**BÀI 2: YOLO: HOME Ngoại vi cơ bản trên Yolo:Bit**

**1) Giới thiệu**

Giống như mô hình máy tính cơ bản, Yolo:Bit là một mạch tích hợp với bộ xử lý trung tâm, các thiết bị xuất dữ liệu và nhập dữ liệu. Nhờ việc thiết kế tích hợp này, bản thân mạch Yolo:Bit có thể dùng để làm nhiều dự án đơn giản mà không cần thêm các thiết bị ngoại vi khác cũng như các kết nối mở rộng. Tuy nhiên, trong phạm vi của giáo trình này, chúng tôi chỉ trình bày sơ lược một số ngoại vi quan trọng trên mạch Yolo:Bit, trước khi tập trung vào các ứng dụng liên quan đến Kết nối vạn vật và Trí tuệ nhân tạo.

Trong đa số các hệ thống điều khiển và xử lý nói chung, ngoại vi sẽ được chia làm 2 loại: Xuất dữ liệu và Nhập dữ liệu. Tuy nhiên ở mức độ cơ bản, chúng tôi tập trung vào 3 phân loại sau đây trên mạch Yolo:Bit: • Màn hình hiển thị: Với 25 đèn hiển thị độc lập, đây sẽ là công cụ hữu ích để làm việc trên mọi dự án và chương trình trên Yolo:Bit.

Màn hình hiển thị là công cụ quan trọng để hiện thị kết quả. Rõ ràng, nó là 1 ngoại vi thuộc nhóm xuất dữ liệu.

• Nút nhấn: Trên mạch Yolo:Bit hỗ trợ 2 nút nhấn tín hiệu là A và B. Nút nhấn Reset không thuộc trong nhóm này, nó là 1 dạng nút nhấn thuộc về hệ thống và không thể lập trình được. Nút nhấn thuộc nhóm nhập dữ liệu.

• Cảm biến: cũng thuộc nhóm nhập dữ liệu. Tuy nhiên, chúng tôi tách nó thành 1 nhóm mới để việc hướng dẫn được tập trung hơn. Cảm biến là các thiết bị cung cấp dữ liệu từ môi trường theo một điều kiện nào đó. Cảm biến chính là thông tin đầu vào cho các ứng dụng thông minh mà chúng ta gặp rất nhiều trong đời sống hiện nay. Một điểm thú vị trong mục này, là các cảm biến hành vi, chẳng hạn như lắc mạch, nghiêng trái hay nghiêng phải, được tích hợp sẵn trên mạch Yolo:Bit

Trong bài hướng dẫn này, chúng tôi sẽ trình bày các câu lệnh chính liên quan đến 3 nhóm thiết bị nói trên. Các chương trình trong bài hướng dẫn này sẽ là một công cụ kiểm tra cần thiết khi bạn đọc tích hợp thêm các tính năng mới và xây dựng một dự án lớn trong tương lai. Các mục tiêu chính trong bài hướng dẫn này được tóm tắt như sau:

• Hiển thị thông tin trên Yolo:Bit

• Làm việc với nút nhấn A và B

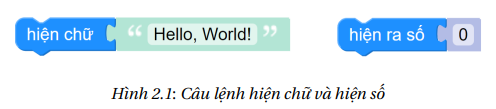
• Truy xuất giá trị cảm biến trên Yolo:Bit Các câu lệnh sử dụng trong bài hướng dẫn này thuộc 2 nhóm chính là CƠ BẢN và NGÕ VÀO. Đây có thể xem là 2 nhóm lệnh liên quan đến xuất kết quả và đọc dữ liệu đầu vào trên mạch Yolo:Bit. Đối với việc lập trình trên các mạch điện phần cứng, việc nắm rõ vai trò của một thiết bị (đầu vào hay đầu ra) là rất quan trọng để phát triển các chương trình trong tương lai

**2) Hiển thị thông tin trên Yolo:Bit**

Nói chung, các ngoại vi xuất kết quả thường có độ phức tạp thấp hơn nhiều so với các ngoại vi nhập dữ liệu. Trên mạch Yolo:Bit cũng không phải là ngoại lệ, khi các câu lệnh dùng cho việc hiển thị nằm trong nhóm CƠ BẢN. Các câu lệnh chính mà chúng ta thường sử dụng được trình bày như sau.

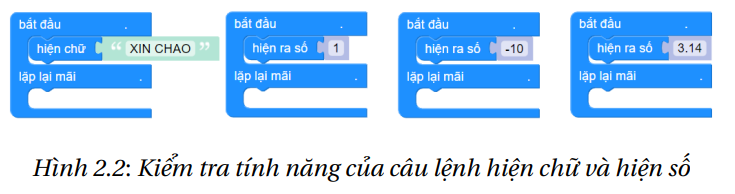
**2.1 Hiện chữ - Hiện số**

Cho dù là chương trình đơn giản hay phức tạp, bạn đọc hãy luôn luôn hiện ra 1 thông tin ra màn hình bằng câu lệnh hiện chữ. Câu lệnh này sẽ được đặt đầu tiên trong phần bắt đầu. Điểm lưu ý quan trọng là Yolo:Bit không hiển thị được tiếng Việt có dấu.



Khác với câu lệnh hiện số, câu lệnh hiện chữ có tầm ảnh hưởng rộng hơn. Phần nội dung của nó có thể được ghép với 1 con số cũng được. Ngược lại, câu lệnh hiện số chỉ có thể xuất được 1 con số ra màn hình mà thôi. Do đó, trong trường hợp không chắc chắn dữ liệu là chuỗi hay là số, bạn có thể dùng câu lệnh hiện chữ cho an toàn.

Để hiểu rõ hơn tính năng của 2 câu lệnh trên, bạn đọc có thể hiện thực các chương trình sau đây để kiểm tra tính năng của nó. Để thay đổi nội dung trong 2 câu lệnh này, chúng ta đơn giản là nhấp chuột vào và gõ nội dung mới

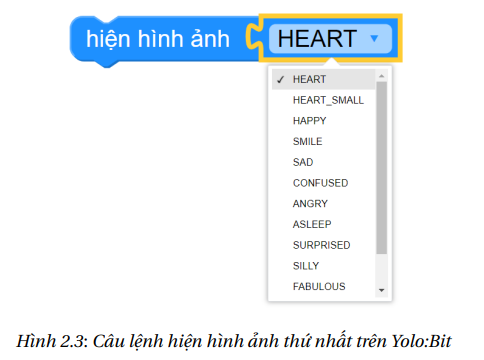


Với 2 câu lệnh này, khi thông tin chỉ có 1 chữ số hoặc 1 kí tự, nó sẽ được hiển thị cố định trên màn hình. Với các thông tin có hơn 2 chữ số hoặc 2 kí tự, nó sẽ được dịch chuyển từ phải qua trái. Câu lệnh hiện số có thể hiển thị được số âm lẫn số thập phân.

**2.2 Hiện hình ảnh**

Hiện hình ảnh là câu lệnh phổ biến tiếp theo khi làm việc với mạch Yolo:Bit. Trên Yolo:Bit, chúng ta có 2 phiên bản của câu lệnh này. Đầu tiên, trong câu lệnh thứ

nhất, chúng ta có một danh sách các hình ảnh có thể được chọn lựa, như trình bày ở hình bên dưới:



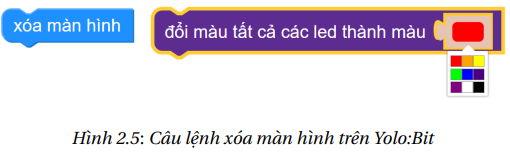
Các hình ảnh thường sử dụng như là mặt cười (HAPPY, SMILE) hoặc mặt khóc (SAD, ANGRY). Bạn đọc hoàn toàn có thể chủ động kiểm tra các hình ảnh này bằng cách sử dụng câu lệnh hiện hình ảnh trong phần bắt đầu của chương trình.

Trong câu lệnh hiện hình ảnh thứ 2, được trình bày như hình bên dưới, chúng ta có một màn hình 25 đèn để có thể chủ động tạo ra hình ảnh cho riêng mình. Thêm nữa, tại mỗi đèn, cũng có rất nhiều màu sắc khác nhau cho chúng ta lựa chọn.



**2.3 Xóa màn hình**

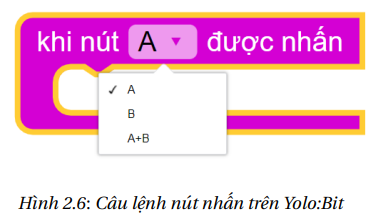
Bên cạnh việc hiển thị thông tin, xóa màn hình cũng là tính năng phổ biến trong các dự án. Chẳng hạn như khi xây dựng 1 ứng dụng nhập mật mã, chúng ta sẽ cho mật mã hiển thị ra trong 1 thời gian ngắn rồi xóa nó đi. Câu lệnh này được trình bày như hình bên dưới.



Bên cạnh câu lệnh thứ nhất, xóa màn hình, câu lệnh thứ 2 cũng là một câu lệnh tiện dụng, câu lệnh đổi màu tất cả các led thành màu. Câu lệnh này nằm trong nhóm LED. Với lựa chọn màu đen, tất cả các đèn sẽ tắt.

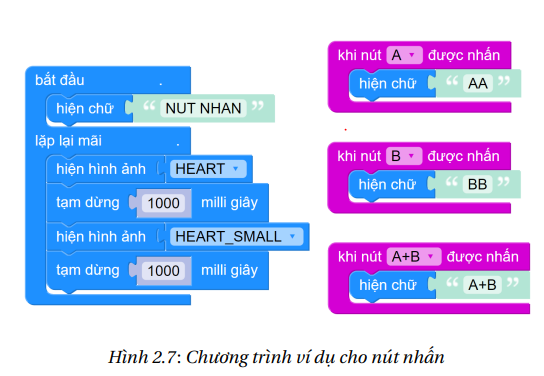
**3) Nút nhấn trên Yolo:Bit**

Trên mạch Yolo:Bit hỗ trợ 2 nút nhấn là A và B. Để có thể biết được khi nào một nút được nhấn, môi trường lập trình hỗ trợ cho chúng ta câu lệnh khi nút được nhấn. Câu lệnh này nằm cuối cùng trong nhóm lệnh NGÕ VÀO, như được trình bày ở hình bên dưới:



Đây là câu lệnh có 3 lựa chọn, dành cho nút A, nút B hoặc khi cả A và B được nhấn đồng thời. Điều đặc biệt của khối lệnh này là nó là một dạng câu lệnh sự kiện. Tức là mặc dù chương trình đang thực hiện tính năng nào đó trong phần lặp mãi mãi, mỗi khi có tín hiệu nút nhấn, chương trình sẽ tạm ngưng lại để thực hiện các câu lệnh trong phần sự kiện.

Cũng bởi vì câu lệnh này là câu lệnh sự kiện, nó có thể đứng riêng lẻ trong môi trường lập trình mà không cần thiết phải được ghép với 1 câu lệnh nào cả, như minh họa ở chương trình sau đây

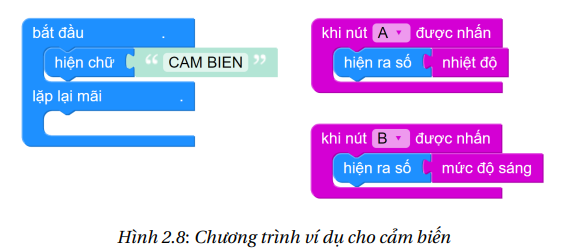


Với chương trình ví dụ này, trong khi hiển thị hình trái tim luân phiên nhau, mỗi khi nút nhấn được nhấn, chương trình sẽ ngay lập tức chuyển sang thực hiện các câu lệnh bên trong sự kiện tương ứng, rồi mới quay lại thực hiện tiếp các câu lệnh trong phần lặp mãi mãi. Chương trình trong các khối sự kiện nút nhấn còn gọi là chương trình ngắt. Đây là một kiến trúc rất đặc trưng trên các hệ thống vi điều khiển nói chung, và mạch Yolo:Bit nói riêng. Nhờ kiến trúc này, trong khi đang thực hiện 1 chức năng trong lặp mãi mãi, nó có thể tạm dừng để thực hiện 1 số chức năng ưu tiên trong các khối lệnh sự kiện khi một nút nhấn được nhấn.

**4) Cảm biến trên Yolo:Bit**

**4.1 Nhiệt độ và Ánh sáng**

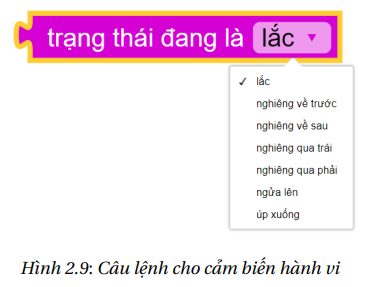
Tích hợp sẵn trên mạch Yolo:Bit là 2 cảm biến vền nhiệt độ và cường độ sáng. Hai thông tin này đều nằm trong mục NGÕ VÀO. Một chương trình để kiểm tra 2 thông tin này được gợi ý như sau:



Do thông tin cảm biến đều là dữ liệu số, do đó chúng ta phải xài câu lệnh hiện số để hiển thị kết quả. Một lưu ý quan trọng dành cho thông tin nhiệt độ, vốn là nhiệt độ trên bo mạch Yolo:Bit và không phải là nhiệt độ của môi trường. Do đó, nó thường sẽ cao hơn nhiệt độ môi trường không khí. Tuy nhiên, trong các ví dụ minh họa, bạn cũng có thể sử dụng thông tin này và trừ đi 1 lượng nhỏ để biển diễn nhiệt độ không khí.

**4.2 Cảm biến hành vi**

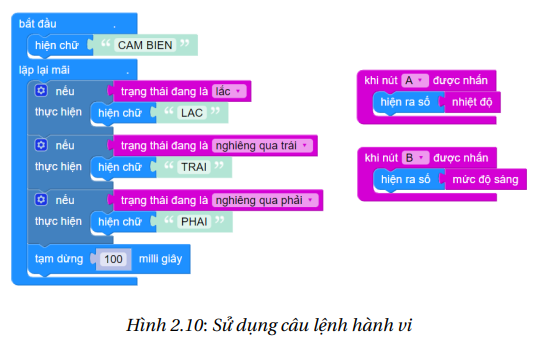
Nhờ cảm biến gia tốc và gốc xoay tích hợp sẵn trên mạch, Yolo:Bit có khả năng nhận biết hành vi tương tác của người dùng. Đây là công nghệ được ứng dụng trong các điện thoại thông minh, khi nó có khả năng nhận biết bạn đang đi bộ, leo câu thang hay đang chạy bộ. Khối lệnh về cảm biến hành vi được trình bày bên dưới, với nhiều lựa chọn khác nhau cho các ứng dụng tương tác.



Tuy nhiên, khối lệnh hành vi này là một dạng điều kiện, với kết quả là Đúng hoặc Sai. Cho nên để sử dụng nó, chúng ta sẽ phải xài kết hợp với câu lệnh nếu ... thực hiện. Câu lệnh này nằm trong mục LOGIC.

Thêm nữa, việc kiểm tra hành vi tương tác sẽ phải làm thường xuyên. Do đó, các câu lệnh nếu ... thực hiện sẽ được đặt trong khối lặp mãi mãi. Tuy nhiên, để hệ thống ổn định hơn, bạn nên đặt thêm 1 khối lệnh tạm dừng 100 mili giây. Đối với người bình thường, 100ms là 1 khoảng thời gian rất nhỏ và khó nhận ra. Tuy nhiên đối với hệ thống xử lý, nó là một khoảng thời gian rất dài để nó có thể khởi tạo lại các thông số cảm biến.

Một chương trình gợi ý để sử dụng câu lệnh này như sau



Với chương trình này, bạn có thể cầm và lắc mạch Yolo:Bit để nó có thể hiện ra dòng chữ LAC hoặc nghiêng mạch sang trái hay sang phải. Trong tương lai, tương tác kiểu hành vi này có thể dùng cho việc hiện thực 1 tay cầm điều khiển Robot từ xa hoặc điều khiển thiết bị bằng hành vi, mà không cần phải nhấn trực tiếp vào nút nhấn.