn có thể đặt tên cho đối tượng mà bạn muốn nhận diện bằng cách nhấn vào biểu tượng thay đổi bên cạnh tên mặc định Class1 và Class2. Thao tác này còn có một tên gọi khác là gán nhãn cho đối tượng.

Bước tiếp theo là cung cấp dữ liệu cho từng đối tượng. Google hỗ trợ 2 phương pháp khác nhau để cung cấp dữ liệu hình ảnh. Chúng ta sẽ bắt đầu trước với việc

sử dụng webcam để cung cấp hình ảnh. Khi nhấn vào biểu tượng webcam, giao diện sau đây sẽ hiện ra:

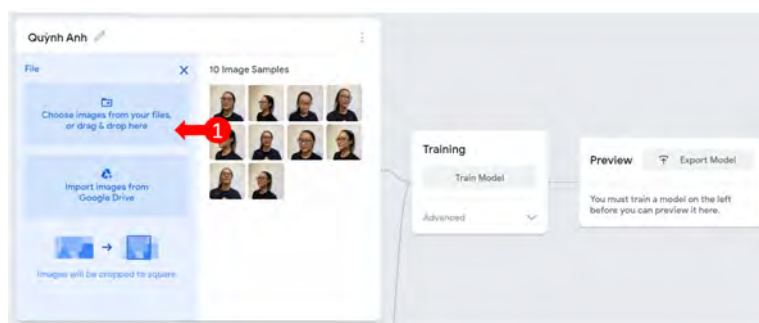


Hình 3.4: Chụp hình ảnh từ webcam

Chúng ta sẽ nhấn nút **Hold To Record** để bắt đầu chụp hình, như minh họa ở Hình 3.4. Càng có nhiều hình cho đối tượng, việc nhận diện sẽ càng chính xác, tuy nhiên thời gian huấn luyện sẽ lâu hơn. Ở đây, chúng tôi chọn mẫu khoảng 10 hình, nhưng bạn hoàn toàn có thể tăng lên 50 hình nếu muốn.

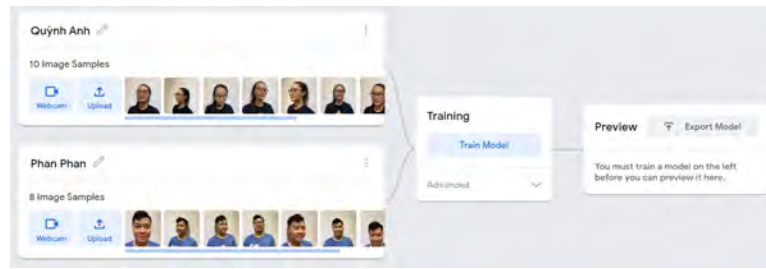
Khi muốn xóa một hình mà bạn không ưng ý, chỉ cần rê chuột vào tấm hình, biểu tượng xóa sẽ xuất hiện để bạn có thể nhấn vào.

Trong một số trường hợp, việc chụp ảnh từ webcam của máy tính không thuận tiện, chúng ta có thể chọn công cụ thứ 2 để cung cấp hình ảnh cho hệ thống trí tuệ nhân tạo. Tắt công cụ webcam đi (nhấn vào dấu X ở góc bên phải khung hình). Khi chọn vào công cụ **Upload**, giao diện sau đây sẽ hiện ra:



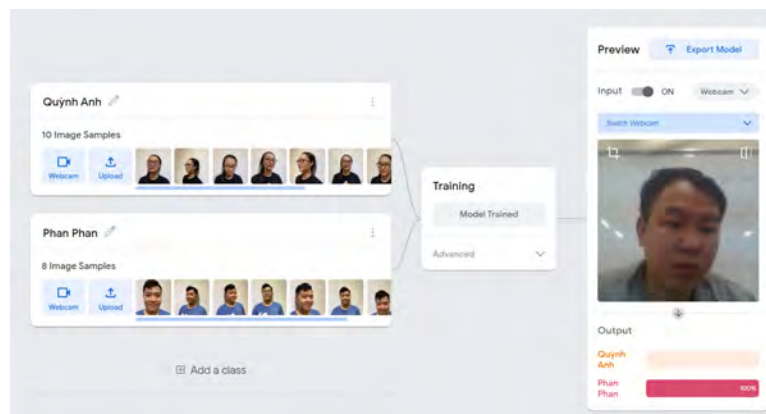
Hình 3.5: Tải thêm hình ảnh từ máy tính

Chọn vào công cụ đầu tiên, bạn đã có thể tải hình ảnh từ máy tính lên hệ thống huấn luyện của Google. Chúng ta sẽ làm tương tự cho nhóm thứ 2, như minh họa ở hình sau đây:



Hình 3.6: Tải dữ liệu cho nhóm thứ 2

Cuối cùng, chúng ta nhấn vào nút **Train Model** để hệ thống bắt đầu quá trình huấn luyện dựa trên tập dữ liệu hình ảnh mà chúng ta cung cấp. Ngay sau khi quá trình huấn luyện kết thúc, hệ thống AI sẽ bắt đầu chạy để nhận dạng. Kết quả của việc nhận dạng là xác suất trùng với các đối tượng đã được huấn luyện, như minh họa ở hình bên dưới:



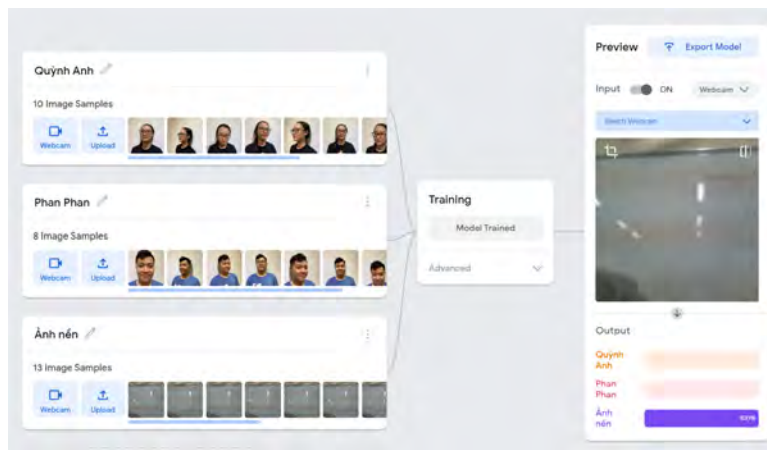
Hình 3.7: Huấn luyện và chạy thử AI

### 3 Huấn luyện dữ liệu sai

Đây là một khái niệm vô cùng quan trọng trong hệ thống trí tuệ nhân tạo. Bên cạnh việc huấn luyện nhận đúng, bạn cũng phải cần huấn luyện các ngữ cảnh không thuộc các đối tượng nhận dạng mong đợi. Ví dụ như trong hình Hình 3.7, mặc dù hình ảnh thực tế hoàn toàn khác xa với tập dữ liệu, hệ thống AI vẫn đang nhận diện là đối tượng **"Phan Phan"**. Một ví dụ khác cho hiện tượng này là khi chúng ta làm một hệ thống điểm danh tự động. Trong hệ thống đã được huấn luyện trước 10 người thật chẳng hạn. Tuy nhiên trong quá trình vận hành, camera có thể quay vào một khoảng không và hệ thống AI vẫn nhận dạng đây là một ai đó.

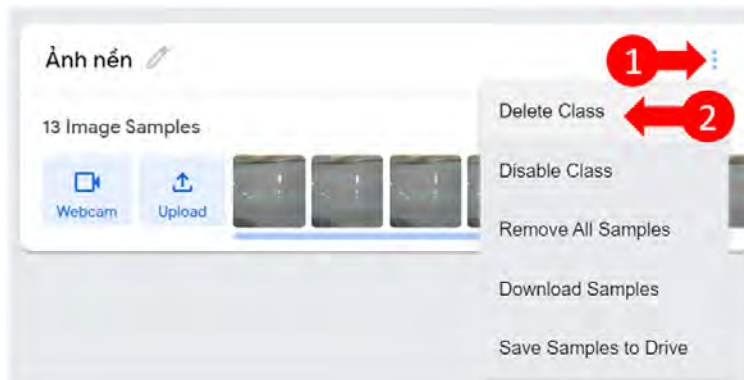
Một kĩ thuật đơn giản để khắc phục hiện tượng này là huấn luyện thêm các ngữ cảnh không mong muốn. Từ Hình 3.7, nhấn vào lựa chọn **Add a class** để thêm một đối tượng mới và cũng cung cấp hình ảnh cho đối tượng này như bình thường. Ở đây, chúng tôi thêm dữ liệu gọi là **Ảnh nền**, như minh họa bên dưới:





Hình 3.8: Thêm đối tượng ảnh nền cho hệ thống

Trong trường hợp muốn xóa một đối tượng, chúng ta nhấn vào biểu tượng **option** ở góc bên phải (bước 1), và chọn tiếp **Delete class** (bước 2), như minh họa ở hình bên dưới:



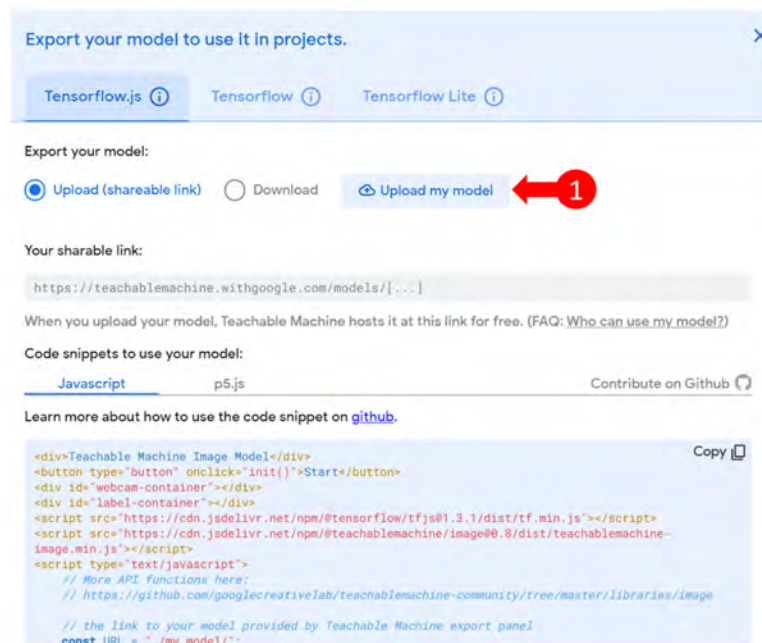
Hình 3.9: Xóa một đối tượng khỏi hệ thống AI

**Lưu ý:** Mỗi khi thay đổi hình hoặc thêm/bớt đối tượng, phải huấn luyện lại hệ thống.

## 4 Xuất kết quả huấn luyện

Đây là bước cuối cùng, để chúng ta có thể tích hợp kết quả của việc huấn luyện vào các ngôn ngữ lập trình khác như Scratch hay Python. Bằng cách nhấn vào nút Export Model, giao diện sau đây sẽ hiện ra:



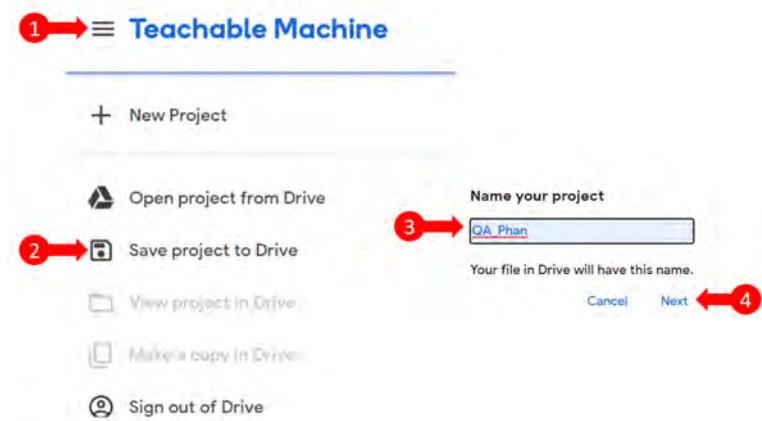


Hình 3.10: Xuất kết quả huấn luyện

Chúng ta sẽ chọn vào mục **Upload my model**, và chờ cho đến khi tác vụ này hoàn tất. Khi việc tải dữ liệu lên server của Google hoàn tất, chúng ta sẽ có một đường dẫn trong mục **Your sharable link**. Bạn hãy sao chép đường dẫn này lại để dùng trong việc lập trình trên Scratch 3.0.

## 5 Lưu dự án vào Google drive

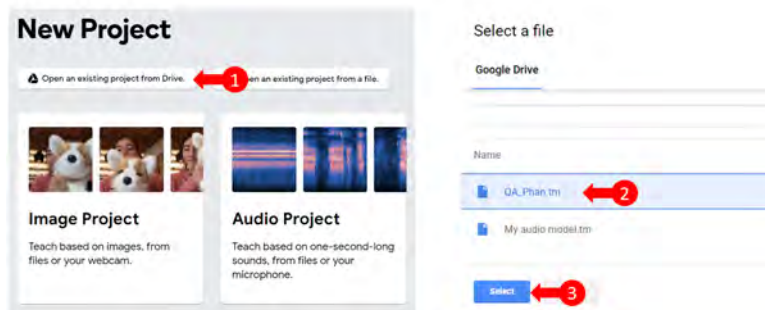
Với những hỗ trợ toàn diện từ phía Google, hệ thống AI của bạn có thể được lưu lại trực tuyến trên Google drive. Bằng các trở lại màn hình chính và làm theo 4 bước hướng dẫn ở hình bên dưới như: Chọn Save project to Drive, Đặt tên cho project và cuối cùng nhấn nút Next.



Hình 3.11: Lưu dự án vào Google drive

## 6 Mở dự án từ Google drive

Trong trường hợp muốn mở lại dự án cũ để cập nhật hoặc làm tiếp, từ trang chủ của Học Máy Google, chúng ta chọn **Open existing project from Drive** rồi lựa chọn trong danh sách các dự án đã lưu, như minh họa ở hình bên dưới.



Hình 3.12: Mở một dự án từ Google drive

Một trong những tiện ích vô cùng lớn của công cụ này, là khi chúng ta cần thay đổi dữ liệu huấn luyện, đường dẫn trực tiếp cho việc lập trình vẫn không thay đổi. Do đó, bạn có thể tự cập nhật ứng dụng của mình mà không làm ảnh hưởng đến việc thực thi của nó.

## 7 Câu hỏi ôn tập

1. Công cụ Học máy với Google có tên là gì?
  - A. Teachable Machine
  - B. Artificial Intelligence
  - C. Deep Learning
  - D. Machine Learning
2. Hiện tại có bao nhiêu công cụ AI đang được hỗ trợ sẵn bởi Google?
  - A. 1
  - B. 2
  - C. 3
  - D. Tất cả đều sai
3. Dữ liệu để huấn luyện cho hệ thống nhận dạng bằng ảnh có thể được lấy từ nguồn nào?
  - A. Trực tiếp từ webcam
  - B. Tải ảnh từ máy tính lên
  - C. Ảnh từ Google Drive của người dùng
  - D. Tất cả các nguồn ảnh trên
4. Sau khi huấn luyện, kết quả thực thi của hệ thống trí tuệ nhân tạo là gì?
  - A. Một hình ảnh dưới dạng xem trực tiếp (live view)
  - B. Xác suất tính theo phần trăm tỉ lệ nhận dạng được
  - C. Một định danh (ID) cho đối tượng nhận dạng
  - D. Tất cả đều đúng
5. Để loại bỏ việc nhận dạng ảnh nền thành một hình nào đó trong danh sách các đối tượng, chúng ta cần:
  - A. Thêm nhiều đối tượng hơn để loại bỏ ảnh nền
  - B. Huấn luyện thêm một đối tượng là ảnh nền
  - C. Huấn luyện lại ảnh cho đối tượng nhiều lần
  - D. Không cần xử lý vì Google đã hỗ trợ
6. Mô hình huấn luyện của Học máy Google hỗ trợ cho những môi trường lập trình nào sau đây?
  - A. Scratch 2.0
  - B. Scratch 3.0 Online
  - C. Scratch JR
  - D. Tất cả đều đúng
7. Để có thể sử dụng mô hình Học máy Google trên môi trường lập trình trên Scratch 3.0 Online, chúng ta cần:
  - A. Lưu lại đường dẫn trực tuyến của mô hình
  - B. Lưu lại file Tensorflow
  - C. Lưu lại file h5
  - D. Tất cả các bước trên

Đáp án

1. A   2. C   3. D   4. B   4. D   5. B   6. B   6. D   7. A



SCRATCH



## CHƯƠNG 4

---

### Tích hợp nhận dạng hình ảnh

---



# 1 Giới thiệu

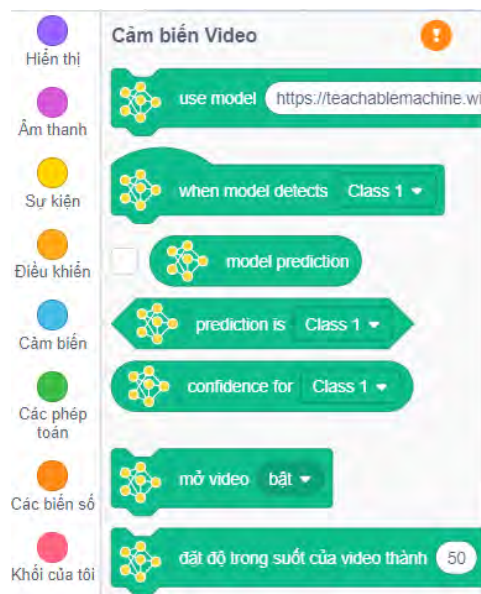
Trong bài hướng dẫn trước, chúng ta đã có thể sử dụng công cụ Học máy với Google để huấn luyện cho hệ thống nhận dạng được một vật thể mong muốn. Trong bài này, chúng ta sẽ tích hợp kết quả huấn luyện được vào một chương trình Scratch.

Thực ra, kết quả huấn luyện ở bài trước, sẽ mở ra một loại ứng dụng mới, gọi là Thị giác Máy tính (Computer Vision). Trong các ứng dụng mới này, camera đóng vai trò như con mắt của máy tính. Dựa vào những hình ảnh đã được huấn luyện trước, nó sẽ phải quyết định xem đó là đối tượng gì. Kỹ thuật này sẽ mở ra rất nhiều ứng dụng tìm năng, khi chỉ nhận dạng khuôn mặt người, nó còn có thể ứng dụng trong lĩnh vực nông nghiệp, như phân biệt các loại hoa quả, kiểm định chất lượng rau có xanh hay bị sâu rầy. Các mục tiêu hướng dẫn của bài này như sau:

- Sử dụng công cụ học máy của Google trên Scratch 3.0
- Xây dựng ứng dụng nhận diện từ Webcam
- Cải thiện độ chính xác của việc nhận dạng

## 2 Công cụ học máy với Google

Công cụ này trên môi trường lập trình được dịch sang tiếng Việt là **Cảm biến Video**. Nhóm lệnh này được trình bày như hình bên dưới:



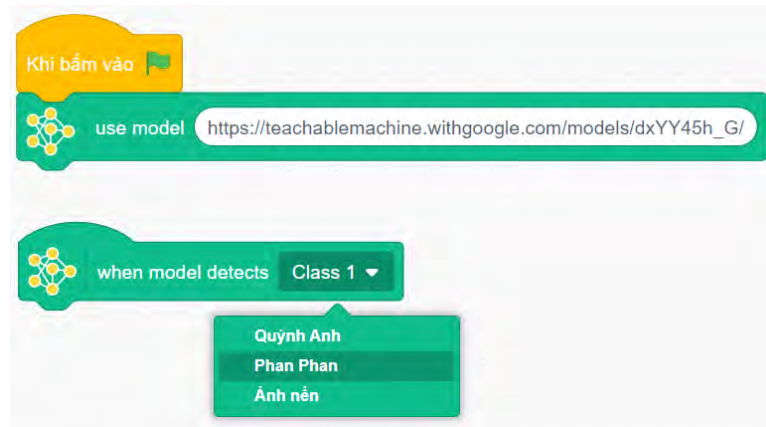
Hình 4.1: Các câu lệnh thuộc nhóm Cảm biến Video

Ban đầu, sẽ có kí hiệu dấu chấm than ở phần này, với ý nghĩa là chúng ta chưa liên kết với công cụ Học máy. Khi câu lệnh đầu tiên được thực thi thành công, biểu tượng này sẽ chuyển sang kí hiệu màu xanh lá.

Khi câu lệnh đầu tiên được thực thi thành công, tùy chọn trong câu lệnh thứ 2, **when model detects** sẽ thay đổi, tùy thuộc vào số lượng đối tượng mà chúng ta huấn luyện ở bài trước. Chúng ta sẽ tập trung sử dụng 2 câu lệnh này trước.

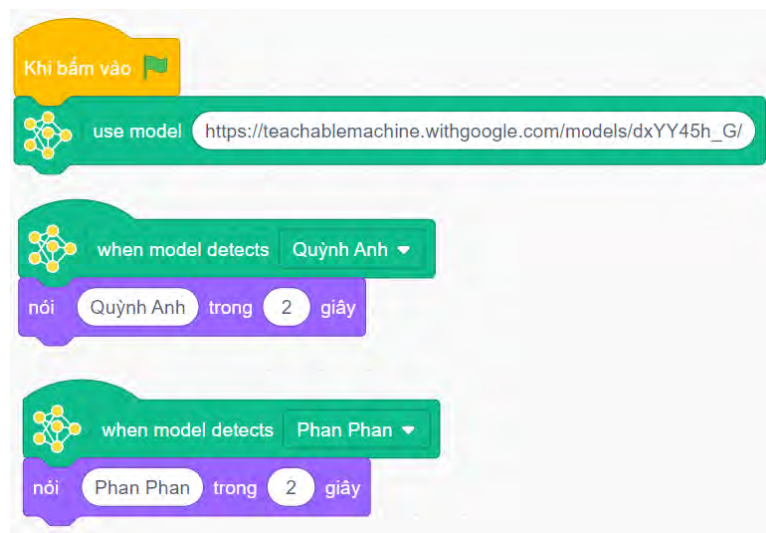
### 3 Nhận dạng hình ảnh

Để bắt đầu chương trình nhận dạng hình ảnh đơn giản, chúng ta sẽ kéo câu lệnh đầu tiên, **use model** ra trước, nhập đường dẫn mô hình của chúng ta huấn luyện ở bài trước, và sau đó nhấn vào nó để thực thi câu lệnh này, như minh họa ở hình bên dưới.



Hình 4.2: Liên kết với mô hình huấn luyện

Khi thực thi câu lệnh **use model** thành công, sẽ có 3 đối tượng có thể được lựa chọn trong câu lệnh tiếp theo. Chúng ta sẽ viết một chương trình đơn giản cho việc nhận diện hình ảnh từ webcam như sau:



Hình 4.3: Chương trình nhận dạng hình ảnh từ webcam

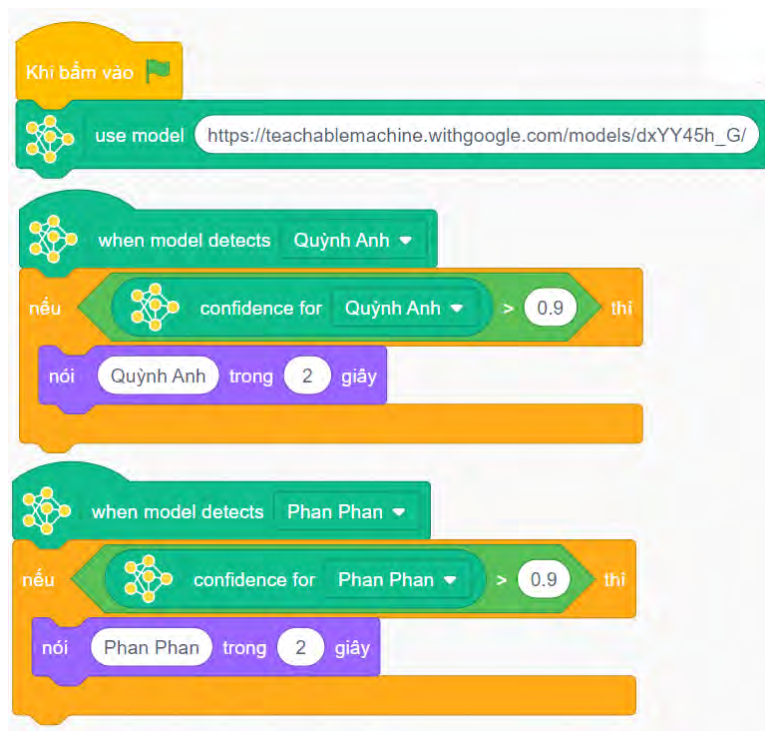
### 4 Cải thiện độ chính xác

Một trong những hạn chế của kỹ thuật huấn luyện hiện tại, là mô hình sẽ phải trả lời là một trong các đối tượng mà chúng ta huấn luyện. Điều đó cũng có nghĩa là



cho dò là ai xuất hiện trong webcam, hệ thống cũng sẽ phải quyết định là **Quỳnh Anh** hoặc **Phan Phan**. Do đó, chúng ta sẽ hiện thực thêm để tăng độ chính xác cho chương trình.

Hệ thống nhận diện của Google hỗ trợ cho chúng ta thông tin về xác suất nhận dạng, tính theo phần trăm, từ 0 cho đến 1, và được lưu trong khối **confident for**. Khối này cũng là một tùy chọn để lựa chọn đối tượng được nhận dạng. Chương trình gợi ý cho chúng ta sẽ như sau:



Hình 4.4: Cải thiện độ chính xác của việc nhận dạng

Thực ra, chương trình để tăng độ chính xác của việc nhận dạng có thể cải tiến phức tạp hơn. Ví dụ như, khi xác suất nhận dạng đối tượng trên 90% liên tục trong 2 giây thì chúng ta mới quyết định. Tuy nhiên, độ phức tạp của chương trình cũng sẽ làm chậm độ đáp ứng của dự án, mà bạn phải cân đối. Trong trường hợp muốn kiểm tra liên tục, bạn đọc cần phải khai báo thêm biến số. Giá trị của nó sẽ tăng dần sau mỗi lần nhận đúng. Và khi giá trị này đủ lớn, chúng ta mới quyết định là nhận dạng ra đối tượng đó. Phần này sẽ xem như là bài tập thêm cho bạn đọc.



## 5 Câu hỏi ôn tập

1. Nhóm lệnh liên quan đến nhận diện hình ảnh trên Scratch 3.0 là gì?
  - A. Teachable Machine
  - B. Cảm biến Video
  - C. Video AI
  - D. Tất cả các khối trên
2. Câu lệnh nào dùng để kết nối với mô hình nhận dạng của Google?
  - A. Sự kiện lá cờ màu xanh
  - B. use model
  - C. when model detects
  - D. Tất cả các thông tin trên
3. Tham số đưa vào câu lệnh use model là gì?
  - A. Tên file zip
  - B. Tên file h5
  - C. Đường dẫn trực tuyến từ Học máy của Google
  - D. Tất cả đều sai
4. Để câu lệnh when model detects có thể sử dụng được, những câu lệnh nào cần được thực thi trước?
  - A. Sự kiện lá cờ màu xanh
  - B. use model
  - C. Kết hợp 2 câu lệnh trên
  - D. Tất cả đều sai
5. Khi nhận dạng ra một đối tượng, xác suất nhận dạng đúng được lưu trong khối lệnh nào?
  - A. use model
  - B. when model detects
  - C. confidence for
  - D. Tất cả đều đúng
6. Câu lệnh confidence for chỉ có thể sử dụng sau khi thực thi câu lệnh nào?
  - A. Sự kiện lá cờ màu xanh
  - B. use model
  - C. Kết hợp 2 câu lệnh trên
  - D. Tất cả đều sai
7. Để kiểm tra đối tượng A có xác suất nhận dạng đúng trên 90%, câu lệnh nào sẽ được dùng?
  - A. nếu confident > 90
  - B. nếu confident > 0.9
  - C. nếu confident for A > 90
  - D. nếu confident for A > 0.9

Đáp án

1. B   2. B   3. C   4. C   5. C   6. C



SCRATCH



## CHƯƠNG 5

---

# Học máy với Google - AI Audio

---



# 1 Giới thiệu

Công cụ thứ 2 trong bộ công cụ Học máy với Google liên quan đến nhận dạng âm thanh. Đây thực sự là một công nghệ rất cập nhật mà Google cung cấp cho chúng ta. Sử dụng công cụ này, chúng ta có thể cho hệ thống nhận dạng những lệnh ngắn, dưới 2 giây, để làm một tác vụ nào đó.

Ban đầu, bạn đọc có thể nghĩ rằng 2 giây là quá ít để ghi âm điều gì đó. Tuy nhiên, đây là công nghệ để xây dựng nên từ lệnh cho các hệ thống nhà thông minh (Google Home) hoặc trợ lý ảo (Google Assistant) trên điện thoại. Khi cần kích hoạt những hệ thống này, bạn thường hay nói "OK Google".

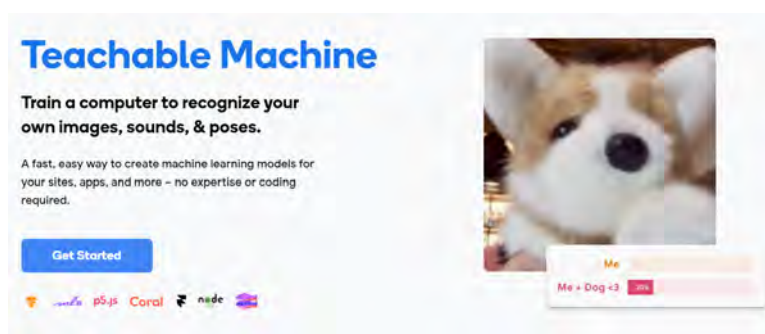
Hiện tại, đa phần các thiết bị điện tử, từ máy tính cho tới tivi thông minh, đều rất ít khi được tắt nguồn hoàn toàn. Nó chỉ ở chế độ chờ (stand by) và sẽ kích hoạt trở lại khi có tín hiệu gửi đến. Do đó, giọng nói theo kiểu ra lệnh sẽ là hướng tiếp cận cho các ứng dụng như vậy.

Bài hướng dẫn này sẽ tập trung vào các mục tiêu sau đây:

- Sử dụng được công cụ Học máy với Google liên quan tới âm thanh
- Hiểu được các khái niệm âm thanh nền, âm thanh lệnh
- Huấn luyện và chạy thử nghiệm một hệ thống trí tuệ nhân tạo về âm thanh

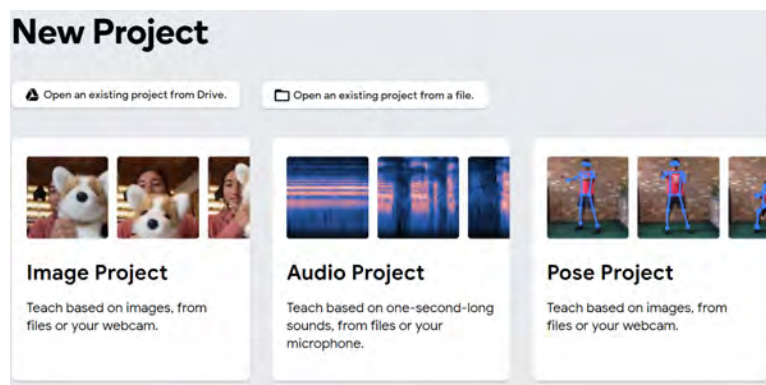
## 2 Huấn luyện hệ thống âm thanh

Học Máy Google đã được xây dựng thành một công cụ trực tuyến. Chúng ta bắt đầu bằng việc vào đường dẫn <https://teachablemachine.withgoogle.com/>, giao diện sau đây sẽ hiện ra:



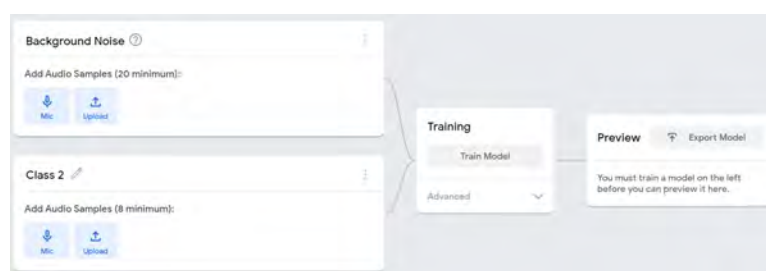
Hình 5.1: Trang web online của Học máy Google

Tại trang web này, chúng ta có thể xem trước một số hình ảnh cũng như video giới thiệu của Google về công nghệ Học máy. Tiếp theo, nhấn vào nút **Get Started** để bắt đầu, giao diện của công cụ học máy với Google được hiện ra như sau:



Hình 5.2: Công cụ Học máy với Google - Audio Project

Lần này, chúng ta sẽ chọn vào **Audio Project** để bắt đầu xây dựng hệ thống trí tuệ nhân tạo cho âm thanh. Giao diện sau đây sẽ hiện ra:



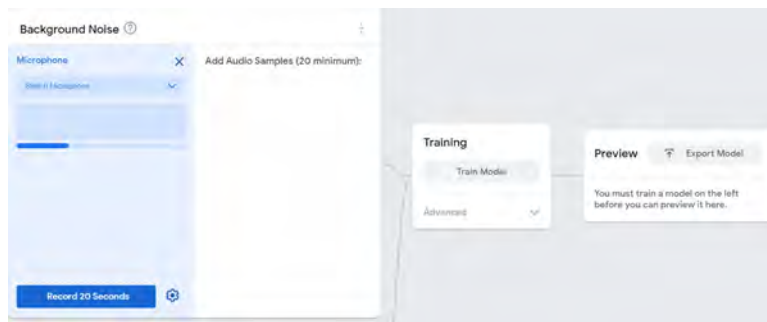
Hình 5.3: Giao diện để huấn luyện âm thanh

### Bước 1: Cung cấp âm thanh nền

Ở giao diện này, trước khi huấn luyện âm thanh, chúng ta cần phải cung cấp âm thanh nền (Background Noise), hay còn gọi là âm nhiễu. Thực sự đây là điều không dễ dàng vì âm nhiễu thường không thể xác định và cũng không có quy luật. Tuy nhiên đó là trường hợp khi chúng ta muốn phát hiện ra âm khi nói nhỏ. Trong trường hợp nói lớn hơn một chút thì ảnh hưởng của âm nền sẽ giảm đi.

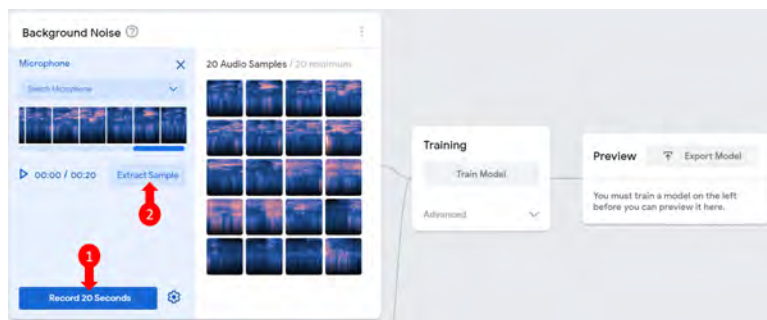
Chúng ta cũng có 2 cách để cung cấp âm thanh nền: Thu âm trực tiếp từ micro hoặc là tải âm nền đã có lên hệ thống huấn luyện. Đối với âm thanh, việc **tải một âm có sẵn lên hệ thống sẽ không tối ưu để huấn luyện**, đơn giản nó sẽ khác rất nhiều với âm thu từ micro của hệ thống. Với cách thứ nhất, giao diện sau đây sẽ hiện lên như sau:

Mặc định, hệ thống sẽ ghi âm 20 giây cho âm nền. Bạn chỉ cần nhấn vào nút **Record for 20 second** là hệ thống bắt đầu ghi âm. Trong trường hợp muốn ghi âm nhiều hơn, bạn có thể nhấn vào nút Setting ở bên cạnh để thay đổi.



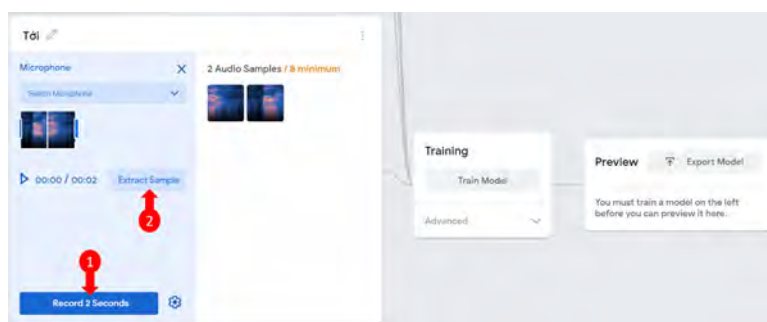
Hình 5.4: Cung cấp dữ liệu cho âm thanh nền

Sau khi ghi âm xong, bạn phải nhấn thêm vào nút Extract Sample để phân tách âm nền ra nhiều mẫu âm thanh nhỏ, mỗi âm thanh là 1 giây, như kết quả ở hình bên dưới:



Hình 5.5: Hoàn thành dữ liệu cho âm nền

Tiếp theo, chúng ta sẽ cung cấp âm thanh cho từ lệnh cần huấn luyện. Các thao tác ở bước này hoàn toàn tương tự như âm thanh nền, chỉ khác là lần này chúng ta sẽ thu mỗi 2 giây, và lưu ý là phải nhấn vào nút **Extract Sample** thì dữ liệu mới được lưu lại cho việc huấn luyện, như minh họa ở hình bên dưới:

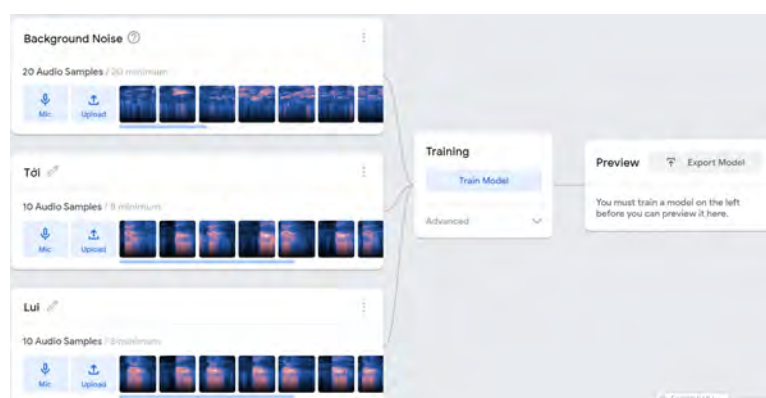


Hình 5.6: Cung cấp dữ liệu cho từ khóa Tôi

## Bước 2: Cung cấp dữ liệu cho từ lệnh

Bạn đọc có thể làm tương tự, để xây dựng hệ thống âm thanh gồm có 4 từ lệnh là Tôi, Lui, Trái và Phải. Ở đây, theo kinh nghiệm của chúng tôi, bạn hãy **nói từ lệnh khớp trong khung hình 1 giây** để việc nhận dạng sau này sẽ chính xác và nhạy hơn. Khác với hình ảnh, âm thanh có sự liên quan giữa dữ liệu trước và sau. Do đó, việc thu từ lệnh ngắn sẽ giảm tải rất nhiều cho hệ thống lúc vận hành. Kết quả cho

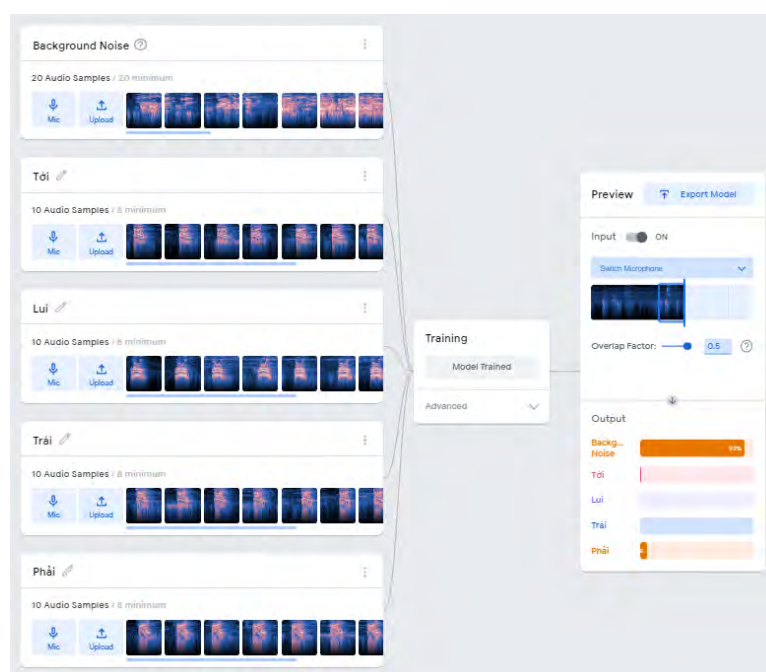
đến bước này sẽ như minh hoạt ở hình bên trên.



Hình 5.7: Toàn bộ hệ thống với 4 từ lệnh Tôi, Lui, Trái và Phải

### Bước 3: Huấn luyện hệ thống

Nhấn nút Train để bắt đầu huấn luyện tập dữ liệu âm thanh. Sau khi quá trình này kết thúc, bạn đã có thể dùng thử hệ thống, như minh họa ở hình bên dưới:

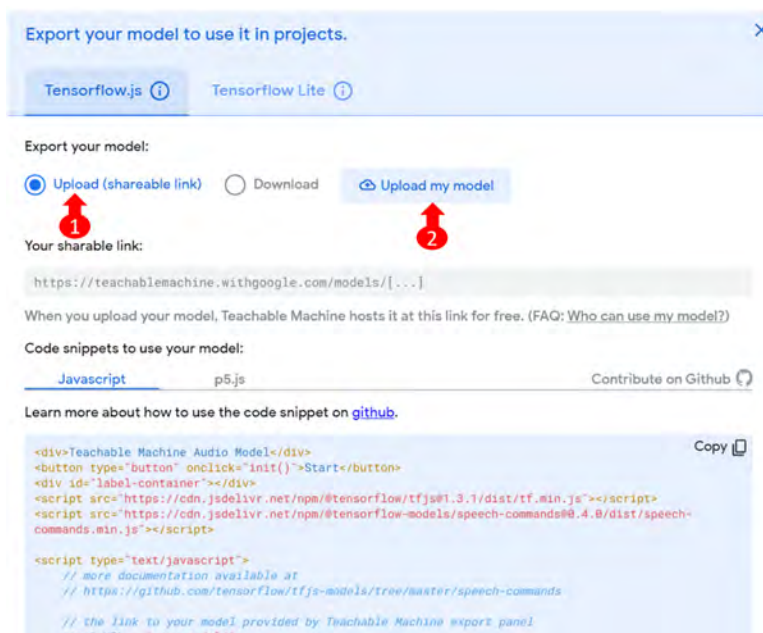


Hình 5.8: Kiểm tra hệ thống nhận diện âm thanh

## 3 Xuất kết quả huấn luyện

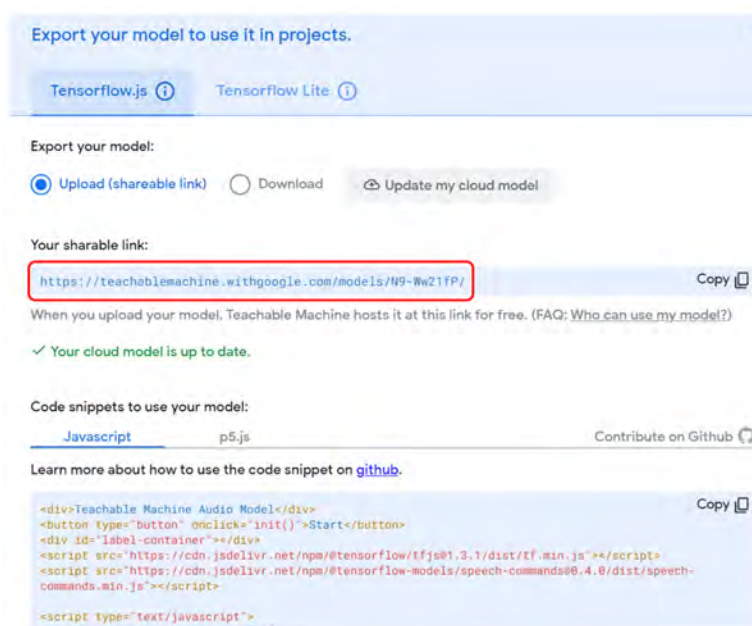
Tương tự như phần huấn luyện hình ảnh, chúng ta cũng cần phải xuất kết quả huấn luyện ra những định dạng khác để có thể tích hợp vào nhiều nền tảng khác nhau, như lập trình trên Scratch hoặc Python chẳng hạn. Bằng cách nhấn vào nút **Export Model**, giao diện sau đây sẽ hiện ra:





Hình 5.9: Tải kết quả huấn luyện lên server Google

Chúng ta sẽ chọn vào mục và chọn Upload (shareable link) và chọn tiếp vào **Upload my model** như minh họa ở hình bên dưới. Chức năng này sẽ tải toàn bộ kết quả huấn luyện lên server của Google và trả về cho chúng ta một đường dẫn để lập trình trong môi trường Scratch. Khi việc tải lên hoàn tất, màn hình dưới đây sẽ hiện ra và bạn có thể sao chép lại đường dẫn ở mục **Your shareable link**.

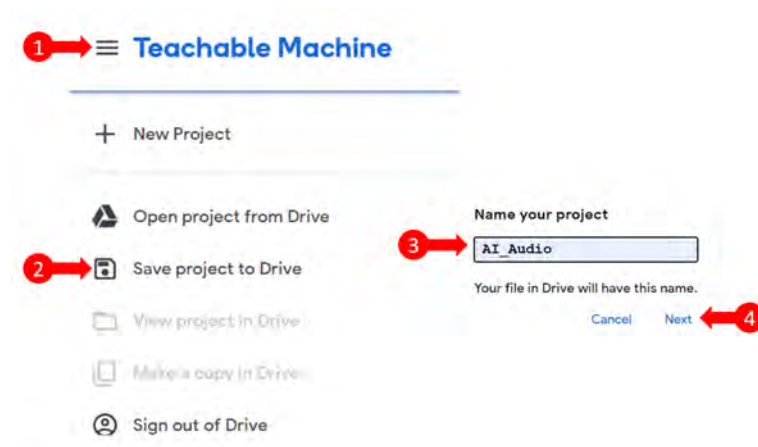


Hình 5.10: Hệ thống đã được tải thành công lên server Google



## 4 Lưu dự án vào Google drive

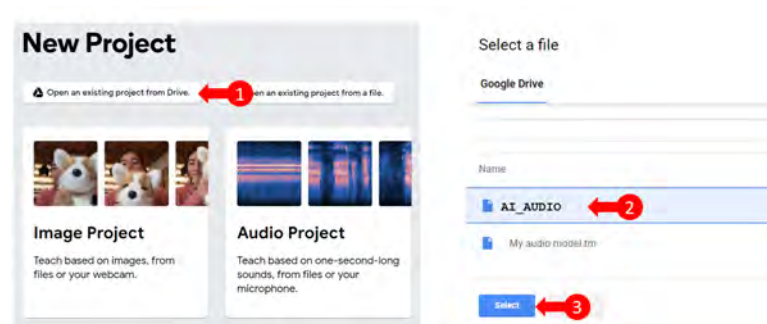
Với những hỗ trợ toàn diện từ phía Google, hệ thống AI của bạn có thể được lưu lại trực tuyến trên Google drive. Bằng các trở lại màn hình chính và làm theo 4 bước hướng dẫn ở hình bên dưới như: Chọn Save project to Drive, Đặt tên cho project và cuối cùng nhấn nút Next.



Hình 5.11: Lưu dự án vào Google drive

## 5 Mở dự án từ Google drive

Trong trường hợp muốn mở lại dự án cũ để cập nhật hoặc làm tiếp, từ trang chủ của Học Máy Google, chúng ta chọn **Open existing project from Drive** rồi lựa chọn trong danh sách các dự án đã lưu, như minh họa ở hình bên dưới.



Hình 5.12: Mở dự án từ Google drive

## 6 Câu hỏi ôn tập

1. Công cụ huấn luyện âm thanh có tên là gì?
  - A. Teachable Machine
  - B. Artificial Intelligence
  - C. Video Project
  - D. Audio Project
2. Đối tượng đầu tiên trong công cụ huấn luyện audio là gì?
  - A. Âm thanh nền (Background Noise)
  - B. Class 1
  - C. Class 2
  - D. Tất cả đều sai
3. Dữ liệu để huấn luyện cho hệ thống nhận dạng bằng ảnh có thể được lấy từ nguồn nào?
  - A. Trực tiếp từ micro của máy tính
  - B. Tải âm thanh từ máy tính lên
  - C. Ảnh từ Google Drive của người dùng
  - D. Tất cả các nguồn âm thanh trên
4. Trong các nguồn âm thanh dưới đây, nguồn âm nào là tốt nhất để huấn luyện hệ thống?
  - A. Micro của máy tính
  - B. Tải âm thanh từ máy tính lên
  - C. Ảnh từ Google Drive của người dùng
  - D. Không thể so sánh được
5. Đường dẫn trực tuyến của mô hình huấn luyện âm thanh hỗ trợ cho môi trường lập trình nào sau đây?
  - A. Scratch 2.0
  - B. Scratch 3.0
  - C. MakeCode
  - D. Tất cả đều đúng
6. Các từ lệnh trong hệ thống huấn luyện âm thanh có độ dài tốt nhất là:
  - A. 1 giây
  - B. 2 giây
  - C. 20 giây
  - D. Tất cả đều sai
7. Khi huấn luyện âm thanh nền, mặc định hệ thống cần bao nhiêu giây?
  - A. 1 giây
  - B. 2 giây
  - C. 20 giây
  - D. Tất cả đều sai

Đáp án

1. D 2. A 3. D 4. A 5. B 6. A 7. C

SCRATCH



## CHƯƠNG 6

---

### Tích hợp nhận dạng âm thanh

---



# 1 Giới thiệu

Cũng như dự án về Video, mô hình nhận dạng của Google cũng hỗ trợ cho chúng ta định dạng để tích hợp vào Scratch 3.0 Online. Với cách thức sử dụng hoàn toàn giống với video, nhưng việc tương tác sẽ dựa vào âm thanh.

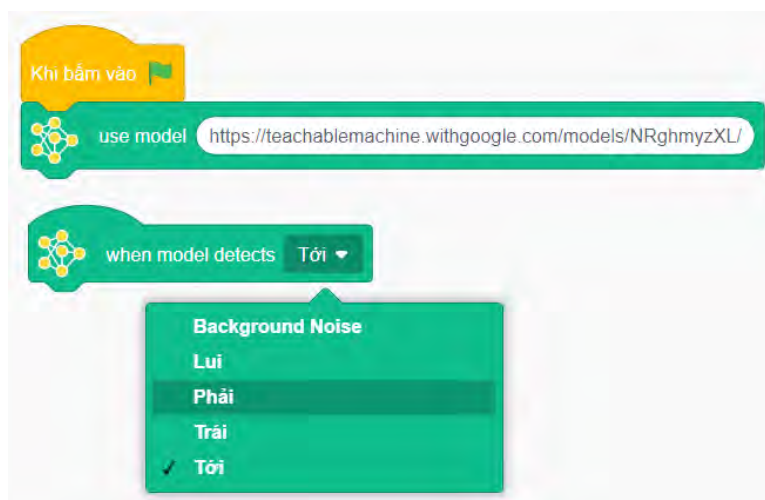
Trong bài này, chúng ta sẽ tích hợp mô hình nhận dạng đã được huấn luyện vào một chương trình nhỏ trên Scratch 3.0. Với dữ liệu ở bài trước, là các từ lệnh di chuyển, chúng ta sẽ điều khiển nhân vật trên sân khấu di chuyển theo từ lệnh đã huấn luyện.

Bên cạnh đó, chúng tôi cũng giới thiệu cho bạn đọc một công cụ rất mạnh mẽ và đang được hỗ trợ sẵn trên Scratch 3.0 Online, đó là Text Classification. Bước đầu, chúng ta sẽ sử dụng tính năng nhận diện giọng nói, còn được gọi là Speech To Text. Mặc dù ngôn ngữ đang hỗ trợ hiện tại là tiếng Anh, nhưng đây là mô hình đã được huấn luyện sẵn bởi Google. Đây là công nghệ được sử dụng trong tìm kiếm bằng giọng nói ở các thiết bị di động hay tivi thông minh. Các mục tiêu của bài hướng dẫn này như sau:

- Áp dụng mô hình huấn luyện giọng nói vào Scratch 3.0
- Hiện thực chương trình sử dụng giọng nói
- Sử dụng công cụ nhận dạng giọng nói Text Classification

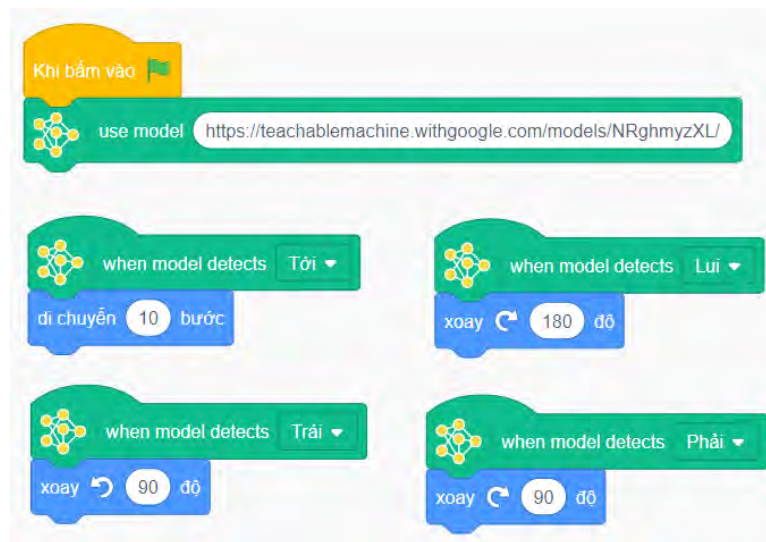
## 2 Nhận dạng âm thanh

Mặc dù nhóm lệnh liên quan đến trí tuệ nhân tạo được dịch là **Cảm biến Video**, nó hoàn toàn có thể sử dụng cho Audio với quy trình gần như tương tự. Chúng ta sẽ cho liên kết với mô hình bằng câu lệnh **use model** và sau đó tiến hành nhận dạng bằng câu lệnh **when model detects**, như minh họa ở hình bên dưới:



Hình 6.1: Liên kết với mô hình huấn luyện giọng nói

Chương trình đơn giản để minh họa việc điều khiển di chuyển bằng giọng nói có thể được hiện thực như sau:



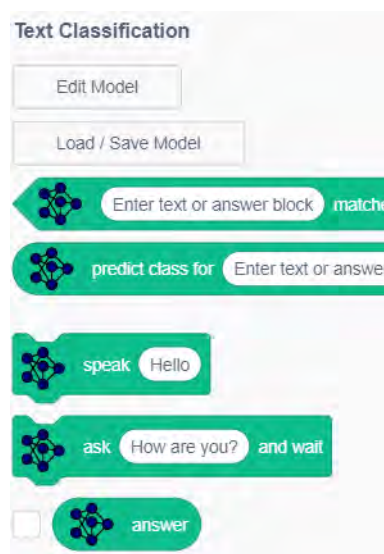
Hình 6.2: Điều khiển nhân vật bằng giọng nói

Cũng tương tự với dự án video, để tăng tính tin cậy của việc nhận dạng, bạn có thể kết hợp thêm thông tin liên quan đến xác suất (câu lệnh confident for). Phần này xem như là phần mở rộng để bạn đọc tự phát triển.

Tuy nhiên, các từ huấn luyện ở đây đa phần là từ lệnh, có nội dung khá ngắn. Trong trường hợp muốn dùng từ lệnh dài hơn, chúng ta cần phải sử dụng một kỹ thuật cao cấp hơn, được trình bày ở phần tiếp theo.

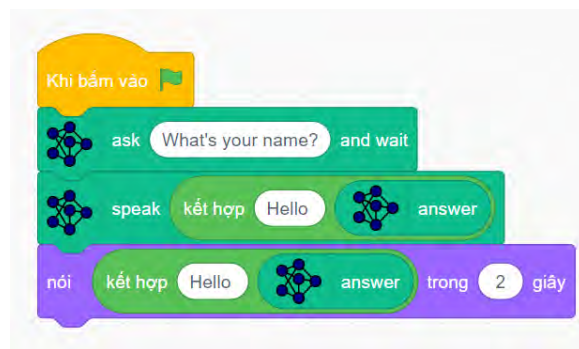
### 3 Text Classification

Đây là khối lệnh cao cấp cuối cùng được hỗ trợ sẵn trong môi trường lập trình Scratch 3.0 Online. Các câu lệnh trong nhóm này được trình bày như sau:



Hình 6.3: Các câu lệnh trong nhóm Text Classification

Trước khi đi sâu vào các câu lệnh của phần này, chúng ta có thể hiện thực một chương trình mẫu như sau:



*Hình 6.4: Chương trình đơn giản với giọng nói*

Chương trình này cũng giống với việc chúng ta chờ nhận người dùng nhập dữ liệu từ bàn phím. Tuy nhiên, ở đây là chờ người dùng nói với máy tính. Chương trình trên cũng sẽ trả lời lại bằng giọng nói cho chúng ta. Bạn đọc cần phải lưu ý, ngôn ngữ hỗ trợ hiện tại cho phần này là tiếng Anh mà thôi.

Với sự hỗ trợ thêm của giọng nói, bạn có thể làm những ứng dụng trắc nghiệm tiếng Anh bằng cách nhận dạng giọng nói trực tiếp từ người dùng, hoặc thậm chí bạn có thể hiện thực một Trợ lý ảo, sẽ được trình bày trong Phần 2 - Dự án của giáo trình này. Các câu lệnh khác trong phần này cũng sẽ được trình bày sau.

## 4 Câu hỏi ôn tập

1. Khối lệnh dùng để lập trình nhận dạng giọng nói đã được huấn luyện là gì?  
A. Cảm biến Audio  
B. Cảm biến Video  
C. Text Classification  
D. Tất cả đều sai
2. Để liên kết với mô hình huấn luyện audio bằng câu lệnh use model, thông tin gì sẽ được sử dụng?  
A. File âm thanh mẫu  
B. File javascript lưu trong máy tính  
C. Đường dẫn trực tuyến của mô hình  
D. Tất cả các thông tin trên
3. Ngôn ngữ hỗ trợ cho khối lệnh Cảm biến Video là gì?  
A. Tiếng Việt  
B. Tiếng Anh  
C. Cả 2 ngôn ngữ trên  
D. Ngôn ngữ nào cũng được, tùy việc huấn luyện
4. Khối nhận diện giọng nói trên Scratch 3.0 Online có tên là gì?  
A. Cảm biến Audio  
B. Cảm biến Video  
C. Text Classification  
D. Không hỗ trợ khối này
5. Ngôn ngữ hỗ trợ cho khối lệnh Text Classification là gì?  
A. Tiếng Việt  
B. Tiếng Anh  
C. Cả 2 ngôn ngữ trên  
D. Ngôn ngữ nào cũng được, tùy việc huấn luyện
6. Để có thể phát ra âm thanh từ một dòng chữ (text), câu lệnh nào sau đây sẽ được sử dụng?  
A. nói ... trong 2 giây  
B. nói ...  
C. speak  
D. answer
7. Thông tin của việc nhận diện giọng nói được lưu trong khối lệnh nào?  
A. Trả Lời  
B. answer  
C. speak  
D. Tất cả đều sai

Đáp án

1. B   2. C   3. D   4. C   5. B   6. C   7. B





# CHƯƠNG 7

---

SCRATCH



## Kết nối với Microbit

---



# 1 Giới thiệu

Một tính năng cải tiến và đã có sẵn trên môi trường lập trình Scratch 3.0 Online là khả năng kết nối với một mạch phần cứng đang rất thông dụng trong giáo dục hiện nay, mạch BBC Microbit. Hơn thế nữa, môi trường lập trình Scratch 3.0 Online còn hỗ trợ một số câu lệnh để có thể điều khiển một Robot sử dụng mạch Microbit làm trung tâm điều khiển.

Trong bài hướng dẫn này, chúng ta sẽ tập trung vào việc kết nối và lập trình truy xuất đơn giản giữa Scratch 3.0 Online và mạch Microbit. Các mục tiêu cần đạt được trong bài hướng dẫn này như sau:

- Nạp được phần mềm chạy nền cho mạch Microbit
- Kết nối giữa Scratch 3.0 Online và Microbit
- Lập trình nhập/ xuất đơn giản trên Microbit

## 2 Phần mềm chạy nền - Firmware

Để có thể kết nối giữa mạch Microbit và môi trường lập trình Scratch 3.0, chúng ta cần một phần mềm chạy nền trên mạch Microbit. Phần mềm này, thực chất là để mở một cổng kết nối thông qua Bluetooth và kết nối với máy tính của chúng ta.

Tuy nó là một phần mềm, các chức năng của nó chủ yếu liên quan đến thiết bị phần cứng bên dưới. Vai trò của phần mềm này, là để chuyển các lệnh từ máy tính và thực thi nó trên phần cứng, là mạch Microbit. Do đó, nó còn có một tên gọi chuyên ngành khác, là Firmware. Thực ra, rất khó định nghĩa ra một ranh giới rõ ràng giữa Firmware và Software (phần mềm). Thông thường, phần mềm mà chức năng liên quan nhiều tới phần cứng hoặc thiết bị, sẽ được gọi là Firmware, dịch nôm na là "phần dẻo".

Để nạp firmware cho mạch Microbit, chúng ta cần tải nó về từ đường dẫn sau đây:

[https://ubc.sgp1.cdn.digitaloceanspaces.com/DARIU/bbc\\_firmware.hex](https://ubc.sgp1.cdn.digitaloceanspaces.com/DARIU/bbc_firmware.hex)

Bạn đọc có thể thường xuyên kiểm tra các phiên bản mới của firmware từ đường dẫn trên. Tuy nhiên, trong bài tiếp theo, chúng tôi sẽ hướng dẫn bạn đọc tự điều chỉnh lại firmware cho ứng dụng của mình để có thêm các tính năng mới.

Sau khi tải firmware về dưới dạng 1 file hex, bạn có thể kết nối mạch Microbit với máy tính, như minh họa ở hình bên dưới:

Mạch Microbit sẽ được nhận dạng như một ổ USB bình thường, bạn chỉ cần chép file hex đã tải ở trên vào ổ đĩa **MICROBIT(F:)** là việc nạp firmware sẽ bắt đầu. Khi đèn nguồn ở mạch Microbit ngừng chớp tắt, là việc nạp đã thành công.

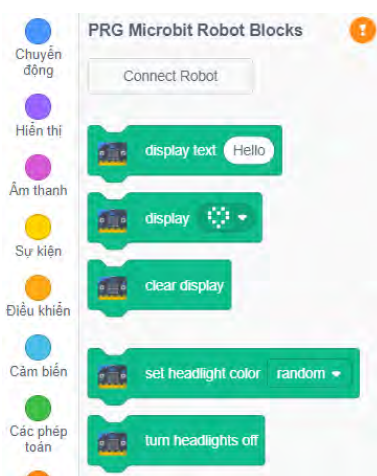


Hình 7.1: Kết nối mạch Microbit và Máy tính

Tiếp theo, chương trình bên trong mạch Microbit sẽ được thực thi. Bạn sẽ thấy **tên của mạch Microbit sẽ được chạy ngang qua màn hình hiển thị** của nó trước khi biểu tượng chữ S xuất hiện. Bạn hãy ghi nhớ lại tên này cho bước tiếp theo.

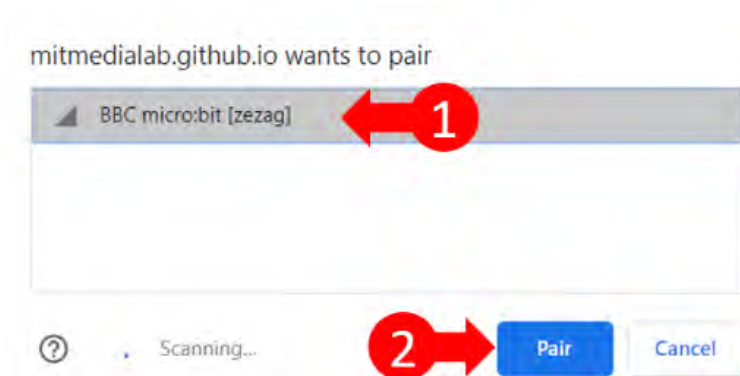
### 3 Kết nối Scratch 3.0 và Microbit

Các câu lệnh để lập trình cho Microbit được nằm ở nhóm **Microbit Robot Blocks**, như minh họa ở hình bên dưới:



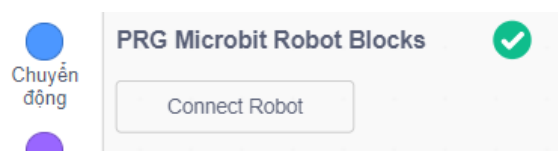
Hình 7.2: Các câu lệnh về Microbit trên Scratch 3.0

Thực ra các khối này được tạo ra để điều khiển một Robot. Tuy nhiên chúng ta sẽ làm việc với mạch Microbit, trước khi kết nối vào Robot. Ở phía đầu của nhóm lệnh, chúng ta nhấn vào nút **Connect Robot**, giao diện sau đây sẽ hiện ra:



Hình 7.3: Kết nối với Microbit

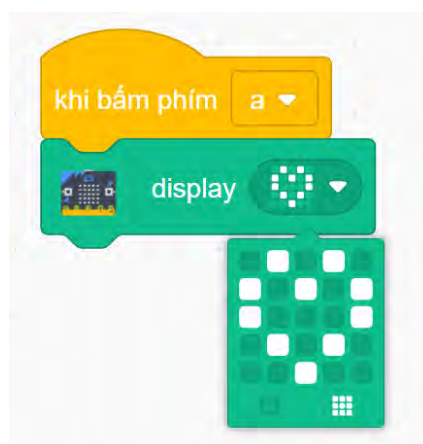
Máy tính sẽ hỏi bạn mở chức năng Bluetooth nếu nó chưa mở. Scratch 3.0 Online sẽ dùng kết nối này để liên kết với mạch Microbit. Bạn đọc hãy chọn cho đúng thiết bị của mình, trước khi nhấn nút **Pair**. Khi việc kết nối hoàn tất, biểu tượng màu xanh lá sẽ xuất hiện thay thế cho biểu tượng màu cam ở góc bên phải của khối lệnh Microbit Robot Blocks, như minh họa ở hình bên dưới:



Hình 7.4: Kết nối với Microbit thành công

## 4 Hiện thị trên Microbit

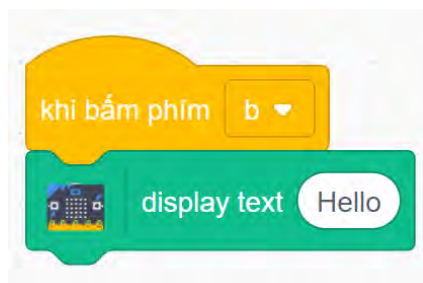
Trên Scratch 3.0 có 2 câu lệnh chính cho việc hiển thị. Chúng tôi gọi câu lệnh đầu tiên là câu lệnh icon, với chương trình minh họa bên dưới:



Hình 7.5: Hiện thị icon trên Microbit

Đây là một câu lệnh tùy chọn, để bạn đọc tùy chỉnh 25 đèn hiển thị trên Microbit và tạo ra hình ảnh mình mong muốn. Bạn đọc có thể sắp xếp nhiều câu lệnh này liên tục để tạo ra các hiệu ứng đẹp, mỗi khi nhấn phím A trên bàn phím máy tính chẳng hạn.

Câu lệnh thứ 2 về hiển thị, là dùng để hiển thị một chuỗi kí tự, như minh họa ở hình bên dưới:

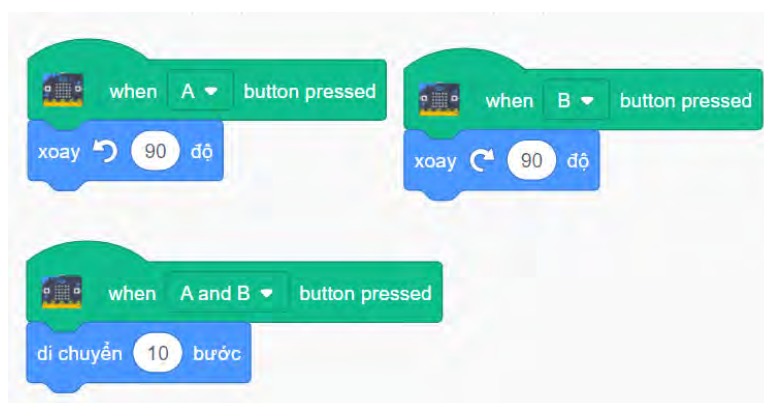


Hình 7.6: Hiển thị kí tự trên Microbit

Cần lưu ý là, chúng ta chỉ hiển thị được kí tự không dấu trên Microbit mà thôi. Khi có 1 kí tự, nó sẽ đứng yên trên màn hình. Từ 2 kí tự trở lên, dòng kí tự sẽ cuộn từ trái qua phải trên màn hình hiển thị.

## 5 Nút nhấn trên Microbit

Mạch Microbit hỗ trợ 2 nút nhấn là A và B. Trên môi trường lập trình Scratch 3.0, chúng ta sẽ nhận sự kiện mỗi khi nút được nhấn. Scratch 3.0 cũng hỗ trợ cho việc phát hiện 2 nút được nhấn cùng lúc. Một chương trình ví dụ để sử dụng chức năng nút nhấn như minh họa ở bên dưới:



Hình 7.7: Nút nhấn trên Microbit

Bạn đọc có thể sử dụng chương trình ở trên để thiết kế một tay game điều khiển từ xa cho nhân vật của chúng ta trên máy tính. Mạch Microbit có thể được cấp nguồn rời mà không cần phải kết nối trực tiếp với máy tính.

## 6 Câu hỏi ôn tập

1. Khối lệnh dùng để lập trình mạch Microbit có tên là gì?
  - A. BBC Microbit
  - B. BBC Robot
  - C. PRG Microbit Robot Blocks
  - D. Tất cả các khối trên
2. Mạch Microbit sau khi nạp firmware sẽ xuất hiện những thông tin gì?
  - A. Tên của mạch Microbit
  - B. Ký hiệu sáng đèn với ký tự S
  - C. Hiện thị ký hiệu Bluetooth B khi kết nối thành công
  - D. Tất cả các thông tin trên
3. Kết nối giữa Scratch 3.0 Online và mạch Microbit có tên là gì?
  - A. Kết nối Radio
  - B. Kết nối Wifi
  - C. Kết nối Bluetooth
  - D. Tất cả các nguồn âm thanh trên
4. Câu lệnh này sau đây dùng để hiển thị icon trên Microbit?
  - A. display
  - B. display icon
  - C. display text
  - D. clear display
5. Câu lệnh nào được hiển thị ký tự trên Microbit?
  - A. display
  - B. display icon
  - C. display text
  - D. clear display
6. Câu lệnh nào để xử lý sự kiện như nhấn nút A trên mạch Microbit?
  - A. Khi nhấn phím A
  - B. When A button pressed
  - C. When A button clicked
  - D. Tất cả đều sai
7. Câu lệnh nào để xử lý sự kiện như nhấn cả 2 nút A và B trên mạch Microbit?
  - A. Khi nhấn phím A và B
  - B. When A and B button pressed
  - C. When A or B button pressed
  - D. When neither A nor B button pressed

Đáp án

1. C 2. D 3. C 4. A 5. C 6. B 7. B



SCRATCH



## CHƯƠNG 8

---

# Tùy chỉnh firmware cho Microbit

---



# 1 Giới thiệu

Trong một số trường hợp, chúng ta cần sử dụng nhiều tương tác hơn từ mạch Microbit, thay vì chỉ có nút nhấn A và B như hiện tại. Chẳng hạn như, bạn muốn thiết kế một tay cầm điều khiển từ xa để điều khiển nhân vật trên Scratch bằng cách nghiêng trái hoặc nghiêng phải mạch Microbit chẳng hạn. Với phiên bản Scratch 3.0 Online hiện tại, chức năng này chưa được hỗ trợ.

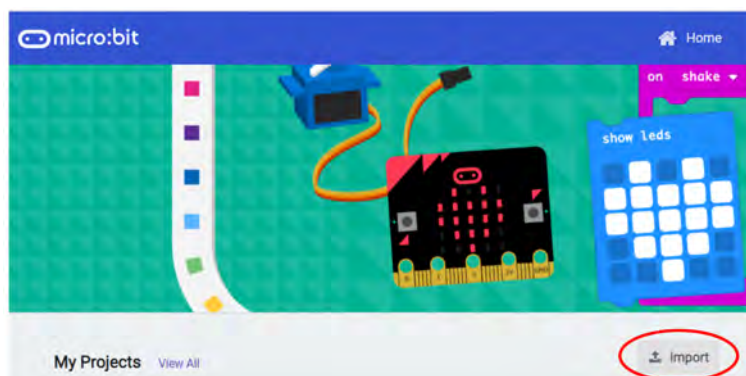
Trong bài hướng dẫn này, chúng ta sẽ hiệu chỉnh lại firmware cho mạch Microbit để có thể phát hiện nó đang nghiêng trái hay nghiêng phải. Chúng ta sẽ tận dụng những gì đang có trên môi trường lập trình Scratch 3.0 để hiện thực điều này. Vì đây là thông tin sẽ truyền từ mạch Microbit lên môi trường lập trình Scratch 3.0, chúng ta cần phải tìm một câu lệnh có chiều dữ liệu như vậy.

Trong khối lệnh **PRG Microbit Robot Blocks**, câu lệnh cuối cùng là **read distance** là thỏa mãn yêu cầu này. Mặc dù khối lệnh này được thiết kế để đo khoảng cách từ Robot tới vật cản, chúng ta có thể tận dụng để truyền thông tin khác, trong trường hợp này là việc nghiêng mạch Microbit qua trái hay phải. Các mục tiêu trong bài hướng dẫn này như sau:

- Hiệu chỉnh lại firmware cho mạch Microbit trên MakeCode
- Gửi thông tin khi mạch nghiêng trái và nghiêng phải
- Áp dụng vào việc điều khiển nhân vật trên Scratch 3.0 Online

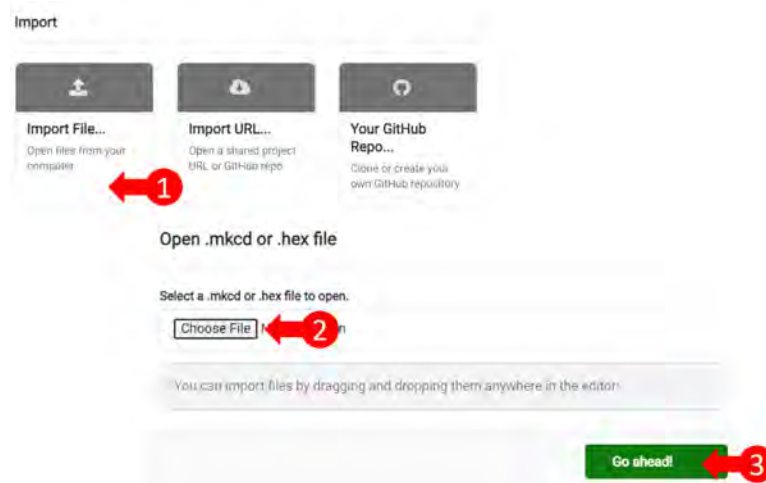
## 2 Môi trường lập trình MakeCode

Đây là môi trường thuận tiện nhất để hiệu chỉnh firmware cho mạch Microbit. Nếu như bạn đọc đã có kinh nghiệm lập trình Microbit từ trước, phần này sẽ không có gì phức tạp. Chúng ta sẽ sử dụng môi trường MakeCode trực tuyến, bằng cách vào đường dẫn <https://makecode.microbit.org/>. Giao diện sau đây sẽ hiện ra



Hình 8.1: Môi trường lập trình MakeCode online

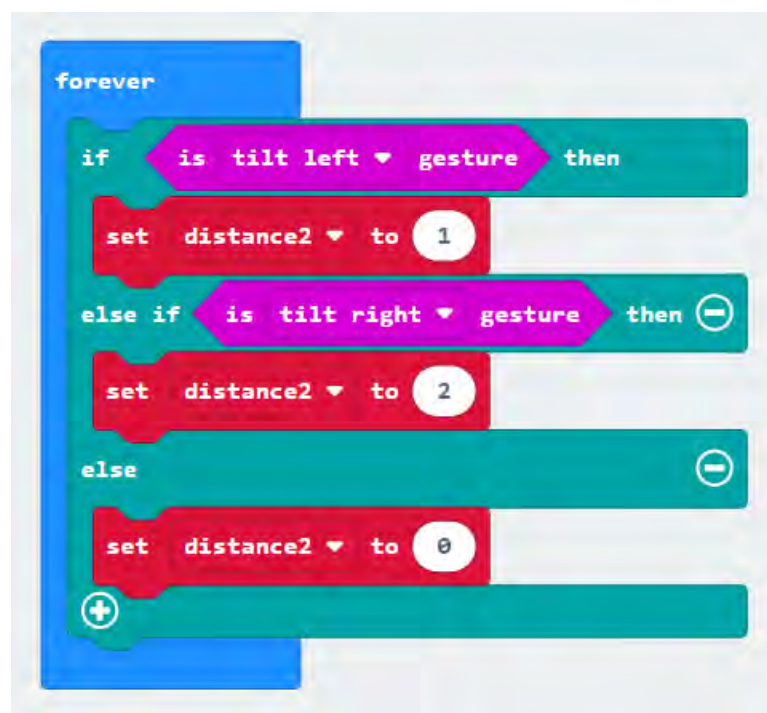
Tuy nhiên, thay vì tạo mới một ứng dụng, chúng ta sẽ nhấn vào nút Import để mở lại file firmware đã tải về trong bài trước. Giao diện sau đây sẽ hiện ra:



Hình 8.2: Mở file firmware (.hex) đã tải ở bài trước

### 3 Thêm sự kiện hành vi

Giá trị trả về của khối lệnh **read distance** là giá trị của **distance2** trong firmware của Microbit. Do đó, chúng ta sẽ hiện thực thêm một khối forever để thay đổi giá trị của biến này, như gợi ý như sau:



Hình 8.3: Thay đổi biến distance2 trong firmware

Theo chương trình ở trên, mỗi khi nghiêng mạch sang bên trái, chúng ta sẽ nhận được giá trị là 1, bên phải sẽ là 2. Các trường hợp còn lại, chúng ta sẽ gửi 0 lên phần mềm Scratch 3.0 Online.

Chúng ta sẽ nạp firmware mới lại cho mạch Microbit, bằng cách nhấn vào nút **Download** và lưu file hex mới vào mạch Microbit, đang nhận như 1 USB trong

máy tính.

Chúng ta sẽ kiểm tra việc hiện thực có đúng hay không, bằng cách kết nối lại mạch Microbit và Scratch 3.0, sau đó bật biến **read distance**, như minh họa ở hình bên dưới:

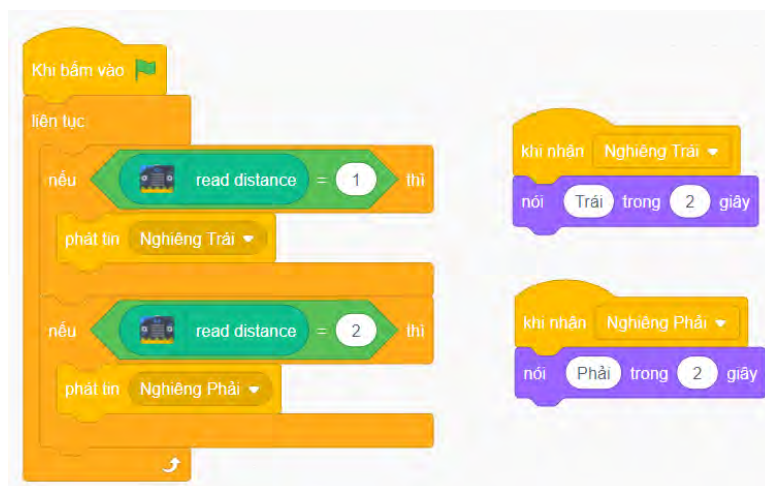


Hình 8.4: Kiểm tra biến distance với firmware mới

Khi nghiêng mạch Microbit lần lượt sang trái và phải, chúng ta sẽ nhận được giá trị là 1 và 2, đúng với mục tiêu lập trình. Tuy nhiên, **chúng ta sẽ không nhận được giá trị là 0 mà sẽ là -1**. Bạn đọc cần lưu ý, việc này không phải do chúng ta lập trình sai. Lý do là phần mềm Scratch 3.0 cũng có xử lý cho trường hợp giá trị là 0, nó tự đổi thành -1, với ý nghĩa là giá trị không hợp lệ.

## 4 Hiện thực sự kiện trên Scratch 3.0

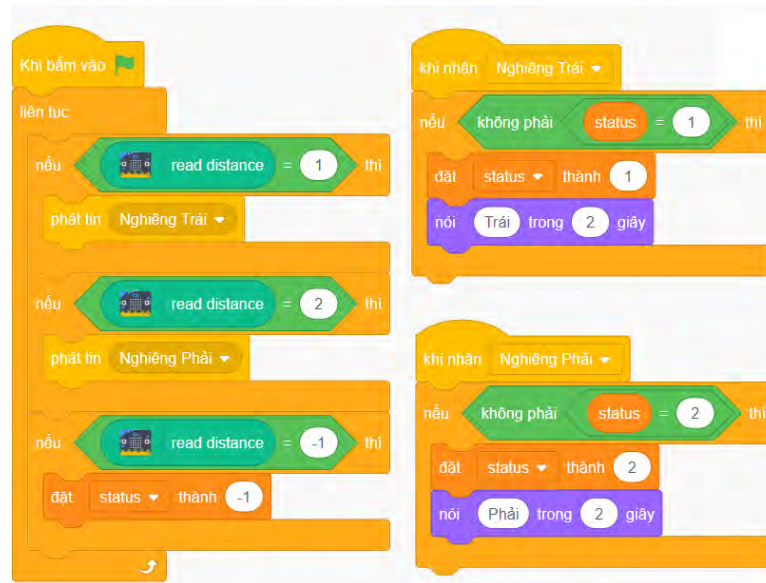
Khi đã nhận được đúng giá trị từ biến read distance, chúng ta sẽ hiện thực thêm lệnh trên Scratch 3.0, để tạo ra những khối tương tự như khi nhấn nút A hoặc B trên mạch Microbit. Kỹ thuật mà chúng ta sẽ sử dụng là câu lệnh tin nhắn. Bước đầu, chúng ta hiện thực chương trình như sau:



Hình 8.5: Dùng kỹ thuật tin nhắn để nhận diện thông tin từ Microbit

Chúng ta bắt buộc phải dùng một vòng lặp vô tận để kiểm tra giá trị gửi lên từ mạch Microbit. Với chương trình như trên, mỗi khi mạch Microbit nghiêng sang bên trái, thông tin này sẽ được gửi liên tục. Điều này được minh họa rằng đối tượng sẽ nói "Trái" mãi mãi mà không bao giờ dừng.

Tùy vào mục đích của việc thiết kế chương trình mà bạn đọc cần phải chủ động. Nếu như muốn khi nghiêng trái, chúng ta chỉ xử lý 1 lần rồi thôi thì chương trình sẽ phải hiện thực thêm như sau:



Hình 8.6: Thay đổi chức năng của câu lệnh xử lý sự kiện

Ý tưởng ở đây là khai báo thêm 1 biến số có tên là **status** để lưu trạng thái trước đó của câu lệnh nhận tin nhắn. Khi có sự thay đổi giá trị của status thì chúng ta mới thực hiện lệnh. Cần lưu ý ở đây là khi nhận giá trị -1, chúng ta phải cho biến status thành một giá trị khác, để đảm bảo khi nghiêng chương trình vẫn xử lý.

Cuối cùng, bạn đọc có thể tự hiện thực thêm nhiều thông tin khác có thể gửi lên từ mạch Microbit. Thậm chí, bạn có thể kiểm tra nhiệt độ của mạch Microbit và gửi thông tin cảnh báo lên Scratch 3.0 để phát ra báo động trên máy tính.

## 5 Câu hỏi ôn tập

1. Để gửi thêm thông tin từ mạch Microbit lên Scratch 3.0, chúng ta đang dựa trên khối lệnh nào?  
A. read distance  
B. read sensor  
C. read buttons  
D. Tất cả các khối trên
2. Để hiện thực thêm sự kiện nghiêng trái hoặc phải, khối lệnh nào sau đây đã được sử dụng trên firmware của Microbit?  
A. on start  
B. forever  
C. on button A pressed  
D. Tất cả các thông tin trên
3. Khi distance2 của Microbit là 0, giá trị nhận được trên Scratch 3.0 Online là bao nhiêu?  
A. 0  
B. -1  
C. Báo lỗi  
D. Tất cả đều sai
4. Việc kiểm tra biến read distance trên Scratch 3.0 được hiện thực trong khối lệnh nào?  
A. Sự kiện lá cờ  
B. Lặp 10 lần  
C. Lặp mãi mãi  
D. Tất cả đều sai
5. Khi mạch Microbit nghiêng sang trái, giá trị 1 được gửi bao nhiêu lần?  
A. Chỉ 1 lần duy nhất  
B. Gửi mãi mãi  
C. Tùy vào việc hiện thực firmware, trong bài này là gửi mãi mãi  
D. Tất cả đều đúng
6. Dựa vào biến read distance, chúng ta có thể tạo ra bao nhiêu sự kiện?  
A. Chỉ 2 sự kiện  
B. Chỉ 11 sự kiện  
C. Rất nhiều sự kiện  
D. Tất cả đều sai
7. Kỹ thuật được sử dụng để tạo ra một sự kiện như nút nhấn trên Scratch 3.0 là gì?  
A. Sử dụng câu lệnh tin nhắn  
B. Sử dụng câu lệnh sự kiện  
C. Sử dụng câu lệnh điều khiển  
D. Sử dụng câu lệnh lặp

Đáp án

1. A   2. B   3. B   4. C   5. C   6. C   7. A



SCRATCH



## CHƯƠNG 9

---

### Điều khiển Robot Microbit

---





# 1 Giới thiệu

Với sự phát triển của khoa học công nghệ ngày nay, các thành tựu của Robot đang được đưa dần vào cuộc sống. Với sự hỗ trợ tuyệt vời của trí tuệ nhân tạo, các hành vi của Robot ngày càng tiến gần đến mức tự hành (tự vận động) hơn. Trên Scratch 3.0 Online, nhóm lệnh PRG Microbit Robot Blocks dùng để minh họa những ý tưởng như vậy.



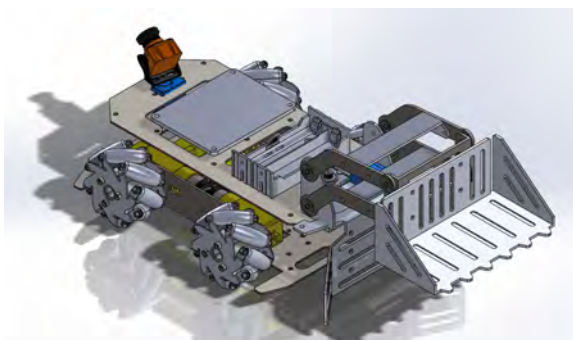
*Hình 9.1: Robot mặc định trên Scratch 3.0 Online*

Robot được đề xuất bởi môi trường Scratch 3.0 Online có tên là CuteRobot, như trình bày ở Hình 9.1. Đây là một robot nhỏ có thể cầm trên lòng bàn tay. Tuy nhiên, với việc có thể tùy chỉnh firmware trong mạch Microbit, chúng ta có thể rất chủ động trong việc lựa chọn Robot. Tất cả những gì cần làm là hiệu chỉnh làm các hàm dùng để điều khiển động cơ cho tương thích mà thôi. Các mục tiêu chính trong bài hướng dẫn này như sau:

- Sử dụng Robot 4 bánh Mecanum
- Nạp firmware cho Robot
- Lập trình điều khiển Robot di chuyển
- Đọc dữ liệu từ cảm biến khoảng cách của Robot

## 2 Nạp firmware cho Robot

Robot mà chúng ta sẽ sử dụng trong bài hướng dẫn này có đến 4 động cơ riêng biệt, và sử dụng 4 bánh cũng đặc biệt, có tên mà Mecanum, như hình vẽ bên dưới:



Hình 9.2: Robot 4 bánh Mecanum

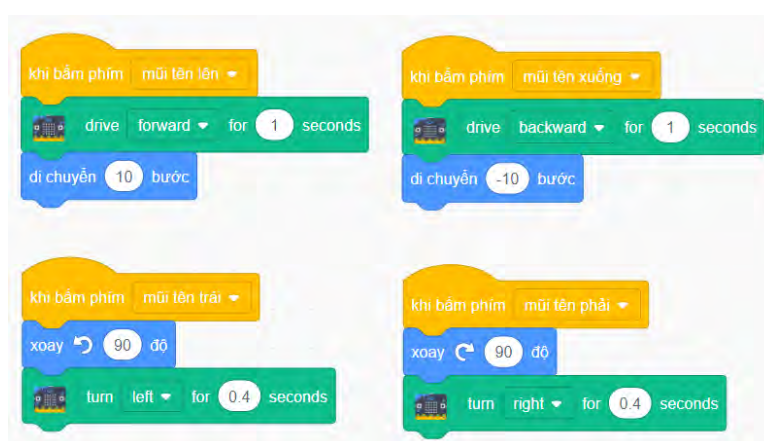
Mỗi bánh xe của Robot sẽ tạo ra một lực phát động 45 độ. Do đó, robot có thể di chuyển ngang hay xéo 45 độ tùy vào việc điều khiển. Với những đặc thù như vậy, firmware cho Robot đã được chúng tôi thay đổi lại, bạn đọc có thể tải từ đường dẫn sau đây:

[https://ubc.sgp1.cdn.digitaloceanspaces.com/DARIU/mecanum\\_robot.hex](https://ubc.sgp1.cdn.digitaloceanspaces.com/DARIU/mecanum_robot.hex)

Tuy nhiên, bạn đọc cũng hoàn toàn có thể tự thay đổi firmware này để tương thích với Robot của mình.

### 3 Điều khiển Robot di chuyển

Chúng ta sẽ viết một chương trình ngắn để điều khiển Robot di chuyển bằng 4 phím mũi tên trên bàn phím. Chương trình gợi ý như sau:



Hình 9.3: Chương trình di chuyển đơn giản cho Robot

Với chương trình gợi ý ở trên, bạn đọc cần hiệu chỉnh lại thời gian quay trái và quay phải cho phù hợp với cơ khí của Robot. Để robot có thể quay chính xác 90 độ, cần phải có những kỹ thuật khó hơn và phải can thiệp tới firmware của Robot. Trong chương trình ở trên, chúng ta cũng sẽ cho nhân vật Robot trên Scratch di chuyển theo Robot thật.

## 4 Đọc dữ liệu từ cảm biến

Hiện tại, firmware của Robot hỗ trợ 2 loại cảm biến khác nhau, là cảm biến khoảng cách và cảm biến dò theo đường kẻ.

### 4.1 Cảm biến khoảng cách

Cảm biến này được dựa trên thiết bị HC04, một cảm biến siêu âm rất thông dụng trên thị trường. Nhờ cảm biến này, chúng ta sẽ có thông tin về khoảng cách giữa Robot và vật cản (tính theo centimet).

Cảm biến này được thường xuyên cập nhật trong khối lệnh **read distance**, mà chúng ta có thể sử dụng nó như một biến số. Để kiểm tra giá trị của cảm biến này, chúng ta đơn giản là kích hoạt nó lên, như minh họa bên dưới:



Hình 9.4: Kiểm tra biến distance của Robot

### 4.2 Cảm biến đường kẻ

Cảm biến này được lưu trong khối lệnh điều kiện line detected on, như minh họa ở hình bên dưới:



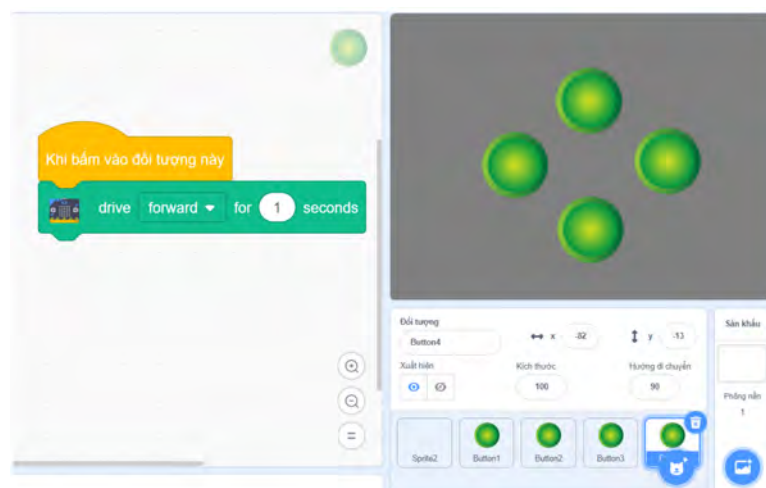
Hình 9.5: Cảm biến đường kẻ trên Robot

Theo thứ tự của câu lệnh ở trên, Robot đang bị lệch phải, lệch trái, hoàn toàn không có cảm biến nào nằm trong đường kẻ và 2 cảm biến đều nằm trong đường kẻ. Khối lệnh này cung cấp thông tin quan trọng để hiện thực việc di chuyển của Robot theo đường kẻ, sẽ được trình bày kỹ hơn ở bài sau.

Các câu lệnh liên quan đến hiển thị đèn (**set headlight color**) và phát nhạc trên Robot (**play song**) bạn đọc có thể chủ động tự tìm hiểu thêm. Hai câu lệnh này cũng là những công cụ tốt để tìm lỗi khi lập trình trên Robot.

## 5 Scratch 3.0 trên di động

Một điều thú vị là bạn đọc cũng có thể dùng điện thoại di động, truy cập vào trang web Scratch 3.0 Online **bằng trình duyệt** và thiết kế một giao diện lập trình trên đó để điều khiển từ xa Robot. Với chương trình trên điện thoại, bạn có thể thiết kế 4 nút tượng trưng cho 4 chiều di chuyển. Sau đó, hiện thực chương trình khi nhấn vào từng nút, như gợi ý ở hình bên dưới:



Hình 9.6: Chương trình di chuyển cho Robot trên di động

Chương trình trên hoàn toàn giống với một dự án trên Scratch. Bạn đọc có thể làm cho nó trở nên hấp dẫn hơn bằng cách thêm câu lệnh về âm thanh hay hiệu ứng hình ảnh, mỗi khi nhấn vào nút điều khiển Robot.

Lưu ý: Điện thoại của bạn phải hỗ trợ chuẩn kết nối Bluetooth 4.0 và phải vào trang lập trình online bằng trình duyệt web Chrome mới có thể sử dụng đầy đủ các tính năng nhất.

## 6 Câu hỏi ôn tập

- Scratch 3.0 có thể kết nối được với Robot nào?
  - Chỉ robot CuteRobot
  - Chỉ robot 2 bánh
  - Chỉ robot mecanum
  - Tất cả Robot Microbit có thể tùy chỉnh firmware
- Khi dùng điện thoại để lập trình Scratch 3.0 Online, điện thoại cần có chức năng gì?
  - Kết nối mạng
  - Kết nối Bluetooth 4.0
  - Hỗ trợ trình duyệt web
  - Tất cả các tính năng trên
- Cảm biến khoảng cách giữa Robot và vật cản được lưu trong biến nào?
  - read distance
  - line detected on
  - Cả 2 khối trên
  - Tất cả đều sai
- Cảm biến dò theo đường kẻ được lưu vào khối nào?
  - read distance
  - line detected on
  - Cả 2 khối trên
  - Tất cả đều sai
- Khi câu lệnh line detected on right side trả về đúng, nghĩa là:
  - Robot lệch phải
  - Robot lệch trái
  - Robot rẽ phải
  - Robot rẽ trái
- Câu lệnh nào sau đây dùng để phát nhạc trên Robot?
  - phát âm thanh
  - bắt đầu âm thanh
  - play song
  - Tất cả đều sai
- Câu lệnh nào sau đây dùng để phát sáng đèn trên robot?
  - display
  - display text
  - set headlight color
  - Tất cả các câu lệnh trên

Đáp án

1. D 2. D 3. A 4. B 5. A 6. C 7. C

# CHƯƠNG 10

---

SCRATCH



## Hiện thực AGV Robot

---



# 1 Giới thiệu

AGV là tên viết tắt của Autonomous Guided Vehicles - đây là công nghệ xe tự hành có hướng dẫn được tự động hóa một cách linh hoạt, chính xác. Tính tự động hóa được lập trình đồng nghĩa với việc bộ xe tự hành và robot phải dựa vào chính khả năng của chúng để xuất ra những dữ liệu vận hành có ích từ bộ phận cảm biến, từ đó đưa ra quyết định thích hợp. Để đi theo xu hướng sản xuất tự động hóa, các hệ thống phân phối trong các nhà máy dưới dạng xe tự hành có hướng dẫn - AGV đã đóng một vai trò vô cùng quan trọng trong tất cả các đợt giao hàng nguyên liệu, sản xuất các mặt hàng với chất lượng cao.

Thực ra, Robot AGV đã có từ lâu, trong thời đại của cuộc cách mạng lần thứ 3 về tự động hóa. Nhưng khi có cuộc cách mạng khoa học công nghệ lần thứ 4, vai trò của nó càng trở nên hữu dụng. Với các công nghệ mới về Internet, việc điều khiển hoặc giám sát từ xa Robot AGV đã trở nên rất phổ biến.

Trong bài hướng dẫn này, chúng ta sẽ sử dụng môi trường lập trình Scratch 3.0 On-line để điều khiển một Robot di chuyển theo đường kẻ. Tận dụng cảm biến đang có sẵn trên Robot, chương trình sẽ phát hiện được Robot đang ở trạng thái lệch sang bên nào và ra lệnh để Robot bám vào đường kẻ.

Với môi trường lập trình hiện tại, không có câu lệnh nào để Robot có thể di chuyển bám theo đường kẻ. Do đó, chúng ta cần phải hiệu chỉnh lại firmware cho Robot. Dựa vào trạng thái của cảm biến, chúng ta sẽ ra lệnh cho Robot di chuyển. Các mục tiêu trong bài hướng dẫn này như sau:

- Hiệu chỉnh firmware cho Robot
- Ra lệnh cho Robot di chuyển bám theo đường kẻ
- Thử nghiệm và tối ưu việc di chuyển của Robot

## 2 Thay đổi firmware cho Robot

Như đã trình bày ở trên, chúng ta cần một câu lệnh để thông báo cho Robot là nó đang bị lệch. Có tất cả 4 trạng thái khác nhau của cảm biến dò theo đường kẻ, nên câu lệnh của chúng ta cần có tối thiểu 4 trạng thái khác nhau.

Với yêu cầu như trên, 2 câu lệnh **set headlight color** và **play song** đều thỏa mãn, do mỗi câu lệnh đều có nhiều hơn 4 tùy chọn. Trong bài hướng dẫn này, chúng tôi sẽ chọn thay đổi vào câu lệnh **set headlight color**. Ý tưởng ở đây là không chỉ thay đổi màu đỏ, chúng ta còn tận dụng điều này để thông báo rằng Robot đang lệch trái và cần phải di chuyển về hướng bên phải.

Hàm mà chúng ta sẽ thay đổi là **SevenColorLED2** trong firmware của Microbit. Bạn vào đường dẫn <https://makecode.microbit.org/> và chọn tiếp vào **Import**. Sau đó, chọn đường dẫn tới file firmware dành cho Robot đã tải ở bài trước.



Trước khi bắt đầu chỉnh sửa firmware, chúng ta cần phải nắm nguyên tắc giao tiếp giữa hàm set headlight color trên Scratch 3.0 và hàm SevenColorLED2 trên Robot: Hàm set headlight color sẽ gửi một kí tự cho hàm SevenColorLED2, như thống kê ở bảng bên dưới:

Bảng 10.1: Giao tiếp giữa Scratch 3.0 và Robot

set headlight color	SevenColorLED2
red	G
blue	H
cyan	I
green	J
yellow	K
mageta	L
white	M
random	N

Chúng ta sẽ dùng 2 giá trị đầu tiên để báo hiệu cho Robot bắt đầu dò theo đường kẻ là đèn màu xanh (J) và dừng robot với đèn màu đỏ (G). Lệnh "N" cũng sẽ được dùng để dừng Robot. Chương trình gợi ý cho chúng ta như sau:



Hình 10.1: Chương trình bám theo đường kẻ

Theo chương trình gợi ý ở trên, câu lệnh then chốt để điều khiển tốc độ 4 động cơ là **serial write line**. Câu lệnh này có tham số là **"s:motor1:motor2:motor3:motor4"**. Trong đó, **s**: là từ lệnh cho Robot trong trường hợp muốn điều khiển động cơ. Bốn tham số tiếp theo là các tốc độ tương ứng với 4 động cơ, với ý nghĩa **0 là quay lùi cực đại, 200 là quay tới cực đại và 100 là điểm chính giữa, Robot sẽ dừng**. Tốc độ cho một động cơ là chuỗi có 3 kí tự. Do đó, nếu chúng ta muốn truyền tốc độ là 10, thì phải ghi rõ là 010.

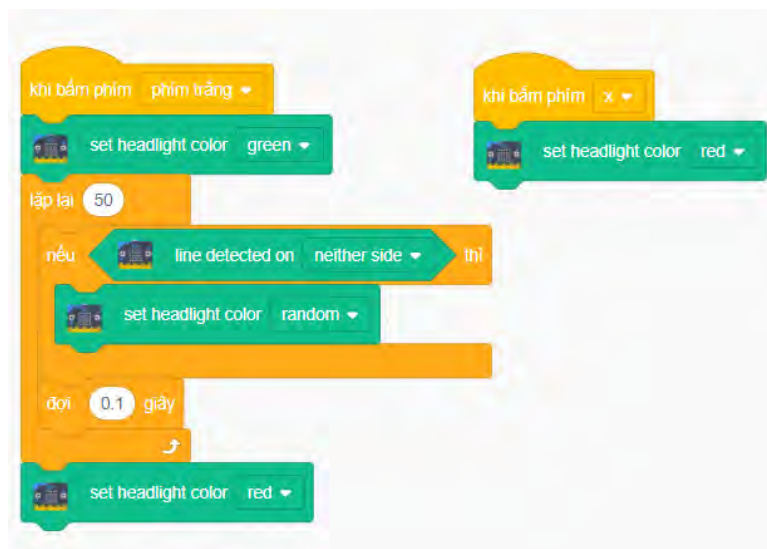
Mặc dù việc lập trình không còn thân thiện như trước nữa, nhưng đây là điều rất bình thường cho việc viết chương trình firmware. Nếu bạn đọc đã từng nghe về việc phát triển driver cho thiết bị (device driver), thì chúng ta cũng đang làm các khái niệm tương tự.

Khi Robot lệch trái, các động cơ bên trái cần phải quay nhanh hơn bên phải. Trong ví dụ ở trên, bên trái sẽ quay với tốc độ 180 và bánh phải là 120. Tùy vào cơ khí của

Robot mà bạn phải cân chỉnh lại các thông số này cho phù hợp. Trong trường hợp nhận N, Robot sẽ đứng im bằng cách gọi hàm **stop\_robot**.

### 3 Di chuyển theo đường kẻ

Tiếp theo, chúng ta sẽ lập trình trên Scratch 3.0 Online để gửi lệnh điều khiển cho Robot di chuyển. Rõ ràng, việc kiểm tra trạng thái của Robot sẽ phải diễn ra liên tục trong suốt quá trình di chuyển. Chương trình gợi ý của chúng tôi như sau:



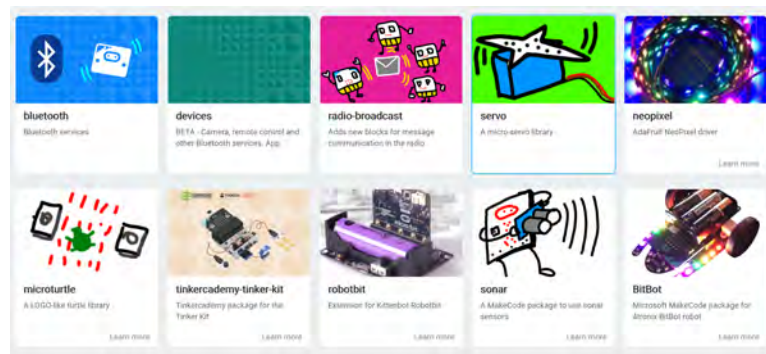
Hình 10.2: Chương trình điều khiển từ Scratch 3.0

Với chương trình ở trên, khi nhấn phím khoảng trắng, Robot sẽ di chuyển theo đường kẻ trong 5 giây. Tuy nhiên, nếu như nó bị lạc (2 cảm biến đều không nằm trên đường kẻ) Robot sẽ dừng lại. Việc kiểm tra 2 cảm biến này cần phải được thực thi thường xuyên, nên chúng ta sẽ dùng khối lệnh lặp 50 lần để hiện thực. Bên cạnh đó, nếu chúng ta nhấn phím X trên bàn phím, Robot cũng dừng. Chức năng này rất cần thiết để hạn chế Robot di chuyển sai.

### 4 Điều khiển cánh tay

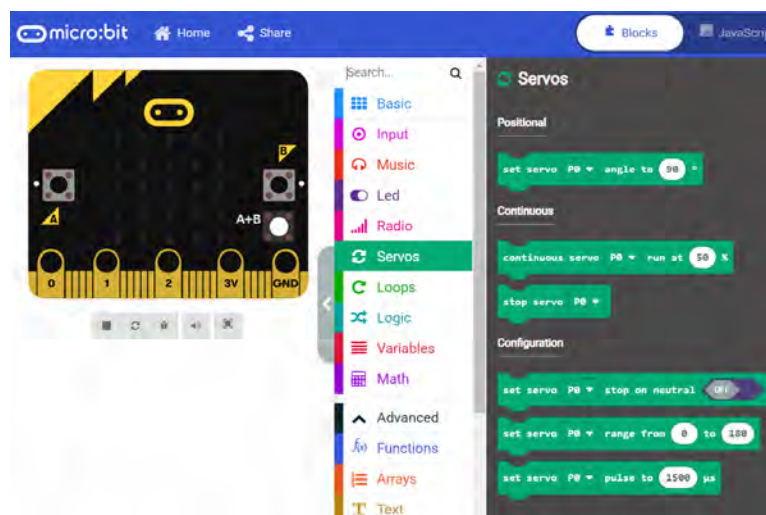
Việc điều khiển cánh tay cũng sẽ được hiện thực tương tự như trên, nhưng nó đơn giản hơn. Một trong những lý do khiến cho điều này dễ dàng là cánh tay thường sẽ sử dụng động cơ RC Servo. Động cơ này được điều khiển bằng góc xoay nên rất dễ dàng để lập trình.

Trước tiên, chúng ta cần phải thêm thư viện cần thiết cho việc lập trình. Bằng cách chọn vào khối lệnh dưới cùng, **Advanced**, rồi chọn tiếp vào **Extensions**, giao diện sau đây sẽ hiện ra:



Hình 10.3: Các thư viện mở rộng trên MakeCode

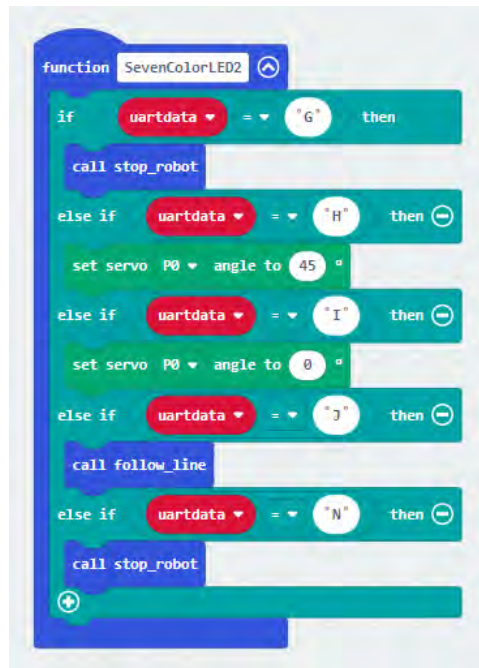
Chúng ta có thể dễ dàng tìm thấy thư viện có tên là **servo** cùng biểu tượng một động cơ. Bạn hãy nhấn vào đó, đợi 1 lúc và sau đó, chúng ta sẽ thấy 1 nhóm lệnh mới, có tên là **Servos** được thêm vào cửa sổ lập trình MakeCode, như minh họa bên dưới



Hình 10.4: Các câu lệnh trong nhóm Servos

Động cơ RC servo khá đặc biệt, vì nó ko xoay tròn 360 độ như những động cơ khác. Thay vào đó, nó sẽ giữ vị trí ở những góc cố định mà người lập trình ra lệnh cho nó. Như ở môi trường MakeCode, câu lệnh đầu tiên cho phép chúng ta chỉnh động cơ ở các vị trí từ 0 đến 180 độ. Do đó, đối với người mới bắt đầu, điều này thuận tiện cho chúng ta rất nhiều ở việc lập trình điều khiển vị trí của động cơ.

Chúng ta có thể dùng đèn màu xanh dương (blue - H) để nâng cánh tay lên còn đèn màu xanh lơ (cyan - I) để hạ cánh tay xuống. Với giả định trạng thái cơ khí của góc 45 độ là nâng cánh tay và 0 độ là hạ cánh tay, chương trình firmware gợi ý của chúng ta như sau:



Hình 10.5: Chương trình điều khiển cánh tay servo

## 5 Câu hỏi ôn tập

- AGV Robot có những đặc điểm nào sau đây?
  - Robot tự động hóa
  - Có cảm biến dò theo đường kẻ
  - Bắt đầu phổ biến từ cách mạng lần thứ 3
  - Tất cả đều đúng
- Câu lệnh nào trên Scratch 3.0 Online có thể được dùng để điều khiển hoạt động của Robot tự hành?
  - set headlight color
  - play song
  - Cả 2 câu lệnh trên
  - Scratch 3.0 không hỗ trợ
- Khi thực thi câu lệnh set headlight color red, firmware trên Robot sẽ nhận được kí tự gì?
  - R
  - G
  - D
  - Tất cả đều sai
- Khi thực thi câu lệnh set headlight color blue, firmware trên Robot sẽ nhận được kí tự gì?
  - B
  - H
  - G
  - Tất cả đều sai
- Để robot di chuyển với tốc độ tối đa, câu lệnh nào sau đây là đúng?

- A. serial write line "s:000:000:000:000"
  - B. serial write line "s:100:100:100:100"
  - C. serial write line "s:180:180:180:180"
  - D. Tất cả đều sai
6. Thư viện mở rộng để điều khiển động cơ servo có tên là gì ?
- A. motor.
  - B. servo.
  - C. Hai câu trên đều đúng.
  - D. Hai câu trên đều sai.
7. Các lợi thế của động cơ RC servo so với các động cơ khác là gì?
- A. Dễ điều khiển nhờ thư viện có sẵn
  - B. Dễ dàng kết nối với mạch MicroBit
  - C. Có mô phỏng trực quan trên MakeCode
  - D. Tất cả đều đúng

### Đáp án

1. D   2. C   3. B   4. B   5. A   6. B   7. D



# **Phần II**

## **Dự án với Scratch 3.0 Online**





# CHƯƠNG 11

SCRATCH



---

## Trợ lý ảo với Text Classification

---



# 1 Giới thiệu

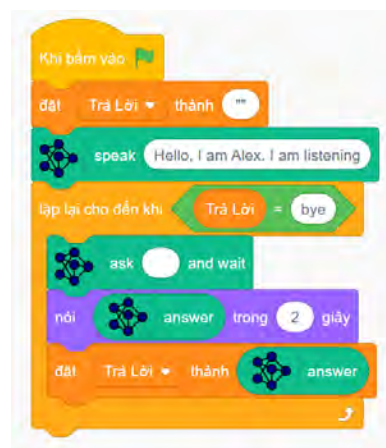
Với các thành tựu trong lĩnh vực nhận dạng giọng nói (Speech To Text), khái niệm trợ lý ảo đã trở nên vô cùng thông dụng trong các thiết bị di động hoặc tivi thông minh. Ban đầu, nó chỉ làm chức năng **tìm kiếm bằng giọng nói**. Tuy nhiên, trải qua một thời gian dài thu thập dữ liệu, các hệ thống này đã được nâng cấp thành hệ thống trợ lý ảo.

Trong bài này, chúng ta sẽ dùng các câu lệnh của khối Text Classification để hiện thực một trợ lý ảo với các chức năng chính như nhận diện giọng nói - phân loại - trả lời lại với người dùng. Chúng ta sẽ sử dụng thêm một công cụ khá cao cấp được hỗ trợ trong công cụ này, đó là phân loại dữ liệu. Chẳng hạn như bạn nói "Good bye" hoặc "bye" hay thậm chí là "bye bye", hệ thống sẽ phải nhận ra tất cả các cụm từ này đều là **Chào tạm biệt** mà xử lý cho đúng. Thực ra, bạn có thể tự hiện thực chức năng này bằng nhiều câu lệnh **nếu.. thì..**, nhưng đó chỉ phù hợp cho các bài toán nhỏ. Các mục tiêu trong dự án này như sau:

- Sử dụng hiệu quả các câu lệnh liên quan đến giọng nói
- Sử dụng được công cụ Text Classification
- Xây dựng hoàn thiện ứng dụng Trợ lý ảo

## 2 Nhập diện giọng nói

Chúng ta sẽ bắt đầu kích hoạt trợ lý ảo khi nhấn vào lá cờ màu xanh. Hệ thống sẽ nói một câu xin chào, sau đó chờ bạn nói với nó. Tùy vào câu nói của bạn mà nó sẽ trả lời lại. Trong trường hợp bạn nói "Bye", trợ lý ảo sẽ kết thúc. Với chức năng như trên, bước đầu chúng ta sẽ hiện thực như sau:



Hình 11.1: Chương trình nhận dạng giọng nói

Đầu tiên, chúng ta phải khai báo thêm một biến, đặt tên là **Trả Lời** để lưu lại giá trị của biến answer. Sở dĩ chúng ta không thể sử dụng trực tiếp biến answer vì không thể đặt lại giá trị ban đầu cho nó. Đây là điều hạn chế rất lớn khi sử dụng một vòng lặp trong chương trình.

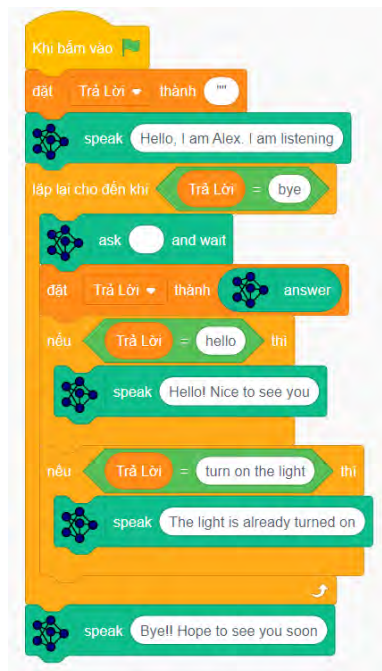
Tiếp theo, là việc chúng ta sẽ sử dụng khéo léo câu lệnh **ask and wait**: không cho nội dung gì vào phần ask cả. Lúc đó, chương trình chỉ lắng nghe mà thôi. Chúng ta cho in ra màn hình những gì chương trình nhận dạng được, để sử dụng cho bước tiếp theo.

### 3 Trí thông minh nhân tạo

Chúng ta sẽ hiện thực một chương trình nhỏ bằng câu lệnh nếu thì để minh họa cho trí thông minh nhân tạo của trợ lý ảo. Tùy vào câu nói của người dùng, máy sẽ trả lời lại, ví dụ như sau:

- Người nói: Hello - Máy nói: Hello, Nice to see you!
- Người nói: Turn on the light - Máy nói: The light is already turned on!
- Người nói: Người nói: I am tired - Máy nói: I play a song for you!
- Người nói: Bye - Máy nói: Bye, hope to see you soon!

Các câu lệnh của chúng ta sẽ được nâng cấp như sau:



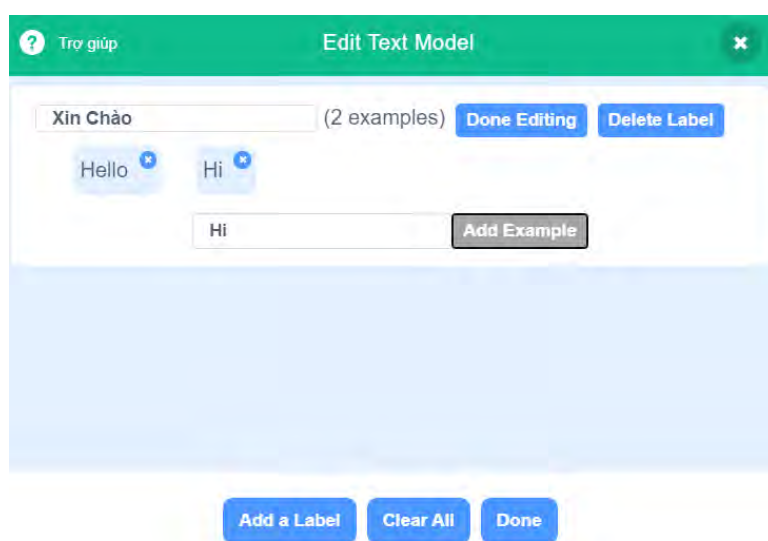
Hình 11.2: Trí tuệ nhân tạo cho Trợ lý ảo

Như vậy, trí tuệ nhân tạo sẽ được hiện thực bằng các câu lệnh nếu và nằm **bên trong vòng lặp**. Khi người dùng nói tạm biệt, câu nói từ phía trợ lý ảo sẽ được hiện thực cuối cùng, nằm bên ngoài vòng lặp. Chúng ta cũng hoàn toàn có thể hiện thực thêm các câu nói khác, hoặc thậm chí là gửi tín hiệu điều khiển bật tắt đèn trên mạch Microbit. Phần này xem như bài tập thêm cho bạn đọc tự hiện thực và sáng tạo.

## 4 Phân loại văn bản

Công cụ này sẽ thực sự rất cần thiết trong việc hiện thực một dự án thật, vì trong ngôn ngữ tự nhiên, có rất nhiều cách nói để diễn tả một mục đích. Chẳng hạn như để nói câu xin chào, trong tiếng Anh chúng ta có rất nhiều cách diễn đạt, Hello hay Hi chỉ là những từ đơn giản nhất.

Chúng ta có thể bắt từ khóa trong câu trả lời của người dùng để hiện thực trí tuệ nhân tạo. Tuy nhiên trong nhóm lệnh này, chúng ta có một công cụ mạnh mẽ hơn, gọi là phân loại văn bản. Bằng cách nhấn vào nút Edit Model, chúng ta sẽ có giao diện sau:

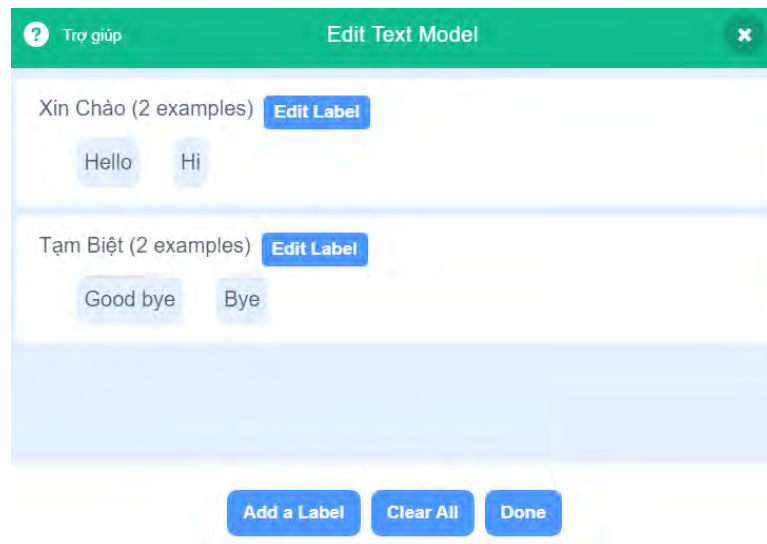


Hình 11.3: Công cụ phân loại văn bản

Một số thông tin quan trọng được tóm tắt sau đây:

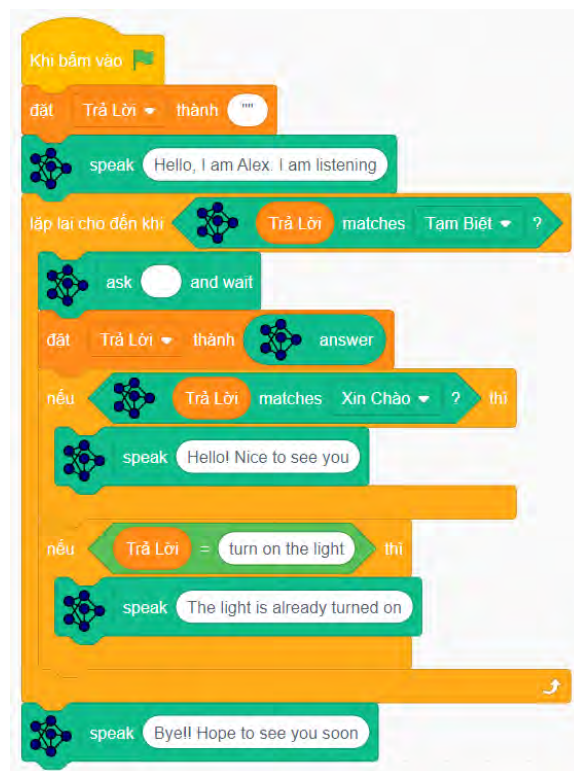
- Tên Class: Nhãn phân loại đối tượng, rất giống với Video AI trên Google. Tất nhiên chúng ta có thể đặt tên là tiếng Việt có dấu.
- Add Example: Giống như dữ liệu huấn luyện bên video, nhưng ở đây là dữ liệu Text
- Add a Label: Thêm một nhãn phân loại đối tượng
- Done: Kết thúc việc xây dựng mô hình phân loại

Bạn đọc có thể tự tạo cho mình một mô hình đơn giản, như minh họa ở hình bên dưới, cho 2 mục đích là Xin Chào và Tạm Biệt:



Hình 11.4: Hai nhãn phân loại Xin Chào và Tạm Biệt

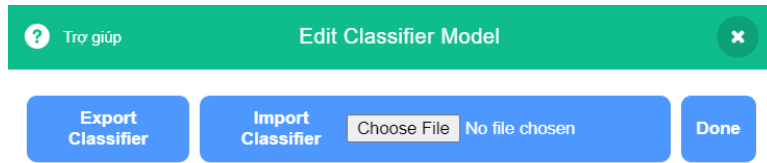
Quay trở lại với chương trình, chúng ta sẽ hiện thực lại 2 chức năng này, bằng cách sử dụng **câu lệnh matches**, **câu lệnh đầu tiên trong nhóm**, như sau:



Hình 11.5: Cải tiến chương trình bằng câu lệnh matches

Như vậy, chúng ta sẽ có nhiều cách nói Xin chào với trợ lý ảo, cũng như nhiều cách nói tạm biệt. Một số hạn chế của hệ thống phân loại này cũng giống như phần Video AI trên Google. Với giá trị khởi tạo cho biến Trả Lời là chuỗi rỗng, nó đang bị liệt kê vào nhóm Tạm Biệt!!! Do đó, bạn cần phải thay đổi giá trị khởi tạo của nó, thành hello chẳng hạn.

Để bổ sung thêm thông tin cho mô hình hiện tại, bạn có thể chọn lại Edit Model. Để lưu mô hình lại và dùng cho các dự án khác, bạn có thể chọn vào nút **Load/Save Model**. Giao diện sau đây sẽ hiện ra:



*Hình 11.6: Lưu và mở lại mô hình huấn luyện phân loại văn bản*

Bạn sẽ chọn tiếp Export Classifier trong trường hợp muốn lưu lại và chọn Import Classifier trong trường hợp muốn mở lại mô hình cũ.



## 5 Câu hỏi ôn tập

1. Câu lệnh nhận dạng giọng nói được nằm trong nhóm lệnh nào?  
A. PRG Microbit Robot Blocks  
B. Cảm biến video  
C. Text Classification  
D. Tất cả đều sai
2. Trí thông minh nhân tạo được hiện thực chủ yếu bằng câu lệnh nào?  
A. lặp mãi mãi  
B. nếu thì  
C. nếu thì nữa  
D. Tất cả đều đúng
3. Để thoát chương trình khi người sử dụng nói "bye", câu lệnh nào được sử dụng  
A. nếu thì  
B. nếu thì nữa  
C. lặp mãi mãi  
D. lặp cho đến khi
4. Để lưu hoặc mở lại mô hình huấn luyện cũ, bạn chọn vào nút nào dưới đây?  
A. Edit Model  
B. Load Model  
C. Save Model  
D. Load/Save Model
5. Với mô hình huấn luyện ở trên, các từ nào sau đây sẽ thuộc vào nhóm Xin Chào?  
A. Xin Chào, Hello, Hi  
B. Hello, Hi, Bye  
C. Hello, Hi  
D. Tất cả đều sai
6. Với mô hình huấn luyện ở trên, các từ nào sau đây sẽ thuộc vào nhóm Tạm Biệt?  
A. Tạm Biệt, Good bye, bye bye  
B. Good bye, bye, bye bye  
C. Hello, Hi  
D. Tất cả đều sai
7. Khi một chuỗi là rỗng, trong mô hình ở bài này, nó được nhận dạng thuộc nhóm nào?  
A. Tạm Biệt  
B. Xin chào  
C. Cả 2 nhóm trên  
D. Không nhóm nào cả

Đáp án

1. C   2. B   3. D   4. D   5. C   6. B   7. A



SCRATCH



## CHƯƠNG 12

---

# Thị giác Máy tính với Video AI

---



# 1 Giới thiệu

Trong bài cuối cùng của giáo trình này, chúng tôi sẽ chủ yếu đưa ra ý tưởng cho các dự án liên quan đến **Thị giác máy tính** (computer vision). Kỹ thuật chủ yếu trong các dự án liệt kê ở bài này sẽ là phân loại hình ảnh bằng công cụ Teachable Machine của Google. Bạn đọc có thể tự sáng tạo để kết hợp thêm với phần cứng, như mạch Microbit hoặc thậm chí là phát ra giọng nói bằng công cụ Text Classification.

Trong các ý tưởng dự án được trình bày, sẽ có 2 loại khác nhau: Camera đặt cố định (dự án điểm danh - nhận diện khuôn mặt) và Camera di chuyển trong quá trình hoạt động (ví dụ như Robot). Đối với dạng thứ 2, chúng tôi sẽ trình bày một số máy tính nhỏ **All in One** mà bạn đọc có thể dùng để triển khai hệ thống.

Thông qua các dự án ở đây, bạn đọc có thể hình dung rằng, bất cứ điều gì chúng ta phân biệt bằng mắt, về mặt lý thuyết là có thể huấn luyện cho hệ thống để phân biệt bằng Camera. Chính vì đặc điểm này, mà chủ đề này có tên là thị giác máy tính. Thêm nữa, giải pháp này sẽ rất tiết kiệm nếu so sánh với hệ thống tự động dựa trên cảm biến. Ví dụ như để phát hiện ô còn trống trong bãi xe ô tô, sẽ là rất tốn kém nếu trang bị mỗi ô đỗ xe là một cảm biến. Trong trường hợp này, hệ thống dùng camera sẽ tiết kiệm hơn rất nhiều. Mặc dù độ chính xác của camera là không thể nào đảm bảo tính chính xác 100%, nó vẫn đang dùng cho những bãi đỗ xe tiên tiến trên thế giới.

## 2 Dự án điểm danh

Với dự án này, bạn có thể áp dụng cho việc điểm danh trong lớp học, kiểm tra người ra vào xe bus hoặc phòng họp có nằm trong danh sách được chỉ định sẵn, hoặc thậm chí là kiểm tra xem người đã ra hết khỏi xe bus hay chưa.

Đầu tiên, để hiện thực dự án này, bạn phải có đủ dữ liệu về gương mặt của toàn bộ danh sách cần nhận dạng. Trong dự án này, bạn phải lưu ý một thành viên "vô danh" là ảnh nền khi không có người, để tăng độ chính xác của việc nhận dạng. Các dự án này thường có hệ thống camera lắp cố định, nên việc thu thập ảnh nền cũng không phải là vấn đề quá khó khăn.

Sau khi đã huấn luyện bằng công cụ Teachable Machine, bạn cần tạo ra một danh sách để lưu các thành viên, thì mới có đủ cơ sở để trả lời các câu hỏi như có bao nhiêu thành viên đã có mặt, gồm những ai, bao nhiêu thành viên còn vắng mặt và cụ thể là ai. Theo kinh nghiệm của tôi, bạn nên có ít nhất 2 danh sách: danh sách gốc và danh sách người đang có mặt. Khi phát hiện ra một người, bạn thêm người đó vào danh sách người đang có mặt.

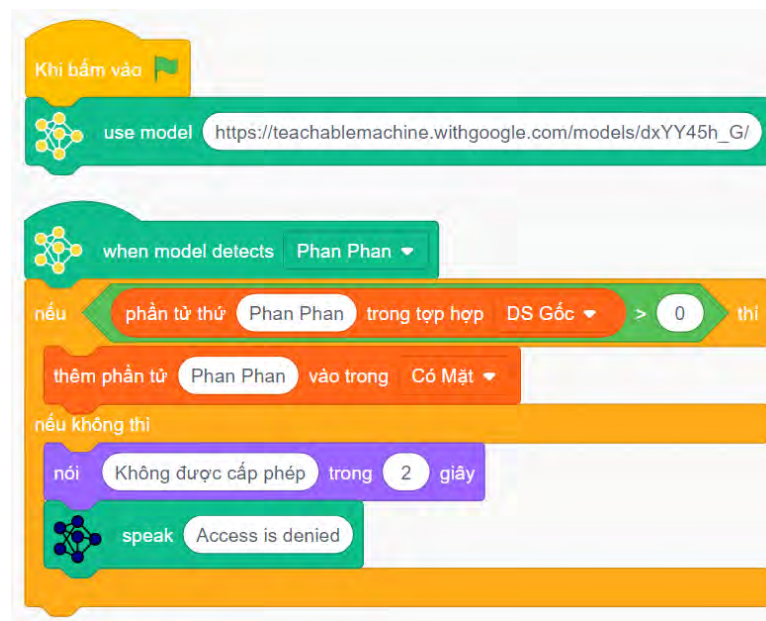
Để tạo một danh sách, bạn vào mục **Các biến số** và chọn tiếp vào **Tạo một danh sách**, như minh họa ở hình bên dưới:

Sau khi tạo xong một danh sách, Scratch sẽ tự động sinh ra hàng loạt các câu lệnh tương ứng để thao tác trên danh sách, bao gồm thêm một phần tử, xóa một phần tử và quan trọng nhất, là duyệt qua danh sách. Thao tác trên danh sách là tác vụ



Hình 12.1: Tạo danh sách cho dự án điểm danh

không đơn giản, nhưng đã có từ phiên bản Scratch 2.0 nên chúng tôi sẽ không trình bày chi tiết ở đây. Chương trình bên dưới là một ví dụ, như phát hiện có mặt một người, hệ thống sẽ duyệt qua danh sách gốc. Nếu người này có trong danh sách gốc, mới được thêm vào danh sách Có Mặt. Một chương trình gợi ý sẽ như sau:



Hình 12.2: Hiện thực chức năng điểm danh

Rất may mắn là trên Scratch 3.0 có hỗ trợ cho chúng ta kiểm tra một phần tử có nằm trong danh sách hay không. Thậm chí, nó còn cho biết phần tử đó nằm ở vị trí nào trong danh sách. Bình thường, chúng ta sẽ phải duyệt qua từng phần tử trong danh sách gốc và so sánh nó với phần tử hiện tại. Việc thêm một phần tử vào trong danh sách Có Mặt cũng nên kiểm tra, để đảm bảo phần tử đó chưa tồn tại trong danh sách Có Mặt.

### 3 Dự án phân loại rau quả

Đây cũng là một dự án rất tiềm năng cho chủ đề Thị giác máy tính. Bạn có thể phân biệt được rau xanh bị sâu và rau xanh còn tốt, phân loại cà chua còn sống hay cà chua đã chín, hoặc xa hơn, là dự đoán cà chua đã đến kì thu hoạch hay chưa. Tất cả các chủ đề này đều cần việc thu thập dữ liệu tỉ mỉ và đa dạng để huấn luyện hệ thống. Các dự án này cũng có đặc điểm giống bài toán điểm danh, là camera được thiết lập cố định. Do đó, bạn có thể dễ dàng thu thập và huấn luyện dữ liệu cho ảnh nền, một điều không thể thiếu để nâng cao tính chính xác của hệ thống nhận dạng.

Một ý tưởng mà chúng tôi muốn chia sẻ, là hệ thống có thể thay thế cho nhân viên cân rau quả ở các siêu thị. Hiện tại, nhân viên này phải nhớ mã của nhiều loại rau quả để có thể tính ra giá thành. Chúng ta có thể dùng camera để nhận dạng, và gợi ý cho nhân viên đó chọn vào rau quả đang được cân. Xa hơn, hệ thống có thể thay thế hoàn toàn nhân viên của siêu thị, để cho người mua tự cân món hàng mà mình muốn mua.

### 4 Robot tự hành

Đây cũng là một chủ đề vô cùng hấp dẫn ở thời điểm hiện tại. Nhờ các công nghệ của lĩnh vực thị giác máy tính, mà các sản phẩm về ô tô và Robot tự hành ngày càng khả thi. Thực ra, đã và đang có những sản phẩm mẫu về chủ đề này đang được sử dụng trong thực tế.

Hãy tưởng tượng rằng, robot của bạn được trang bị một webcam ở trên cao. Chúng ta hoàn toàn có thể chụp hình về đường đi, để nhận dạng ra là Robot có đi ở giữa đường hay không. Việc điều khiển của nó cũng tương tự như ý tưởng dò theo đường kẻ, tuy nhiên chúng ta cần một bộ xử lý đủ mạnh thì Robot mới có thể đi nhanh được, vì chụp hình và xử lý nhận dạng là các tác vụ tương đối phức tạp.

Tiếp theo, chúng ta cũng hoàn toàn có thể nhận diện ra biển báo giao thông, cũng bằng những kĩ thuật tương tự: chụp hình và huấn luyện nhiều biển báo cho hệ thống. Một bộ xử lý trong trường hợp này có thể sẽ không đủ, bạn sẽ phải cần 2 bộ xử lý độc lập: một để đi đúng đường và một để nhận biết biển báo giao thông.

Một hạn chế khi triển khai công nghệ Thị giác máy tính trên Robot là hình ảnh từ camera là hình ảnh động và di chuyển theo Robot. Việc đặt một máy tính lên Robot là không ổn về giải pháp, do máy tính của chúng ta không được thiết kế để vừa hoạt động vừa di chuyển, và nó cũng khó có thể hoạt động 24/7. Trong trường hợp này, có thể bạn sẽ cần tới những máy tính công nghiệp hoặc máy tính mini, sẽ được trình bày ở phần tiếp theo.

### 5 Triển khai với Mini PC

Trong tất cả các dự án trên, dù sao thì máy tính vẫn không phải là thiết bị thích hợp để triển khai hàng loạt, do nó không bền và cũng khó để bảo trì. Vì vậy, chúng

tôi sẽ giới thiệu một số máy tính mà bạn đọc có thể xem xét. Theo thứ tự từ trên xuống là mức độ phổ biến nhưng giá thành cao, cho tới máy tính chuyên dụng với giá thành thấp.

## 5.1 Máy tính NUC

Bạn có thể tìm kiếm từ khóa NUC PC hoặc Mini PC để tham khảo các sản phẩm này. Thực ra, sức mạnh của nó ngang ngửa với những máy tính để bàn thông dụng, nhưng kích thước lại rất nhỏ gọn và có thể cầm trên lòng bàn tay. Tùy vào cấu hình mà giá thành có thể dao động từ 3 triệu cho tới hơn 5 triệu VNĐ. Một hình ảnh minh họa cho sản phẩm này như sau:



Hình 12.3: Máy tính mini NUC

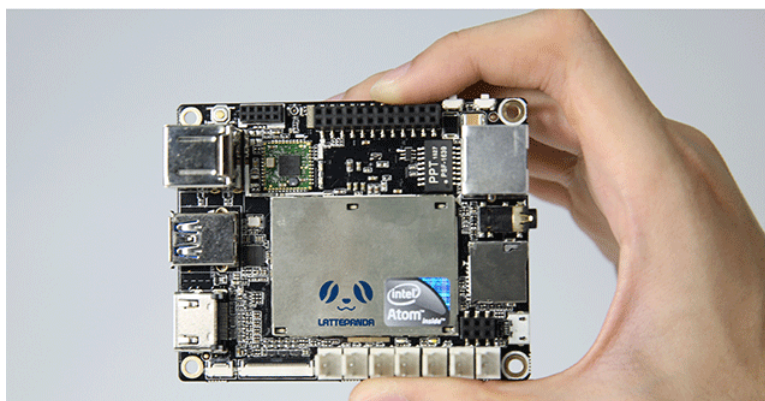
Thực ra, để chạy một giải thuật nhận dạng trên nền tảng web, bạn cũng không cần một máy tính quá mạnh. Hãy lựa chọn một cấu hình vừa phải, ưu tiên xài nguồn tiết kiệm năng lượng hoặc thậm chí là nguồn 5V để bạn có thể dùng cục sạc điện thoại để vận hành.

## 5.2 Máy tính nhúng LettePanda

Sản phẩm này được thiết kế theo chuẩn công nghiệp hơn, cũng xài điện một chiều 5V nên rất dễ để triển khai. Bên cạnh việc cài đặt hệ điều hành Windows, máy tính nhúng còn dễ dàng kết nối với các phần cứng phụ trợ khác. Thêm nữa, giá thành cũng rất phù hợp cho các ứng dụng tầm trung. Hình ảnh của máy tính này như bên dưới:

Sử dụng toàn bộ linh kiện là chip tích hợp, máy tính nhúng sẽ có độ bền cao, chịu được va đập và hoạt động trong điều kiện khắc nghiệt hơn máy tính NUC. Giá





LATTEPANDA : A SINGLE BOARD FULL WINDOWS 10 COMPUTER 

*Hình 12.4: Máy tính nhúng LettePanda*

thành của máy tính nhúng này dao động tầm 3 triệu VNĐ. Với lợi thế nhỏ gọn và tiết kiệm năng lượng, bạn có thể dùng cho các dự án liên quan đến Robot tự hành với LettePanda.

Hiện tại, máy tính nhúng LettePanda xài hệ điều hành Windows 10 đã được chúng tôi kiểm tra và hoàn toàn chạy được các giải thuật trí tuệ nhân tạo từ Google. Việc nhận diện âm thanh từ Micro có sẵn cũng có hiệu suất chấp nhận được.

### **5.3 Điện thoại di động**

Đây cũng là một thiết bị vô cùng tiềm năng để triển khai các ứng dụng đơn giản. Mặc dù hiện tại, điện thoại di động chưa có độ tương thích cao với mã nguồn mở Scratch 3.0 Online. Tuy nhiên chúng tôi kì vọng trong một ngày không xa, những phiên bản tiếp theo của Scratch sẽ chạy được mượt mà trên điện thoại di động. Với lợi thế là đa số chúng ta đều có sẵn điện thoại di động, và điện thoại di động thì luôn có sẵn camera lẫn micro. Điện thoại di động cũng có sẵn nguồn pin có thể xài trong thời gian dài mà không cần sạc. Do đó, trong tương lai nếu có thể dùng điện thoại di động cho các ứng dụng mà khung hình chuyển động sẽ là giải pháp tiết kiệm nhất.

Cùng xài hệ điều hành Android với điện thoại, trong tương lai, hy vọng các TIVI Box cũng có thể áp dụng được các giải thuật về trí tuệ nhân tạo.

## 6 Câu hỏi ôn tập

1. Những ứng dụng nào sau đây thuộc nhóm Thị giác máy tính?
  - A. Nhận dạng từ khóa
  - B. Nhận dạng bảng giao thông
  - C. Điều khiển nhà thông minh
  - D. Tất cả đều đúng
2. Các thiết bị nào sau đây là máy tính nhúng?
  - A. Laptop PC
  - B. Android Phone
  - C. Android Box
  - D. LettePanda
3. Trên máy tính nhúng, trình duyệt nào có thể thực thi đầy đủ nhất các tính năng của Scratch 3.0 Online?
  - A. Internet Explorer
  - B. FireFox
  - C. Microsoft Edge
  - D. Google Chrome
4. Cảm biến dò theo đường kẻ được lưu vào khối nào?
  - A. read distance
  - B. line detected on
  - C. Cả 2 khối trên
  - D. Tất cả đều sai
5. Trong dự án điểm danh, cần phải sử dụng cấu trúc dữ liệu nào?
  - A. Cấu trúc cây
  - B. Cấu trúc hàng đợi
  - C. Cấu trúc danh sách
  - D. Tất cả các cấu trúc trên
6. Máy tính nhúng LettePanda dùng cho ứng dụng nào sau đây là thích hợp nhất?
  - A. Điểm danh
  - B. Nhận diện hoa quả
  - C. Phân biệt rau xấu - tốt
  - D. Robot tự hành
7. Đối tượng hình nền là cần cho các ứng dụng nào sau đây?
  - A. Điểm danh
  - B. Nhận diện hoa quả
  - C. Phân biệt rau xấu - tốt
  - D. Tất cả các ứng dụng trên

Đáp án

1. B   2. D   3. D   4. B   5. C   6. D   7. D