



THE DARIU FOUNDATION

HƯỚNG DẪN LẬP TRÌNH MICRO:BIT CHO HỌC SINH

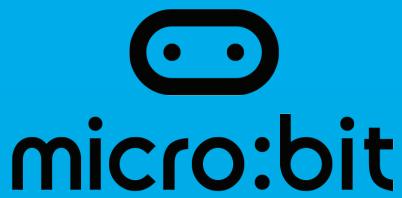




THE DARIU FOUNDATION

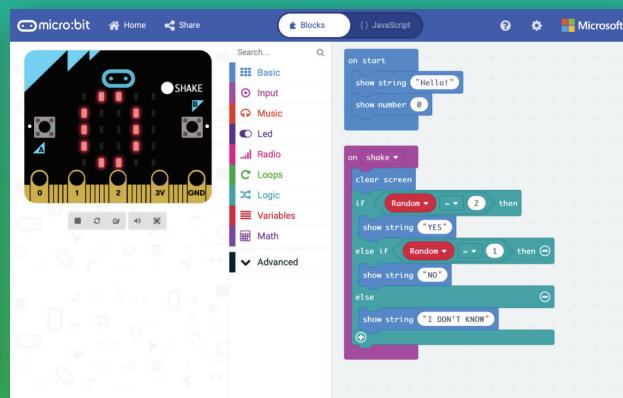
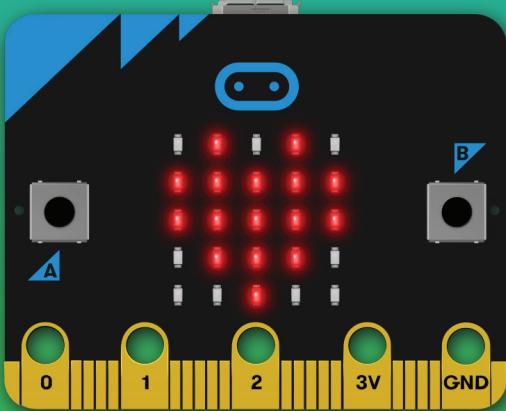
NỘI DUNG

| | |
|--|----|
| Bài 1: Làm quen với mạch lập trình MicroBit | 3 |
| Bài 2: Tương tác với màn hình hiển thị trên MicroBit | 10 |
| Bài 3: Tổng hợp các câu lệnh về hiển thị | 16 |
| Bài 4: Điều khiển nút nhấn trên MicroBit | 20 |
| Bài 5: Lập trình MicroBit bằng điện thoại thông minh | 24 |
| Bài 6: Tương tác giữ MicroBit và hành vi người dùng | 35 |
| Bài 7: Cảm biến trên MicroBit | 43 |
| Bài 8: Gửi dữ liệu không dây giữa các mạch MicroBit | 47 |
| Bài 9: Các cấu trúc lặp và Điều kiện trên MicroBit | 51 |
| Bài 10: Đồng hồ thông minh trên MicroBit | 55 |



Bài 1:

Làm quen với mạch lập trình Micro:Bit

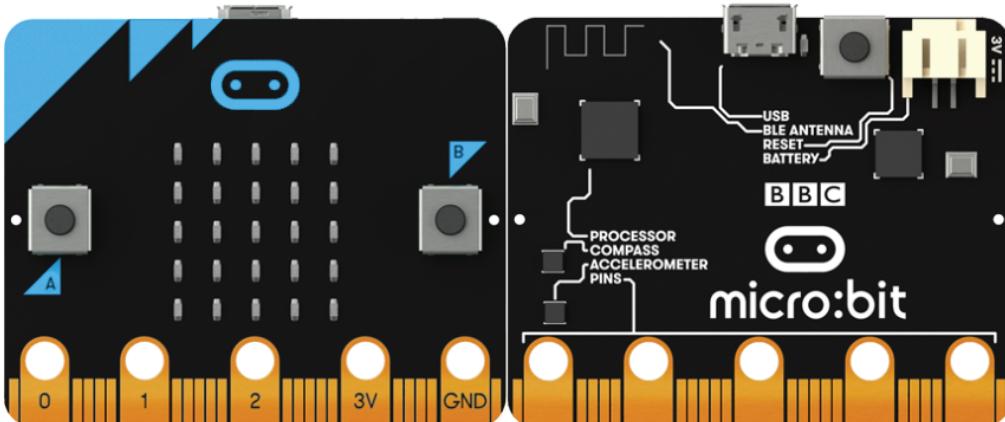


1. Mục tiêu

- Học sinh hiểu được các bước cơ bản trong việc lập trình MicroBit
- Học sinh nắm được các thao tác cơ bản trên MakeCode
- Học sinh viết được chương trình đơn giản trên MicroBit
- Học sinh nắm được việc sử dụng chương trình mô phỏng của MicroBit

2. Nội dung chi tiết

2.1. Tổng quan về MicroBit



Hình 1: Hình ảnh mặt trên và mặt dưới của MicroBit

MicroBit là một thiết bị nhỏ gọn có thể cầm trên tay. Chi tiết hơn, nó thực sự là một máy tính thu nhỏ và có khả năng lập trình được. Tức là người dùng có thể thay đổi chức năng và hoạt động của nó để tạo ra các ứng dụng hấp dẫn và thu hút, từ điều khiển các hiệu ứng đèn chớp tắt, cho đến các ứng dụng phức tạp như là điều khiển Robot không dây, nhà thông minh, thậm chí là các ứng dụng cao cấp như Internet vạn vật (còn gọi là Internet of Things).

Hiện tại MicroBit được sử dụng trong trường học khá rộng rãi, và trở nên rất phổ biến ở các nước trên thế giới như Phần Lan, Iceland hay Singapore. Với việc hỗ trợ ngôn ngữ lập trình theo kiểu “kéo-thả” cùng một bộ thư viện khá đầy đủ, các em học sinh có thể dễ dàng tiếp cận và sáng tạo ra rất nhiều các ứng dụng thú vị.

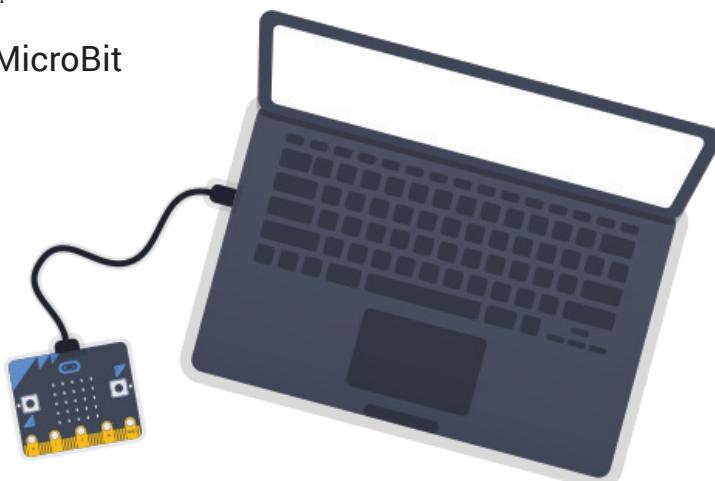
Trong giáo trình này, chúng tôi tập trung vào việc hướng dẫn các thao tác cơ bản để người đọc có thể làm được những ứng dụng đơn giản và làm nền tảng cho việc tự tìm hiểu cũng như sáng tạo các ứng dụng trên MicroBit.

2.2. Các bước cơ bản để lập trình trên MicroBit

Bước 1: Kết nối với MicroBit

Việc kết nối với MicroBit thực sự rất đơn giản, chỉ cần một dây micro USB vào máy tính. Hệ điều hành trên máy tính sẽ tự động nhận ra mạch MicroBit như một USB bình thường.

Hình 2 mô phỏng việc kết nối giữa máy tính và mạch MicroBit.

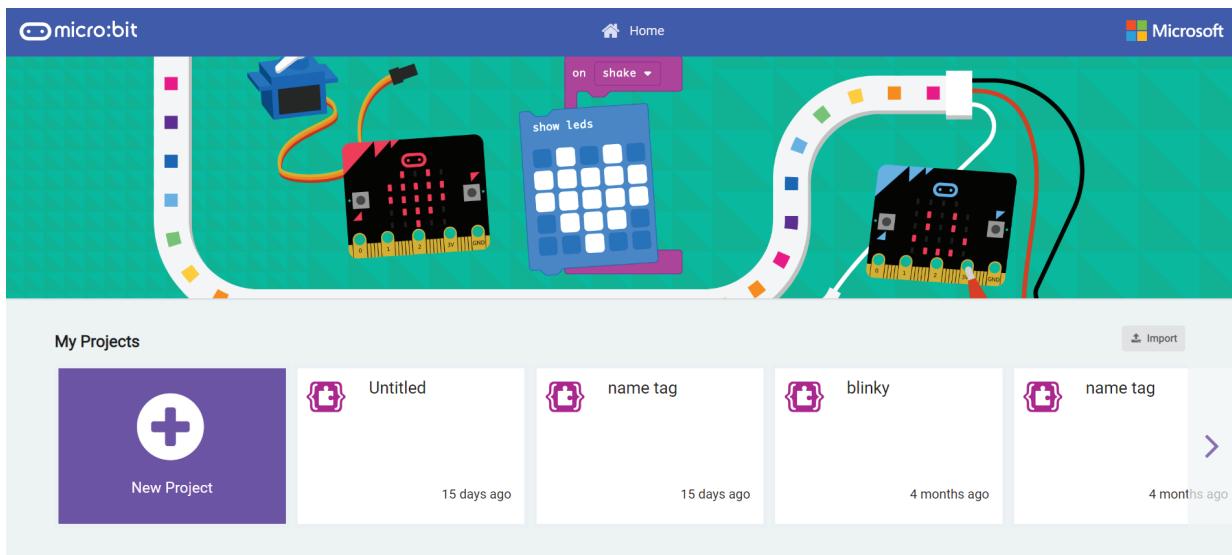


Hình 2: Kết nối mạch MicroBit với máy tính thông qua dây USB

Bước 2: Viết chương trình

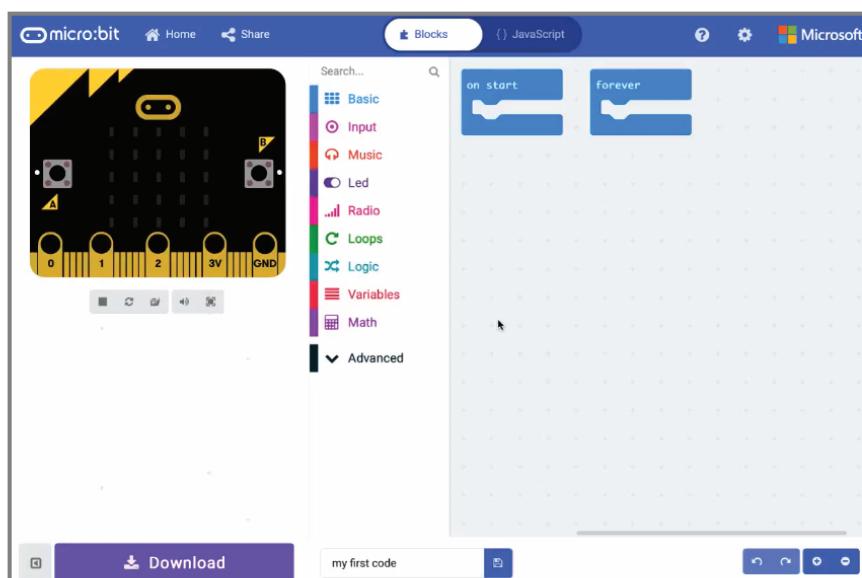
Để lập trình cho MicroBit, có rất nhiều công cụ hỗ trợ. Tuy nhiên trong giáo trình này, chúng tôi sử dụng môi trường lập trình trực tuyến, gọi là MakeCode. Một lợi thế rất lớn mà MakeCode có được là việc mô phỏng chương trình trước khi nạp trực tiếp vào mạch MicroBit. Chức năng này sẽ tiết kiệm nhiều thời gian cho việc kiểm tra chương trình.

Vào đường dẫn sau đây, và chọn vào New Project <https://makecode.microbit.org/>



Hình 3: Vào trang web lập trình trực tuyến MakeCode

Giao diện sau đây sẽ hiện ra để chúng ta có thể bắt đầu lập trình:

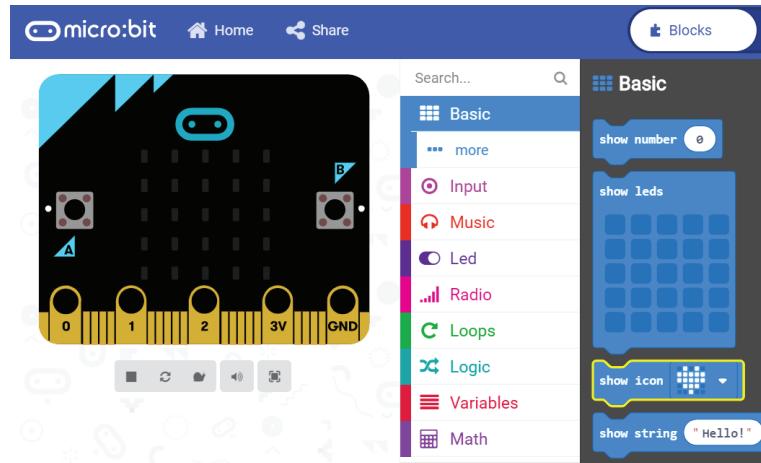


Hình 4: Giao diện lập trình cơ bản

Chương trình của MicroBit mặc định sẽ có 2 khối cơ bản như sau:

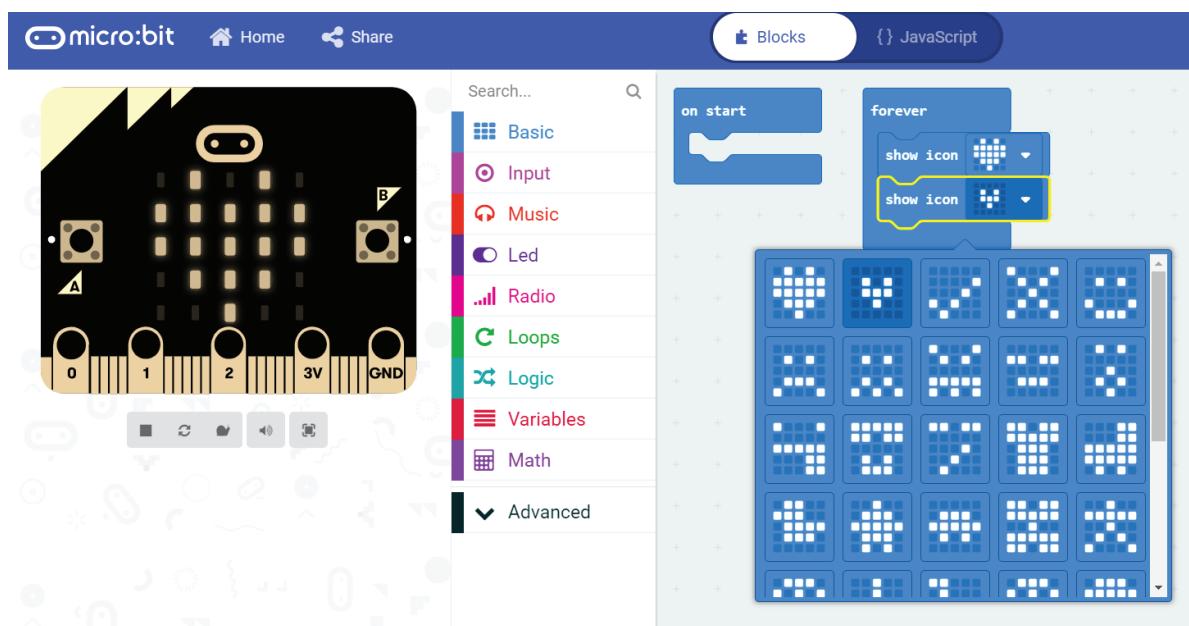
- on start: Những câu lệnh trong khối này sẽ được thực hiện đầu tiên khi mới bật nguồn, tuy nhiên nó chỉ được thực hiện duy nhất một lần.
- forever: Những câu lệnh trong khối lệnh này sẽ được thực hiện sau khối on start, tuy nhiên nó được lặp đi lặp lại mãi mãi khi mạch MicroBit còn được cấp nguồn điện.

Đầu tiên, để cho đơn giản, chúng ta dùng câu lệnh show icon nằm trong nhóm lệnh Basic (xem Hình 5)



Hình 5: Viết chương trình đầu tiên bằng lệnh show icon

Chúng ta có thể ghép 2 câu lệnh show icon như Hình 6. Câu lệnh show icon là một câu lệnh có thể thay đổi nội dung hiển thị bằng cách chọn vào dấu mũi tên bên phải.

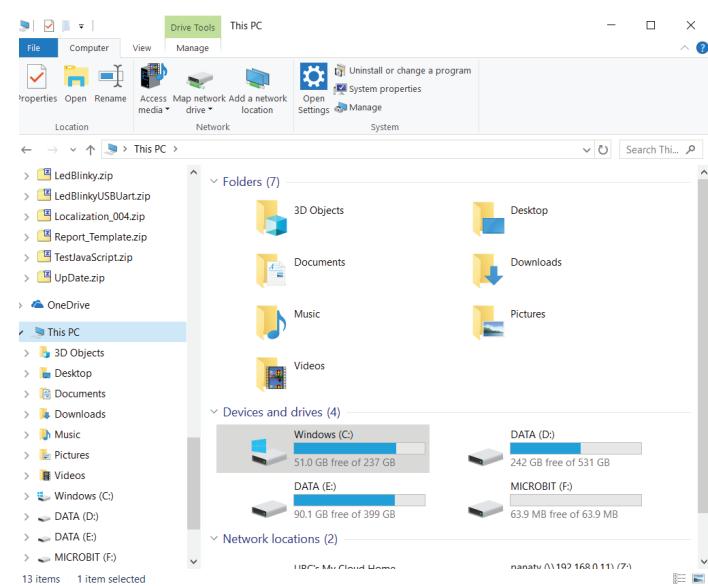


Hình 6: Tạo chương trình bằng 2 câu lệnh show icon

Bước 3: Nạp chương trình

Việc nạp chương trình vào mạch MicroBit thực sự rất tiện lợi do bo mạch này được nhận diện như một thiết bị USB bình thường. Như hình bên dưới, mạch MicroBit được nhận dạng là ổ đĩa F.

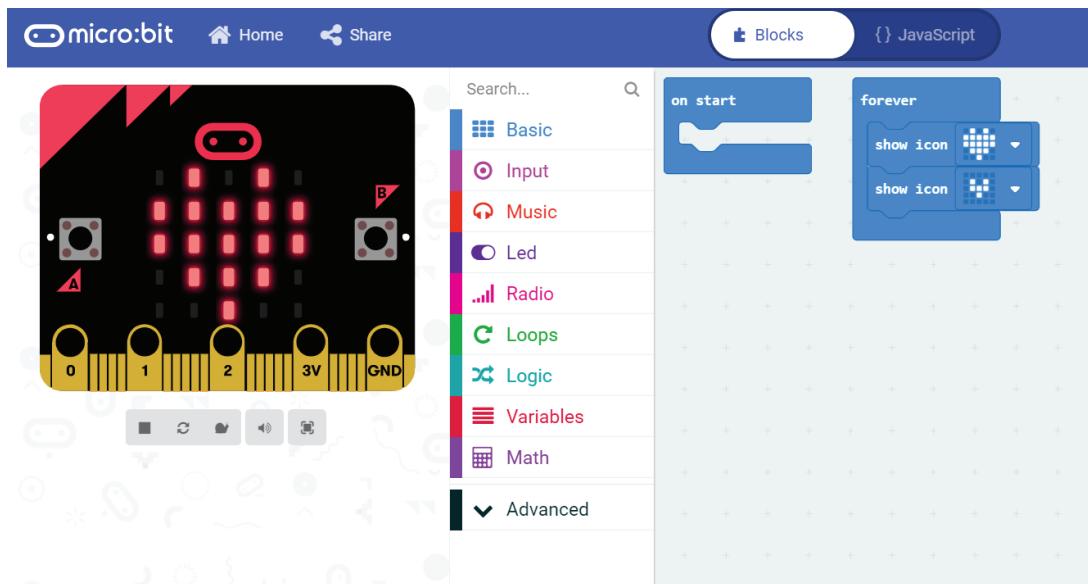
Do đó, chúng ta chỉ cần nhấn vào nút Download và chọn đường dẫn tới mạch MicroBit là xong. Quá trình nạp sẽ mất vài giây. Trong suốt quá trình nạp, đèn nguồn bên cạnh khe cắm USB sẽ chớp tắt liên tục. Khi đèn ngừng chớp tắt cũng là lúc chương trình nạp xong và bo mạch MicroBit sẽ bắt đầu chạy chương trình mà chúng ta vừa hiện thực.



Hình 7: Mạch MicroBit được nhận dạng là một ổ đĩa trên máy tính

2.3. Các bước cơ bản để lập trình trên MicroBit

Đây là một đặc tính hết sức thú vị của chương trình lập trình hỗ trợ trên bo mạch MicroBit. Ở khung cửa sổ bên trái, mạch mô phỏng lại chương trình mà chúng ta vừa viết. Nhờ chức năng này mà chúng ta có thể tự tìm hiểu chức năng của một câu lệnh, giống như việc nhập vào câu lệnh và xem đáp ứng của chú mèo bên ngôn ngữ Scratch. Trong ví dụ bên dưới, mạch mô phỏng sẽ luôn hiển thị hình ảnh 2 trái tim, tương ứng với chương trình của chúng ta hiện thực ở khung chương trình bên phải.

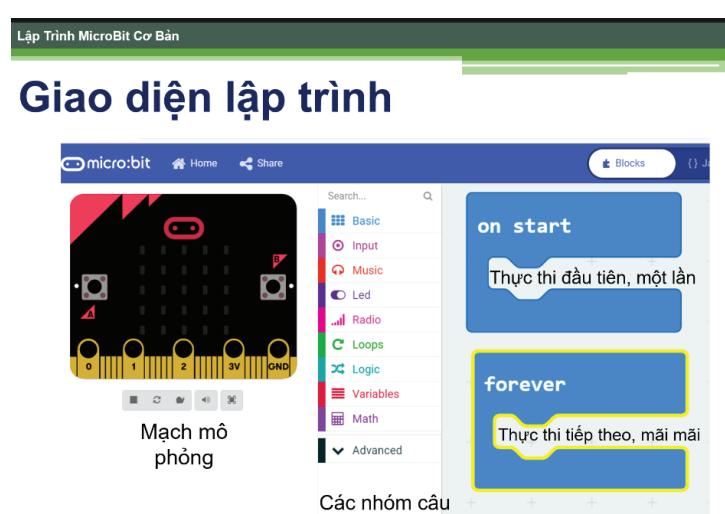


Hình: Chương trình mô phỏng trên MicroBit

Nhờ chức năng mô phỏng này, chúng ta có thể viết và kiểm tra sơ bộ chương trình trước khi nạp vào mạch MicroBit. Bên cạnh đó, học sinh có thể tự viết những chương trình ở nhà, không cần mạch MicroBit, mà vẫn có thể thấy được kết quả của mình.

2.4. Các thao tác cơ bản trên MicroBit

Tương tự như ngôn ngữ kéo thả Scratch, môi trường lập trình trên MicroBit cũng được tổ chức thành 3 phần như trình bày ở Hình 8. Thay thế cho sân khấu với nhân vật chú mèo, chúng ta có hình ảnh của mạch MicroBit. Thực chất, đây chỉ là hình ảnh mô phỏng việc thực thi chương trình của chúng ta trên mạch MicroBit. Về cơ bản, việc mô phỏng khá chính xác với chương trình sẽ chạy thực tế trên mạch MicroBit.



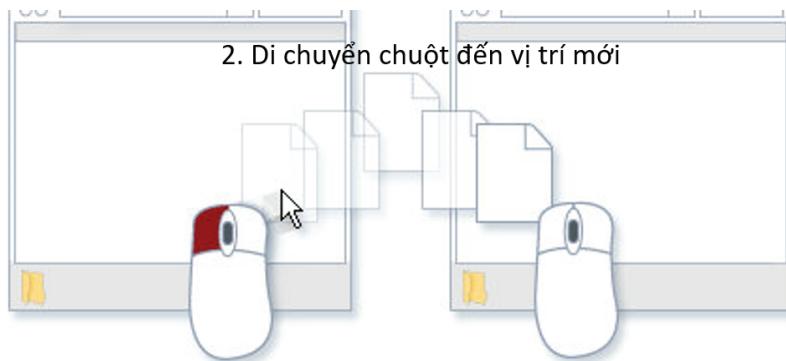
Hình 8: Giao diện lập trình trên MicroBit

Ở khung cửa sổ thứ 2, chúng ta có các câu lệnh trên MicroBit, được tổ chức theo từng nhóm, mỗi nhóm có 1 màu khác nhau. Cũng như môi trường lập trình trên Scratch, việc tổ chức này giúp chúng ta dễ dàng tìm kiếm các câu lệnh dựa vào màu sắc của chúng. Cuối cùng, là khung chương trình, dùng để ghép nối các khối lệnh trên MicroBit.

2.4.1 Ghép nối các câu lệnh

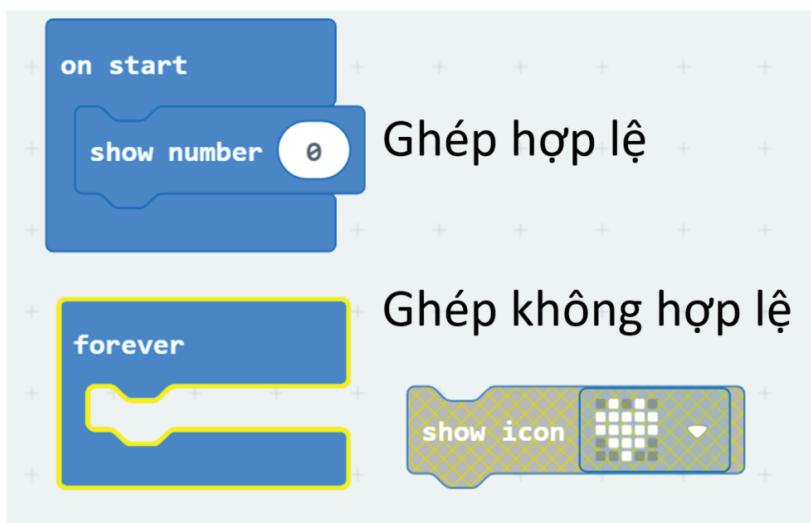


Hoàn toàn tương tự với Scratch, việc ghép nối các câu lệnh trên MicroBit được dựa vào nguyên lý “kéo-thả”, như minh họa ở Hình 9.



Hình 9: Ba bước để thực hiện việc kéo thả

Tuy nhiên, trên môi trường MicroBit, khi một câu lệnh được ghép nối không hợp lệ, nó sẽ bị tô xám. Chỉ khi nào được ghép nối một cách hợp lệ, nó mới hiển thị màu của câu lệnh đó, như minh họa ở Hình 10.



Hình 10: Ghép nối hợp lệ và không hợp lệ

2.4.2 Thay đổi thông số của một câu lệnh

Hoàn toàn tương tự như trên Scratch, có 2 cách để chúng ta có thể thay đổi thông số của một câu lệnh:

- Nhấp đôi vào giá trị trong ô màu trắng, sau đó nhập giá trị mới
- Chọn vào một danh sách có sẵn, như minh họa ở Hình 11.

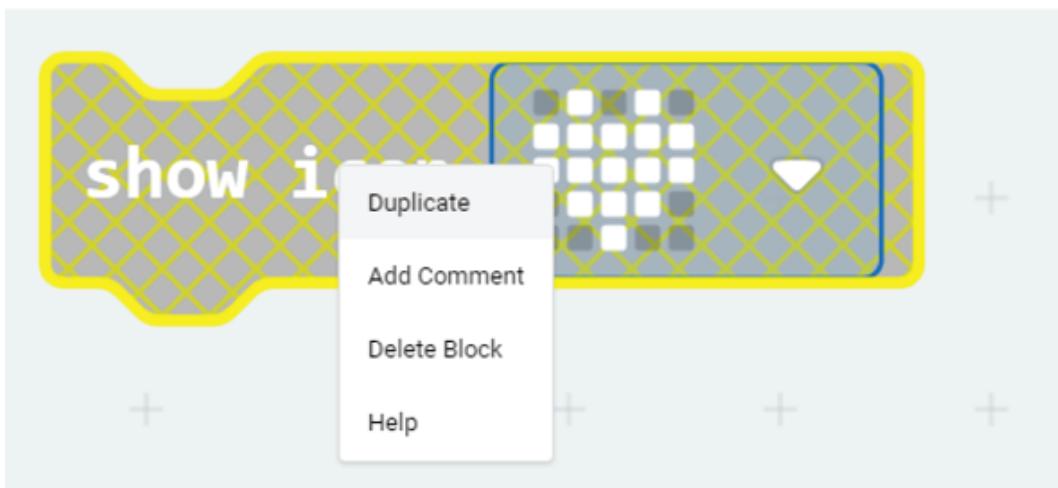


Hình 11: Thay đổi thông số của một câu lệnh

Trong một số trường hợp, mặc dù là danh sách chọn, chúng ta vẫn có thể nhập một giá trị ngoài danh sách có sẵn. Ví dụ như ở Hình 11, chúng ta có thể nhập vào con số 5000 nếu muốn tạo hiệu ứng đợi 5 giây.

2.4.3 Nhân bản – Xóa một câu lệnh

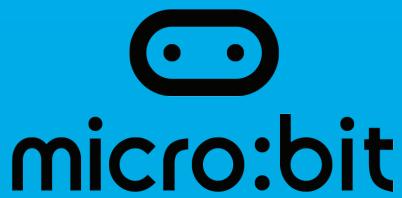
Để thực hiện 2 chức năng này, chúng ta nhấp chuột phải vào câu lệnh đó, và chọn Duplicate hoặc Delete Block, như minh họa ở Hình 12.



Hình 12: Nhân bản - Xóa một câu lệnh

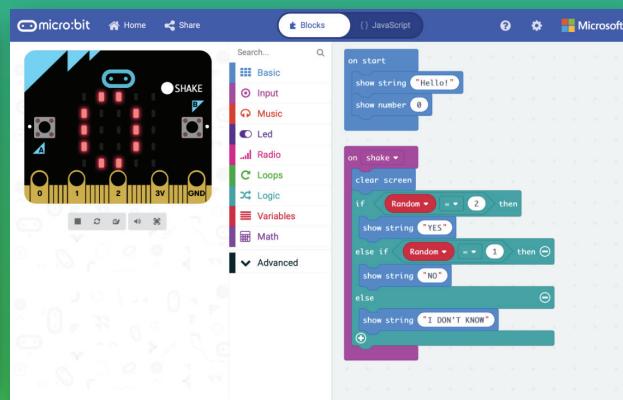
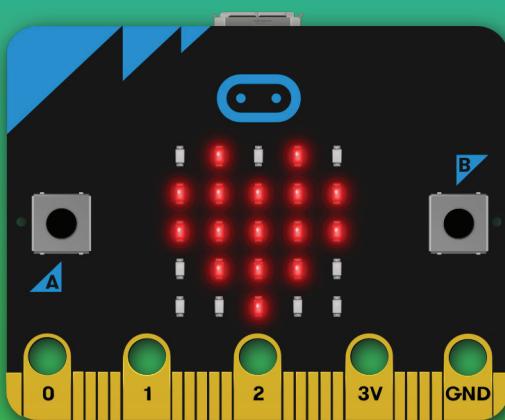
3. Bài tập trên lớp

Học sinh tự thiết kế thêm những hình ảnh sinh động khác trong kịch bản hiển thị của MicroBit bằng câu lệnh show icon.



Bài 2:

Tương tác với màn hình hiển thị trên MicroBit

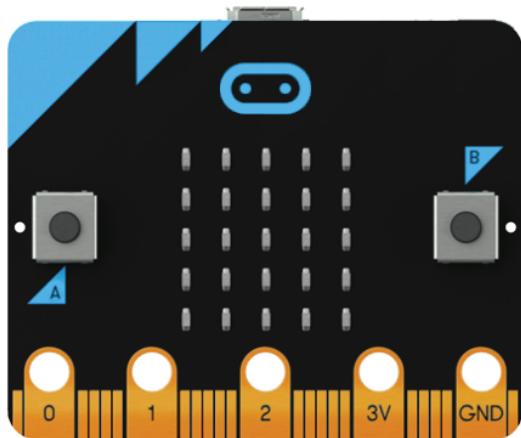


1. Mục tiêu

- Học sinh hiểu được các bước cơ bản trên màn hình hiển thị
- Học sinh viết được chương trình tạo hiệu ứng hiển thị trên MicroBit
- Học sinh nắm được việc tổ chức chương trình của MicroBit
- Học sinh phối hợp được các câu lệnh để tạo ra hiệu ứng hiển thị đẹp
- Học sinh có thể lưu và mở lại chương trình của mình

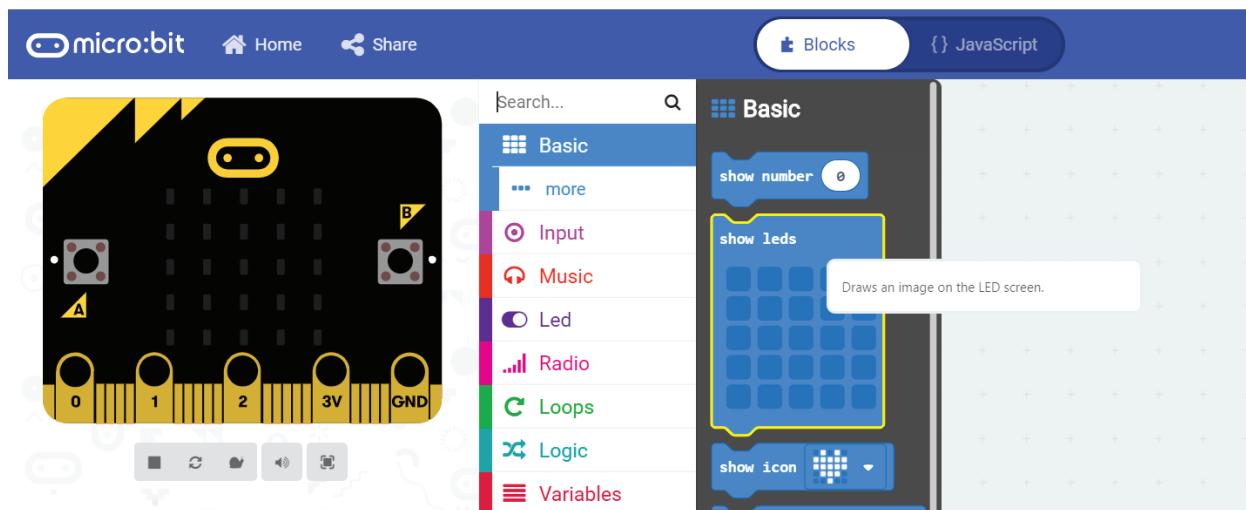
2. Nội dung chi tiết

2.1. Giới thiệu về màn hình hiển thị



Hình 1: Màn hình hiển thị với 25 đèn nhỏ ở mặt trước của MicroBit

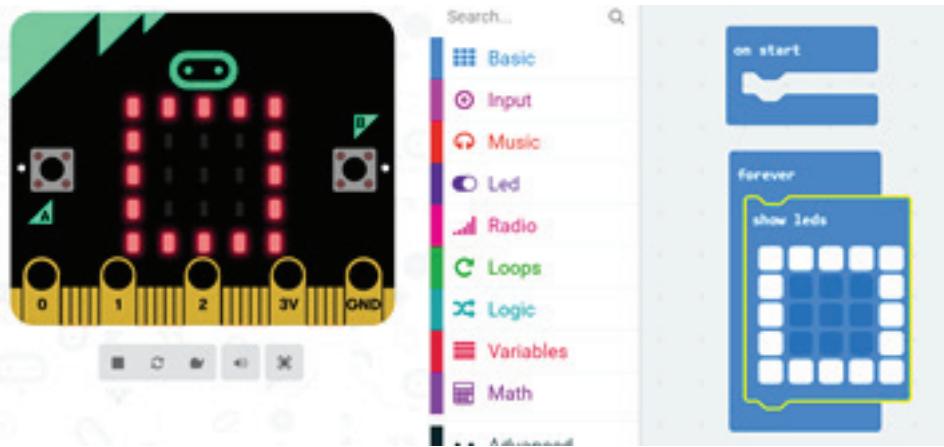
MicroBit có tất cả 25 đèn nhỏ (còn gọi là LED) được tổ chức thành 5 hàng và 5 cột như Hình 1. Có rất nhiều cách để hiển thị nội dung mong muốn ra màn hình hiển thị này. Tuy nhiên trong bài học này chúng ta sẽ tập trung vào việc sử dụng câu lệnh show leds. Câu lệnh này nằm trong mục Basics, có màu xanh dương, như hình bên dưới:



Hình 2: Câu lệnh show leds phục vụ cho việc hiển thị

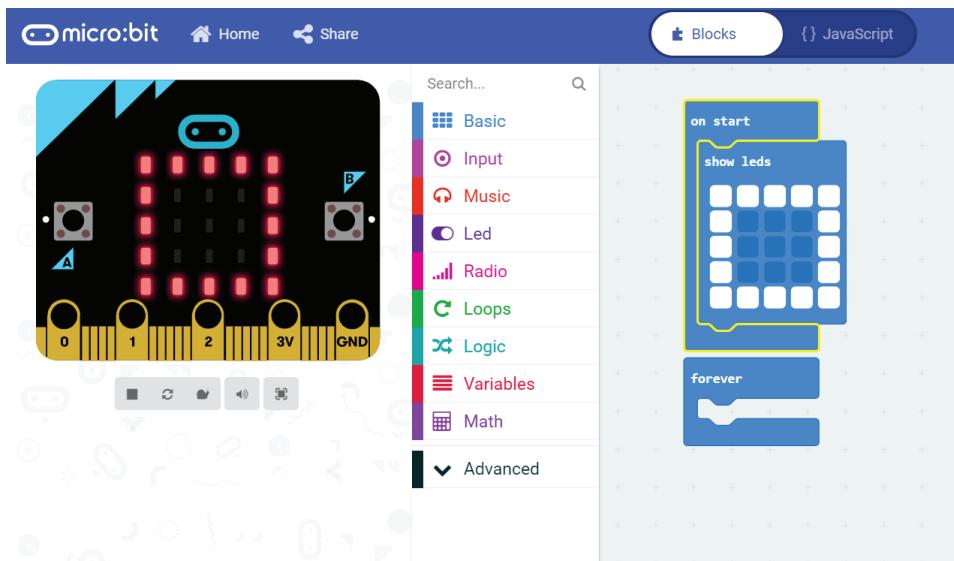
2.2. Nguyên tắc sử dụng câu lệnh show leds

Câu lệnh này đưa ra một giao diện tương tác rất sinh động với màn hình hiển thị 25 led của MicroBit. Chúng ta muốn bóng đèn nào sáng, chỉ cần nhấp chuột vào nó. Khi muốn bóng đèn đó tắt, chúng ta nhấp thêm một lần nữa. Trong ví dụ dưới đây, một hình vuông sẽ được hiển thị trên mạch MicroBit



Hình 3: Sáng một hình vuông trên màn hình hiển thị

Câu hỏi trên lớp: Học sinh giải thích sự giống nhau và khác nhau của đoạn chương trình ở Hình 3 và Hình 4 bên dưới.



Hình 4: Một chương trình khác để hiển thị hình vuông

Giống nhau: Đều hiển thị hình vuông ra màn hình hiển thị.

Khác nhau: Chương trình ở Hình 4 chỉ gửi lệnh hiển thị ra màn hình 1 lần duy nhất mà thôi. Tuy nhiên đèn sẽ sáng mãi mãi cho đến khi nào nó nhận một lệnh tắt đi. Ngược lại, ở Hình 3, đèn sẽ liên tục nhận lệnh phải sáng. Như vậy, chúng ta có thể thấy, chương trình ở Hình 4 là tối ưu hơn so với Hình 3, mặc dù kết quả của chúng là giống nhau.

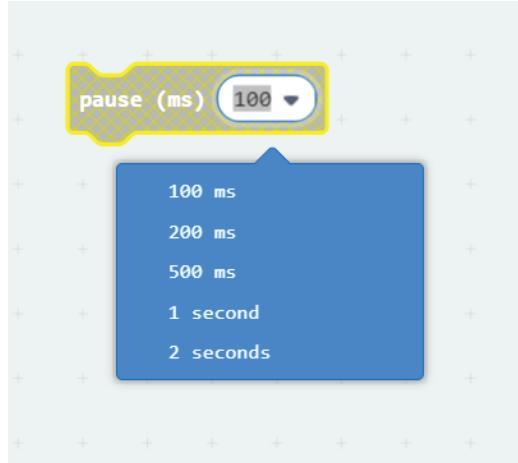
2.3. Câu lệnh tạo hiệu ứng đợi pause

Trong bài này, để có thể hiển thị nhiều nội dung hấp dẫn, chúng ta cần sử dụng thêm câu lệnh đợi. Câu lệnh này có tên là pause và cũng nằm cùng nhóm Basic với câu lệnh show leds.



Hình 5: Câu lệnh tạo hiệu ứng đợi

Đơn vị thời gian trong câu lệnh này là mili giây. Chúng ta có thể thấy đây là câu lệnh có thể lựa chọn thông số bên trong bằng cách nhấn vào phím mũi tên bên phải như hình dưới đây:

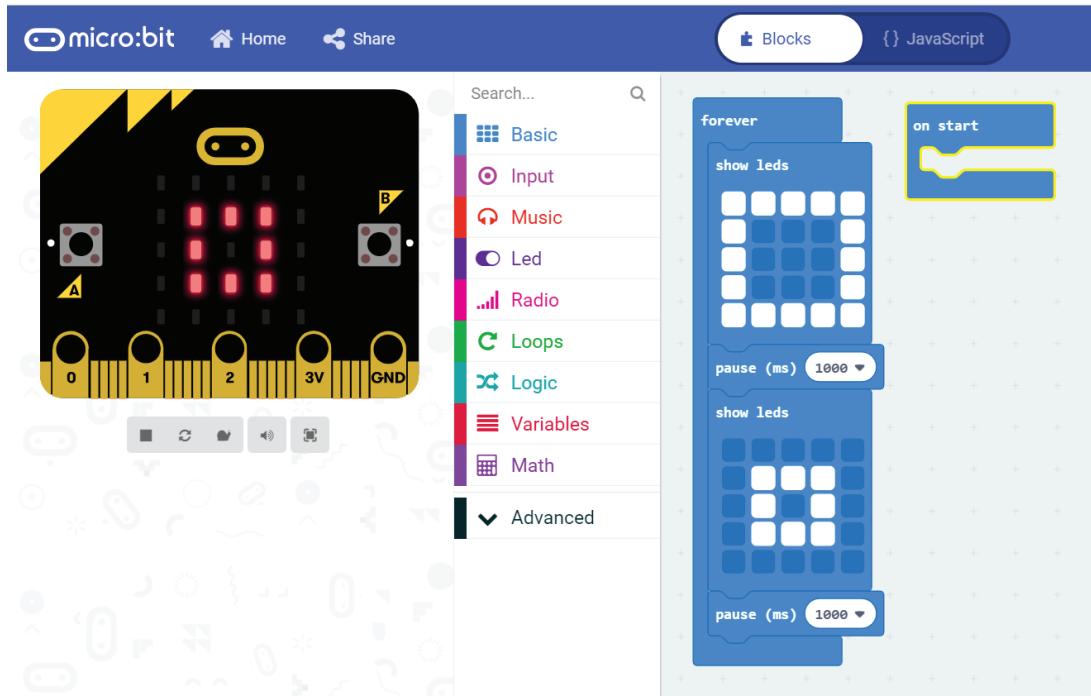


Hình 6: Các thông số được phép lựa chọn của câu lệnh đợi

Như hình bên trên, thời gian đợi tối đa chỉ là 2 giây mà thôi. Nếu chúng ta muốn tạo hiệu ứng đợi lâu hơn, có 2 cách sau đây:

- Ghép nhiều câu lệnh pause với nhau
- Gõ tay vào ô màu trắng thời gian mong muốn. Ví dụ muốn đợi 5 giây, chúng ta sẽ gõ vào 5000, do đơn vị ở đây là mili giây.

Một chương trình gợi ý về việc tạo ra một hiệu ứng hiển thị trên MicroBit như sau:

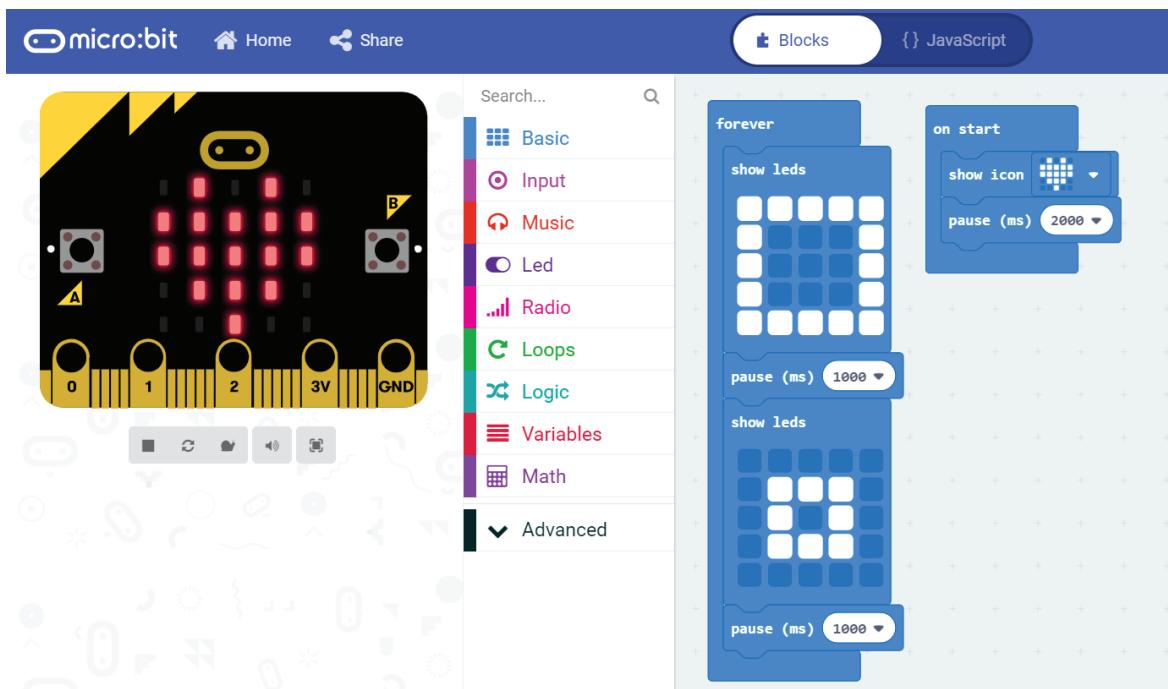


Hình 7: Một chương trình tạo hiệu ứng đơn giản

3. Bài tập trên lớp

Học sinh viết một chương trình hiển thị hình trái tim (câu lệnh show icon) trong 2 giây đầu tiên. Sau đó, thiết kế hiệu ứng pháo hoa lặp đi lặp lại trên màn hình hiển thị.

Đáp án: Do hình trái tim chỉ hiển thị 1 lần, nên câu lệnh show icon sẽ được dùng trong phần on start. Các câu lệnh tạo hiệu ứng pháo hoa sẽ được hiện thực trong phần forever. Một gợi ý cho chương trình này như sau:

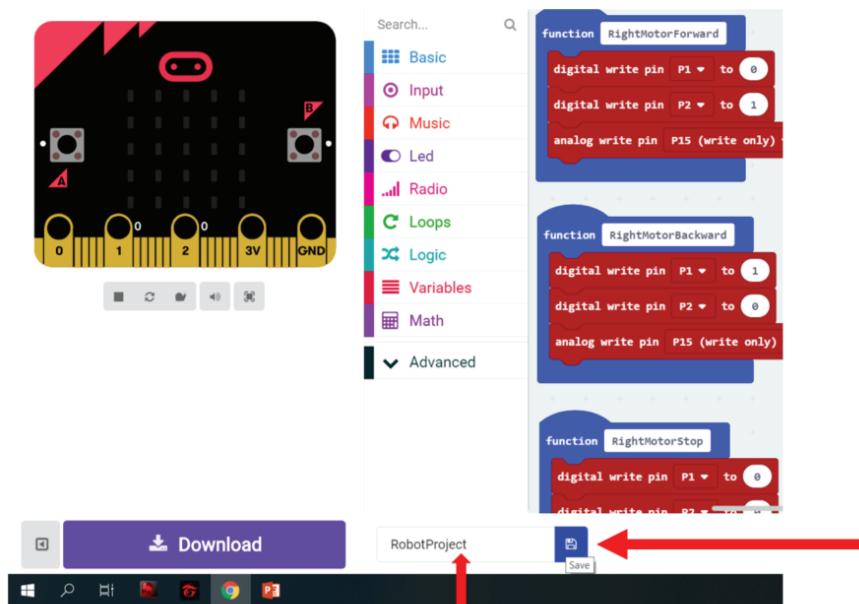


Hình 8: Đáp án gợi ý

4. Lưu và mở lại chương trình

Một nhu cầu tất yếu của việc lập trình là Lưu và Mở lại chương trình. Trước tiên, để lưu chương trình hiện tại, chúng ta có 2 bước cơ bản, như minh họa ở Hình 9:

1. Đặt tên cho chương trình cần lưu
2. Nhấn vào nút Save và chọn đường dẫn để lưu



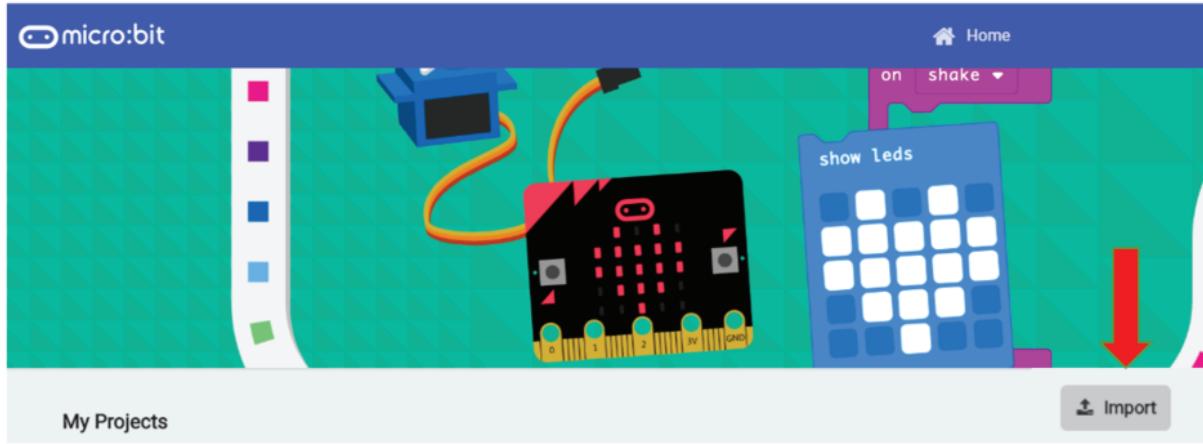
1. Đặt tên cho chương trình

2. Nhấn vào nút Save

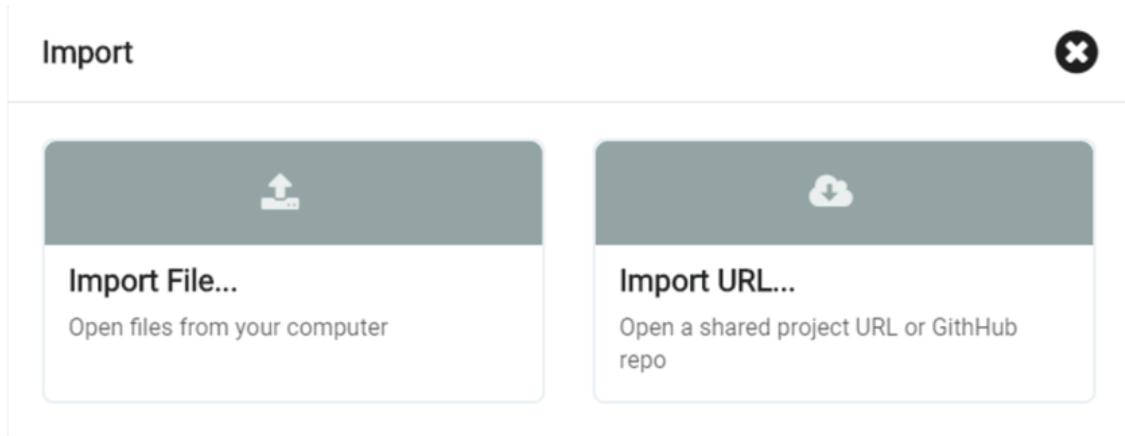
Hình 9: Các bước cơ bản để lưu chương trình

Lưu ý: Chương trình sẽ được lưu lại với định dạng file .hex

Để mở lại chương trình cũ và tiếp tục lập trình, từ màn hình chính, chúng ta chọn Import ở góc bên phải màn hình, như mô tả ở Hình 10:



Hình 10: Chọn vào Import để mở lại chương trình cũ

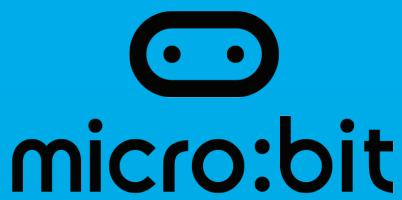


Hình 11: Chọn tiếp Import File và chọn tới đường dẫn file đã lưu trước đó

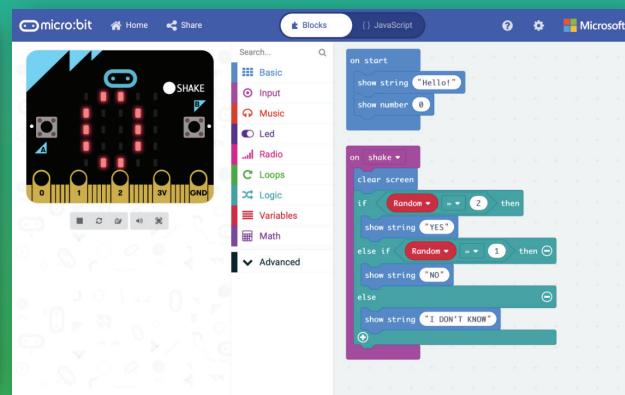
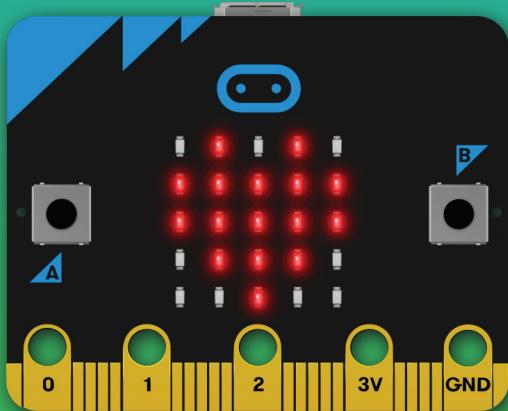
Một giao diện như Hình 11 sẽ hiện ra, chúng ta chọn tiếp Import File và chọn tới đường dẫn file hex đã lưu trước đó.

5. Bài tập về nhà

Học sinh thiết kế các hiệu ứng đẹp trên màn hình Microbit, lưu lại chương trình và trình bày cho cả lớp vào hôm sau.



Bài 3: Tổng hợp các câu lệnh về hiển thị



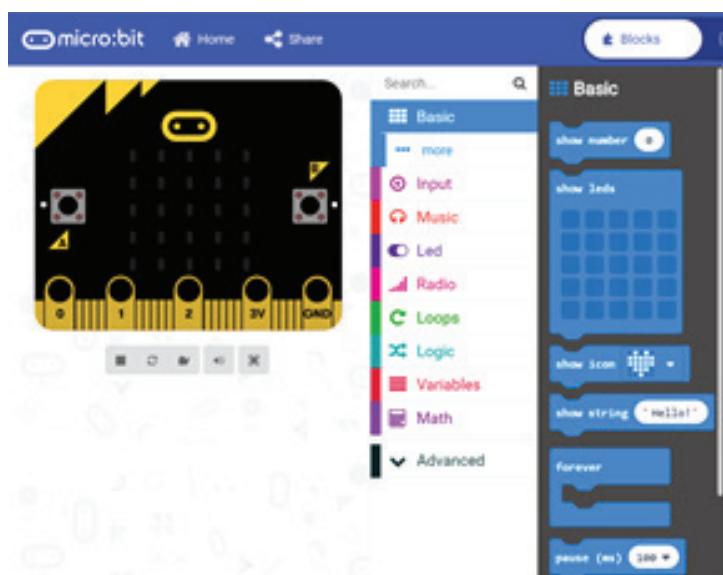
1. Mục tiêu

- Học sinh tự tìm hiểu các câu lệnh về hiển thị trên MicroBit
- Học sinh có khả năng phối hợp nhiều câu lệnh
- Học sinh nắm vững được nguyên lý hiển thị trên MicroBit

2. Nội dung chi tiết

2.1. Giới thiệu tổng quan về nhóm lệnh hiển thị

Các câu lệnh về hiển thị được nêu chính trong mục Basics. Tuy nhiên khi bấm vào nút more ngay bên dưới, chúng ta có thể 2 câu lệnh thuộc nhóm này nữa. Tất cả các câu lệnh này đều được MicroBit hỗ trợ để người dùng có thể hiển thị nhiều hình ảnh đẹp trên màn hình gồm 25 LED đơn.



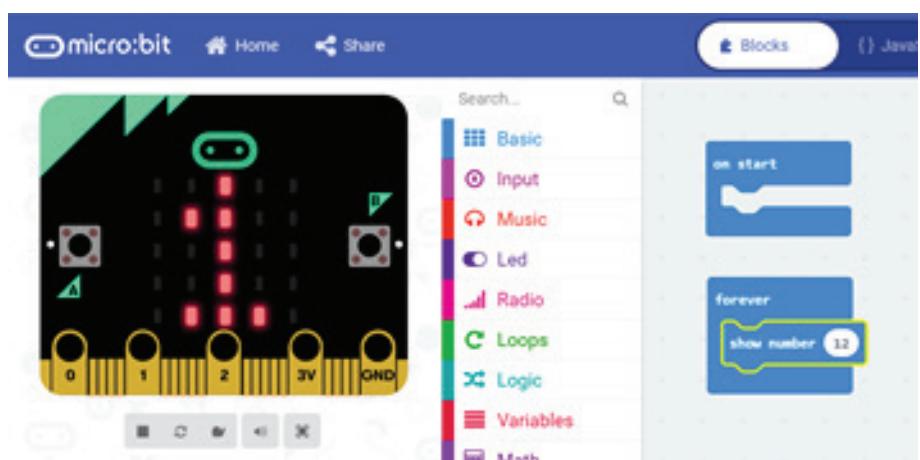
Hình 1: Các câu lệnh về hiển thị được nêu trong mục Basics

3. Giới thiệu chức năng của các câu lệnh

Cũng giống như các câu lệnh Scratch, chúng ta chỉ cần kéo thả câu lệnh này và đặt vào khung lệnh on start hoặc forever, thì nó sẽ được thực thi. Dựa vào kết quả của màn hình mô phỏng, chúng ta có thể tự tìm hiểu chức năng của câu lệnh này. Phần bên dưới sẽ trình bày chi tiết chức năng của từng câu lệnh. Tuy nhiên giáo viên cũng có thể cho học sinh tự tìm hiểu mà không cần phải trình bày trước các nội dung bên dưới.

3.1. Câu lệnh show number

Câu lệnh này sẽ hiển thị một số ra màn hình. Tuy nhiên, nó chỉ hiện ra số có 1 chữ số từ 0 đến 9. Nếu số này có 2 chữ số trở lên, nó sẽ chạy ngang qua bên trái.

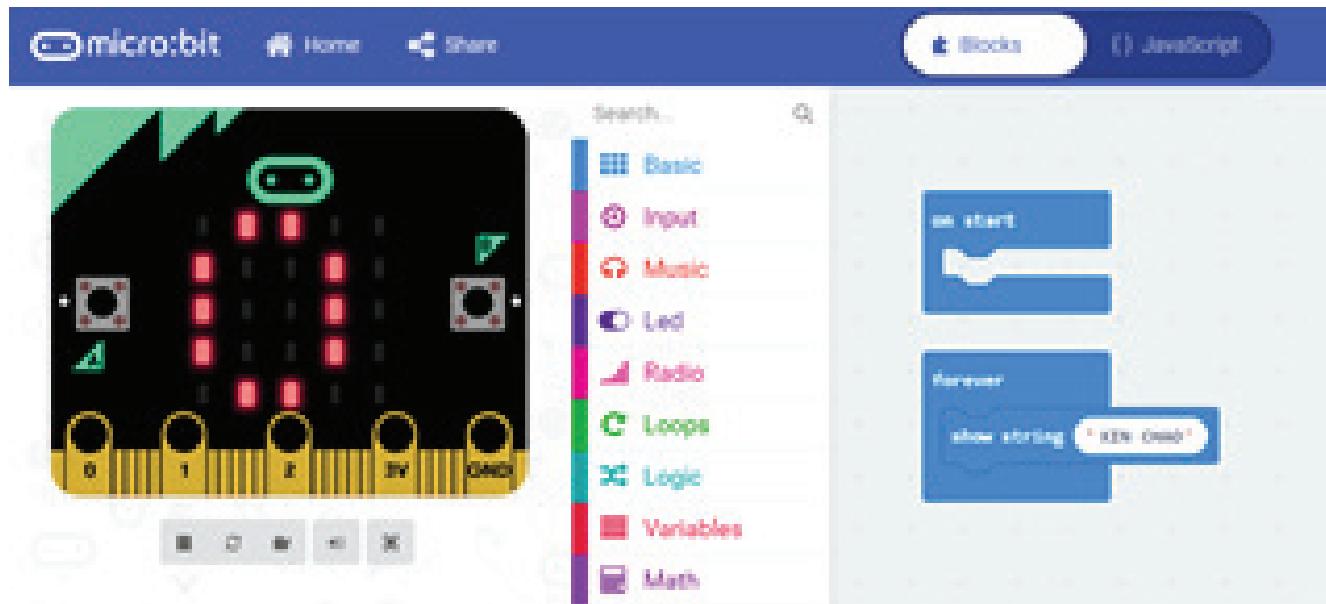


Hình 2: Ví dụ về câu lệnh show number

Ví dụ ở trên hiển thị số 12 ra màn hình. Tuy nhiên nó chỉ hiển thị ra số 1, sau đó dịch chuyển dần sang bên trái để hiển thị tiếp số 2.

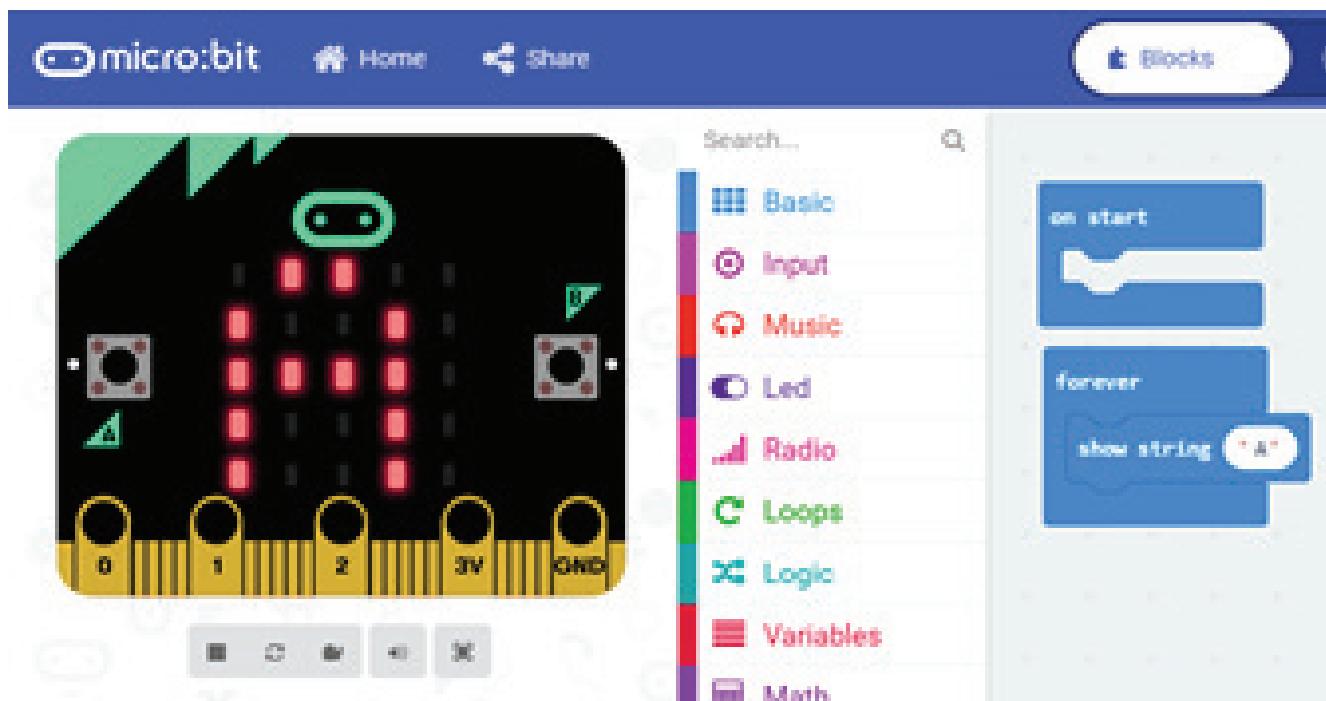
3.2. Câu lệnh show string

Hai câu lệnh tiếp theo là show leds và show icon đã được trình bày ở 2 bài trước, nên sẽ không trình bày lại ở bài này. Câu lệnh tiếp theo, show string sẽ cho phép một chuỗi dài các kí tự dịch chuyển sang trái. Tuy nhiên, câu lệnh này không hỗ trợ tiếng việt có dấu.



Hình 3: Ví dụ về câu lệnh show string

Tuy nhiên, nếu tham số bên trong câu lệnh show string này chỉ có 1 kí tự, nó chỉ hiển thị kí tự này ra màn hình mà thôi và không có hiệu ứng dịch chữ sang bên trái. Ví dụ ở chương trình bên dưới, chỉ hiển thị một kí tự A ra màn hình mà thôi.



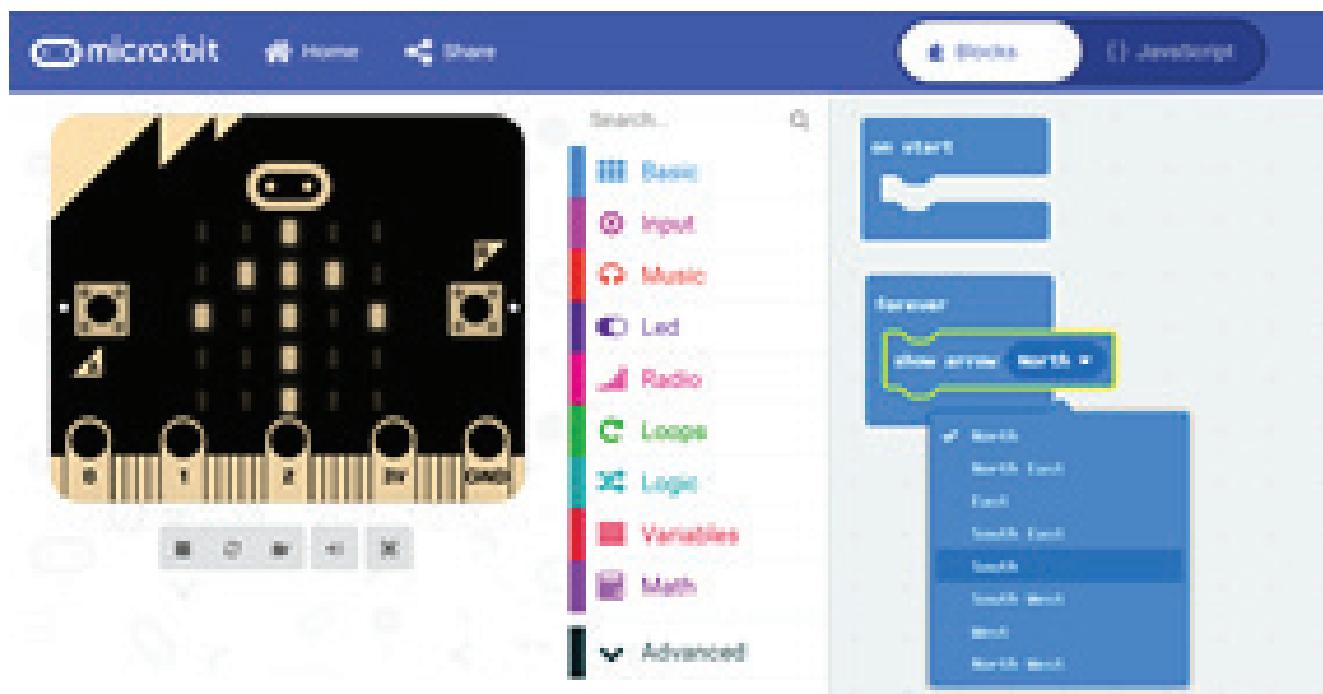
Hình 4: Câu lệnh show string khi có 1 kí tự sẽ không có hiệu ứng dịch trái

3.3. Câu lệnh clear screen

Đây là câu lệnh đầu tiên phần more. Chức năng câu lệnh này là xóa toàn bộ màn hình.

3.4. Câu lệnh show arrow

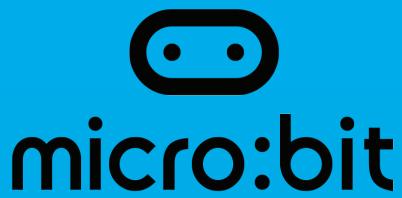
Cũng giống như câu lệnh show icon, câu lệnh này định nghĩa sẵn một số hình ảnh về mũi tên. Người dùng cũng có thể chọn từ danh sách hỗ trợ sẵn của MicroBit.



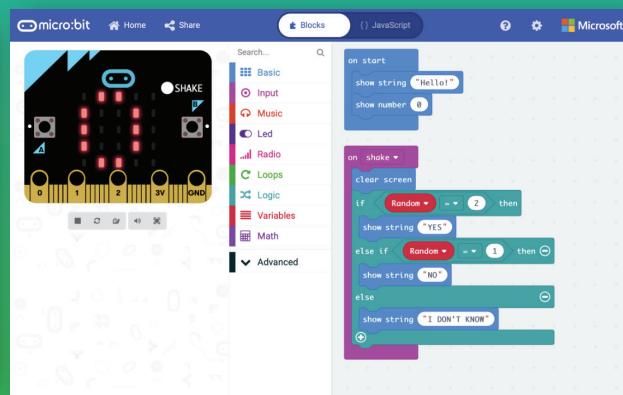
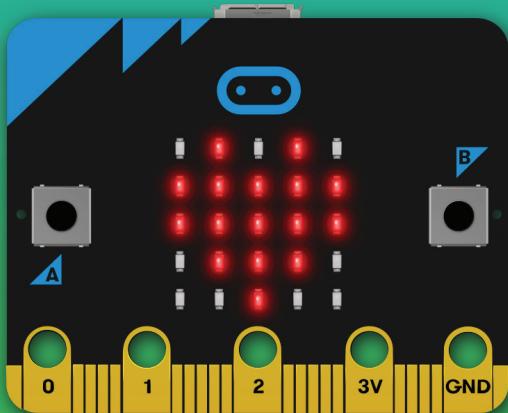
Hình 5: Câu lệnh show arrow

4. Bài tập trên lớp

Học sinh phối hợp các câu lệnh để tạo ra các hiệu ứng đẹp trên màn hình hiển thị.



Bài 4: Điều khiển nút nhấn trên MicroBit



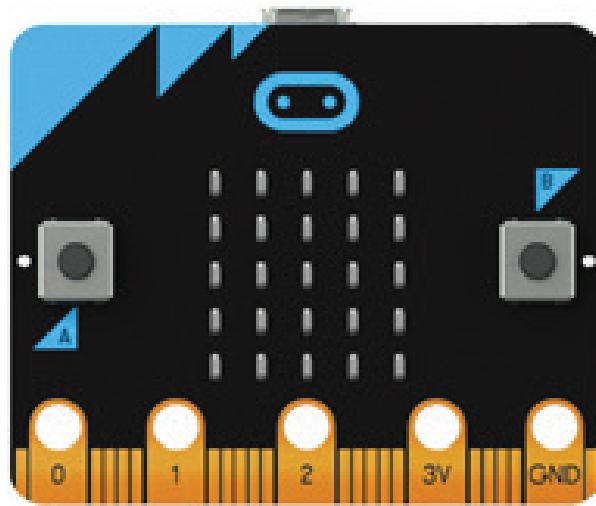
1. Mục tiêu

- Học sinh hiểu được nguyên lý hoạt động của nút nhấn
- Học sinh có khả năng sử dụng các câu lệnh liên quan đến nút nhấn
- Học sinh kết hợp được nhiều câu lệnh trên MicroBit

2. Nội dung chi tiết

2.1. Giới thiệu tổng quan về nút nhấn

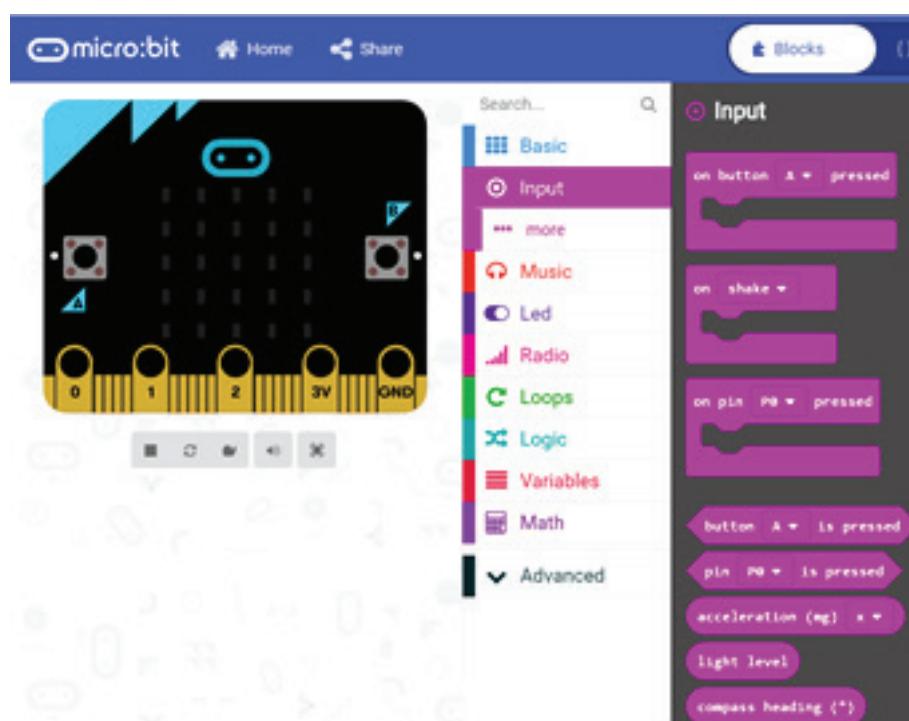
Trên mạch MicroBit được hỗ trợ sẵn 2 nút nhấn, có tên gọi là A và B như Hình 1 bên dưới. Cũng giống như bàn phím máy tính, hai nút nhấn này đóng vai trò là các thiết bị nhập, gửi dữ liệu đầu vào.



Hình 1: Hai nút nhấn A và B trên MicroBit

3. Lập trình điều khiển nút nhấn

Các câu lệnh để điều khiển nút nhấn thuộc nhóm Input, nhóm thứ 2 sau nhóm Basic.



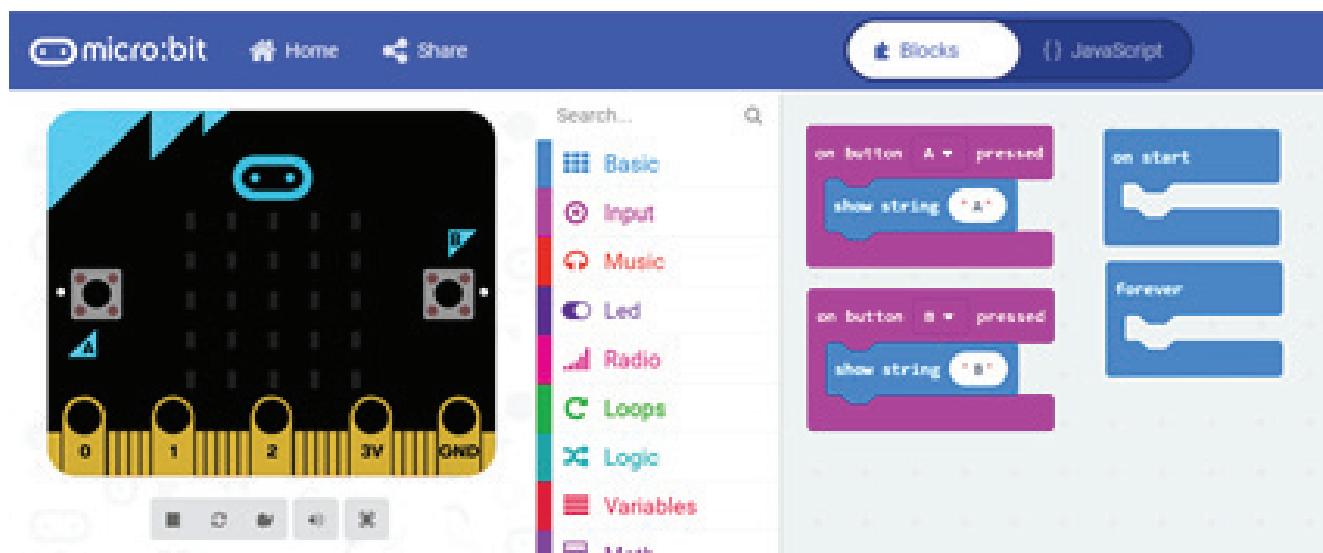
Hình 2: Các câu lệnh thuộc nhóm Input để điều khiển nút nhấn

Khác với kiến trúc chương trình ở các bài trước, các câu lệnh được thực hiện đầu tiên trong khối on start, sau đó chuyển qua phần forever và lặp đi lặp lại, việc điều khiển nút nhấn được thực hiện trong các sự kiện. Đây là một kiến trúc lập trình rất hiện đại, cải thiện rất nhiều hiệu suất của hệ thống.

Chúng ta hiện thực một chương trình đơn giản đầu tiên, khi nhấn phím A sẽ hiện ra chữ A trên màn hình, nhấn phím B sẽ hiện ra chữ B. Câu lệnh chính trong yêu cầu này là on button pressed, như hình bên dưới.

Trong câu lệnh này, chúng ta có thể lựa chọn sự kiện tương ứng cho từng nút nhấn A, B hoặc cả 2 nút nhấn cùng một lúc.

Chương trình của chúng ta đơn giản như sau:



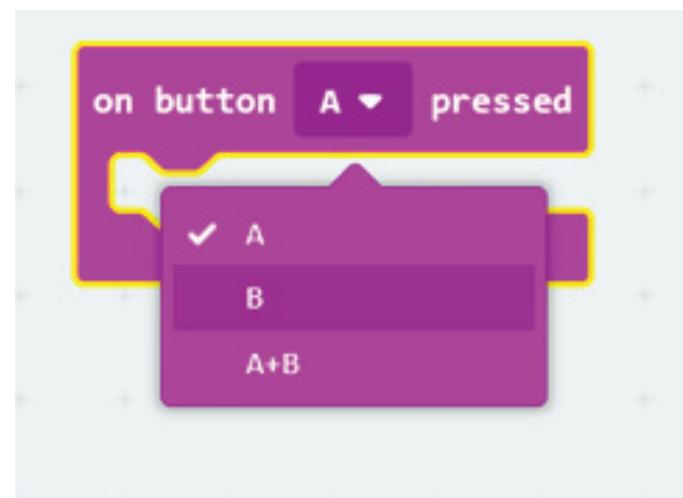
Hình 4: Chương trình đơn giản tương tác với nút nhấn

Như chúng ta có thể thấy, chương trình thực hiện hoàn toàn độc lập với 2 khối lệnh on start và forever. Những chương trình này hoàn toàn có thể mô phỏng được trực tuyến.

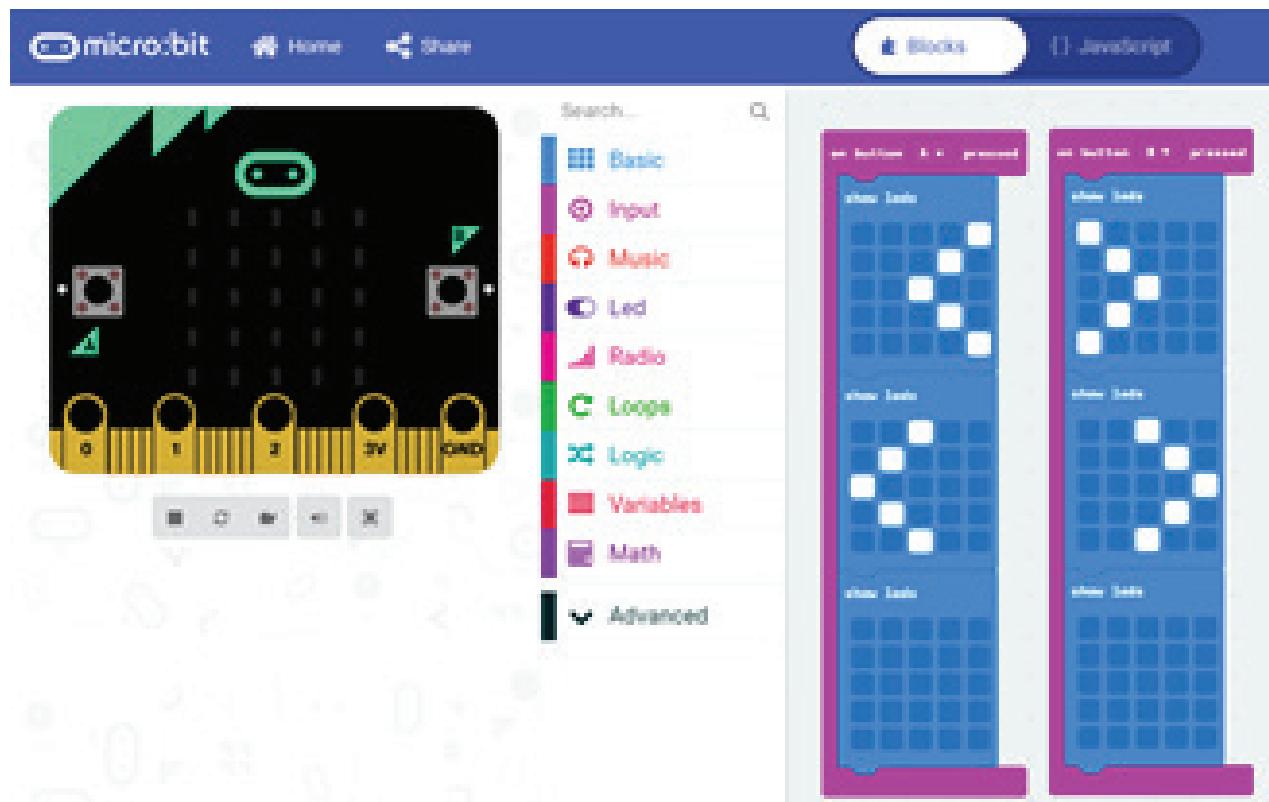
4. Bài tập trên lớp

Thiết kế một hiệu ứng là dấu mũi tên. Khi nhấn phím A, phím mũi tên này di chuyển sang trái. Khi nhấn phím B, phím mũi tên này di chuyển sang phải. Đáp án gợi ý của chương trình như Hình 5.

Giải thích: Hiệu ứng thực ra là nhiều màn hình hiển thị xuất hiện cách nhau 1 khoảng thời gian đủ nhỏ. Chẳng hạn như video mà chúng ta đang xem, thực ra là hình tĩnh cách nhau 1/30s (30 hình một giây). Chúng ta sẽ sử dụng nguyên lý này để định nghĩa ra hàng loạt các khung hình bằng câu lệnh show leds. Sau đó đặt các màn hình này liên tiếp nhau, chúng ta sẽ thấy một hiệu ứng.



Hình 3: Câu lệnh sự kiện cho nút nhấn
với 3 lựa chọn khác nhau

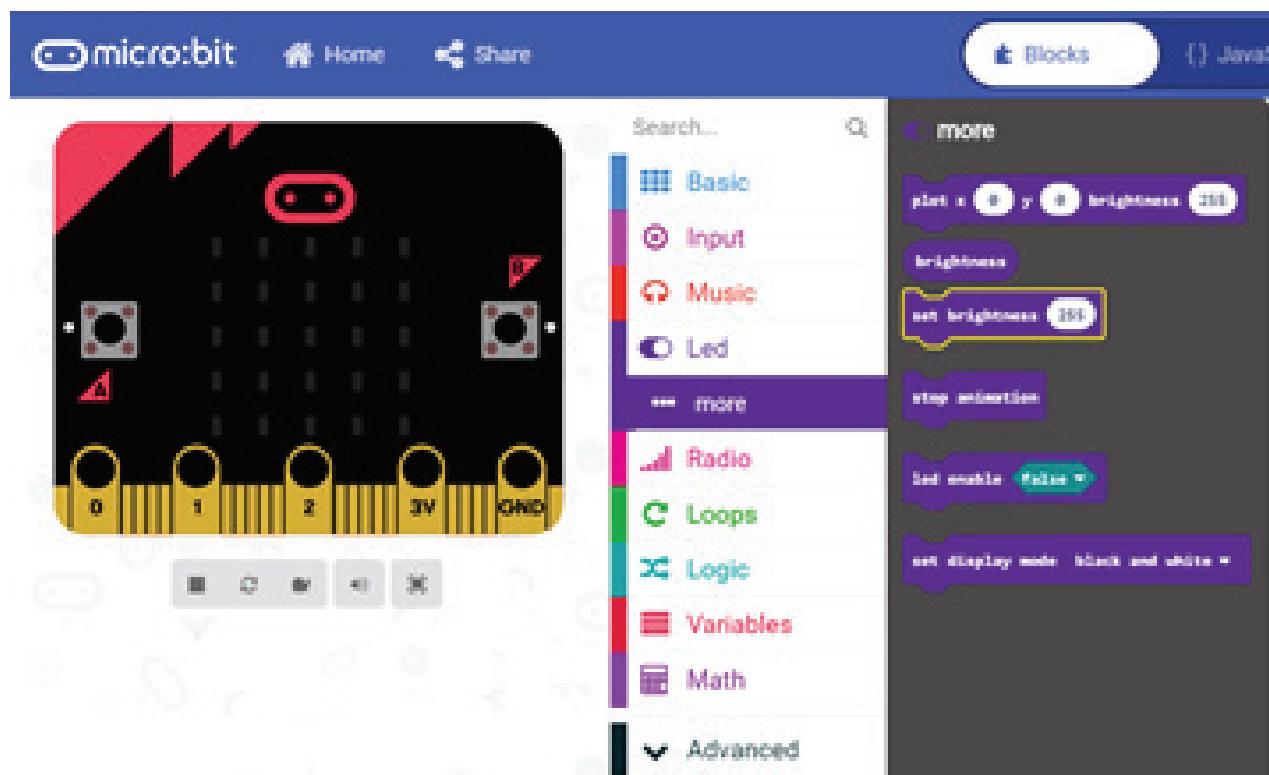


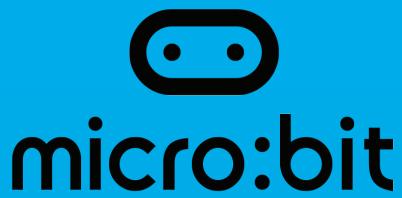
Hình 5: Hiệu ứng mũi tên qua trái qua phải

4. Bài tập trên lớp

Học sinh tự thiết kế các hiệu ứng khác nhau ví dụ chữ dịch lên, dịch xuống, chữ nhấp nháy và thay đổi hiệu ứng của chữ bằng cách nhấn phím A hoặc B hoặc kết hợp cả 2 phím. Học sinh trình bày kết quả của mình vào buổi hôm sau.

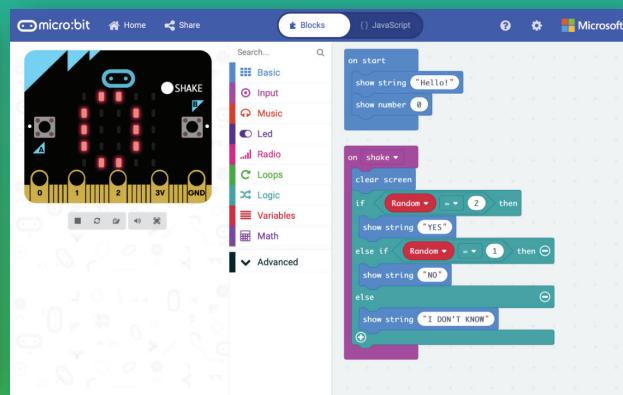
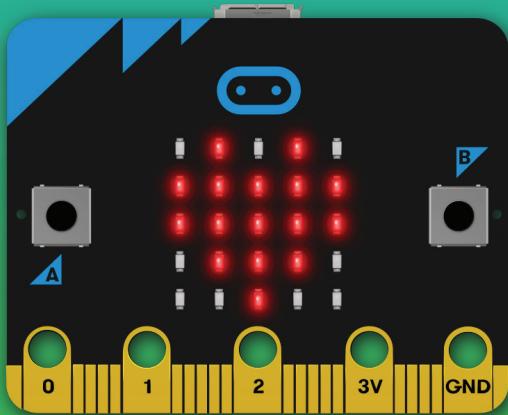
Để có thể làm thêm các hiệu ứng đẹp, học sinh có thể sử dụng thêm 1 lệnh trong phần Led, chọn tiếp vào more. Chúng ta có lệnh set brightness để chỉnh độ sáng của mà hình hiển thị. Học sinh có thể dùng câu lệnh này để cho chữ sáng từ mờ sang rõ hoặc ngược lại.





Bài 5:

Lập trình MicroBit bằng điện thoại thông minh



1. Mục tiêu

- Học sinh nắm được thao tác kích hoạt mạch MicroBit vào chế độ Bluetooth
- Học sinh có thể viết và nạp chương trình bằng điện thoại Android
- Học sinh có thể viết và nạp chương trình bằng điện thoại iOS

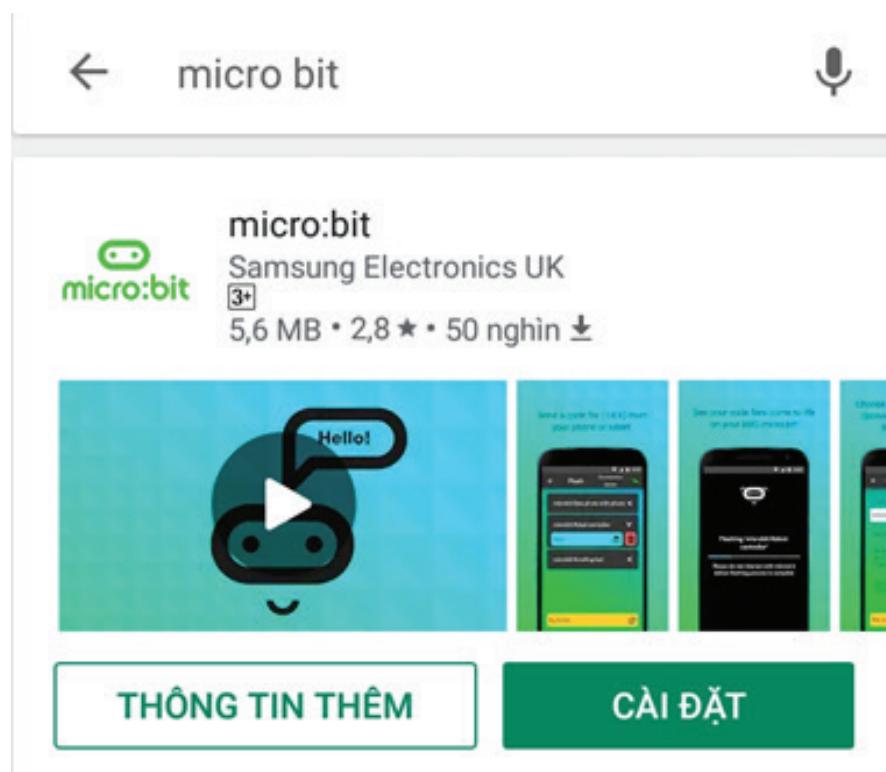
2. Nội dung chi tiết

2.1. Giới thiệu

Trong các bài đã giới thiệu, chúng ta đang lập trình bằng cách dùng máy tính, kết nối vào mạng Internet và vào trang web www.makecode.makecode.org để lập trình. Tuy nhiên trong 1 số trường hợp, việc sử dụng máy tính là khá cồng kềnh và không thuận tiện. Trong những trường hợp này, chúng ta có thể sử dụng điện thoại thông minh để lập trình cho mạch MicroBit. Mặc dù hiện tại, các thao tác trên điện thoại còn chưa được thuận tiện, tuy nhiên nó cũng là một công nghệ mới, đáng ghi nhận của board mạch MicroBit. Trong bài hướng dẫn này, chúng tôi sẽ trình bày các bước cơ bản để lập trình cho mạch MicroBit bằng điện thoại thông minh, cho cả 2 nền tảng là Android và iOS.

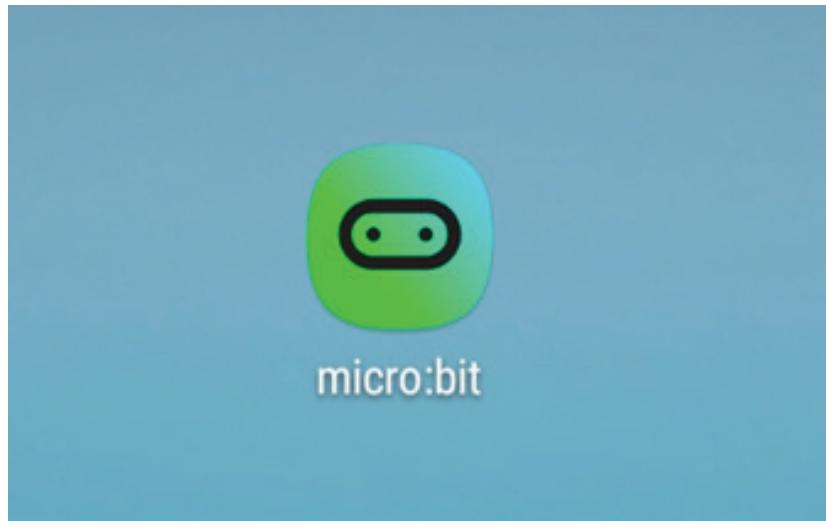
2.2. Lập trình bằng điện thoại Android

Để có thể lập trình trên Android, chúng ta cần vào kho ứng dụng Google Play để tải phần mềm micro:bit. Chúng ta sẽ tìm kiếm bằng từ khóa microbit như hình minh họa bên dưới, sau đó nhấn nút CÀI ĐẶT để tải ứng dụng này về.

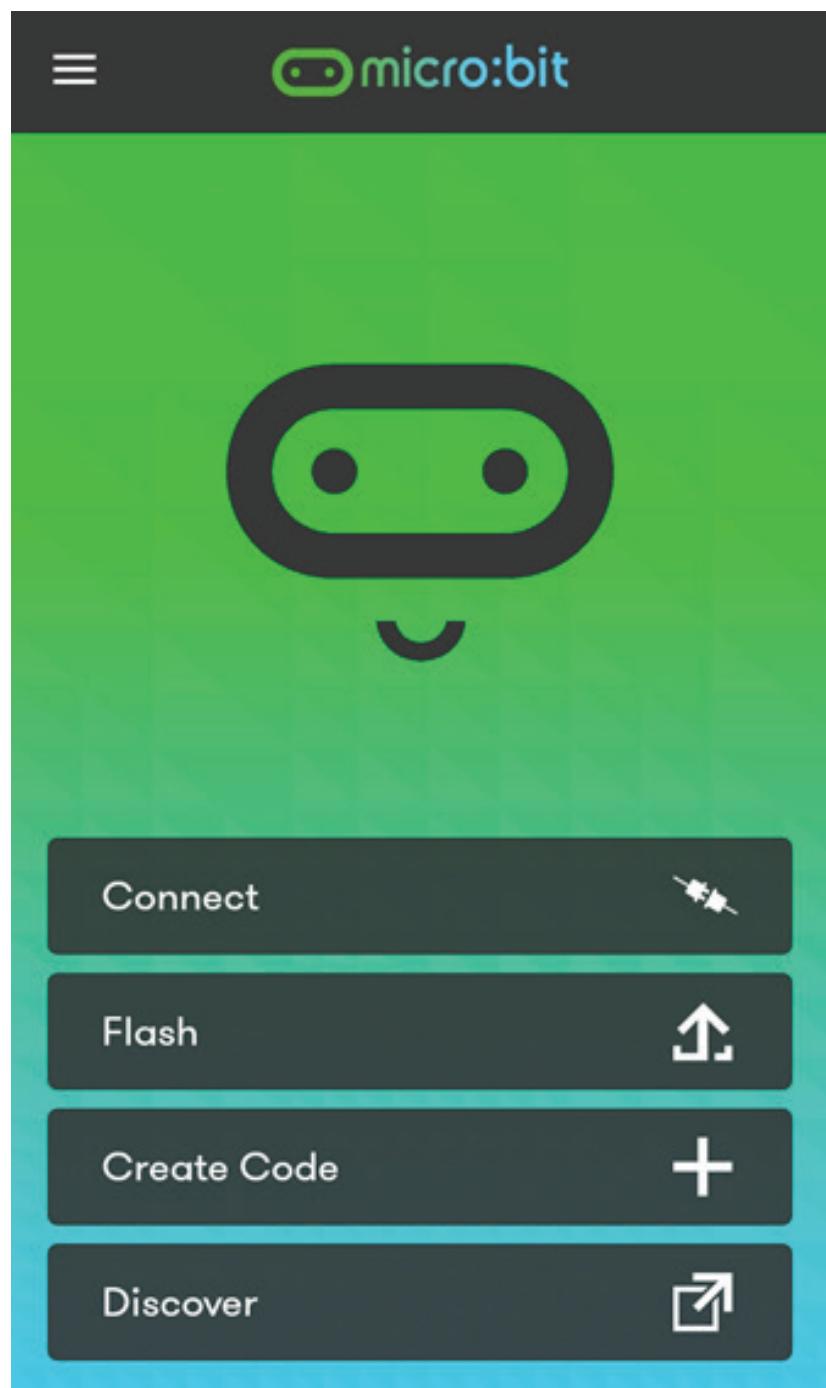


Hình 1: Tải và cài đặt ứng dụng MicroBit

Khi cài đặt xong, trên màn hình điện thoại sẽ xuất hiện ứng dụng để lập trình cho mạch MicroBit (xem Hình 2). Khi nhấn vào icon này, ứng dụng sẽ được mở ra như Hình 3. Bạn hãy chọn Đồng ý hoặc Cho phép, để ứng dụng được cấp quyền thực thi trên điện thoại Android



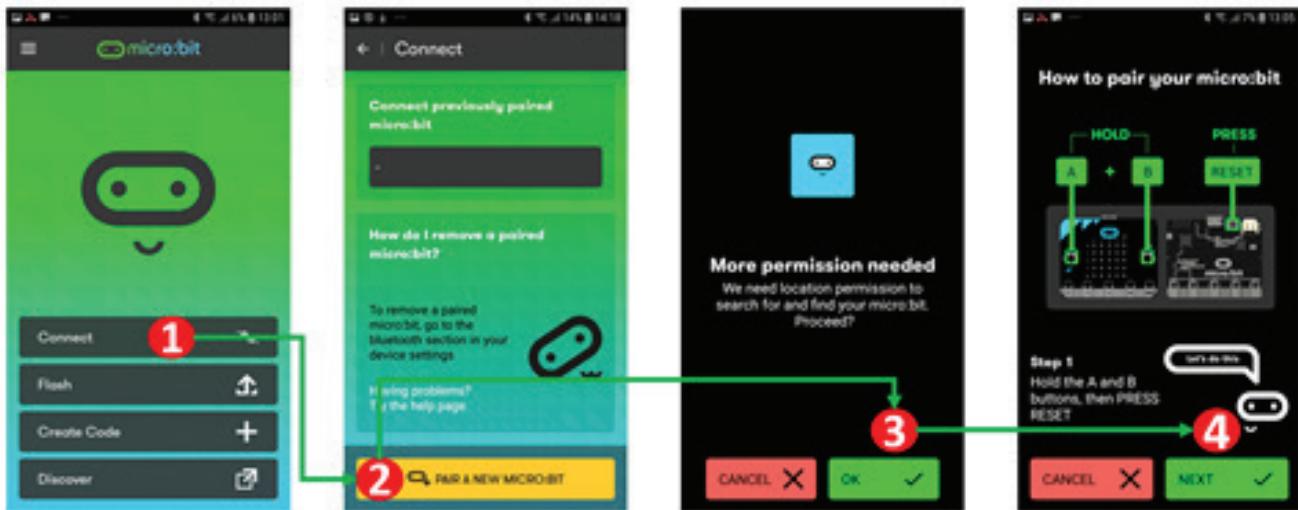
Hình 2: Icon của ứng dụng MicroBit



Hình 3: Giao diện của ứng dụng

2.2.1 Ghép đôi điện thoại và mạch MicroBit

Đây là bước đầu tiên mà chúng ta phải làm để có thể lập trình trên MicroBit bằng điện thoại. Do việc nạp chương trình được dựa trên nền tảng giao tiếp Bluetooth. Tuy nhiên bước này chỉ cần thực hiện một lần mà thôi. Để có thể ghép đôi với mạch MicroBit, chúng ta cần bật nguồn cho mạch MicroBit. Sau đó, nhấn vào nút Connect trên điện thoại.



Hình 4: Các thao tác trên điện thoại để ghép đôi thiết bị

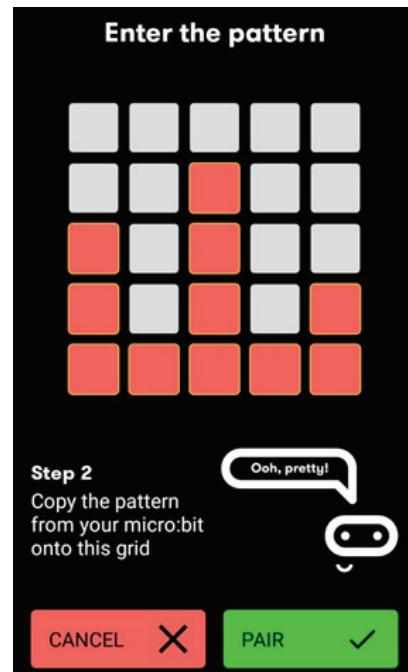
Trình tự ghép đôi được minh họa ở Hình 4, được mô tả như sau:

- Bước 1: Nhấn vào nút Connect
- Bước 2: Nhấn tiếp vào nút PAIR A NEW MICRO:BIT
- Bước 3: Nhấn vào nút OK hoặc Đồng ý để cấp quyền cho ứng dụng

Màn hình ở Bước 4 sẽ hiện lên, cung cấp hướng dẫn cho chúng ta để kích hoạt chế độ Bluetooth trên board mạch MicroBit. Chúng ta nhấn giữ đồng thời 2 nút A và B trong khoảng 2 giây, sau đó nhấn vào nút Reset, thả nút Reset nhưng vẫn tiếp tục giữ 2 nút A và B. Chúng ta sẽ thấy một số hiệu ứng hiển thị trên màn hình MicroBit cho đến khi màn hình ổn định ở một ký hiệu. Đến lúc này, chúng ta có thể thả 2 nút A và B ra.

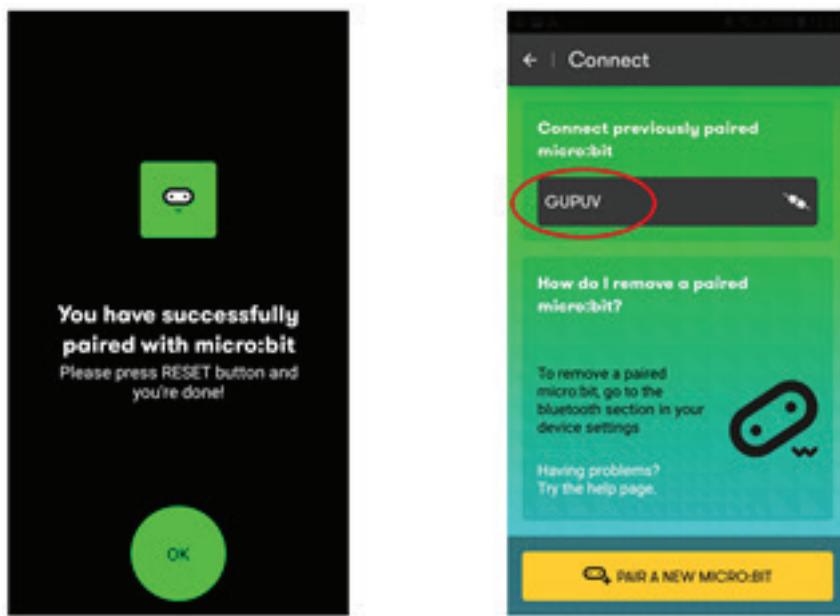
Toàn bộ quy trình ở trên là để kích hoạt chế độ ghép nối Bluetooth. Màn hình cuối cùng chính là mật mã để kết nối với board mạch MicroBit. Đến lúc này, chúng ta có thể nhấn vào nút Next ở bước 4 trên điện thoại.

Giao diện sau đây trên điện thoại sẽ hiện ra, để chúng ta nhập mật mã.



Hình 5: Nhập mật mã kết nối với MicroBit

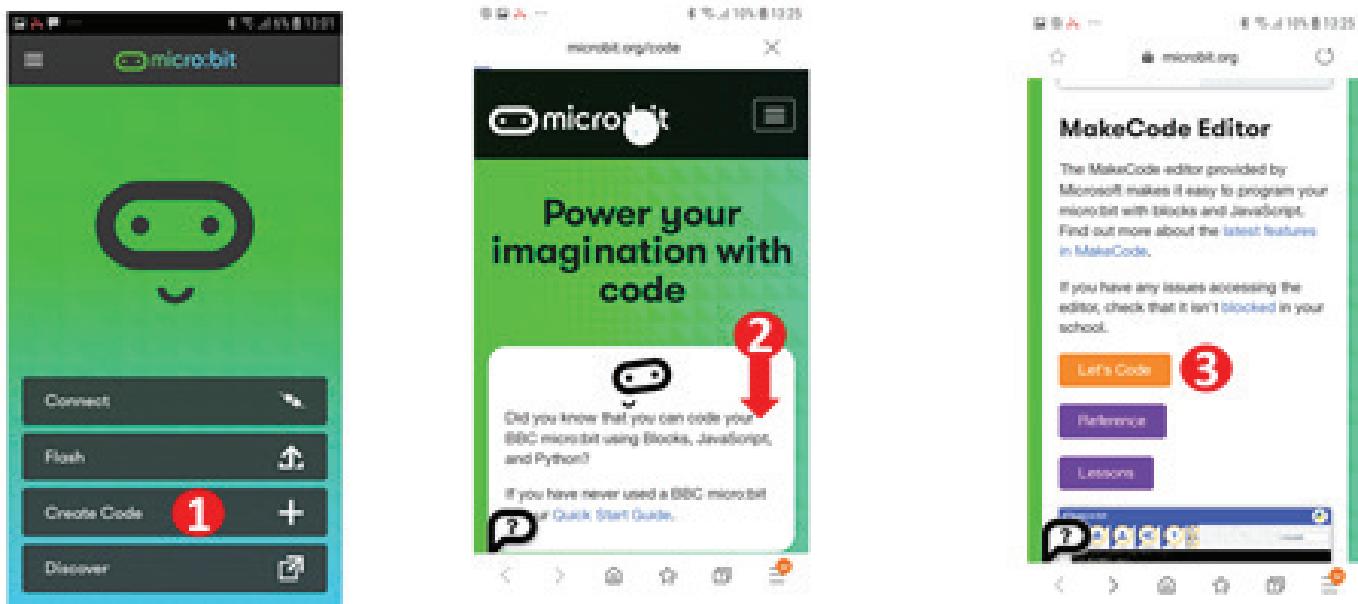
Hãy để ý hình dạng của các bóng đèn trên MicroBit, sau đó nhấp vào màn hình điện thoại để có hình dạng tương ứng. Cuối cùng nhấn nút PAIR. Khi việc kết nối là thành công, màn hình sau đây sẽ hiện ra, chúng ta có thể nhấn nút OK. Một màn hình khác sẽ hiện ra, cùng với tên của board mạch MicroBit của chúng ta.



Hình 6: Kết nối thành công, tên thiết bị hiện ra trên mục ghép nối

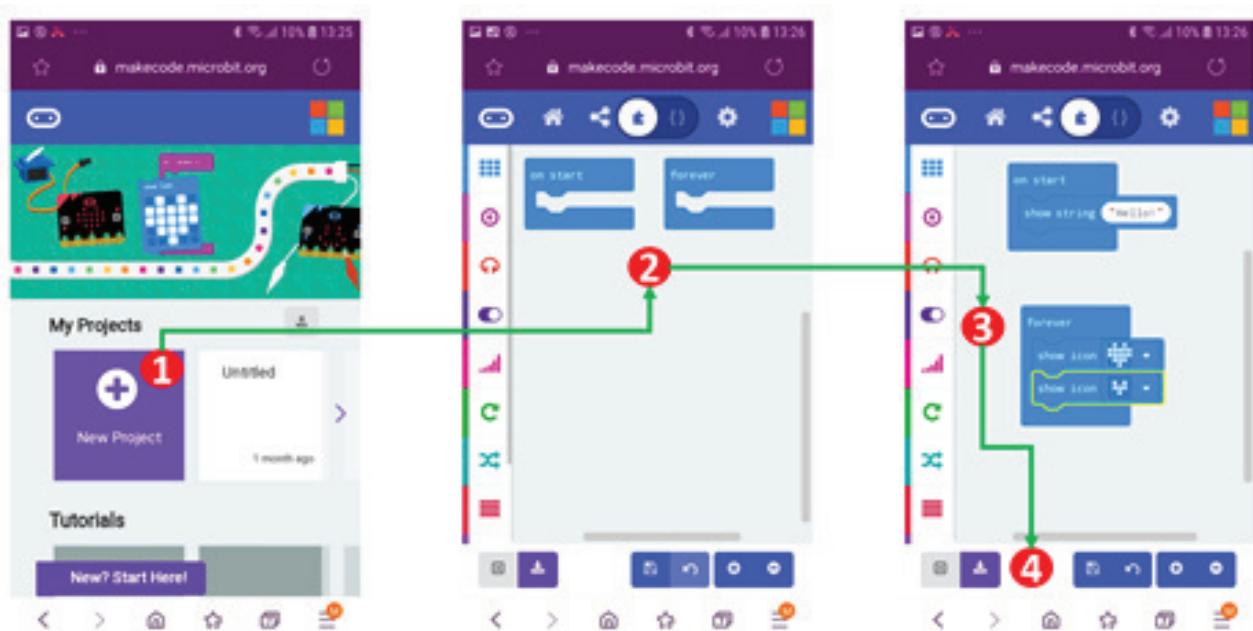
2.2.2. Lập trình trên điện thoại

Sau khi ghép đôi xong, chúng ta có thể nhấn nút BACK trên điện thoại để trở về giao diện chính. Tại đây, chúng ta có thể nhấn vào nút Create Code để bắt đầu lập trình cho mạch MicroBit.



Hình 7: Bắt đầu lập trình cho MicroBit

Toàn bộ quá trình này được minh họa ở Hình 7, nhấn vào nút Create Code (Bước 1), kéo màn hình xuống phần MakeCode Editor (Bước 2), và chọn Let's Code (Bước 3).



Hình 8: Các bước để viết chương trình trên điện thoại

Giao diện để lập trình trên điện thoại hoàn toàn tương tự với phiên bản trên máy tính, chúng ta cũng chọn vào New Project (Bước 1), một giao diện mặc định sẽ hiện ra để bắt đầu lập trình (Bước 2). Chúng ta chọn các câu lệnh và ghép nối chúng lại (Bước 3). Cuối cùng nhấn vào nút Tải về (Bước 4) để tải chương trình về điện thoại. Chúng ta cũng có thể đổi tên file tải về cho gợi nhớ (xem Hình 9). Trong ví dụ này, chúng tôi đổi file hex mặc định có tên là test.hex

Tải file về?

Đã có một file với tên bên dưới. Hãy đổi tên file.

Kích thước: 595,4KB

Tên
microbit-Untitled.hex

Tên file đã được sử dụng.

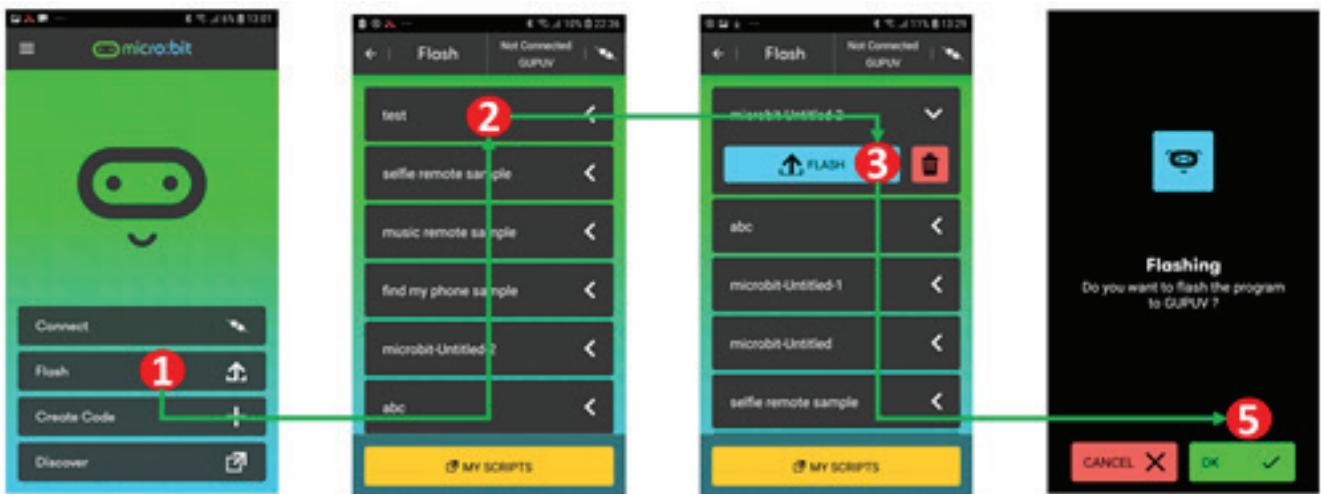
Thoát

Tải về

Hình 9: Đặt tên cho chương trình tải về

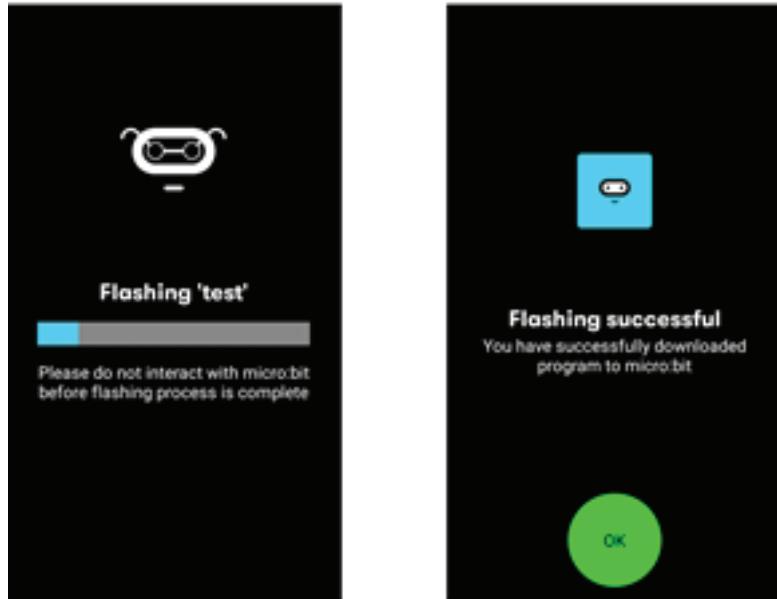
2.2.3 Nạp chương trình vào mạch MicroBit

Để nạp chương trình cho mạch MicroBit, chúng ta trở về màn hình chủ của điện thoại (nhấn nút HOME hoặc nhấn BACK nhiều lần). Từ màn hình chính, chúng ta có thể làm tuần tự như mô tả ở Hình 10.



Hình 10: Các bước để nạp chương trình cho MicroBit

Đầu tiên, chúng ta nhấn vào nút Flash, sau đó chọn vào file hex đã lưu trước đó (trong ví dụ này là file test). Tiếp theo chúng ta nhấn và chọn Flash. Tuy nhiên, sau bước thứ 3 này, chúng ta cần phải bật mạch MicroBit sang chế độ Bluetooth, trước khi nhấn vào nút OK ở bước 5. Làm tương tự như bước ghép đôi mạch MicroBit và điện thoại, chúng ta nhấn đè 2 nút A và B, nhấn và thả nút Reset, vẫn tiếp tục nhấn đè 2 nút A và B cho tới khi hiện ứng báo hiệu kết nối Bluetooth sẵn sàng trên mạch MicroBit, chúng ta thả 2 nút A và B ra. Cuối cùng, chúng ta mới nhấn nút OK ở bước 5. Các màn hình dưới đây sẽ lần lượt xuất hiện, báo hiệu việc nạp chương trình cho mạch MicroBit thành công.

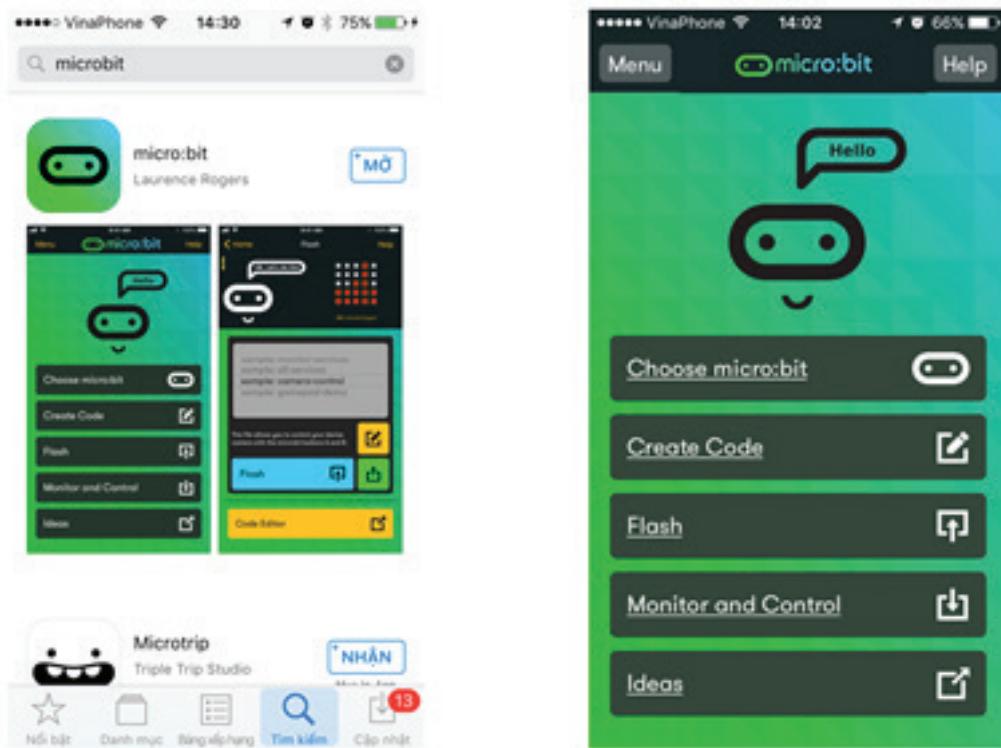


Hình 11: Quá trình tải chương trình lên mạch MicroBit và báo hiệu nạp thành công

Đến bước này, chương trình sẽ hỏi chúng ta có muốn tiếp tục kết nối với mạch MicroBit nữa hay không. Chúng ta có thể chọn NO vì việc này không cần thiết nữa. Nếu chương trình chạy không đúng ý, chúng ta sẽ quay lại mục 2 để soạn thảo lại chương trình, tải nó về và nạp lại chương trình. Trước khi nhấn OK để nạp, đừng quên trình tự nhấn đè 2 nút A và B, nhấn và thả nút Reset, chờ chế độ Bluetooth kích hoạt và thả 2 nút A và B.

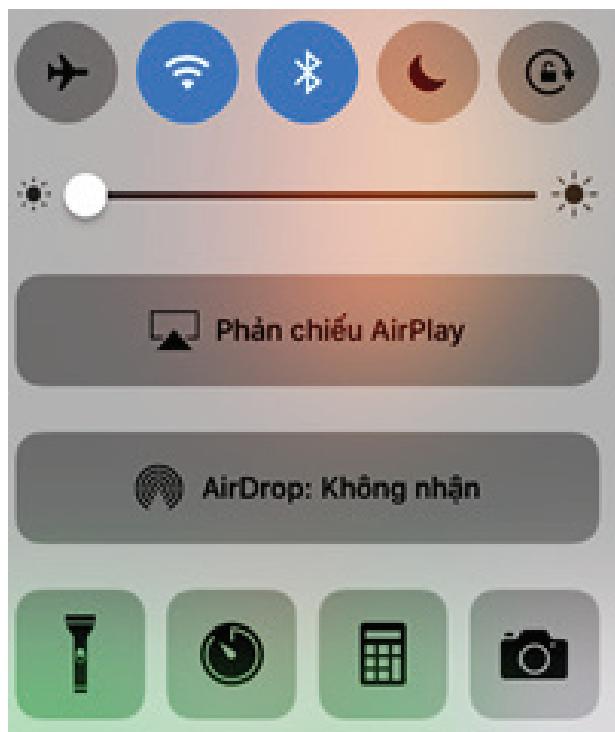
2.3. Lập trình bằng điện thoại iOS

Hoàn toàn tương tự với phiên bản chạy trên Android, chúng ta cũng vào kho ứng dụng AppStore, tìm kiếm từ khóa microbit để tải ứng dụng này về điện thoại. Sau khi tải và cài đặt, chúng ta có thể mở chương trình này lên. Giao diện của nó rất giống với phiên bản chạy trên Android.



Hình 12: Tìm kiếm và tải ứng dụng MicroBit về thiết bị iOS

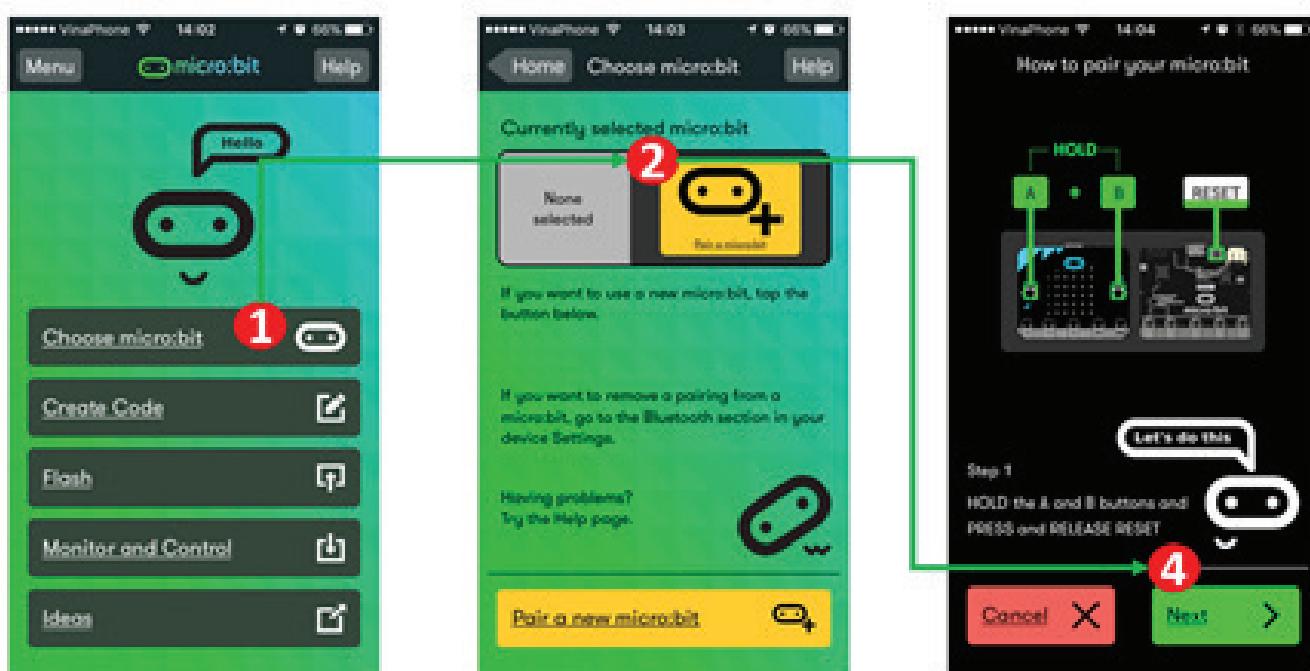
Tất nhiên, để có thể sử dụng được kết nối Bluetooth, chúng ta phải bật Bluetooth của điện thoại lên trước, bằng cách vào mục Setting, rồi bật kí hiệu Bluetooth (nằm bên cạnh kí hiệu wifi) lên như minh họa ở Hình 13.



Hình 13: Bật kết nối Bluetooth của điện thoại iPhone

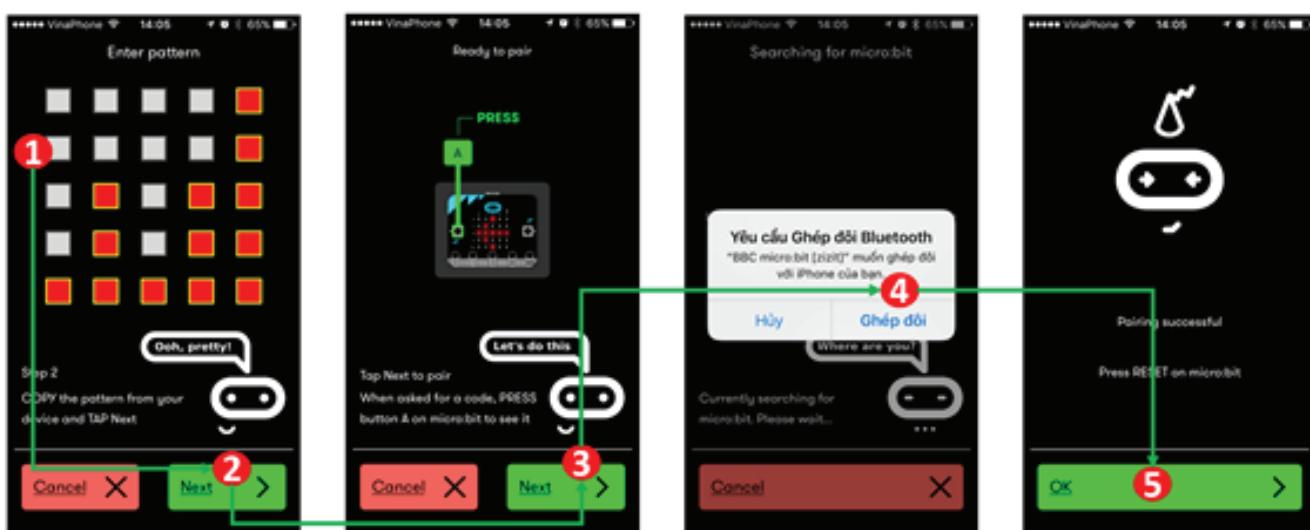
2.3.1. Ghép đôi thiết bị

Để thực hiện chức năng này, như minh họa ở Hình 12, chúng ta chọn vào lựa chọn đầu tiên là Choose micro:bit, chọn tiếp Pair a micro:bit. Tới đây, tương tự như khi làm việc với thiết bị Android. Chúng ta khoan bấm nút Next (Bước 4) mà chuyển sang mạch MicroBit, nhấn theo trình tự: Đè 2 nút A và B, nhấn và thả nút RESET, chờ cho đến khi tín hiệu kết nối Bluetooth kích hoạt, thả 2 nút A và B. Tới đây, chúng ta có thể quay lại điện thoại, để nhấn tiếp nút OK.



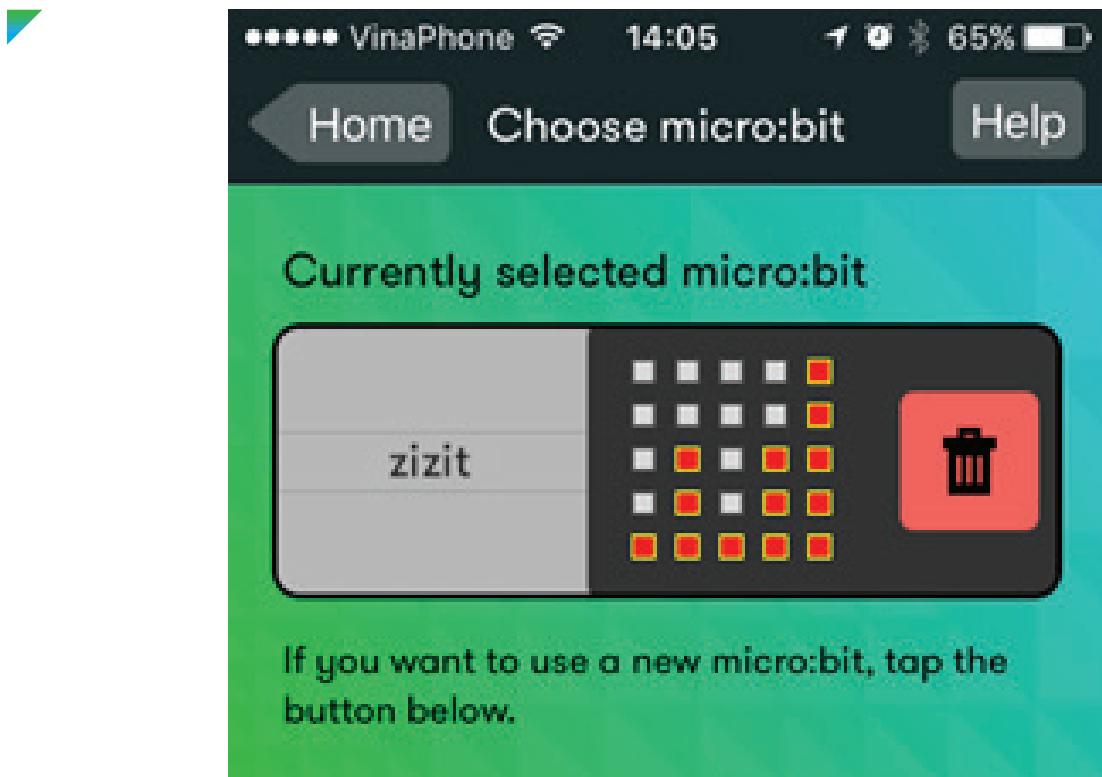
Hình 14: Ghép đôi một mạch MicroBit trên iOS

Các bước tiếp theo được minh họa ở Hình 15, bao gồm nhập mã hiển thị trên MicroBit (Bước 1), nhấn Next (Bước 2), tiếp tục nhấn Next (Bước 3), chọn Ghép đôi khi có thông báo hiển thị (Bước 4), và cuối cùng, chọn OK để kết thúc việc ghép đôi.



Hình 15: Nhập mã để ghép đôi thiết bị

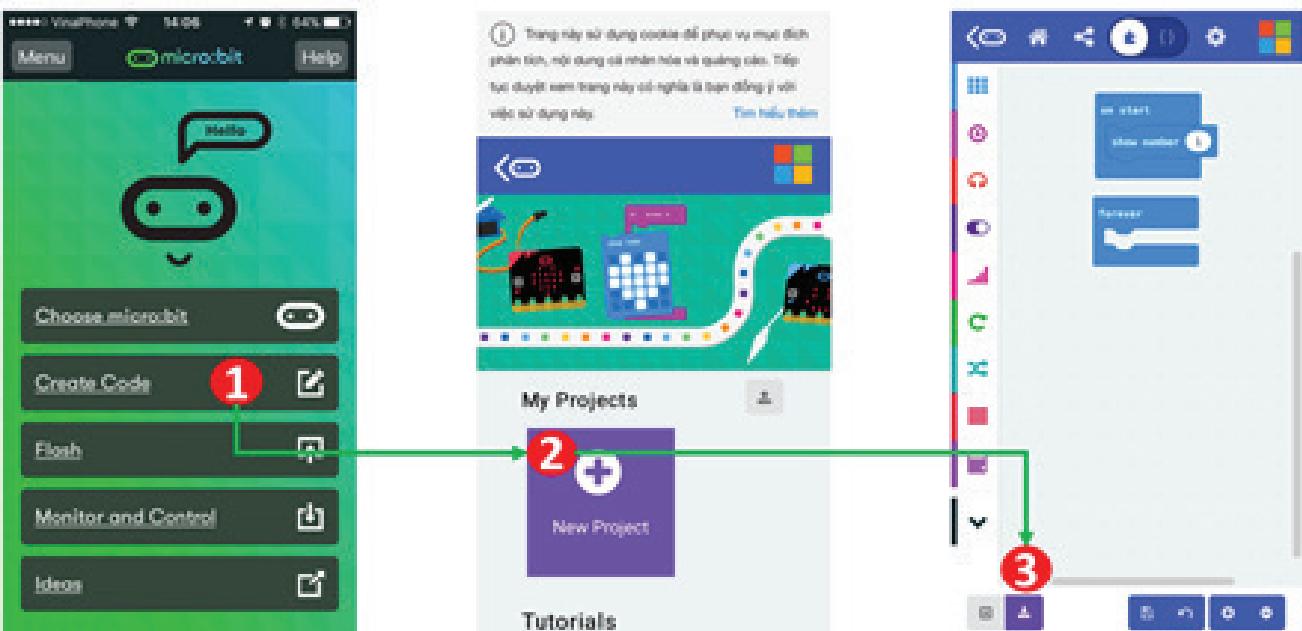
Sau khi nhấn OK ở bước 5, chúng ta sẽ được quay trở lại màn hình trong Bước 2 của Hình 14. Tuy nhiên, lúc này trong danh sách của chúng ta đã có 1 thiết bị MicroBit. Tất nhiên chúng ta có thể xóa thiết bị này đi và ghép nối với một thiết bị khác bằng cách nhấn vào biểu tượng Remove.



Hình 16: Một thiết bị MicroBit đã được thêm vào trong danh sách

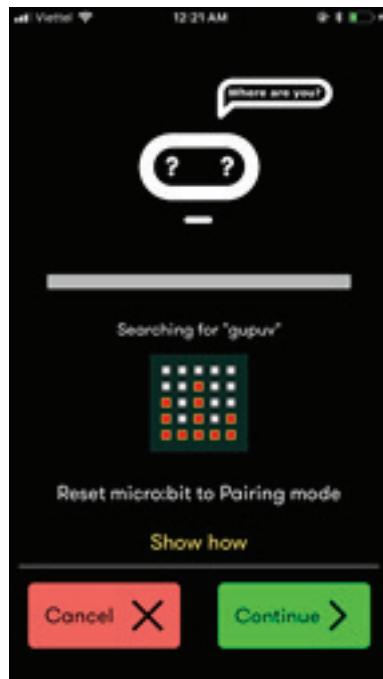
2.3.2. Lập trình và nạp chương trình lên MicroBit

Trở về màn hình Home của ứng dụng này, chúng ta có thể nhấn vào tùy chọn thứ 2, Create Code (xem Hình 17). Một giao diện quen thuộc với chúng ta sẽ hiện ra để chúng ta lập trình.



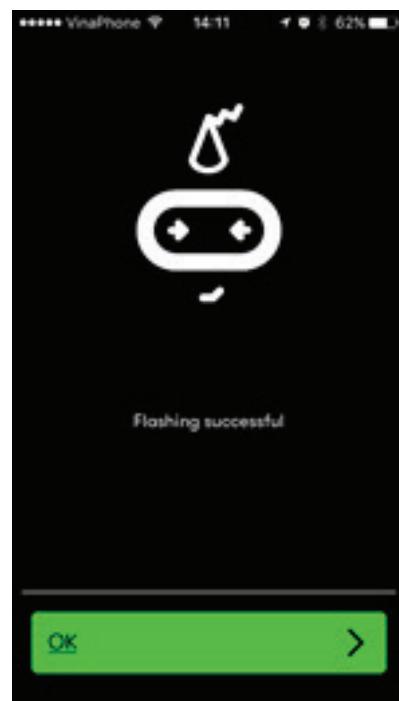
Hình 17: Viết chương trình cho MicroBit

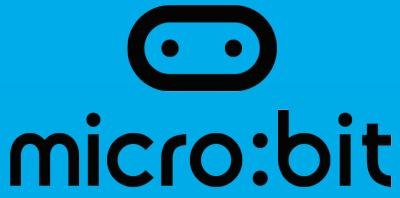
Cuối cùng, sau khi chương trình đã hoàn chỉnh, chúng ta có thể nhấn và nút tải về (Bước 3 trên Hình 17). Giao diện sau đây (Hình 18) sẽ hiện ra, yêu cầu chúng ta chuyển mạch MicroBit sang chế độ Bluetooth. Chúng ta cần phải một lần nữa, nhấn đè A và B, nhấn thả RESET, tiếp tục đè 2 nút A và B cho đến khi ký hiệu Bluetooth xuất hiện trên mạch MicroBit thì thả ra. Cuối cùng, chúng ta có thể quay lại điện thoại, để nhấn nút Continue.



Hình 18: Chuyển mạch MicroBit sang chế độ Bluetooth trước khi nhấn nút Continue

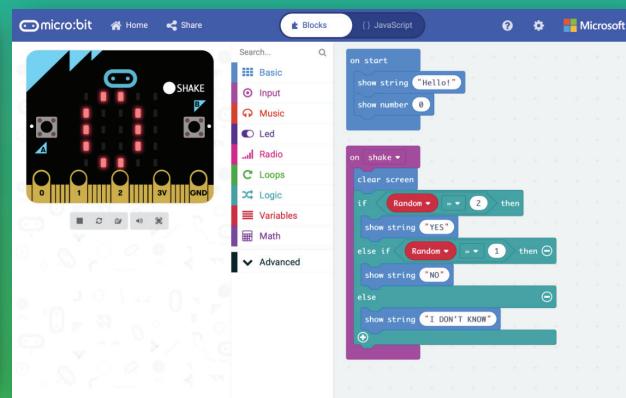
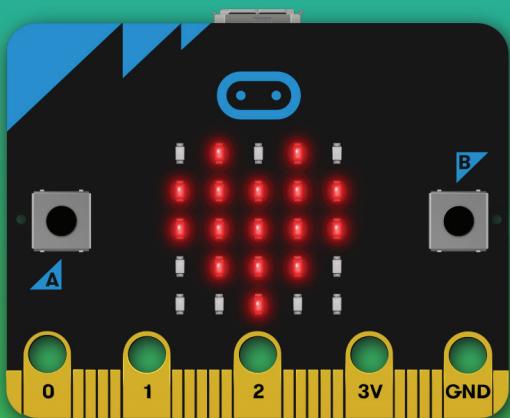
Đến đây thì chương trình sẽ tự động làm hết cho chúng ta phần còn lại, kết nối và nạp chương trình. Ở bước này, phiên bản trên iOS thực sự tốt hơn so với phiên bản trên Android. Việc nạp dữ liệu diễn ra nhanh hơn và thao tác cũng bớt rườm rà hơn phiên bản trên Android. Giao diện bên dưới sẽ hiện ra, báo hiệu việc nạp đã thành công. Trong trường hợp muốn soạn lại chương trình, chúng ta chỉ cần nhấn OK và chương trình sẽ tự động quay về giao diện lập trình.





Bài 6:

Tương tác giữa MicroBit và hành vi người dùng



1. Mục tiêu

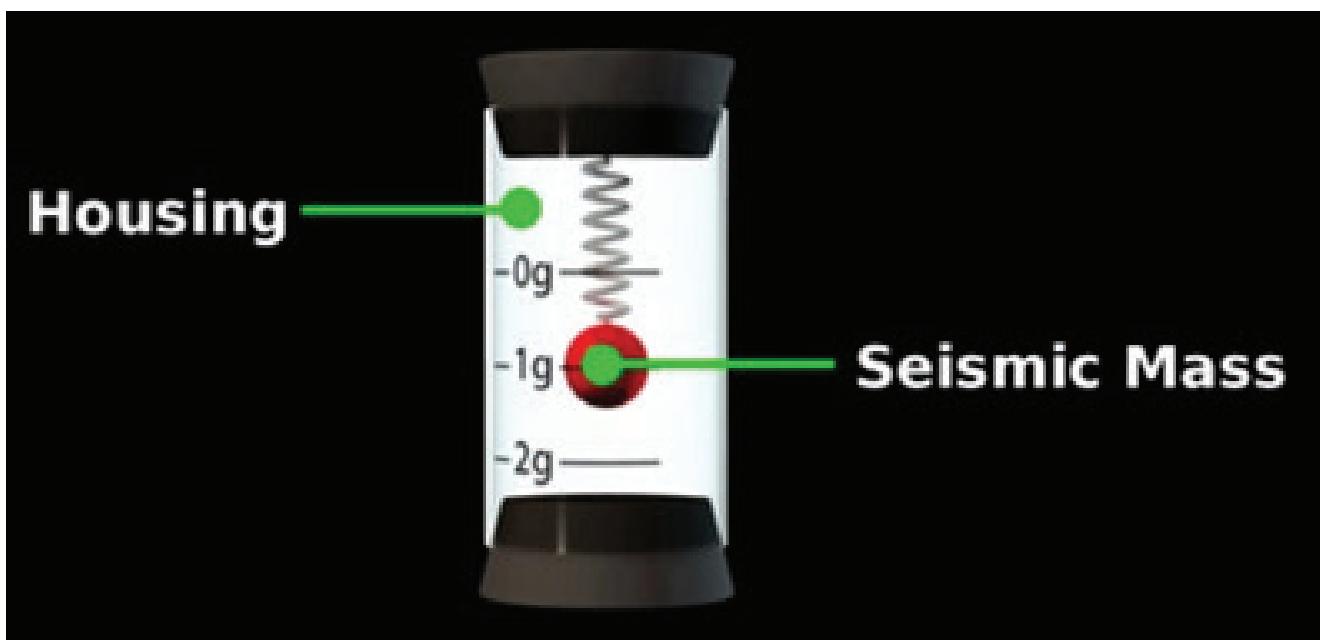
- Học sinh hiểu được nguyên lý phát hiện hành vi người dùng dựa trên cảm biến gia tốc
- Học sinh sử dụng được các câu lệnh để phát hiện hành vi người dùng
- Học sinh kết hợp các câu lệnh này để tạo ra một ứng dụng đơn giản

2. Nội dung chi tiết

2.1. Nguyên lý phát hiện hành vi của người dùng

Đối với các ứng dụng hiện đại, việc tương tác với người dùng đã trở nên ngày càng thông minh. Cụ thể, trên MicroBit, người dùng có thể lắc tay, nghiêng sang trái hay sang phải để tạo ra các sự kiện gửi đến MicroBit.

Nguyên lý để phát hiện ra hành vi của người dùng trên MicroBit được dựa trên cảm biến gia tốc, có tên tiếng Anh là Accelerometers. Quy tắc hoạt động của cảm biến này có thể được minh họa như Hình 1, nó bao gồm 2 phần: khoang chứa hình trụ được gắn liền vào vật thể mà chúng ta cần đo gia tốc, còn quả bóng là vật có thể di chuyển một chiều bên trong khoang chứa. Khi chúng ta di chuyển khoang chứa, quả bóng cũng sẽ di chuyển bên trong khoang chứa, khiến lò xo co giãn ra. Dựa vào độ co giãn của lò xo, chúng ta có thể đoán biết được lực và gia tốc của chuyển động.



Hình 1: Nguyên lý hoạt động của cảm biến gia tốc

Để có thể đo đạc được trong không gian 3 chiều, mô hình trên được nhân lên gấp 3, tương ứng cho 3 trục không gian X, Y và Z. Tuy nhiên, tất cả các nguyên lý phức tạp này đã được MicroBit đơn giản đi rất nhiều bởi các câu lệnh sự kiện, sẽ được trình bày trong phần tiếp theo.

2.2. Các câu lệnh phát hiện hành vi

Tất cả các câu lệnh để tương tác với hành vi của người dùng được hỗ trợ sẵn bởi MicroBit được trình bày ở Hình 2. Chỉ bằng một câu lệnh on shake trong phần Input, đã có rất nhiều hành vi được hỗ trợ sẵn cho chúng ta. Trong phần tiếp theo, một số lựa chọn thông dụng cho câu lệnh on shake sẽ được trình bày.



Hình 2: Các câu lệnh phát hiện hành vi

2.3. Hành vi on shake

Đây là hành vi căn bản, và cũng là hành vi mặc định của câu lệnh on shake. Khi chúng ta lắc nhẹ mạch MicroBit, sự kiện này sẽ xảy ra. Chúng ta có thể hiện thực một chương trình nhỏ để thử nghiệm sự kiện này như Hình 3. Chương trình này hoàn toàn có thể mô phỏng được mà không cần nạp trực tiếp vào mạch MicroBit. Chúng ta chỉ cần rê chuột qua lại trên mạch MicroBit mô phỏng, thì sự kiện on shake sẽ xảy ra.



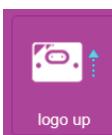
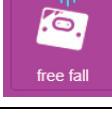
Hình 3: Chương trình thử nghiệm sự kiện on shake

Bài tập trên lớp: Giáo viên cho học sinh hiện thực lại chương trình trên và giải thích chức năng của nó.

Trả lời: Khi lắc mạch, sẽ hiển thị một icon trong 2 giây.

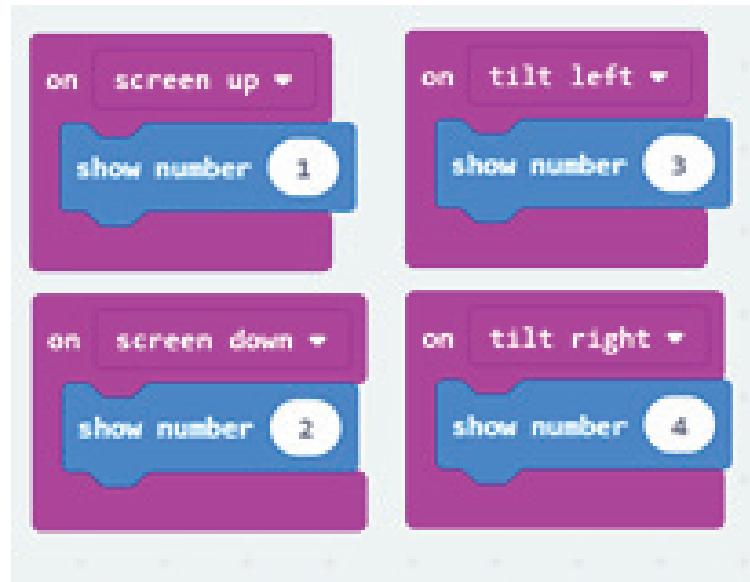
2.4. Các hành vi khác

Trong phần này, ý nghĩa của các hành vi khác sẽ được tóm tắt lại trong bảng bên dưới. Tuy nhiên, để có thể hiểu chức năng của nó cặn kẽ hơn, chúng ta chỉ cần hiện thực một chương trình nhỏ như Hình 3 ở trên.

| Câu lệnh | Chức năng |
|---|--|
|  | Ở góc bên trái của mạch MicroBit có một Logo, khi Logo này hướng lên trên, sự kiện logo up sẽ xảy ra |
|  | Tương tự như câu lệnh ở trên, sự kiện này xảy ra khi logo hướng xuống dưới |
|  | Trên mạch MicroBit có 1 màn hình gồm 25 đèn hiển thị. Khi màn hình này hướng lên trên, sự kiện screen up xảy ra |
|  | Ngược lại với sự kiện screen up, sự kiện này xảy ra khi màn hình hiển thị úp xuống |
|  | Sự kiện này xảy ra khi chúng ta hơi nghiêng mạch về bên trái |
|  | Sự kiện này xảy ra khi chúng ta hơi nghiêng mạch về bên phải |
|  | Khi thả mạch MicroBit rơi tự do, sự kiện này sẽ được gọi |
|  | Nhóm sự kiện cuối cùng mô phỏng một va chạm. Nó cũng tương đương như việc chúng ta giật mạnh tay mạch MicroBit rồi cho nó dừng lại đột ngột. Các thông số 3g, 6g và 8g tượng trưng cho mức độ dừng lại đột ngột của mạch MicroBit. |

3. Bài tập trên lớp

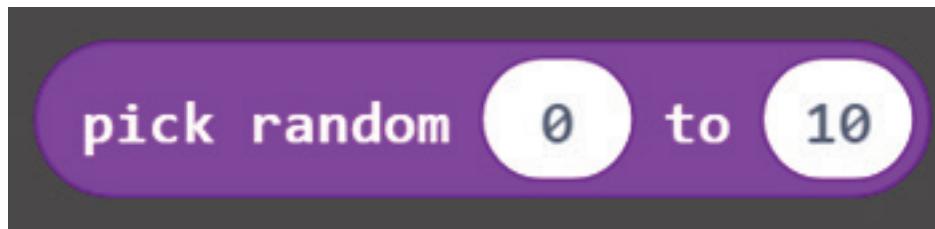
1. Học sinh hiện thực chương trình hiển thị số 1, 2, 3 và 4 trên màn hình hiển thị tương ứng với 4 sự kiện màn hình hiển thị hướng lên, xuống, nghiêng sang trái và sang phải. Chương trình gợi ý như sau:



Hình 4: Chương trình gợi ý

2. Học sinh hiện thực một chương trình “Xúc xác điện tử”: Lắc mạch MicroBit sẽ hiển thị ngẫu nhiên 1 con số từ 1 đến 6.

Gợi ý: Học sinh sử dụng câu lệnh lấy số ngẫu nhiên ở Hình 5.



Hình 5: Lệnh lấy giá trị ngẫu nhiên

Đáp án:



Hình 6: Đáp án cho chương trình "Xúc xác điện tử"

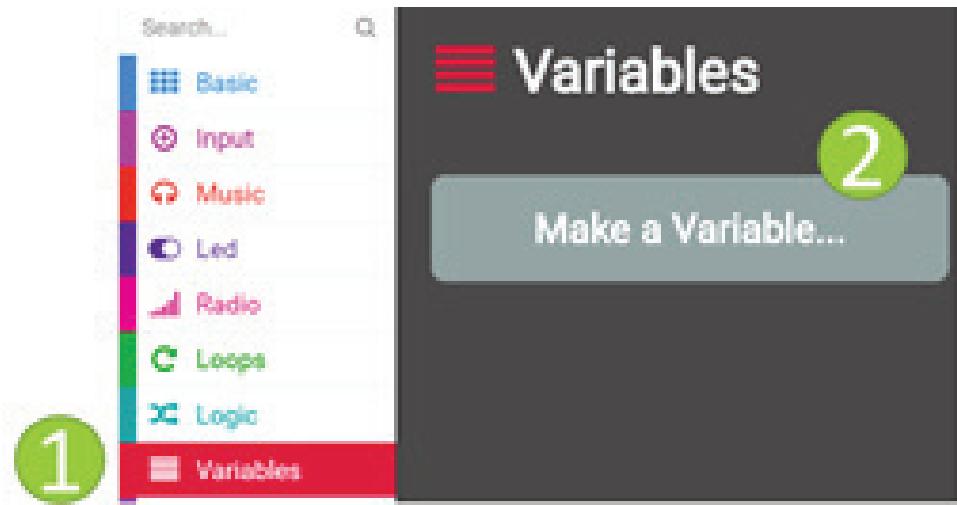
3. Học sinh hãy cải tiến chương trình “Xúc xác điện tử”, thay vì hiển thị các con số từ 1 đến 6, hãy hiển thị một dấu chấm, hai dấu chấm, ... trên màn hình như 1 xúc xắc bình thường.

Gợi ý:

- Tạo một biến số lưu giá trị số ngẫu nhiên
- Sử dụng thêm câu lệnh điều kiện trong mục Logic

Các bước chi tiết để hiện thực bài tập này như sau:

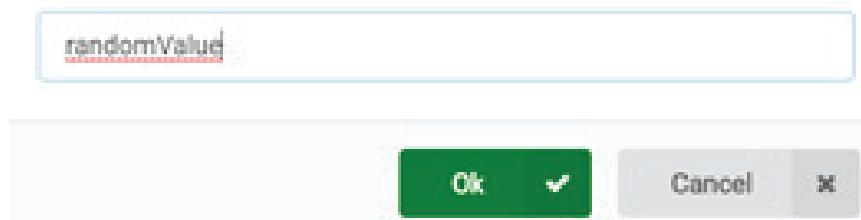
Bước 1: Vào mục Variables (1), nhấn vào Make a Variable (2)



Hình 7: Tạo một biến số cho chương trình

Bước 2: Đặt tên cho biến (ví dụ như randomValue), rồi nhấn OK.

New variable name:



Bước 3: Hiện thực chương trình bằng cách sử dụng câu lệnh if – else trong mục Logic



Hình 9: Đáp án gợi ý cho chương trình "Xúc xắc điện tử"

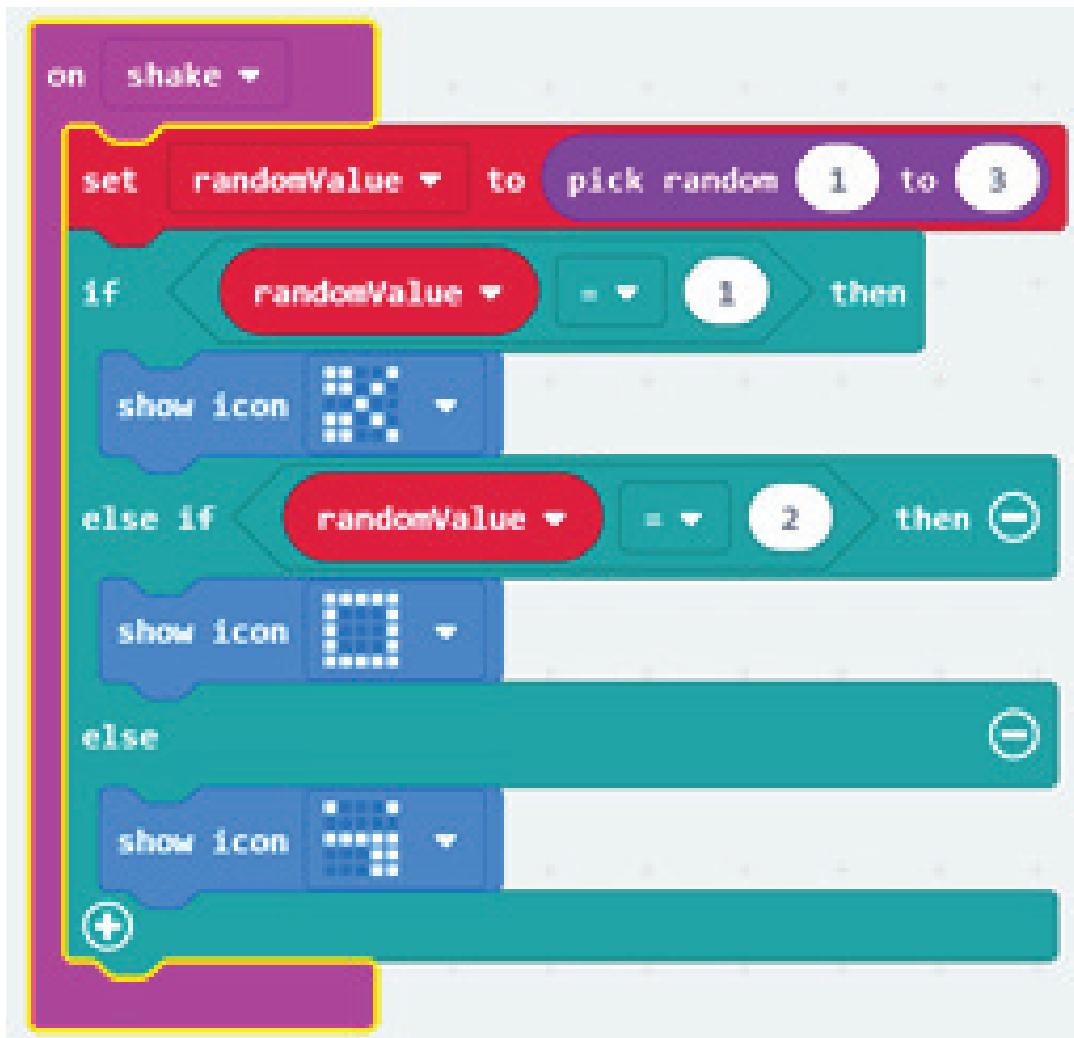
4. Bài tập về nhà

Học sinh hiện thực trò chơi "Oẳn tù tì" điện tử bằng cách sử dụng mạch MicroBit. Mạch sẽ hiển thị các biểu tượng ngẫu nhiên 1 trong 3 hình ảnh Búa - Dao - Kéo.

Gợi ý:

- Lấy ngẫu nhiên từ 1 đến 3: 1 – Búa, 2- Dao, 3 – Kéo
- Sử dụng câu lệnh show icon

Đáp án:



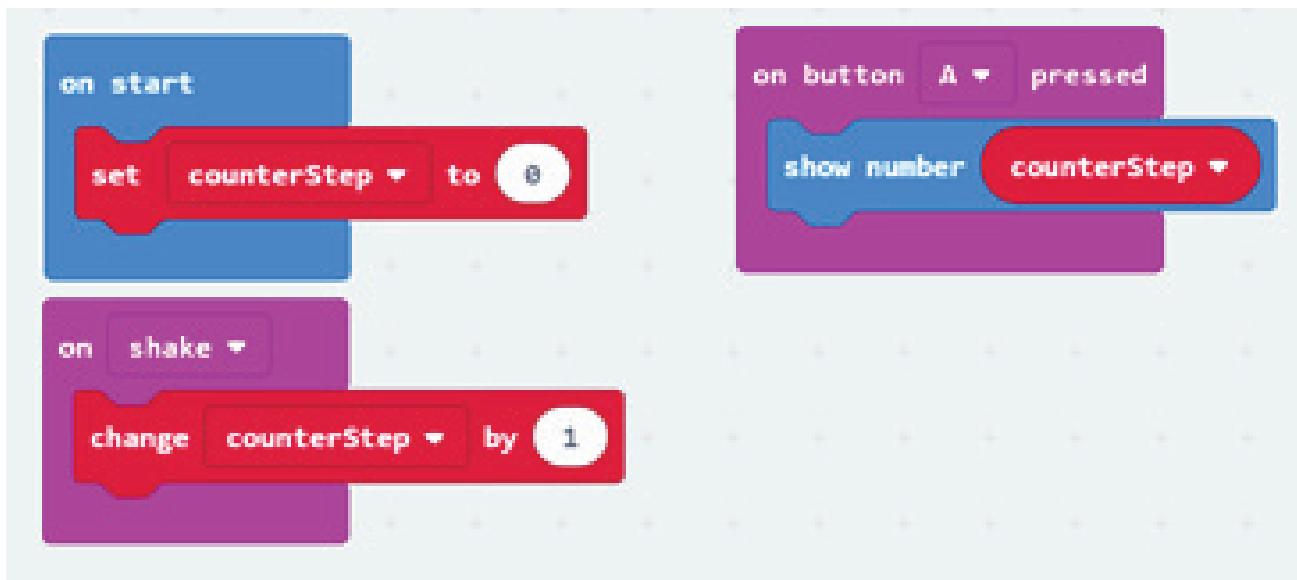
Hình 10: Đáp án gợi ý cho chương trình "Oẳn tù tì"

Học sinh hiện thực một chương trình đếm số bước chân khi di chuyển. Mạch MicroBit sẽ được gắn vào chân (không để dưới đế giày!!!!). Bình thường, mạch không hiện thị gì cả, chỉ khi nhấn vào nút A, mạch sẽ hiển thị lên số bước chân đã di chuyển.

Gợi ý:

- Tạo một biến để lưu trữ số bước chân (counterStep)
- Tăng biến này lên 1 trong sự kiện on shake
- Hiển thị ra khi nhấn nút A

Đáp án:

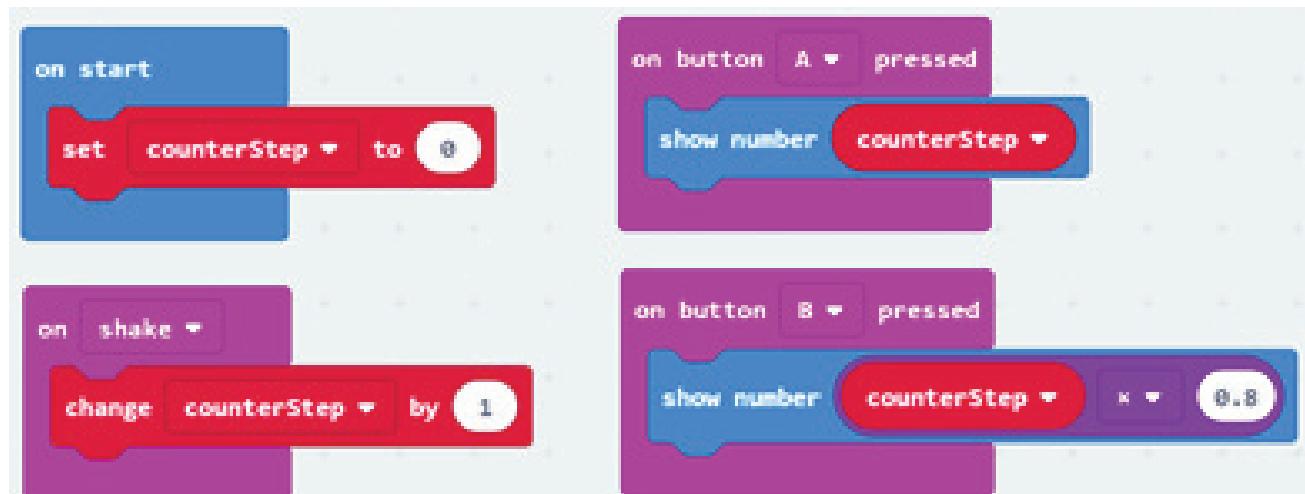


Hình 11: Chương trình đếm số bước chân di chuyển

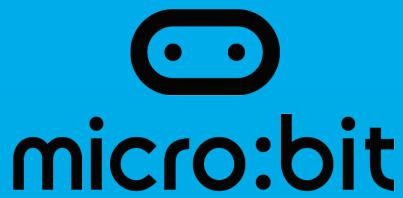
Học sinh cải thiện bài tập trinh, khi nhấn vào nút B, mạch MicroBit sẽ hiển thị quãng đường đã di chuyển.

Gợi ý: Sử dụng số bước chân, nhân với khoảng cách gần đúng của một bước chân (khoảng 0.8m)

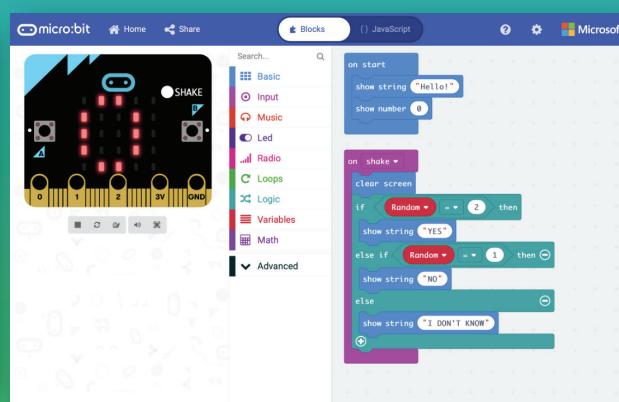
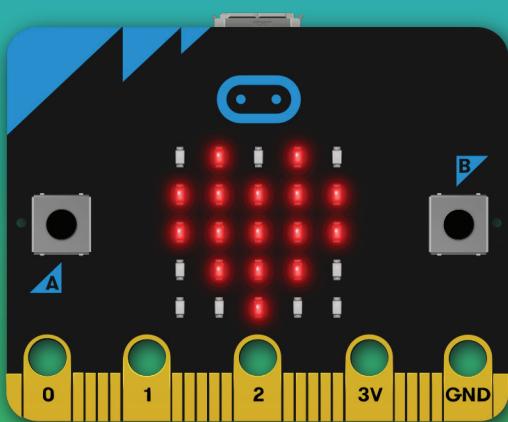
Đáp án:



Hình 12: Chương trình đếm số bước chân và quãng đường đã di chuyển



Bài 7: Cảm biến trên MicroBit



1. Mục tiêu

- Học sinh hiểu được cảm biến là gì
- Học sinh sử dụng được các câu lệnh liên quan đến các cảm biến đơn giản
- Học sinh kết hợp các câu lệnh này để tạo ra một ứng dụng đơn giản

2. Nội dung chi tiết

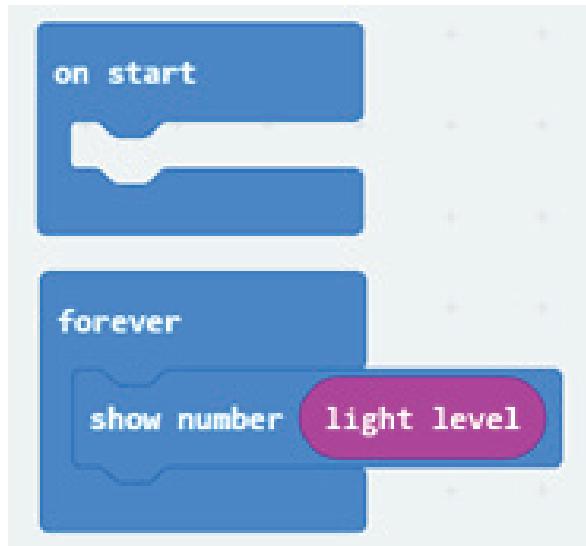
2.1. Cảm biến là gì?

Cảm biến là thiết bị điện tử cảm nhận những trạng thái hay quá trình vật lý hay hóa học ở môi trường cần khảo sát, và biến đổi thành tín hiệu điện để thu thập thông tin về trạng thái hay quá trình đó. Đơn giản hơn, cảm biến điện tử giống như các giác quan của con người, giúp hệ thống điện tử có thể nhận biết được trạng thái của môi trường xung quanh.

Hiện tại, trên bo mạch MicroBit hỗ trợ 3 loại cảm biến cơ bản là cảm biến ánh sáng, cảm biến nhiệt độ và cảm biến la bàn.

2.1.1. Cảm biến ánh sáng

Cảm biến ánh sáng trong MicroBit được hỗ trợ sẵn bởi biến light level, nằm trong mục Input. Một chương trình nhỏ để kiểm tra giá trị của biến này được minh họa như

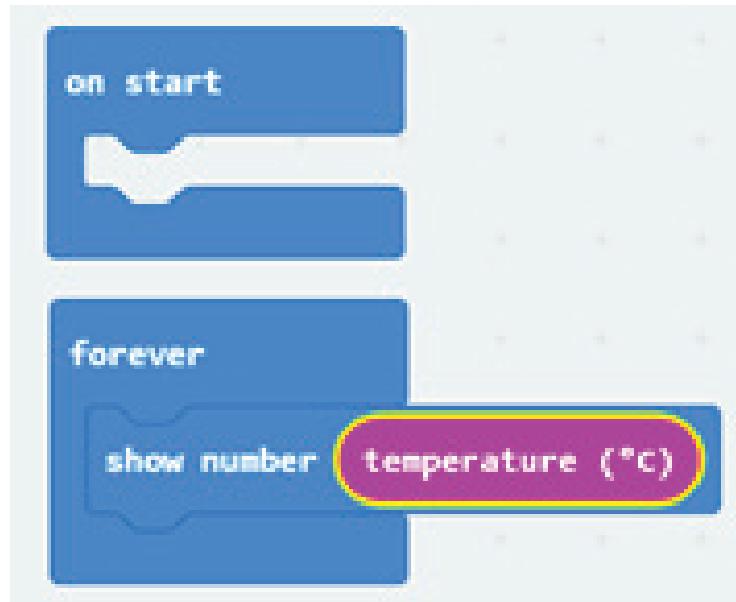


Hình 1: Chương trình kiểm tra giá trị cảm biến ánh sáng

Thực ra, giá trị của cảm biến ánh sáng được thực hiện bởi chính màn hình hiển thị. Mỗi một bóng đèn trên màn hình hiển thị có 2 chu kỳ hoạt động. Ở chu kỳ thứ nhất, nó làm nhiệm vụ phát sáng. Ở chu kỳ còn lại, nó trở thành 1 cảm biến nhận dạng độ sáng. Do đó, việc xuất giá trị cảm biến này ra chính màn hình hiển thị, làm cho giá trị light level không ổn định do nó bị ảnh hưởng bởi các bóng đèn xung quanh. Ở bài sau, học sinh sẽ được hướng dẫn để gửi giá trị cảm biến này qua một mạch MicroBit khác. Lúc đó, giá trị của cảm biến ánh sáng mới trở nên trung thực và chính xác.

2.1.2 Cảm biến nhiệt độ

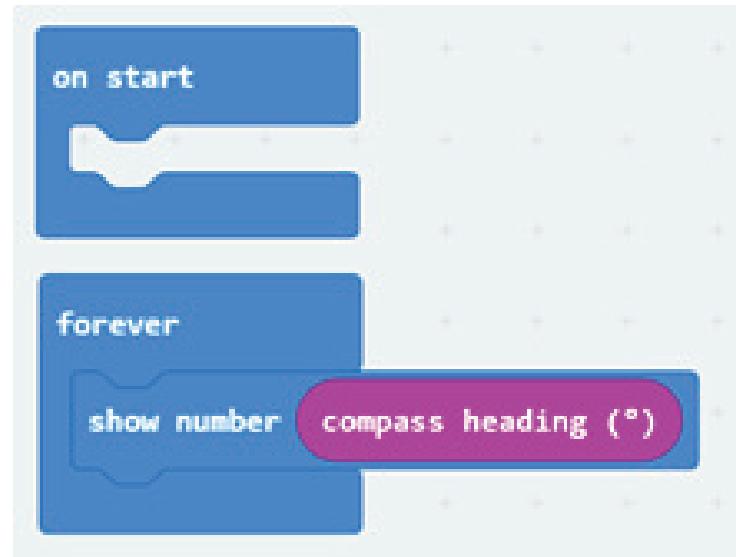
Tương tự như cảm biến ánh sáng, cảm biến nhiệt độ được hỗ trợ sẵn trên MicroBit thông qua biến temperature. Chúng ta có thể hiện thực một chương trình đơn giản như Hình 2 để kiểm tra giá trị của biến này. Khác với cảm biến ánh sáng, giá trị của cảm biến nhiệt độ sẽ rất ổn định và phản ánh nhiệt độ của môi trường xung quanh.



Hình 2: Chương trình kiểm tra cảm biến nhiệt độ

2.1.3. Cảm biến la bàn

Chúng ta có thể hiện thực nhanh một chương trình để kiểm tra cảm biến la bàn như Hình 3.



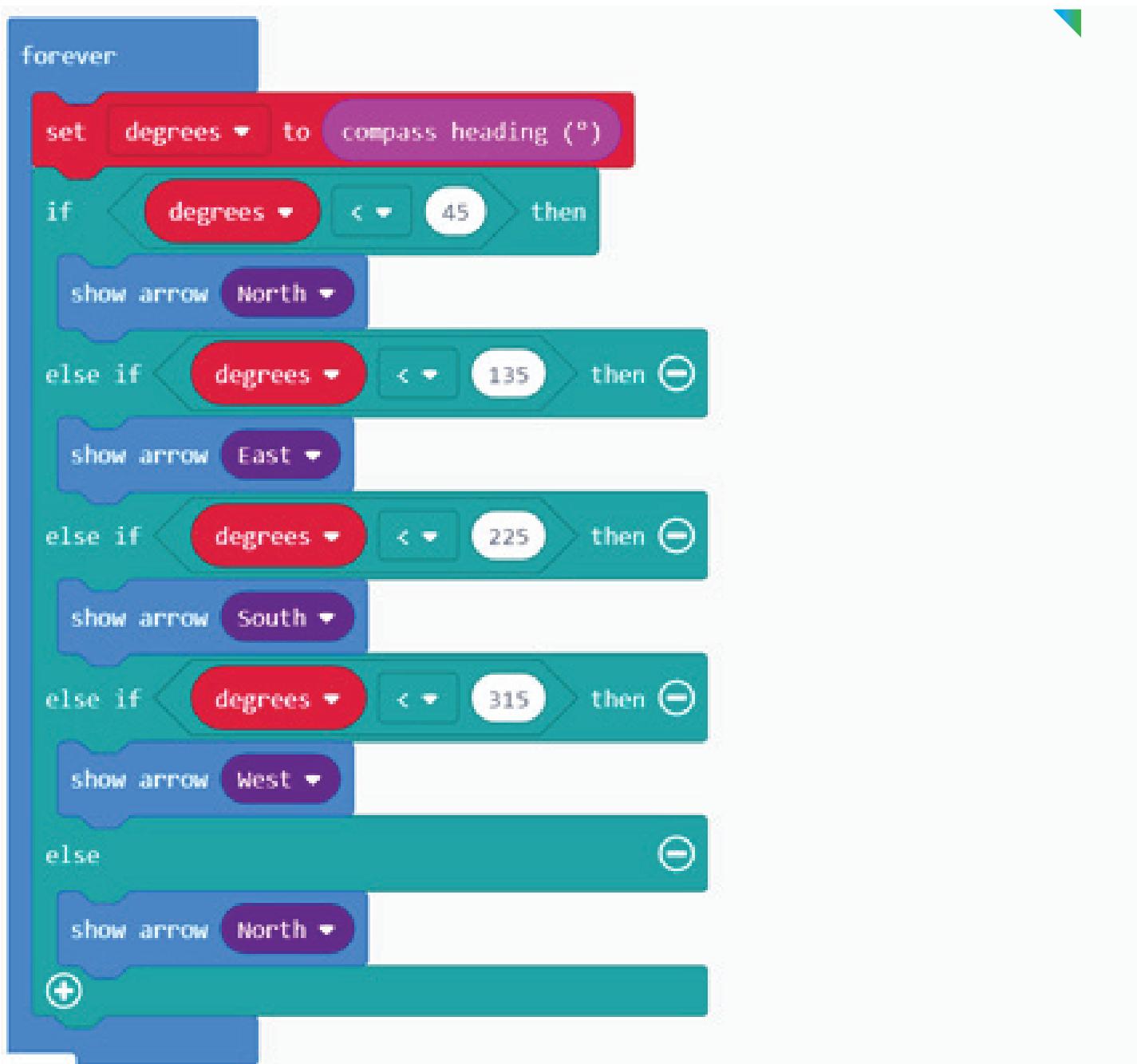
Hình 3: Chương trình kiểm tra cảm biến la bàn

Tuy nhiên, khác với hai cảm biến trên. Khi sử dụng cảm biến la bàn, MicroBit yêu cầu chúng ta phải chỉnh lại thước đo của nó. Một thông báo hiện lên, yêu cầu chúng ta xoay mạch MicroBit để tất cả các đèn đều phải sáng. Sau khi hoàn thành bước này, chương trình của chúng ta mới được phép chạy.

Sau khi việc cân chỉnh hoàn thành, giá trị của la bàn sẽ nằm từ 0 cho đến 359. Nhỏ hơn 45 độ là hướng Bắc, từ 45 đến 135 là hướng Tây, từ 135 đến 225 là hướng Nam, 225 đến 315 là hướng Đông. Một phần giá trị từ 315 đến 0 vẫn là hướng Bắc.

2.1.4. Bài tập trên lớp

Học sinh hiện thực một chương trình để vẽ ra bốn hướng Đông Tây Nam Bắc bằng cảm biến la bàn. Đáp án gợi ý cho bài tập này được minh họa ở Hình 4.



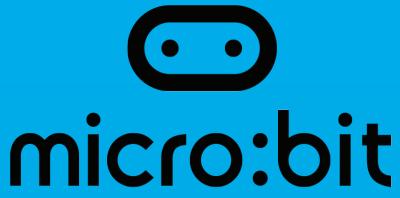
Hình 4: Đáp án gợi ý về cảm biến la bàn

3. Bài tập về nhà

Học sinh tự tìm hiểu một cảm biến khác được hỗ trợ sẵn bởi MicroBit là acceleration. Đây là cảm biến gia tốc, đã được trình bày ở Bài 6. Học sinh hãy so sánh giá trị của cảm biến này so với các sự kiện hành vi đã được trình bày ở Bài 6.

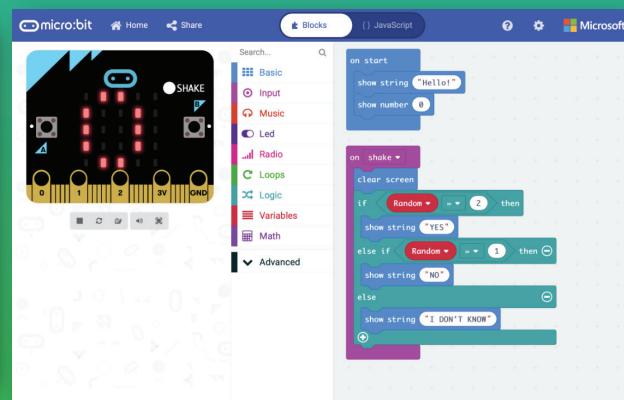
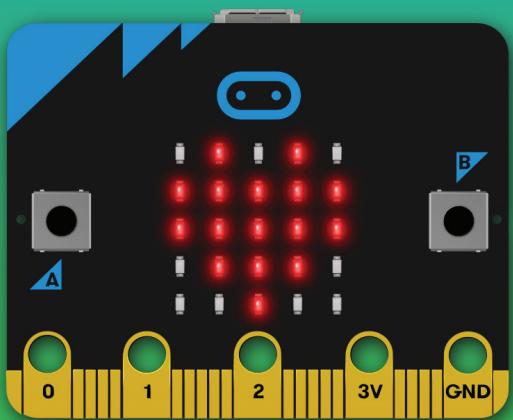


Hình 5: Cảm biến gia tốc



Bài 8:

Gửi dữ liệu không dây giữa các mạch MicroBit



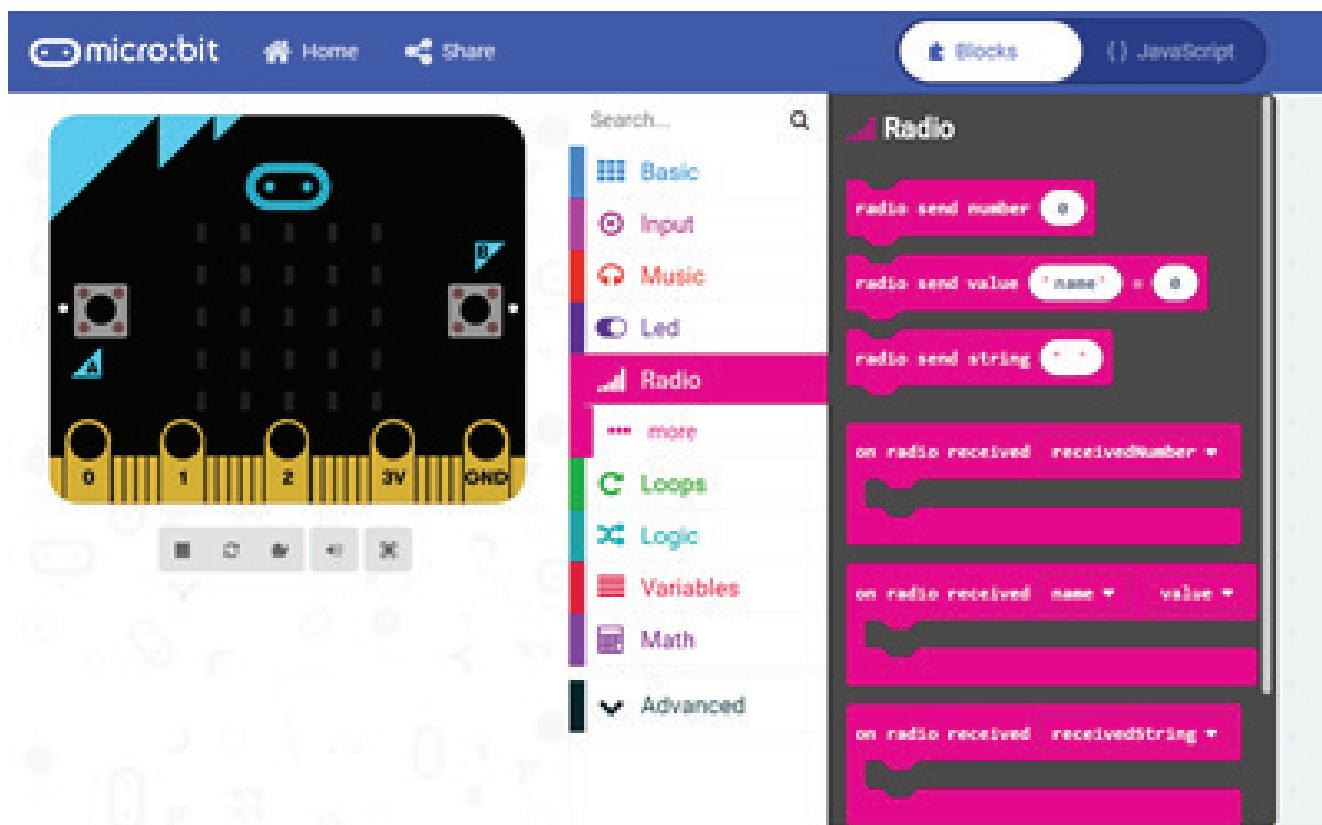
1. Mục tiêu

- Học sinh hiểu được nguyên lý gửi nhận dữ liệu không dây.
- Học sinh sử dụng được các câu lệnh gửi nhận dữ liệu không dây.
- Học sinh phối hợp được nhiều câu lệnh để tạo thành chương trình mới.

2. Nội dung chi tiết

2.1. Sự cần thiết của gửi nhận dữ liệu không dây

Trong cuộc sống hiện tại, chúng ta đang đón nhận những thành tựu của cuộc cách mạng công nghiệp 4.0. Một trong những đặc điểm quan trọng của cuộc cách mạng lần này là việc trao đổi thông tin không dây giữa các thiết bị. Trên bo mạch MicroBit cũng hỗ trợ sẵn cho chúng ta các câu lệnh rất đơn giản để chúng ta có thể gửi dữ liệu từ mạch này sang mạch khác. Các câu lệnh này được nằm trong nhóm Radio, như .



Hình 1: Các câu lệnh thuộc nhóm Radio để truyền nhận dữ liệu không dây

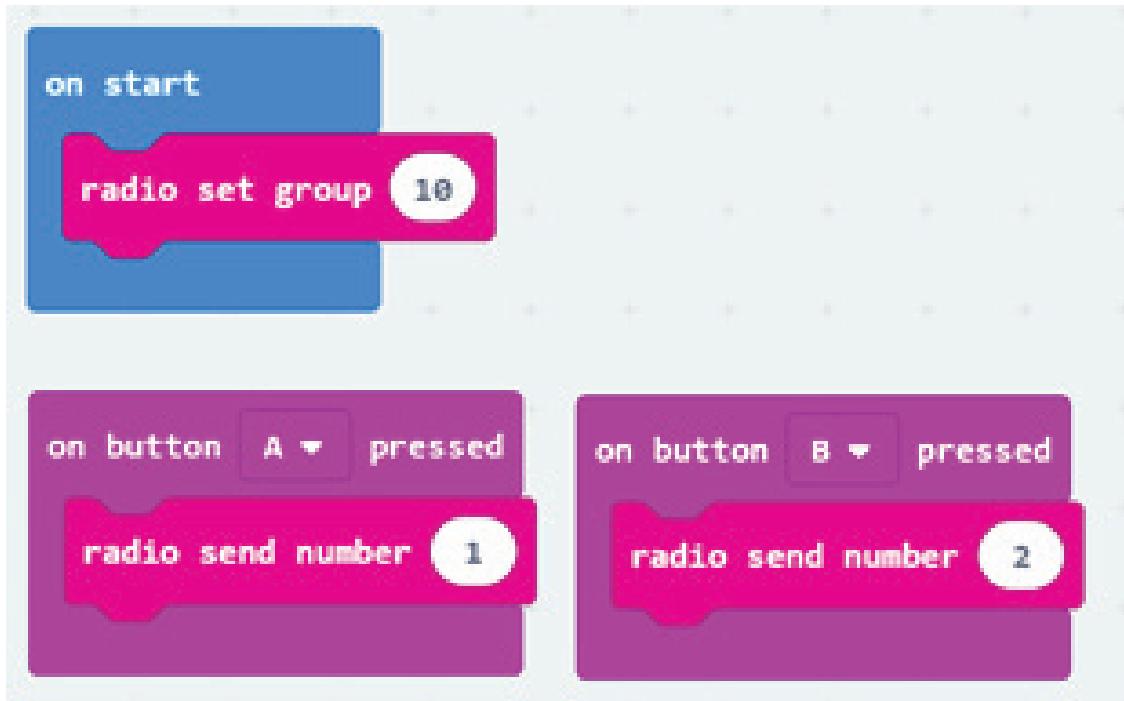
Một nguyên lý cơ bản của việc giao tiếp không dây trên MicroBit là các mạch phải có cùng nhóm với nhau. Mặc định, nhóm của các mạch MicroBit là 1. Tuy nhiên để đảm bảo an toàn, chúng ta sẽ chỉnh lại nhóm của chúng theo con số mà chúng ta quy định.

Hiển nhiên, để có thể giao tiếp được giữa 2 mạch MicroBit, chúng ta cần hiện thực 2 chương trình, một cho nốt truyền và chương trình còn lại cho nốt nhận. Phần tiếp theo sẽ trình bày chi tiết việc hiện thực chương trình cho từng nốt.

2.2. Chương trình nốt gửi

Để đơn giản, khi nhấn phím A, chúng ta sẽ gửi một con số cho nốt nhận, ví dụ số 1. Nhấn nút B, chúng ta sẽ gửi số 2. Chúng ta sẽ chọn nhóm cho 2 nốt là 10. Tuy nhiên, học sinh có thể chọn bất kì một con số nào đó cho hệ thống.

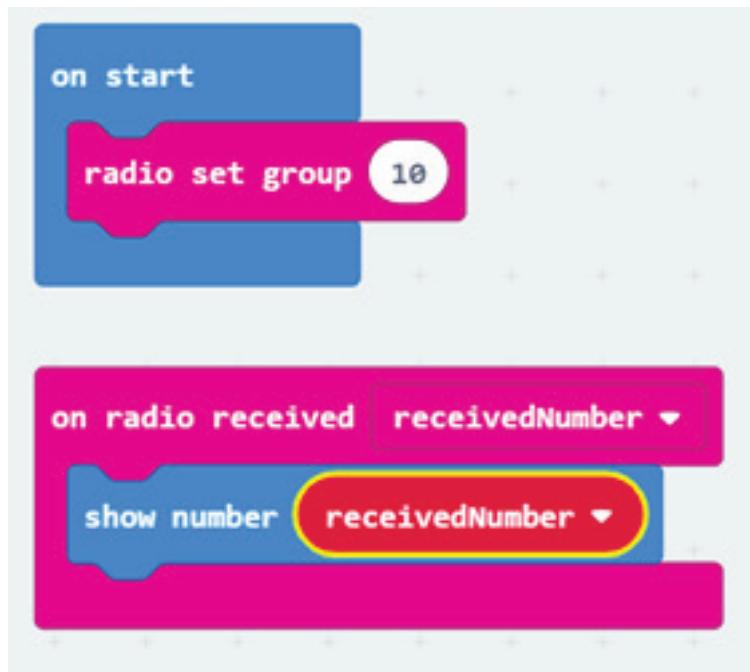
Việc chỉ định nhóm cho hệ thống sẽ được hiện thực trong phần on start. Và việc gửi dữ liệu sẽ được hiện thực bằng sự kiện nhấn nút A hay B (được trình bày ở Bài 4). Chương trình gợi ý của chúng ta như sau:



Hình 2: Chương trình cho nốt truyền

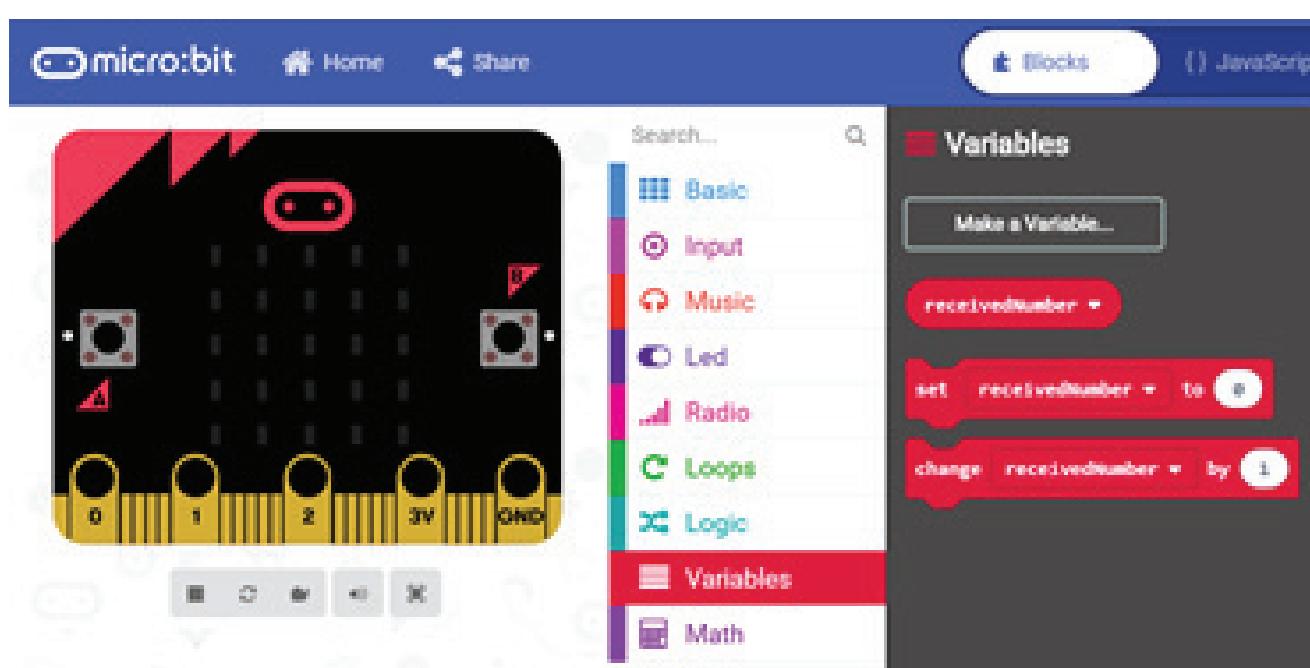
2.3. Chương trình cho nốt nhận

Cũng tương tự như nốt truyền, trong hàm on start, chúng ta cũng chỉ định nhóm cho nốt nhận là 10. Tiếp sau đó, chúng ta sẽ dùng câu lệnh on radio received. Câu lệnh này là câu lệnh sự kiện, cũng giống với nút nhấn A hay B, trong trường hợp này, khi nhận được một dữ liệu gì đó, sự kiện này sẽ xảy ra. Chương trình của chúng ta như sau:



Hình 3: Chương trình cho nốt nhận

Trong chương trình trên, khi nhận được bất kì dữ liệu số nào, chúng ta sẽ hiện thị số đó ra màn hình bằng câu lệnh show number. Kết quả nhận được sẽ được lưu trữ mặc định trong biến receivedNumber, nằm trong mục Variable như Hình 4.



Hình 4: Biến receivedNumber trong mục Variable

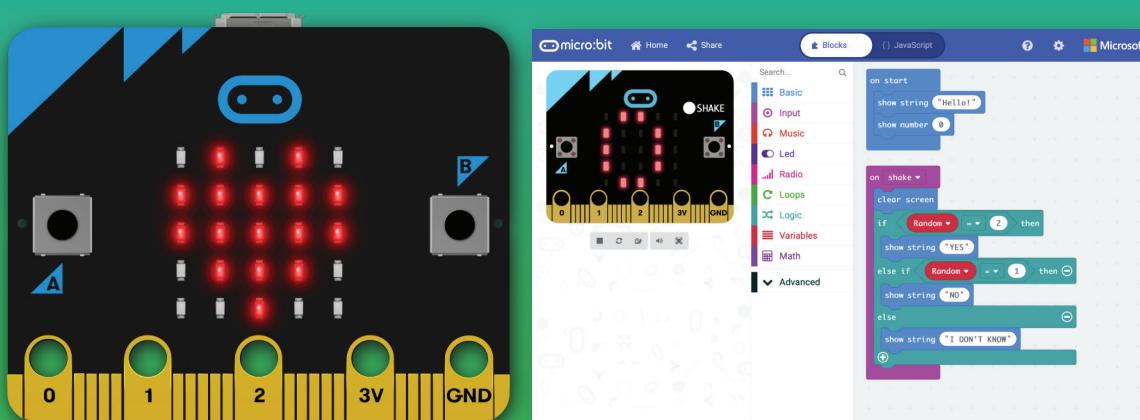
Bây giờ, học sinh có thể kết hợp chương trình của 2 mạch MicroBit để thấy được việc gửi dữ liệu không dây.

3. Bài tập trên lớp

Học sinh hãy hiện thực nhiều câu lệnh hơn từ nốt gửi, bằng cách sử dụng các câu lệnh liên quan đến hành vi của người dùng, ví dụ như nghiêng qua trái, nghiêng qua phải. Bên nốt nhận, thay vì chỉ hiển thị các con số, học sinh có thể hiển thị các dấu mũi tên hoặc các icon khác.



Bài 9: Các cấu trúc lặp và Điều kiện trên MicroBit



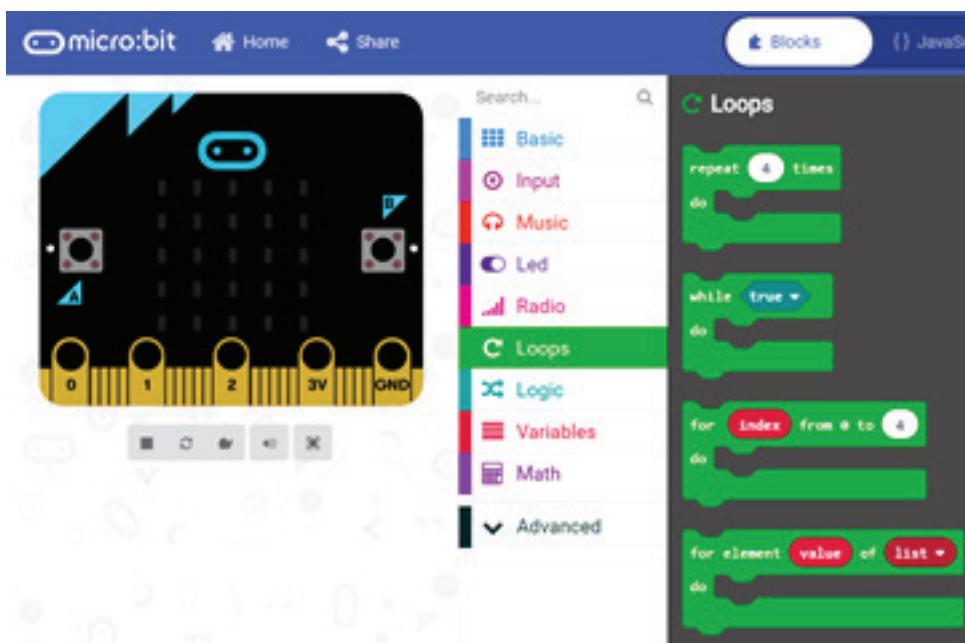
1. Mục tiêu

- Học sinh hiểu được cấu trúc lặp và điều kiện trên MicroBit
- Học sinh sử dụng được các câu lệnh lặp và điều kiện trên MicroBit
- Học sinh phối hợp được nhiều câu lệnh để tạo thành chương trình mới.

2. Nội dung chi tiết

2.1. Cấu trúc lặp

Cấu trúc lặp dùng để đơn giản việc hiện thực một chương trình có nhiều câu lệnh lặp đi lặp lại nhiều lần. Cấu trúc này chúng ta đã được gấp ở phần ngôn ngữ Scratch và sẽ không trình bày quá chi tiết ở đây. Các câu lệnh thuộc cấu trúc lặp này được nằm trong phần Loop như Hình 1.



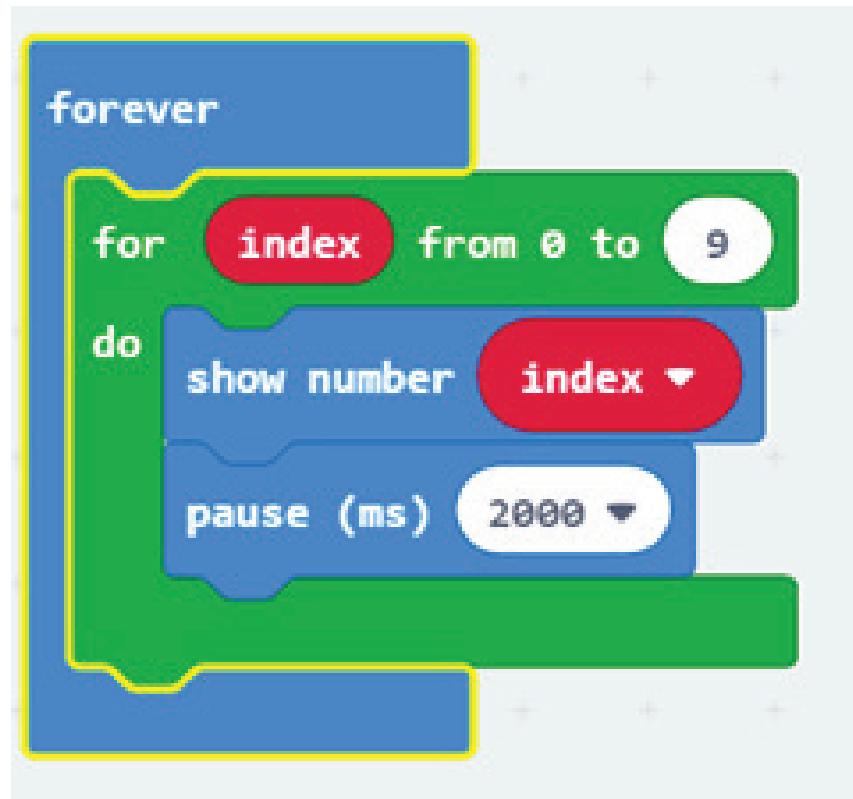
Hình 1: Cấu trúc lặp trong MicroBit

Trong phần cơ bản này, chúng ta chỉ tập trung vào 2 câu lệnh đầu tiên, với các chức năng được trình bày trong bảng dưới đây.

| Câu lệnh | Chức năng |
|-------------------------------|---|
| repeat (4 times) do | Lặp lại các câu lệnh bên trong 4 lần. Số lần lặp có thể thay đổi được, ví dụ đổi 4 lần thành 10 lần. |
| while [true] do | Lặp lại các câu lệnh bên trong chừng nào điều kiện vẫn còn đúng. Khi điều kiện sai, cấu trúc lặp này dừng lại |
| for [index from 0 to 4] do | Lặp lại 5 lần, từ 0 đến 4. Tuy nhiên với mỗi lần lặp, biến index sẽ có giá trị. Số lần lặp có thể thay đổi |

Bài tập trên lớp: Học sinh cho xuất hiện các con số từ 0 đến 9 ra màn hình MicroBit. Mỗi lần xuất hiện, con số đó hiện ra trong 2 giây trước khi đổi sang số tiếp theo.

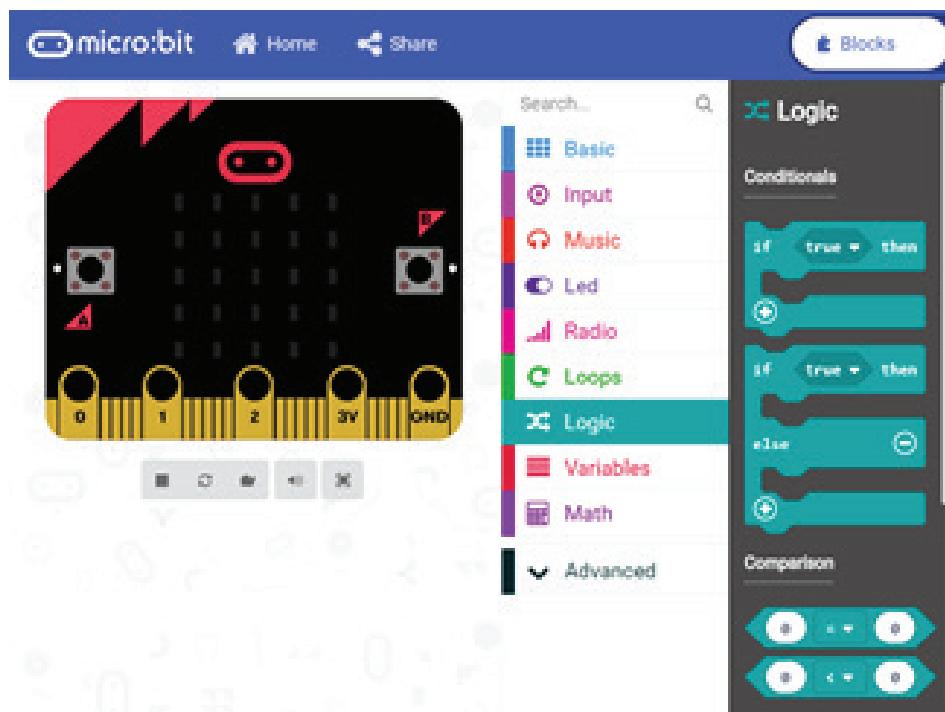
Đáp án: Câu lệnh tối ưu cho bài tập này được trình bày ở bên dưới. Tuy nhiên 2 câu lệnh còn lại hoàn toàn có thể thực hiện được chức năng này. Giáo viên có thể cho phần hiện thực bằng 2 câu lệnh đầu tiên như là một bài tập nâng cao.



Hình 2: Một chương trình sử dụng cấu trúc lặp

2.2. Cấu trúc điều kiện

Cấu trúc điều kiện, về cơ bản chính là câu lệnh nếu thì nữa trong ngôn ngữ Scratch. Ý nghĩa của nó cũng hoàn toàn giống với cấu trúc này bên Scratch. Nếu điều kiện đúng thì các câu lệnh bên trong nó mới được thực hiện. Các câu lệnh điều kiện này nằm trong mục Logic.

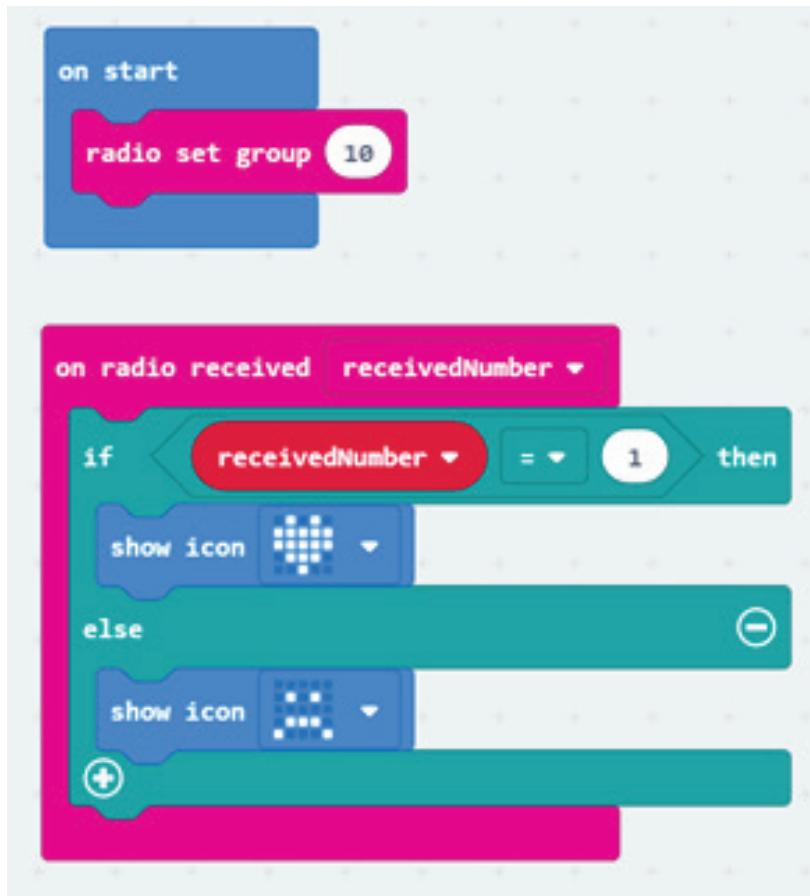


Hình 3: Câu lệnh điều kiện

Trong việc hiện thực câu lệnh điều kiện, một điều quan trọng nhất là xây dựng các câu lệnh điều kiện (có hình thoi). MicroBit hoàn toàn hỗ trợ sẵn các điều kiện này, nằm ngay bên dưới các câu lệnh điều kiện if else.

Bài tập trên lớp: Học sinh cải tiến lại chương trình của nốt nhận ở bài trước. Thay vì hiển thị các con số, hãy hiện thị nó bằng một hình ảnh nào đó.

Đáp án: Một đáp án gợi ý như Hình 4. Ở đây chúng ta sẽ nhận 2 giá trị khác nhau là 1 và 2. Do vậy chúng ta chỉ cần kiểm tra điều kiện nhận số 1, tất cả các giá trị còn lại hiển nhiên là giá trị 2.

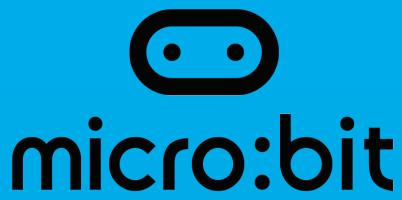


Hình 4: Một ví dụ về sử dụng câu lệnh điều kiện

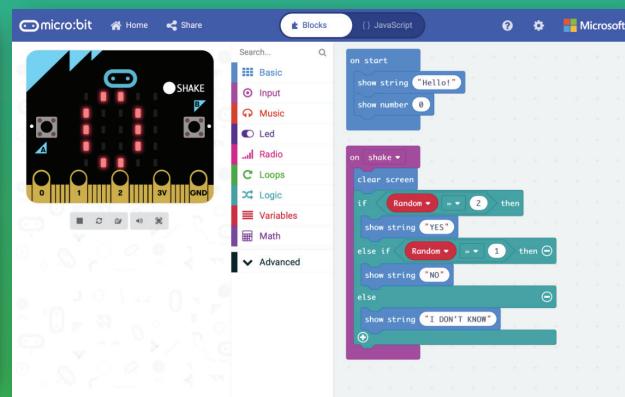
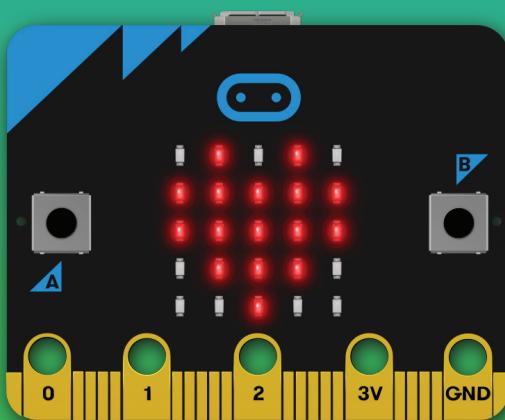
2.3. Bài tập về nhà

Học sinh có thể chủ động sáng tạo phối hợp các câu lệnh đã học để viết ra một chương trình hấp dẫn. Một số gợi ý như sau:

- Giám sát nhiệt độ trong phòng, nếu nhiệt độ lớn hơn một ngưỡng nhất định (40 độ), thì chớp tắt đèn để cảnh báo.
- Giám sát độ sáng trong phòng, nếu độ sáng nhỏ hơn một ngưỡng nhất định, thì bật đèn lên.
- Điều khiển từ xa sử dụng MicroBit: Học sinh tự định nghĩa thêm các câu lệnh, tương ứng với các con số khác nhau, để gửi từ nốt truyền sang nốt nhận. Tương ứng với một câu lệnh, nốt nhận sẽ thực hiện một chức năng nào đó.



Bài 10: Đồng hồ thông minh trên MicroBit



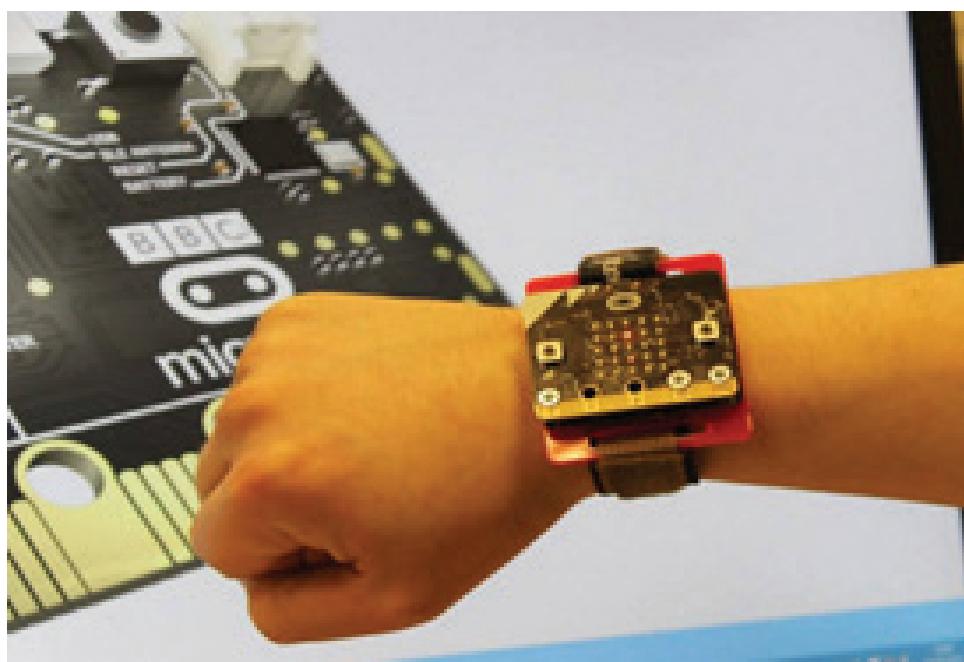
1. Mục tiêu

- Học sinh hiểu được nguyên lý tổ chức chương trình trong MicroBit.
- Học sinh có khả năng hiện thực một dự án nhỏ trên MicroBit.
- Học sinh có khả năng tự mở rộng dự án của mình.

2. Nội dung chi tiết

2.1. Chức năng của đồng hồ thông minh

Trong bài học này, chúng ta sẽ dùng mạch MicroBit để mô phỏng một đồng hồ thông minh. Hãy tưởng tượng rằng bạn có một dây đeo mạch MicroBit vào tay và dùng nó như một chiếc đồng hồ, như minh họa ở Hình 1. Ngoài chức năng hiển thị giờ hiện tại, chúng ta còn có thể xem thêm các thông tin khác như nhiệt độ, cường độ ánh sáng hay thậm chí là la bàn.



Hình 1: Đồng hồ thông minh với MicroBit

Trong bài hướng dẫn này, chúng ta sẽ hiện thực các chức năng như sau:

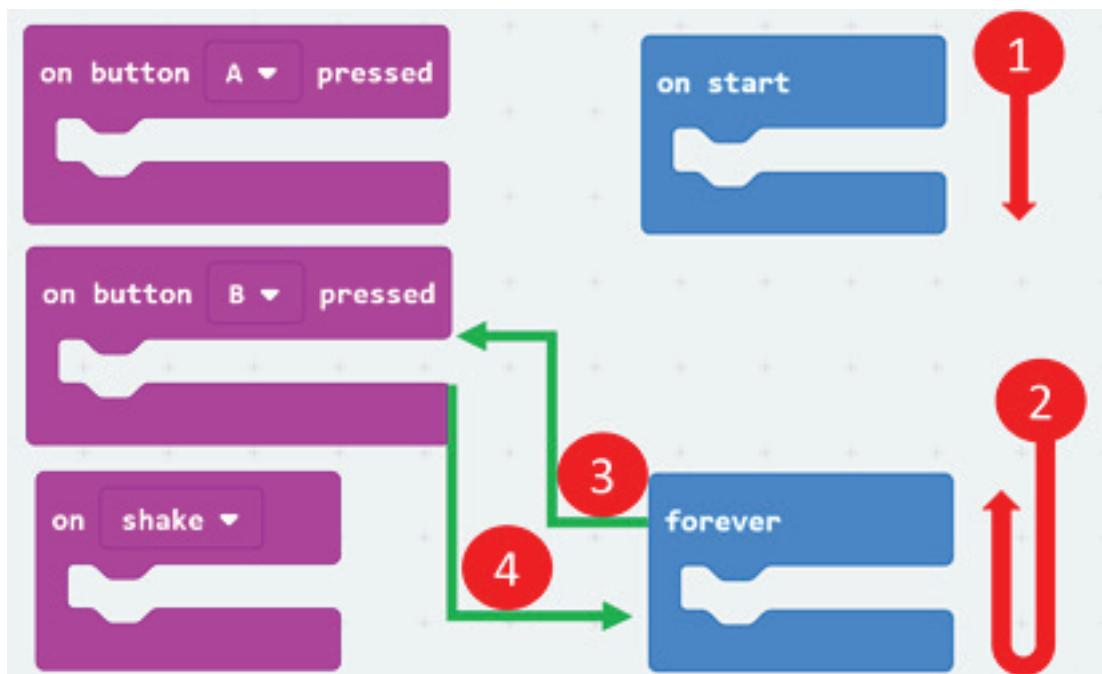
- Để tiết kiệm năng lượng, bình thường đồng hồ không hiển thị gì ra màn hình
- Khi người dùng muốn xem giờ, chỉ cần lắc tay, thông tin giờ phút giây sẽ được hiện ra
- Khi người dùng nhấn nút A, thông tin về nhiệt độ hiện tại sẽ được hiển thị ra màn hình
- Khi người dùng nhấn nút B, thông tin về cường độ ánh sáng sẽ được hiện ra
- Tất cả các thông tin hiển thị ra chỉ tồn tại trong 5 giây, sau đó màn hình sẽ tắt để tiết kiệm pin

2.2. Nguyên lý thực thi chương trình trên MicroBit

Việc nắm vững nguyên lý thực thi chương trình trên MicroBit là rất quan trọng để thực hiện các dự án với độ phức tạp ngày càng cao. Khác với các ngôn ngữ lập trình truyền thống như Pascal hoặc thậm chí là Scratch, chương trình trên MicroBit sẽ thực thi các câu lệnh trong khối on start trước, và sau đó lặp đi lặp lại ở khối forever. Trong khi đó, các ngôn ngữ lập trình khác, đa số các câu lệnh được thực thi theo thứ tự từ trên xuống dưới cho đến câu lệnh cuối cùng.

Điểm khác biệt thứ 2, một khác biệt khá lớn giữa ngôn ngữ lập trình trên MicroBit và các ngôn ngữ lập trình trên máy tính, là các chương trình hiện thực trong các sự kiện (ví dụ: sự kiện nhấn nút A, nhấn nút B hay lắc nhẹ mạch MicroBit). Khi một sự kiện xảy ra, việc thực thi lệnh trong khối forever sẽ tạm dừng lại, để cho chương trình bên trong khối lệnh được thực thi. Sau đó, MicroBit sẽ quay lại và tiếp tục thực hiện các câu lệnh được lặp đi lặp lại trong khối forever.

Hình 2 minh họa nguyên lý thực thi chương trình trên MicroBit. Bình thường, chương trình sẽ thực hiện hết các câu lệnh trong khối on start, sau đó chuyển sang khối forever và lặp đi lặp lại các câu lệnh trong khối này. Tuy nhiên, khi có 1 sự kiện nào đó xảy ra, việc thực thi chương trình trong forever sẽ tạm ngưng, và bắt đầu lại khi các câu lệnh trong khối sự kiện được thực thi hết.



Hình 2: Nguyên lý thực thi chương trình trên MicroBit

2.3. Hiện thực chức năng của đồng hồ

2.3.1 Đếm giờ của đồng hồ

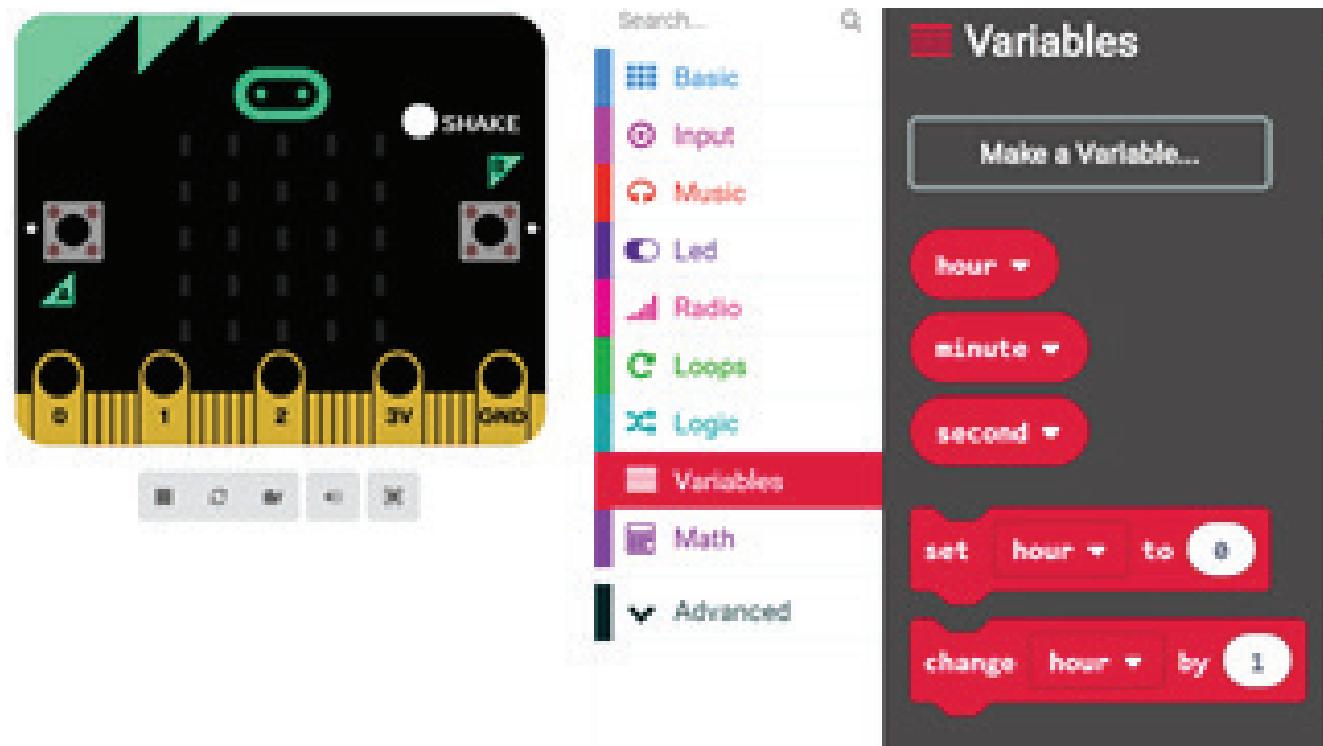
Đầu tiên, chúng ta hiện thực chức năng cơ bản của một đồng hồ điện tử, là tự động cập nhật sau mỗi giây. Rõ ràng, việc này sẽ được thực hiện lặp đi lặp lại, do đó chúng ta sẽ hiện thực chức năng này trong khối lệnh forever.

Trước tiên, chúng ta cần khai báo 3 biến số là second, minute và hour, để lưu giữ thông tin về giây, phút và giờ. Để làm việc này, chúng ta vào mục Variable, chọn Make a Variable, đặt tên cho biến số và nhấn OK, như minh họa ở Hình 3.



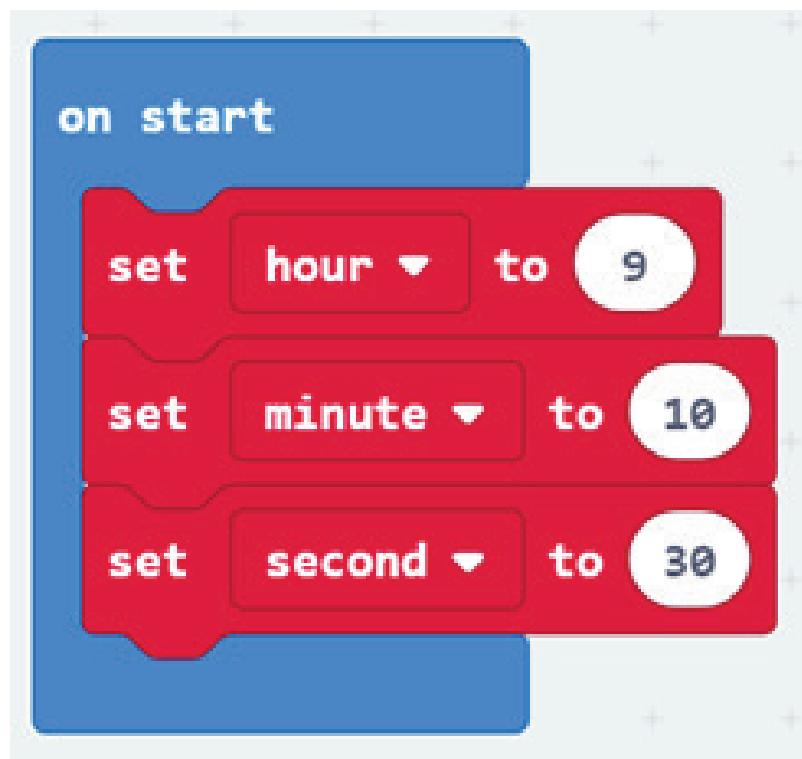
Hình 3: Tạo một biến trên MicroBit

Chúng ta tiếp tục lặp lại thao tác này cho 2 biến còn lại, là minute và hour. Cuối cùng, chúng ta có thêm các khối lệnh mới và các biến số mới vừa được tạo ra như Hình 4.



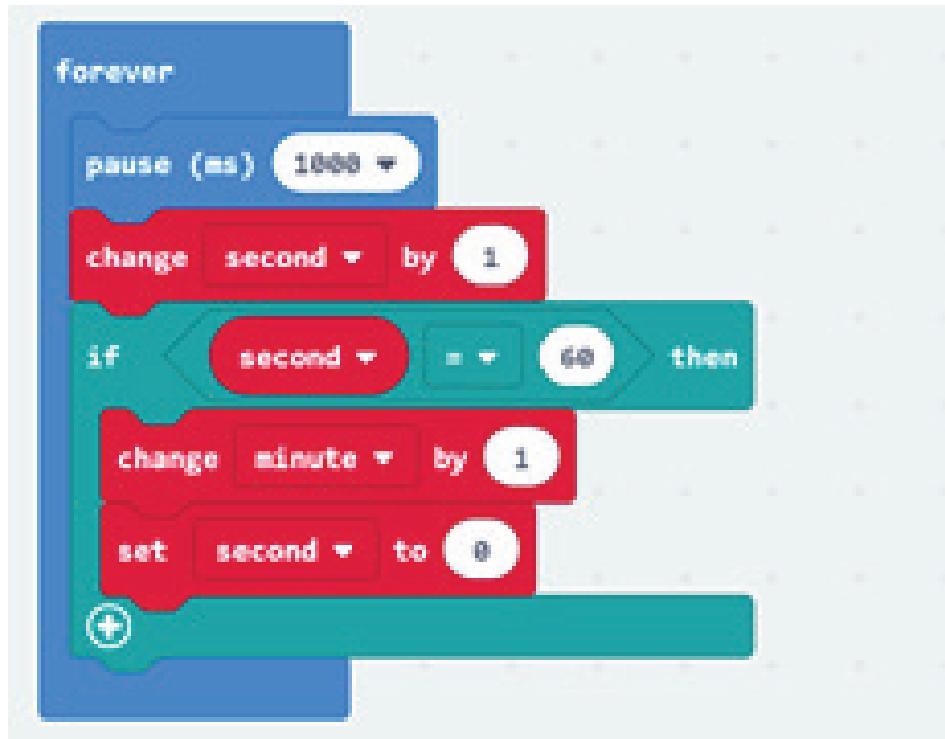
Hình 4: Chương trình sau khi tạo ra 3 biến số

Ở đầu chương trình, trong khối lệnh on start, chúng ta có thể chỉnh giá trị của 3 biến số này gần giống với giờ hiện tại của chúng ta. Ví dụ như hiện tại là 9 giờ, 10 phút và 30 giây, chúng ta sẽ hiện thực như sau:



Hình 5: Chỉnh trạng thái cho đồng hồ gần với giờ hiện tại

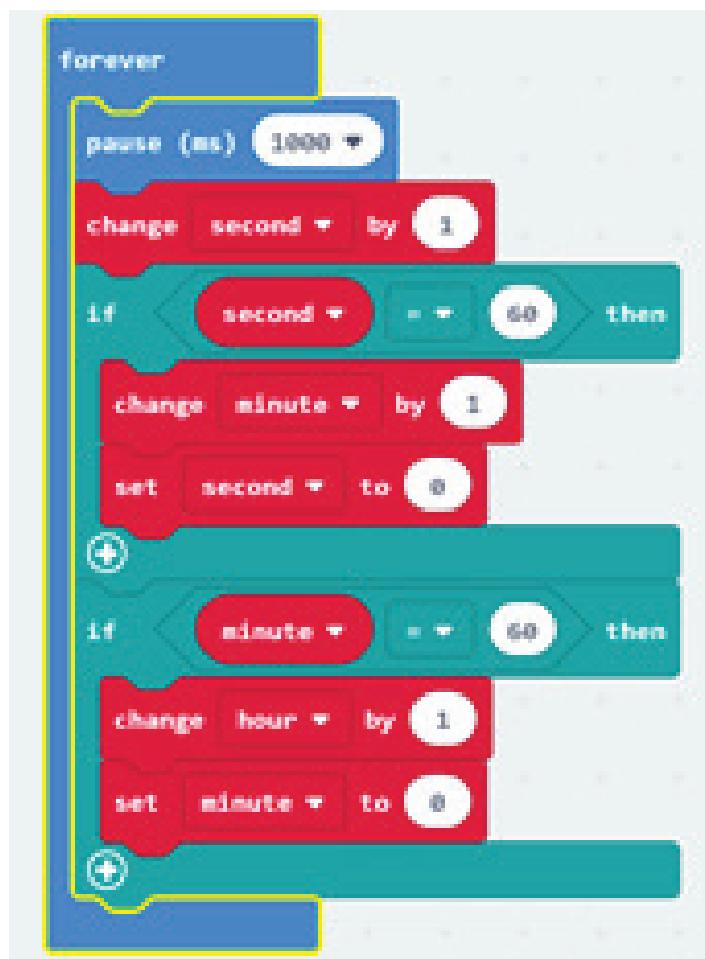
Tiếp theo trong phần forever, chúng ta sẽ cho giây (biến second) tăng lên 1 đơn vị sau mỗi giây (dùng câu lệnh delay). Tuy nhiên sau đó, chúng ta phải kiểm tra biến second này, nếu nó là 60, chúng ta cho nó về 0 và tăng phút (biến minute) lên một đơn vị, đúng theo nguyên tắc của đồng hồ. Chương trình gợi ý cho chúng ta như sau:



Hình 6: Thay đổi giây, và cập nhật phút khi đã hết 60 giây

Bài tập trên lớp: Giáo viên cho học sinh tham khảo và hiện thực lại chương trình trên. Học sinh lưu ý vào màu của các câu lệnh để tìm kiếm các câu lệnh trong danh mục có màu tương ứng.

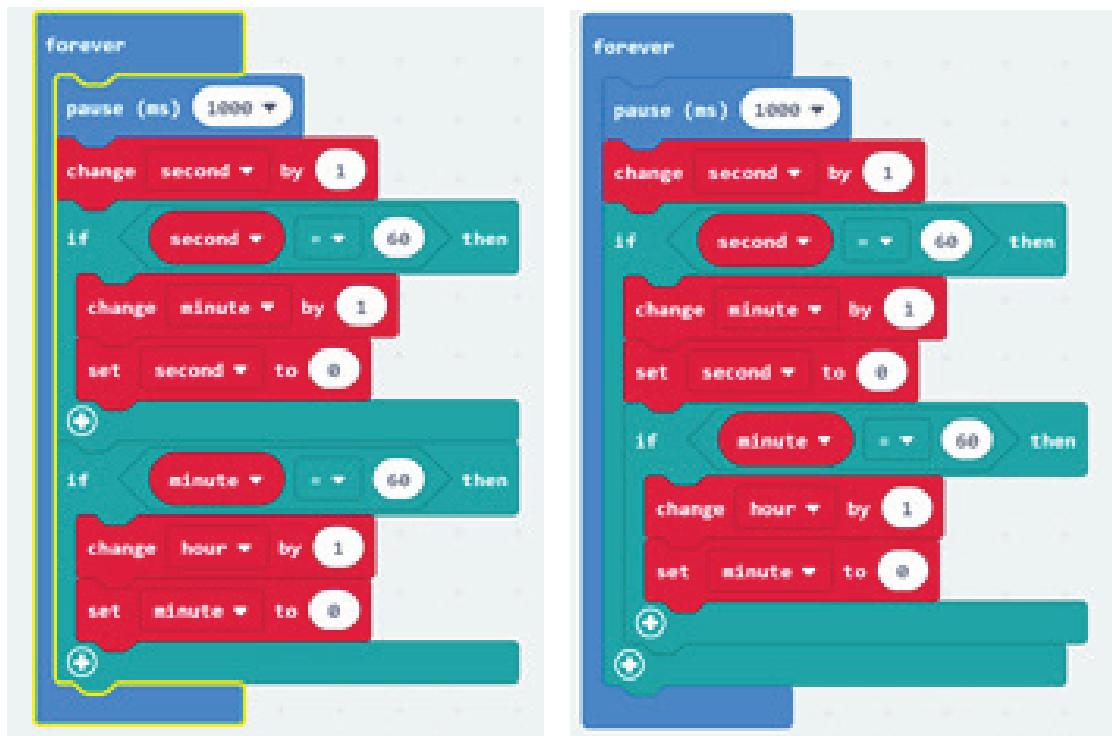
Bài tập trên lớp: Giáo viên cho học sinh hiện thực tiếp chức năng tiếp theo, là tăng giờ mỗi 60 phút. Đáp án gợi ý sẽ như sau:



Hình 7: Tăng giờ mỗi 60 phút

Bài tập trên lớp:

Giáo viên cho học sinh phân biệt sự khác nhau giữa 2 chương trình sau đây:



Hình 8: Phân biệt sự khác nhau giữa 2 chương trình

Đáp án: Về chức năng, 2 chương trình này hoàn toàn giống nhau. Tuy nhiên chương trình bên phải sẽ thực hiện tối ưu hơn, do việc kiểm tra phút để tăng giờ chỉ được thực hiện mỗi 60 giây 1 lần. Ở chương trình bên trái, việc kiểm tra phút được thực hiện mỗi giây một lần. Tuy nhiên, chương trình bên trái sẽ dễ đọc hiểu hơn là bên phải.

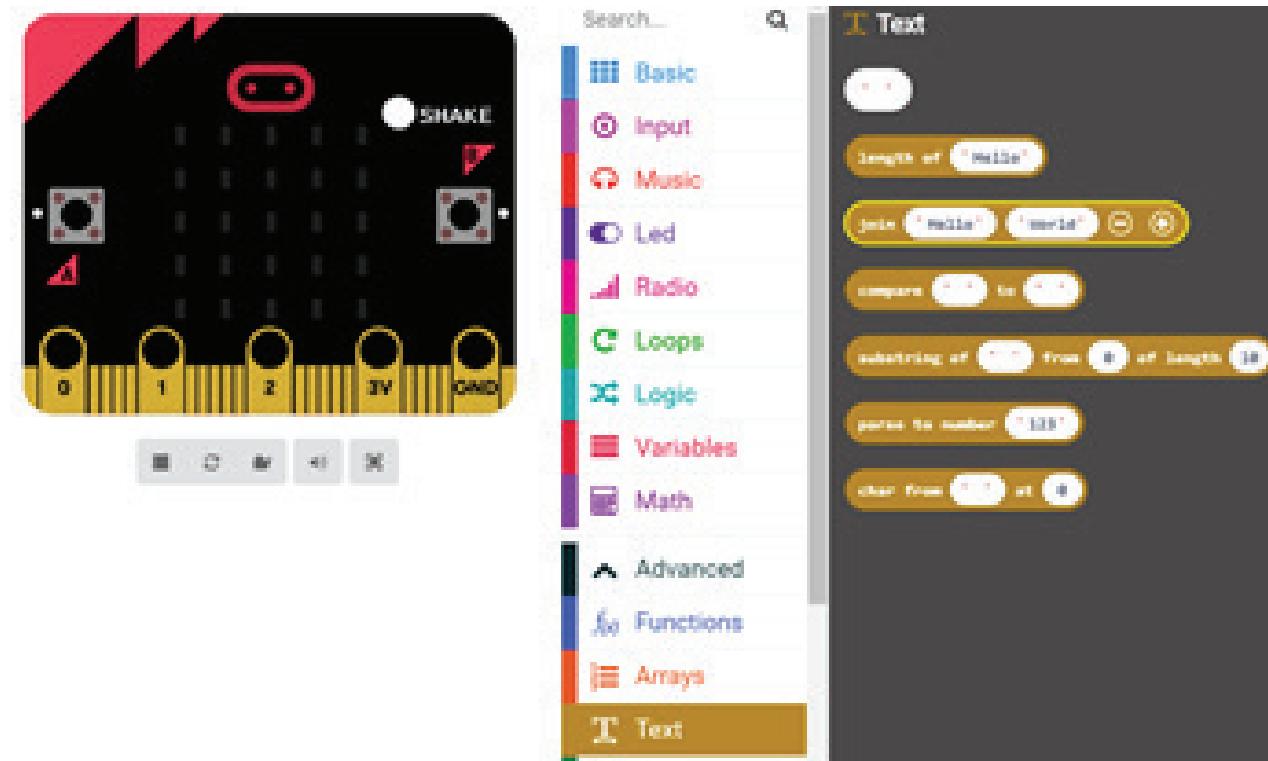
Bài tập trên lớp: Học sinh tiếp tục hiện thực tương tự cho giờ, khi đủ 24 giờ, thì cho giờ về lại 0. Đáp án cho bài tập này sẽ như sau:



Hình 9: Chương trình cập nhật giờ hoàn chỉnh

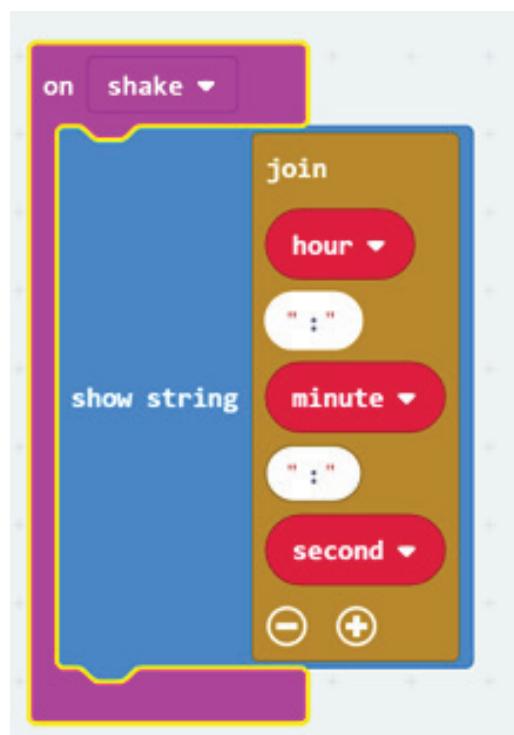
2.3.2 Hiển thị giờ ra màn hình

Để hiển thị thông tin giờ ra màn hình, chúng ta sẽ dùng câu lệnh ghép chuỗi trong mục Advance, Text và sau đó chọn join. Câu lệnh này cho phép chúng ta ghép nhiều chuỗi hiển thị lại với nhau. Để thêm thông tin ghép nối, chúng ta sẽ nhấn vào dấu + ở cuối câu lệnh join.



Hình 10: Câu lệnh join trong mục Advance/ Text

Vì đồng hồ chỉ hiển thị giờ khi chúng ta lắc tay, chương trình hiển thị giờ sẽ được hiện thực trong hàm on shake, như sau:

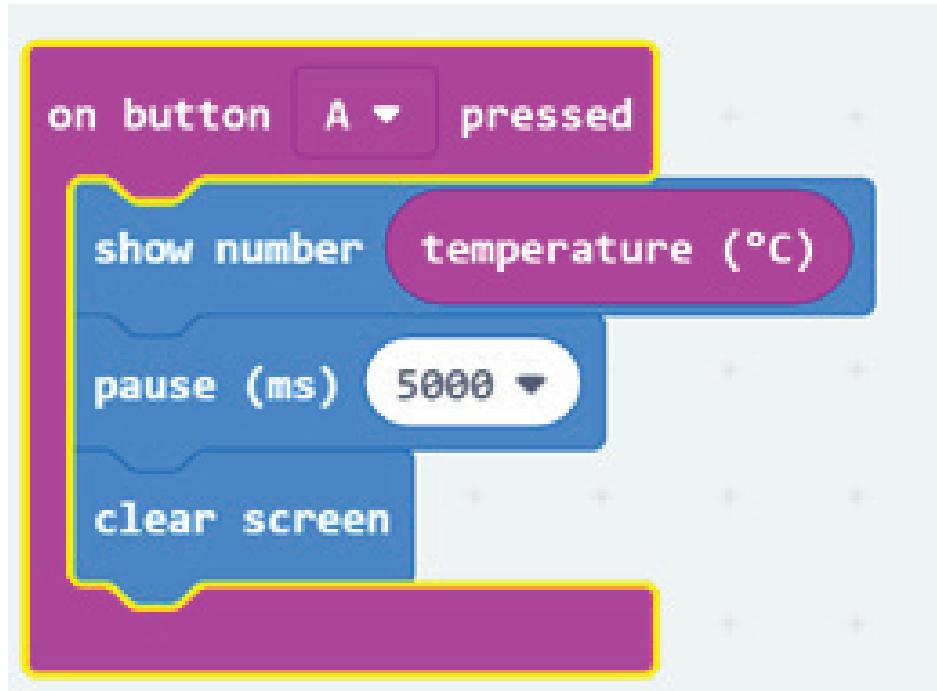


Chúng ta sẽ ghép thông tin giờ phút giây, cách nhau bằng dấu ":" để hiển thị ra màn hình. Khi muốn thêm 1 ô để ghép, chúng ta sẽ nhấn vào dấu + ở cuối lệnh join. Vì chuỗi thông tin này rất dài, nên khi hiển thị xong, đồng hồ sẽ tự động tắt, chúng ta không cần phải chủ động tắt màn hình.

Hình 11: Hiển thị đồng hồ ra màn hình khi lắc tay

2.3.3 Hiển thị nhiệt độ

Chương trình để hiển thị nhiệt độ sẽ được hiện thực khi nhấn vào nút A, như sau:

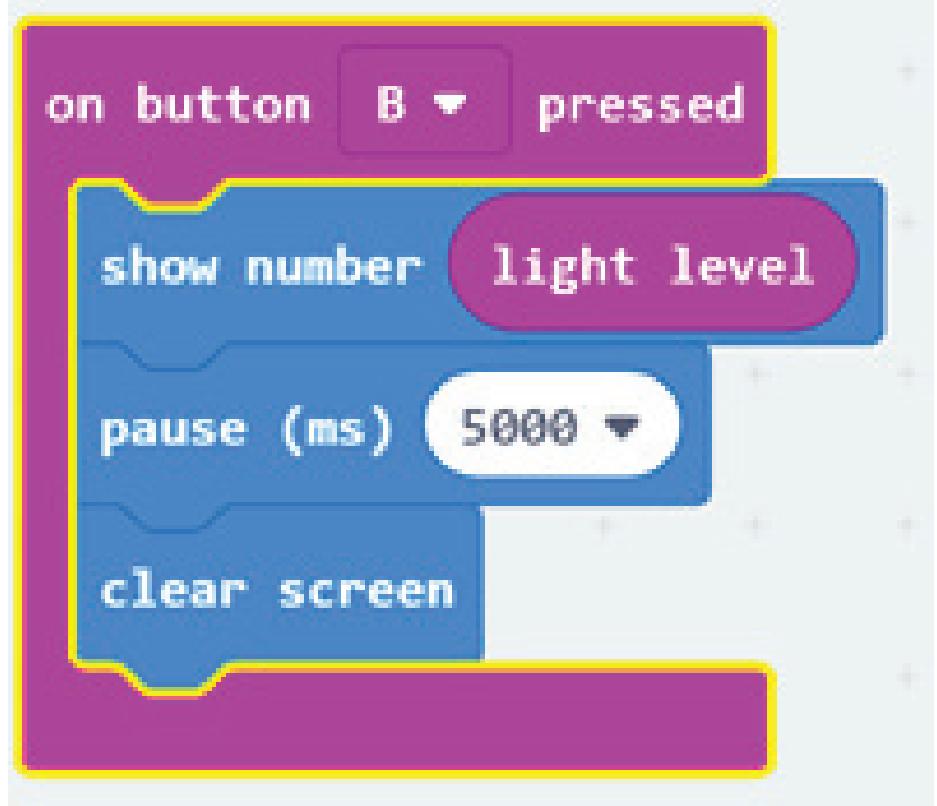


Hình 12: Hiển thị nhiệt độ khi nhấn nút A

Cần lưu ý là để có hiệu ứng chờ 5 giây, chúng ta bắt buộc phải nhập tay vào số 5000. Trong tùy chọn của câu lệnh pause, chúng ta chỉ có hiệu ứng đợi tối đa là 2 giây. Câu lệnh clear screen có thể được tìm thấy trong mục more, thuộc nhóm Basics.

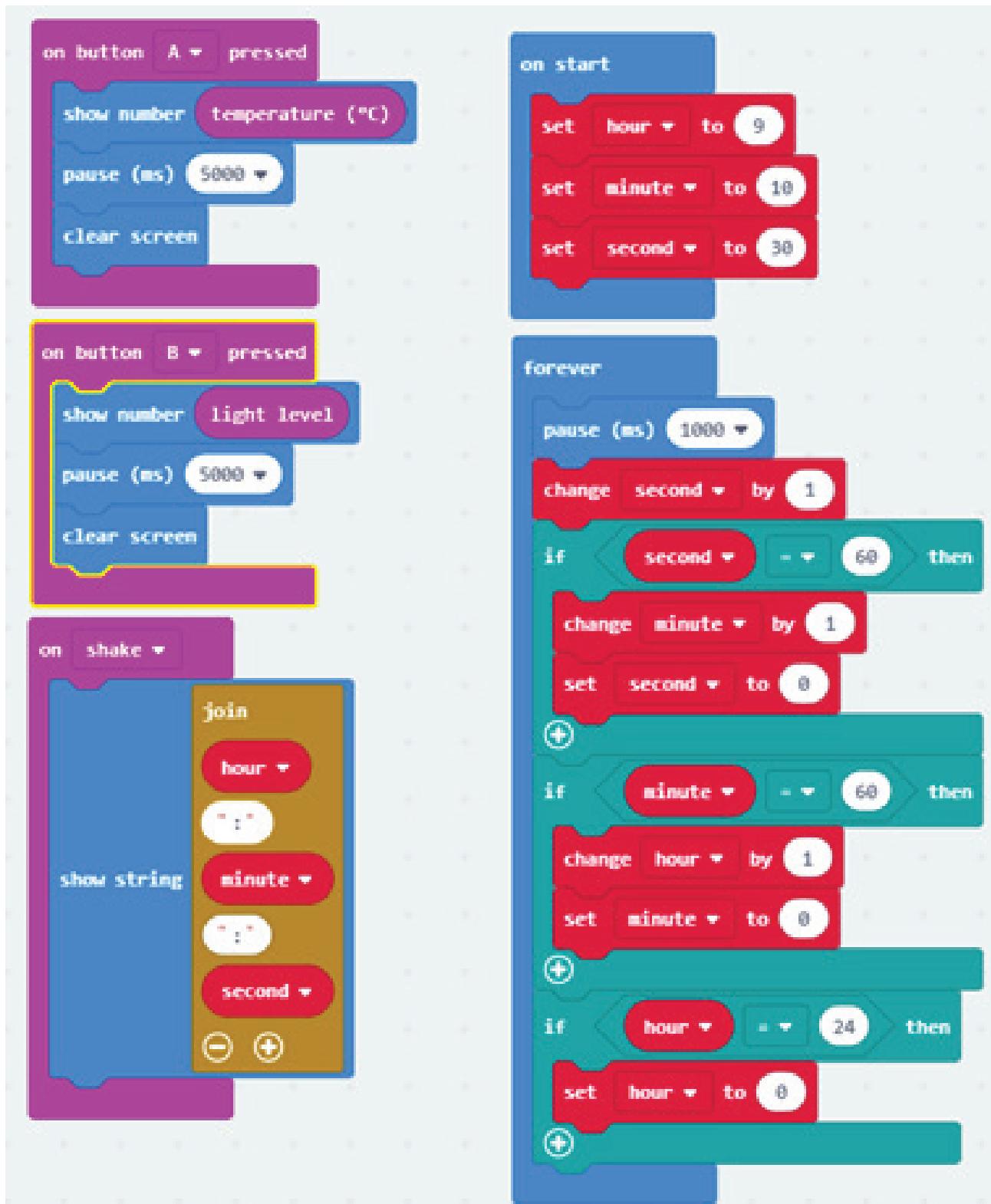
2.3.4. Hiển thị cường độ ánh sáng

Hoàn toàn tương tự như chương trình ở bên trên, chúng ta hiện thực chương trình cho nút B như sau:



Hình 13: Hiển thị cường độ ánh sáng khi nhấn nút B

Cuối cùng, chương trình hoàn thiện của chúng ta sẽ như sau:



Hình 14: Chương trình đồng hồ thông minh hoàn thiện

2.4. Các hướng phát triển

Chúng ta có thể phát triển chương trình này như một chiếc đồng hồ Casio điện tử. Với sức mạnh và sự hỗ trợ của MicroBit, nó hoàn toàn có thể làm được các chức năng như một đồng hồ Casio như chỉnh lại giờ cho đúng, hẹn giờ, bấm giờ thể thao. Trong hướng dẫn này, chúng tôi không trình bày các câu lệnh để hiện thực các chức năng trên. Chúc mọi người may mắn trong việc hiện thực nó!!



THE DARIU FOUNDATION