

# Thông điệp từ team rero EDU@Cytron

Xin chào Jr Maker,

Chào mừng các bạn đã lên tàu!

Chúng tôi là Adam & Anna. Rất vui khi các bạn tham gia vào team Makers. Ở những trang tiếp theo, chúng tôi sẽ hướng dẫn các bước để xây dựng ZOOM.BIT... và rất sớm thôi bạn sẽ có cho mình một chiếc xe hơi robot Bạn sẽ học cách viết chương trình và sử dụng ZOOM:BIT để thực hiện một số thủ thuật khiến bạn bè ngạc nhiên. Chắc chắn rằng các bạn sẽ có rất nhiều niềm vui với nhau nữa đấy!



Nếu bạn gặp phải bất kỳ sự cố nào trong quá trình học tập, hãy liên hệ với chúng tôi qua Telegram **t.me/zoombit\_support** Chúng tôi luôn sẵn sàng hỗ trợ. Bạn đã sẵn sàng chưa? Bắt đầu thôi nào! Adam & Anna



https://t.me /zoombit\_support

## Khám phá STEM & Viết code với ZOOM:BIT

Tác giả Cheryl Ng & SC Lim

> Minh họa **Suhana Oazmi**

Bản dịch tiếng Việt Thai July @ Cytron Việt Nam

Sản xuất bởi



Bản quyền thuộc về Cytron Technologies Mọi quyền được bảo lưu.

# Danh mục

Có gì trong chiếc hộp ?	1
Hãy cùng xây dựng nào !	3
Chương 1 : Hello world ! (LED Ma trận trên micro:bit)	17
Chương 2 : Hát một bài! (Còi Piezo /Loa trên micro:bit)	32
Chương 3 : Bật đèn lên Tadaa! (Đèn pha LED)	43
Chương 4 : Hãy di chuyển nào (Động cơ DC)	50
Chương 5 : Báo hiệu bạn đang đi đâu (LED RGB trên REKA:BIT)	58
Chương 6 : Nhìn sang trái, nhìn sang phải (Động cơ Servo)	63
Chương 7 : Vượt chướng ngại vật (Cảm biến siêu âm)	71
Chương 8 : Đi đúng hướng! (Cảm biến dò line Maker)	80
Chương 9 : Tất cả trong một, thay đổi chế độ	93
Chương bổ sung: Điều khiển từ xa (Truyền thông vô tuyến)	103
Hành trình học tập của tôi	108

# Có gì trong chiếc hộp?





Bản đồ để robot di chuyển



Bo mạch REKA:BIT



Cảm biến siêu âm

Cytron

Cảm biến dò line Maker

PRODUCT OF



2 x Mô-đun đèn LED



4 x Cáp Grove









- Dọn sạch hộp, tháo tất cả các bộ phận và đục lỗ như hình minh họa.
- 2 Gắn REKA:BIT lên đầu hộp; bạn có thể sử dụng các lỗ đinh tán để căn chỉnh vị trí.
  - Chèn bốn (4) đinh tán màu đen qua các lỗ rồi ấn mạnh để cố định bo mạch REKA:BIT vào đúng vị trí.









5

 $\bigcirc$   $\bigcirc$ 

Sử dụng bu lông và đai ốc để siết chặt động cơ DC vào mặt bên của hộp có nhãn MOTOR 2 (bánh xe bên phải) như hình trên.

- Lặp lại để gắn động cơ DC khác vào bên có nhãn MOTOR 1 (bánh xe bên trái).
- Luồn dây điện qua các lỗ có nhãn MOTOR 1 wire (bánh xe bên trái) và MOTOR 2 (bánh xe bên phải).

Các dây phải hướng vào trong và rãnh hướng ra ngoài



Kết nối dây động cơ với các đầu nối MOTOR 1 và MOTOR 2 trên REKA:BIT :-

7

(i) gắn phần lõi của dây điện vào, sau đó

(ii) siết chặt các vít bằng tuốc nơ vít được cung cấp để cố định các kết nối vào vị trí

#### Các kết nối dây

Độ	ng cơ	Đầu nối động cơ
MOTOR	Đen (-)	M1A
1	Đỏ (+)	M1B
MOTOR	Đỏ (+)	M2A
2	Đen (-)	M2B





- 8 Gắn bánh xe chắc chắn vào trục động cơ DC.
  - Xoay hộp lại và định vị bánh xe ở vị trí đã chỉ định.
- Chèn hai (2) đinh tán màu đen qua các lỗ và ấn mạnh để cố định bánh xe vào đáy hộp.









#### Nối một dây cáp Grove tới cảm biến Maker Line

\*Nếu bạn đang sử dụng micro:bit V1, KHÔNG nối dây TRẮNG với cảm biến Maker Line. Để dây màu trắng không được kết nối.

Định vị cảm biến Maker Line ở vị trí được
chỉ định và sử dụng hai (2) đinh tán màu
đen để cố định nó vào vị trí.

#### Các kết nối dây:

2	Cáp Grove	Cảm biến Maker Line
)	🔨 Trắng	CAL (Hiệu chỉnh)
	Vàng	AN (Tương tự)
	Đen	GND (Ground)
	Đỏ	VCC (Nguồn)









Luồn dây cáp qua các lỗ như minh họa.

1 4

9

Kết nối cáp cảm biến Maker Line với cổng P1:P9 trên REKA:BIT.







1 5

Nối dây cáp Grove đến Mô-đun đèn LED

1 6

Làm tương tự với Mô-đun LED còn lại \* Không nối dây trắng

Sử dụng hai miếng băng keo hai mặt để gắn các mô-đun LED vào mặt trước của hộp như hình trên.

#### Các kết nối dây:

Cáp Grove	Mô-đun LED
Trắng	Không kết nối
Vàng	Vào (Đầu vào)
Đen	GND (Ground)
Đỏ	VCC (Nguồn)









Luồn cả hai dây cáp LED bên phải và bên trái qua các lỗ như được dán nhãn.

- 1 9
- Kết nối cáp LED bên trái với cổng P13:P14 trên REKA:BIT.
- Kết nối cáp LED bên phải với cổng P15:P16 trên REKA:BIT.

#### Các kết nối dây:

Mô-đun LED	Cổng REKA:BIT
Trái	P13:P14
Phải	P15:P16



Gắn các bìa cứng trước vào hộp bằng bốn (4) đinh tán màu trắng; sử dụng hai đinh tán ở mỗi bên.

Hạ động cơ servo vào lỗ cho đến khi nó nằm chắc chắn ở vị trí như minh họa ở trên.









- 2 3
- Luồn cáp động cơ servo qua lỗ như hình minh họa.
- 2 4
- Nối cáp động cơ Servo đến cổng Servo S1.
- Dùng 1 miếng băng keo hai mặt để gắn cánh tay động cơ servo hình chữ thập đến bìa cứng như hình minh họa.

#### Các kết nối dây:

Cáp động cơ Servo	Cổng Servo S1
Cam	S (Signal)
Đỏ	+ (Nguồn)
Nâu	- (Ground)







2

Gắn cánh tay vào trục động cơ servo. Sử dụng vít và tuốc nơ vít được cung cấp để cố định miếng bìa cứng vào đúng vị trí.

- 2
  - Gắn một cáp Grove vào cảm biến siêu âm.
  - Gắn cảm biến siêu âm vào miếng bìa cứng như hình minh họa.

#### Các kết nối dây:

Cáp Grove	Cảm biến siêu âm
Đỏ	VCC (Nguồn)
Vàng	Trig (Trigger)
Trắng	Echo (Echo)
Đen	GND (Ground)

14









Luồn cáp cảm biến siêu âm qua lỗ.

- 3 0
- Gấp bìa cứng dọc theo các đường gấp nếp và nhét các miếng bìa vào các khe của chúng để tạo thành phần đầu robot.
- 3 1
- Kết nối cáp cảm biến siêu âm với cổng P2:P12 trên REKA:BIT.









1 Mở trình duyệt và truy cập <u>https://makecode.microbit.org</u>.

2

Nhấp chọn [New Project]. Đặt tên cho dự án của bạn và chọn [Create].





#### Microsoft MakeCode Editor cho phép bạn dễ dàng xây dựng mã của mình bằng

phương pháp kéo và thả.





В

Chọn lập trình bằng Khối, JavaScript hoặc Python

- Xuất bản và chia sẻ dự án của bạn
  - Mở Menu Trợ giúp
- D

Thay đổi cài đặt, thêm tiện ích mở rộng, ghép đôi thiết bị, v.v



**Cửa Sổ Mô phỏng** - Hiển thị micro:bit mô phỏng chương trình của bạn

**Ngăn Kéo Danh Mục -** Nhận các khối mã bạn cần tại đây. Nhấp để xem các khối mã hóa có sẵn cho từng danh mục.



Không gian làm việc - Xây dựng đoạn mã của bạn tại đây bằng cách ghép các khối lại với nhau



Nhấp để tải đoạn mã xuống ZOOM:BIT



Đặt tên và lưu dự án hiện tại vào máy tính của bạn



Tạo GitHub repository



Hoàn tác / Quay lại



Phóng to / Thu nhỏ



<sup>3</sup> Nhấp chọn danh mục [Basic] và lựa chọn khối [show string ("Hello!")].



"Hello!"

show string

4

21

Nhấp vào danh mục **[Basic]** một lần nữa và chọn khối **[show icon]**. Lặp lại quá trình đó và thêm khối **[show icon]**. Nhấn chọn và kéo thả cả 02 khối **[show icon]** lên khối **[forever]**.



5

6



Nhấp vào biểu tượng của khối thứ 2 [show icon] và chọn hình '**trái tim nhỏ**' từ cửa sổ bật lên.







Bạn có thể thấy được mô phỏng mã của mình trong Makecode Editor. Dòng chữ "Hello" xuất hiện và chỉ cuộn qua một lần nhưng hình động trái tim đập cứ lặp đi lặp lại. Bạn có biết tại sao không?





#### Kết nối cáp USB đến máy tính và robot ZOOM:BIT như hình bên dưới.





Nhấp vào nút [**Download]** . Trong cửa sổ bật lên, chọn tải dự án của bạn xuống ổ MICROBIT rồi nhấp vào [**Save]**.

9 Chọn [Done] để đóng cửa sổ bật lên khi nó hiển thị "Download completed".



8

**Lưu ý:** Nếu của sổ không bật lên, điều này có nghĩa rằng tệp tin được tự động tải đến thư mục lưu tệp tin mặc định của trình duyệt. Nhấp chuột phải vào tệp tin .hex ở góc trái của màn hình và chọn "Show in folder".

Chọn và kéo thả tệp tin đã tải lưu "microbit-xxxx.hex" đến ổ MICROBIT như là bạn đang sao chép tệp đến ổ đĩa flash.











Bạn có thấy "Hello!" cuộn qua LED ma trận theo sau là hình động trái tim đang đập? Nếu bạn bỏ lỡ nó, bạn có thể trượt công tắc nguồn sang vị trí OFF, sau đó sang ON để khởi động lại bo mạch.



#### Lưu ý:

mã đến thiết bị. Sau khi đã được kết nối thành công, bạn có thể flash đoạn mã đến ZOOM:BIT chỉ với 1 cái click chuột. Yeah!~

11

Kết nối ZOOM:BIT đến PC của bạn. Nhấp vào ba dấu chấm bên cạnh nút **[Download],** sau đó chọn **[Connect device]**.





Làm theo sự hướng dẫn trên màn hình. Chọn **'BBC micro:bit CMSIS-DAP**' hoặc **'DAPLink CMSIS-DAP**' từ danh sách, sau đó chọn **[Connect]**.



**Lưu ý:** Bạn nên sử dụng trình duyệt Microsoft Edge và Chrome, và cài firmware mới nhất cho micro:bit trước khi ghép đôi. Nếu bạn có các vấn đề liên quan đến kết nối thiết bị hãy truy cập https://makecode.microbit.org/device/usb/webusb/troubleshoot để được hỗ trợ thêm.



# Khám phá thêm nhiều khối





### Và đây là một thử thách THÚ VI dành cho bạn!!

Bạn có thể giải mã những gì ZOOM:BIT đang hỏi không? Lập trình cho robot của bạn trả lời bằng cách sử dụng cùng một mã bí mật.






 Nhấp vào danh mục [Input] sau đó chọn khối [on button (A) pressed]. Nhấp chọn danh mục [Music] sau đó chọn khối [start melody (dadadum) repeating (once)].

E Blocks

🚮 JavaScript 🗸

Hãy dạy ZOOM:BIT hát .. Do Re Mi ~ Bạn có thể tạo một dự án mới hoặc tiếp tục thêm các khối vào mã trước đó của mình.





\* < 0 \*

Hicrosoft | micro:bit

#### 3

#### Nhấp vào [dadadum] và chọn giai điệu 'birthday' từ danh sách sổ xuống.





Nhấp vào nút A của trình mô phỏng trên màn hình của bạn. Bạn có nghe thấy một giai điệu quen thuộc? Hãy thử xem các giai điệu khác nữa nhé~

Lưu ý:

Hãy đảm bảo loa máy tính của bạn đang được BẬT.



**Bạn có biết?** Bên cạnh danh sách các giai điệu cài sẵn, bạn cũng có thể lập trình ZOOM:BIT phát bất kỳ bài hát nào bạn thích. Tuy nhiên, bạn sẽ cần dạy nó từng nốt bằng cách sử dụng các khối [ play tone (middle C) for (1 beat) ] và [ rest (ms) (1 beat) ] từ danh mục [ Music ].





# Hãy thử lập trình ZOOM:BIT để chơi các nốt nhạc mở đầu của nhạc phim STAR WARS~~





Tone	Middle D	Middle D	Middle D	Middle G	High D	
Beat	1/3	1/3	1/3	2	2	
Tone	High C	Middle B	Middle A	High G	High D	
Beat	1/3	1/3	1/3	2	1	Π
Tone	High C	Middle B	Middle A	High G	High D	
<b>Tone</b> Beat	High C 1/3	Middle B 1/3	Middle A 1/3	High G 2	High D 1	J
Tone Beat Tone	High C 1/3 High C	Middle B 1/3 Middle B	Middle A 1/3 High C	High G 2 Middle A	High D 1	

Hai dòng này giống nhau. Bạn có thể sử dụng khối vòng lặp để làm cho mã của bạn gọn hơn.



4

Thêm các khối sau vào mã của bạn. Bạn có thể tìm thấy các khối bạn cần từ các ngăn kéo danh mục có cùng màu.



Bạn có biết ?Tất cả các khối đều đượcphân chia theo màu. Bạn có thể tìm cáckhối có chức năng tương tự từ ngăn kéodanh mục có cùng màu.Nếu bạn cần thêm sự hướng dẫn, hãytruy cậphttps://link.cytron.io/zoombit-tutorial-2



37

 $\bigcirc$ 

anno 🔽 anno







#### Lưu ý:

ZOOM:BIT của bạn (với micro:bit V2) có thể "hát" và tạo nhạc vì nó có loa tích hợp cho phép tạo ra âm thanh. Nếu bạn đang sử dụng micro:bit V1 (không có loa tích hợp), bạn cần cắm loa Grove vào Cổng P0:P1 để phát âm thanh. Bạn có thể tham khảo https://link.cytron.io/zoombit-grove-buzzer để biết thêm chi tiết.





# Khám phá thêm nhiều khối



 $\bigcirc$ 

Bạn có thể lập trình ZOOM:BIT để phát các bài hát khác nếu bạn biết cách đọc nốt nhạc Dưới đây là hướng dẫn đơn giản giúp bạn "giải mã" một bản nhạc.







Vị trí của một nốt nhạc trên khuông nhạc (tức là năm đường kẻ ngang) cho chúng ta biết âm nào sẽ chơi. Nốt nằm trên khuông nhạc càng cao thì cao độ/tần số của âm thanh càng cao và ngược lại.. Các ký hiệu âm nhạc khác nhau được sử dụng để cho chúng ta biết khoảng thời gian(tức là thời lượng) một nốt nhạc sẽ được chơi.

 $\odot$ 

## Và đây là một thử thách THÚ VI dành cho bạn!!

Dạy ZOOM:BIT để "hát" bài hát yêu thích của bạn. Bạn sẽ cần lập trình nó, ghi chú từng nốt nhạc.

Nếu bạn không có một bài hát nào trong đầu, thì hãy thử giai điệu sau:



Nốt	Middle E	Middle G	Middle C	Rest	Middle A	High C	Middle F	Middle A
Nhịp	1	1/2	2	1/2	1	1/2	2	1/2
Nốt	Middle B	Middle G	Middle A	Middle B	High D	High C	Đó là một giai điệu rất quen thuộc. Bạn có	
Nhịp	1/2	1/2	1/2	1/2	1/2	1 1/2	đoàn đượ nào k	chông?





**Bạn có biết?** LED ma trận trên micro:bit cũng có thể hoạt động như cảm biến ánh sáng. Hãy lập trình ZOOM:BIT để tự động BẬT đèn pha khi xung quanh tối và TẮT khi trời sáng.



Tạo một dự án mới trong MakeCode Editor của bạn. Nhấp vào biểu tượng bánh răng 🔅 sau đó chọn **'Extensions'.** \*Bạn cần kết nối Internet để thêm tiện ích mở rộng.



1

Tiện ích mở rộng là tập hợp các khối tùy chỉnh mà chúng ta thêm vào MakeCode Editor để dễ dàng lập trình các phụ kiện cho micro:bit, chẳng hạn như robot ZOOM:BIT!



O

Nhập **'zoombit'** (hoặc https://github.com/CytronTechnologists/pxt-zoombit) vào hộp tìm kiếm và nhấp Enter.

Nhấp để chọn tiện ích mở rộng **'zoombit'** Đợi nó tải và bạn sẽ nhận thấy các ngăn danh mục mới sau đây được thêm vào Trình chỉnh sửa MakeCode của bạn.



45

2



Tải mã xuống ZOOM:BIT của bạn và bật nó lên. Quan sát đèn pha của nó.



#### Đọc mức độ ánh sáng

ZOOM:BIT có bật đèn pha không? Nếu không, hãy thử tạo bóng trên ma trận LED. Tiếp theo, cố gắng chiếu đèn pin sáng vào ma trận LED; bạn quan sát thấy được những gì?

Lưu ý:

5

Mức độ ánh sáng dao động từ 0 (không phát hiện thấy ánh sáng) đến 255 (độ sáng tối đa).





# Khám phá nhiều khối hơn

Trong Trình chỉnh sửa MakeCode của bạn, hãy tạo một dự án mới, thêm các khối bên dưới rồi tải xuống ZOOM:BIT của bạn. Bạn quan sát thấy gì khi nhấn nút A + B? Và cả về nút A nữa?





"Chuyển đổi" có nghĩa là chuyển từ trạng thái này sang trạng thái khác. Nếu trạng thái hiện tại là BÂT thì nó sẽ chuyển sang TẮT và ngược lại.

Khối này làm chậm chương trình để bạn có thể quan sát đèn pha BẬT và TẤT.

(i) Mức độ ánh sáng trong phòng của bạn bây giờ là bao nhiêu? Mức độ ánh sáng đọc được khi bạn chiếu ánh sáng chói vào ma trận LED là bao nhiêu?
\* Để có độ chính xác, hãy ghi lại ít nhất 3-4 lần đọc và sau đó tính giá trị trung bình.
(ii) Bạn có thấy đèn pha nhấp nháy sau khi nhấn nút A không?
Tắt nguồn để dừng lại.



## Và đây là một thử thách THÚ VI dành cho bạn!!

Dạy ZOOM:BIT để giao tiếp bằng Mã Morse. Lập trình ZOOM:BIT của bạn để nháy đèn pha khi nhấn nút A hoặc B.

Bật nút A được nhấn	BẬT cả hai đèn pha trong 500ms rồi TẮT	Chấm
Bật nút B được nhấn	<b>BẬT</b> cả hai đèn pha trong <b>1500ms</b> rồi <b>TẮT</b>	Cách
Tham khảo b được cung cá tin nhắn S.O. đúng trình tự Video demo https://link.c	iểu đồ International Morse Code ấp, bạn có thể lấy ZOOM:BIT để flash S bằng cách nhấn nút A và nút B theo ự không ? có sẵn tại cytron.io/zoombit-morse-code	

# **CHAPTER 4**



https://link.cytron.io /zoombit-chapter-4



Let's Get Moving!

Trước khi chúng ta bắt đầu lập trình ZOOM:BIT để di chuyển, hãy kiểm tra để đảm bảo rằng chúng ta đã kết nối đúng cách.



- Trượt công tắc nguồn sang BẬT.
- Nhấn lần lượt các nút M1A, M1B, M2A và M2B trên REKA:BIT và quan sát hướng quay của các bánh xe.





#### Lưu ý:

Nếu các bánh xe không quay theo hướng như các mũi tên màu đỏ, bạn cần kiểm tra và sửa các kết nối động cơ DC. Bạn có thể tham khảo trang 5-6.

 $\odot$ 

1

Bây giờ chúng ta đã sẵn sàng lập trình ZOOM:BIT để di chuyển xung quanh... Hãy bắt đầu!

Input



Tạo một dự án mới trong MakeCode Editor của bạn và thêm phần mở rộng ZOOM:BIT (bạn có thể tham khảo trang 44-45).

Xây dựng mã dưới đây. Bạn có thể lấy các khối bạn cần từ các ngăn kéo danh mục này:

📇 ZOOM:BIT

Basic

2



Bạn có thể truy cập https://link.cytron.io/zoombit-tutorial-4 để biết hướng dẫn từng bước cách tạo mã.



Tải mã xuống ZOOM:BIT và bật nó lên.

Nhấn Nút A, Nút B, rồi nút A+B cùng một lúc. Quan sát phản hồi của ZOOM:BIT.

Bật nút A được nhấn	Rẽ phải trong 500 mili giây.	
Bật nút B được nhấn	Rẽ trái trong 500 mili giây.	
Bật nút A+B được nhấn	Di chuyển về phía trước trong 1 giây	zip!zip!m!
		CON.BIT.

## Và đây là một thử thách THÚ VI dành cho bạn!!

Tìm một khoảng không rộng rãi và thiết lập một chướng ngại vật bằng cách đặt ngẫu nhiên các đồ vật, chẳng hạn như ghế, sách hoặc hộp các tông, dọc theo đường đi của ZOOM:BIT. Hãy thách thức anh chị em hoặc bạn bè của bạn hướng dẫn thủ công ZOOM:BIT để tìm đường thoát ra.



- Nhấn A+B để đi thẳng.
- Nhấn A để rẽ phải.
- Nhấn B để rẽ trái

Người được thách thức nào mất thời gian ngắn nhất (hoặc số lần di chuyển ít nhất) để dẫn ZOOM:BIT ra khỏi chướng ngại vật là NGƯỜI CHIẾN THẮNG!



 Bạn có biết?

 guay. Hãy thử các giá trị khác nhau để hoàn thành bảng sau.

 90°

 on button A ▼ pressed

128

at speed

500 -

90

....

180°

Tốc độ	Độ trễ	Góc
128	500	
	250	60
255	500	

0°

turn right 🔻

pause (ms)

brake

# Khám phá nhiều khối hơn!



#### Bạn có biết ?

nếu cả hai bánh xe quay với tốc độ khác nhau, ZOOM:BIT sẽ quay về phía bánh xe đang quay với tốc độ thấp hơn. Trong ví dụ trên, ZOOM:BIT sẽ di chuyển về phía trước nhưng lái sang trái theo thời gian vì bánh xe bên trái đang quay ở tốc độ thấp hơn.

Bạn có thể dự đoán hướng ZOOM:BIT sẽ di chuyển nếu chúng ta đặt Tốc độ bên trái thành -150 và Tốc độ bên phải thành -200 không? Hãy kiểm tra nó và xem thử bạn trả lời đúng!





#### Bạn có biết ?

Chắc chắn có một sự khác biệt nhỏ giữa thông số kỹ thuật của động cơ và hiệu suất thực tế của nó. Các động cơ dường như giống hệt nhau có khả năng quay với tốc độ hơi khác nhau mặc dù chúng được cung cấp cùng một điện áp. Nói cách khác, ngay cả khi bạn lập trình ZOOM:BIT của mình để di chuyển thẳng (nghĩa là cùng tốc độ cho cả bánh xe bên trái và bên phải), vẫn có khả năng ZOOM:BIT sẽ hơi xoay sang phải hoặc trái sau một thời gian.

Độ chính xác và nhất quán của các chuyển động của ZOOM:BIT cũng có thể bị ảnh hưởng bởi mức pin cũng như tình trạng của bề mặt mà nó tiếp xúc. ZOOM:BIT có thể di chuyển chậm hơn khi mức pin yếu và khi mặt đất quá mềm hoặc không bằng phẳng.





Có hai đèn LED RGB trên bo mạch REKA:BIT, được gắn nhãn "0" và "1". Bạn có thể lập trình để chúng phát sáng với các màu khác nhau bằng cách sử dụng các khối từ ngăn danh mục [REKA:BIT].



Thêm các khối được đánh dấu sau đây từ các ngăn danh mục [Basic], [Loops] và [REKA:BIT] vào mã của bạn từ bài học trước.



 $\odot$ 

4100 0000

Tải mã xuống ZOOM:BIT và bật nó lên.

2

3

Nhấn Nút A, Nút B, rồi nút A+B cùng một lúc. Quan sát đèn LED RGB trên bo mạch REKA:BIT.



Bạn có nhận thấy đèn LED RGB " 0 " ở bên phải nhấp nháy trước khi ZOOM:BIT rẽ phải không? Và đèn LED RGB " 1 " bên trái nhấp nháy trước khi ZOOM:BIT rẽ trái? Và cả hai đèn LED RGB đều sáng màu xanh lam khi ZOOM:BIT đang tiến về phía trước?



# Khám phá nhiều khối hơn!



 $\odot$ 

## Và đây là một thử thách THÚ VI dành cho bạn!!

Bạn có thể lập trình ZOOM:BIT để nhấp nháy đèn LED RGB giống như đèn khẩn cấp của xe cảnh sát không? Và để có hiệu quả cao hơn, hãy làm cho nó phát ra tiếng còi báo động\*?



\* Đối với tiếng còi
báo động, bạn có thể xen kẽ giữa nốt C
giữa và nốt F# giữa
trong một vòng lặp.
\* Bạn có thể bỏ qua
phần còi báo động
nếu đang sử dụng
micro:bit V1 không
có loa/còi.





Đầu của ZOOM:BIT được gắn với động cơ servo 180 độ. Nói cách khác, bạn có thể lập trình ZOOM:BIT để nhìn thẳng về phía trước và tương tự như vậy, bạn có thể làm cho ZOOM:BIT quay đầu sang trái hoặc phải bằng cách điều khiển servo quay sang góc mong muốn của bạn. Hãy thử!



Tạo một dự án mới trong MakeCode Editor của bạn và thêm phần mở rộng ZOOM:BIT. Ban có thể tham khảo trang 44-45.

Xây dựng mã sau đây. Bạn có thể lấy các khối bạn cần từ các ngăn kéo sau:



Basic

1

CREKA:BIT

 $\odot$ Input



3 Tải mã xuống ZOOM:BIT và bật nó lên.

Nhấn Nút A, Nút B, rồi nút A+B; quan sát hướng mà đầu đang phải đối mặt.



ZOOM:BIT của bạn có nhìn thẳng về phía trước sau khi bạn nhấn Nút A+B không? Nếu nó KHÔNG được căn chỉnh đúng cách, thì bạn cần tháo đầu ra, định vị lại và sau đó gắn lại vào cánh tay động cơ servo.



65



Nếu sau khi điều chỉnh lại đầu theo cách thủ công, bạn vẫn thấy đầu bị xoay nhẹ sang phải/trái trong khi lẽ ra nó phải nhìn thẳng, thì bạn có thể sửa lại bằng cách điều chỉnh mã của mình. Thực hiện theo các bước bên dưới để xác định "hiệu chỉnh góc" cho ZOOM:BIT của bạn.



1

2

Tạo một dự án mới trong MakeCode Editor của bạn và thêm phần mở rộng ZOOM:BIT (bạn có thể tham khảo trang 44-45).

Nhấp vào danh mục [**Variables**] rồi chọn [**Make a Variable**]. Đặt tên cho biến của bạn (ví dụ: "angle\_correction") rồi nhấp vào nút [**Ok].** 





### 3

#### Xây dựng mã sau đây. Bạn có thể lấy các khối bạn cần từ các ngăn kéo này:

I Basic C REKA:BIT I Variables O Input C Logic III Math



Để được hướng dẫn thêm, bạn có thể tham khảo https://link.cytron.io/ zoombittutorial-6a để biết hướng dẫn từng bước xây dựng mã.



 $\odot$
Tải mã xuống ZOOM:BIT và bật nó lên. Nghiêng ZOOM:BIT sang trái (hoặc sang phải, tùy theo điều kiện nào áp dụng) để quay đầu theo hướng đó. Khi bạn hài lòng rằng đầu hướng thẳng về phía trước, hãy giữ ZOOM:BIT với logo micro:bit hướng lên trên (và đầu của ZOOM:BIT hướng xuống dưới) để nhận giá trị "angle\_ Correction".

4

5



Bây giờ bạn đã biết giá trị [angle\_correction] cho ZOOM:BIT của mình, bạn có thể sử dụng giá trị đó trong các dự án tương lai của mình để đảm bảo rằng đầu được quay theo góc bạn muốn.







Đây là mã mẫu bao gồm hiệu chỉnh góc. Khi bật nguồn, ZOOM:BIT sẽ nhìn sang trái, rồi sang phải và cuối cùng nhìn thẳng về phía trước.





### Và đây là một thử thách THÚ VI dành cho bạn!!

Bạn có thể lập trình ZOOM:BIT để nhảy không? Hãy sáng tạo với các bước di chuyển; tạo ZOOM:BIT xoắn và xoay

Với micro:bit V2, bạn có thể sử dụng khối [On Loud Sound] từ danh mục [Input] làm trình kích hoạt để ZOOM:BIT bắt đầu nhảy; và bạn cũng có thể thêm các khối [Music] để làm cho màn trình diễn sống động hơn!



## **CHAPTER 7**



https://link.cytron.io /zoombit-chapter-7



**Obstacle** Detected!



## Giờ đây, ZOOM:BIT đã có thể di động, hãy dạy bạn ấy không va vào chướng ngại vật trên đường đi.

Tạo một dự án mới trong MakeCode Editor của bạn và thêm phần mở rộng ZOOM:BIT (bạn có thể tham khảo trang 44-45).

Xây dựng mã này. Bạn có thể lấy các khối bạn cần từ các ngăn kéo danh mục này:

2



Bạn sẽ cần tạo một biến mới ("khoảng cách") và đặt biến đó luôn nhận giá trị của cảm biến siêu âm. Bạn có thể truy cập https://link.cytron.io/zoombit-tutorial-7 để biết hướng dẫn từng bước cách tạo mã nếu bạn cần thêm hướng dẫn.







#### Tải mã xuống ZOOM:BIT của bạn và bật nó lên.







ZOOM:BIT sẽ tiếp tục di chuyển về phía trước khi không phát hiện thấy chướng ngại vật.

Cố gắng giơ tay trước ZOOM:BIT. Robot của bạn có dừng lại khi cách tay bạn khoảng 10cm không?

Từ từ di chuyển bàn tay của bạn về phía ZOOM:BIT. Quan sát phản ứng của nó khi khoảng cách nhỏ h 10cm.

O C

73

0

 $\odot$ 

Hãy làm cho ZOOM:BIT rẽ phải khi nhấn nút A và rẽ trái khi nhấn nút B khi nó ở chế độ đứng yên, tức là dừng cách chướng ngại vật 10 cm.



Bạn có biết ? Chúng ta có thể biến các khối mã thực hiện một tác vụ cụ thể thành một hàm. Sau khi tạo một hàm, bạn có thể sử dụng hàm đó ở nhiều vị trí trong chương trình của mình mà không cần phải tạo đi tạo lại cùng một khối mã. Ngoài ra, các lập trình viên chuyên nghiêp cũng sử dụng các hàm để làm cho mã của họ dễ đọc hơn đối với người khác.



Nhấp vào [Make a Function], đổi tên doSomething thành 'turn right' rồi nhấp vào nút **[Done].** Một khối **[funtion turn\_right]** sẽ được thêm vào không gian làm việc của bạn.





75

Lặp lại để tạo một chức năng khác và đổi tên nó 'turn\_left'.

6

7

Tiếp tục xây dựng mã của bạn bằng cách thêm các khối sau vào khối [function] turn\_right] và [function turn\_left] của bạn.



76

Cuối cùng, nhấp vào danh mục [Functions] và thêm khối [call turn\_right] và [call turn\_left] vào mã của bạn. Đây là mã hoàn chỉnh:







77

 $\odot$ 

Vâng!! Giờ đây, ZOOM:BIT có thể tự do đi lang thang trong phòng của bạn mà không va vào đồ vật. Khi đường đi của ZOOM:BIT bị chướng ngại vật chặn, bạn có thể nhấn Nút A (để rẽ phải) hoặc Nút B (để rẽ trái) để hướng dẫn ZOOM:BIT vượt qua chướng ngại vật.





Tải mã xuống ZOOM:BIT của bạn và bật nó lên.



Bạn có nghĩ rằng bạn có thể sửa đổi mã để ZOOM:BIT có thể tự động di chuyển khỏi chướng ngại vật mà không cần chờ sự trợ giúp của bạn không? Hãy thử xem nào~



## Và đây là một thử thách THÚ VI dành cho bạn!!

Biến ZOOM:BIT thành đàn piano siêu âm. Lập trình ZOOM:BIT để phát các âm khác nhau để đáp ứng với việc đọc cảm biến siêu âm của nó.





Điều khiển ZOOM:BIT để hát một bài hát. Di chuyển lòng bàn tay của bạn về phía hoặc ra xa khỏi mặt của nó để ZOOM:BIT phát âm thanh mà bạn muốn. Nếu bạn không chắc nó hoạt động như thế nào, hãy quét mã QR để xem video demo.

ZOOM:BIT





Bạn có biết ?ZOOM:BIT có thể được lập trình để chạy theo đường vẽ không?ZOOM:BIT có thể dễ dàng làm được điều đó vì nó được trang bị cảm biến Maker Line.Cảm biến cho phép nó phát hiện một đường kẻ (đen hoặc trắng) trên nền có màu tươngphản.



#### Cảm biến Maker Line Mặt phía trên

Trượt công tắc màu của đường kẻ sang "DARK" vì bản đồ được cung cấp đang sử dụng đường kẻ màu đen (trên nền trắng)

 $\odot$ 

Trước khi bạn bắt đầu lập trình ZOOM:BIT, trước tiên hãy làm theo các bước bên dưới để hiệu chỉnh cảm biến Maker Line. Chỉ cần thực hiện hiệu chuẩn một lần trừ khi chiều cao, đường kẻ hoặc màu nền của cảm biến thay đổi.



1

2

- Bày đường đua đã được cung cấp. Đặt ZOOM:BIT trên đường đua và bật nguồn.
- Nhấn và giữ nút CALIBRATE cho đến khi cả 5 đèn LED sáng lên; chỉ nhả nút khi tất cả các đèn LED đều nhấp nháy (tức là Maker Line đã vào chế độ hiệu chỉnh.)
  - Di chuyển thủ công ZOOM:BIT từ bên này sang bên kia trên đường màu đen. Lặp lại nhiều lần và đảm bảo rằng tất cả các cảm biến đã được tiếp xúc với đường dây.

Nhấn lai nút CALIBRATE để thoát khỏi chế đô hiệu chuẩn.



Sau khi cân chỉnh thành công, bạn sẽ thấy các đèn LED chạy liên tục. Khi đó cảm biến Dò line Maker đã sẵn sàng. Hãy quét mã này để xem cách nó hoạt động.





Tạo một dự án mới trong MakeCode Editor của bạn và thêm phần mở rộng ZOOM:BIT (bạn có thể tham khảo trang 44-45).

<sup>2</sup> Tạo đoạn mã sau để hướng dẫn ZOOM:BIT theo dõi. Bạn có thể lấy các khối bạn cần từ các ngăn danh mục được liệt kê:



#### Tải mã xuống ZOOM:BIT của bạn. Bật nguồn, đặt nó trên rãnh và nhấn nút A.

3



Thật kinh ngạc khi ZOOM:BIT của bạn thu nhỏ và di chuyển xung quanh đường đua sau khi bạn nhấn Nút A. Bạn có thể tìm ra cách thức hoạt động của mã này không?





Khi bắt đầu, hãy hiển thị biểu tượng 📙.

Sau đó không cần làm gì cả miễn là nút A không được nhấn. Nếu nút A được nhấn, thoát khỏi vòng lặp while. Chơi giai điệu 'jump up' một lần.

Liên tục kiếm tra giá trị từ Maker Line và hành động.

Điều kiện	Nhận dạng đường kẻ?	Làm sao để ở lại Quay lại đường đua?
	Giữa	Chạy thẳng
	Trái	Rẽ nhẹ sang bên trái
	Phải	Rẽ nhẹ sang bên phải
	Xa bên trái	Rẽ trái
	Xa bên phải	Rẽ phải

 $\odot$ 

 $\odot$ 4110 000 Đôi khi ZOOMBIT của bạn có đi lạc hướng không, đặc biệt là khi chúng ở ngay đường cua? Khi ZOOM:BIT đang rẽ tại góc cua, cảm biến Maker Line có thể cách xa đường kẻ đó trong giây lát (như hình bên dưới). Khi điều này xảy ra, ZOOM:BIT bị nhầm lẫn vì trong mã của chúng ta trước đó đã không cho ZOOM:BIT biết phải làm gì khi không phát hiện thấy dòng kẻ nào.

Để ngăn ZOOM:BIT đi lang thang, chúng ta cần dạy ZOOM:BIT tìm đường trở lại đúng hướng bằng cách rẽ theo cùng một hướng (như trước khi mất khả năng phát hiện đường thẳng) ... cho đến khi phát hiện lại đường thẳng.

Chúng ta có thể thêm một biến "vị trí" vào mã của mình cho mục đích đó. Chuyển sang trang tiếp theo để tìm hiểu cách cải thiện mã trước đây của chúng tôi.

Nhấn vào danh mục **[Variables]** sau đó chọn **[Make a Variable]**. Hãy đặt tên cho biến của bạn (ví dụ như "position") và sau đó nhấn nút **[Ok]**.



Thêm các khối sau **[set (position) to ( \_\_ )]** từ ngăn kéo danh mục **[Variables]** đến mã của bạn.



Đặt biến [position] là 0 khi ZOOM:BIT được bật lên.



Đặt biến [position] là "1" khi đường kẻ được nhận diện bên trái hoặc xa về phía trái, đặt "2" khi đường kẻ được nhận diện bên phải hoặc xa về phía bên phải.



Nhấn vào biểu tượng 🕞 và them 1 điều kiện khác **"else-if"**. Sau đó, thêm các khối được đánh dấu sau vào mã của bạn.





Tải mã đã hoàn thành xuống ZOOM:BIT của bạn. Bật nguồn, đặt nó trên rãnh và nhấn nút A.



Hãy kiểm tra. Cố gắng đẩy ZOOM:BIT ra khỏi đường dẫn (cho đến khi Maker Line không phát hiện thấy dòng nào). Bạn có nhận thấy ZOOM:BIT điều chỉnh vị trí của nó để trở lại đúng hướng, thay vì lang thang hay không ?





### Và đây là một thử thách THÚ VI dành cho bạn!!

Bạn có thể lập trình ZOOM:BIT để làm các bước sau không :-

- Chạy quanh đường đua khi nhấn nút A
- zip! zip! 200m! Chơi một giai điệu trong 1/2 nhịp bất cứ khi nào nó vượt qua vạch đích
- Hiển thi số vòng chay đã hoàn thành và
- Dừng lại sau khi đã hoàn thành ba (3) vòng.

Mẹo: Chúng ta có thể lập trình ZOOM:BIT để 'biết' rằng nó đã vượt qua vạch đích bằng cách sử dụng [line detected on (all)].



60°0 000

#### Bạn có biết ?

**CO DIET ?** Bên cạnh việc sử dụng đường đua có sẵn, bạn cũng có thể sáng tạo và thiết kế đường đua của riêng mình bằng cách sử dụng băng keo điện màu đen. Bạn có thể dễ dàng mua một cái ở bất kỳ cửa hàng tiện lợi nào. Chúc bạn có thật nhiều khoảnh khắc vui vẻ khi thiết kế đường đua riêng cho ZOOM:BIT~

 $\bigcirc$ 



# OCHAPTER 9

LET'S START!



https://link.cytron.io /zoombit-chapter-9

> Bringing Everything Together. You've Got This!

et's try alli

Chúng ta đã cùng nhau dạy ZOOM:BIT nhiều thủ thuật và anh ấy đã học được từng mẹo một. Bây giờ, hãy đào tạo ZOOM:BIT để sắp xếp tất cả chúng lại - chuyển đổi từ chế độ này sang chế độ khác một cách dễ dàng.





Tạo một dự án mới trong MakeCode Editor của bạn và thêm phần mở rộng ZOOM:BIT (bạn có thể tham khảo trang 44-45). Sau đó, xây dựng mã sau đây cho **chế độ thủ công**.

Bỏ khối phát nhạc này nếu bạn đang sử dụng micro:bit V1.





Tiếp theo, hãy thêm các chế độ khác. Để làm điều đó, chúng ta sẽ sử dụng hàm.



2

Nhấp vào [Advanced] rồi chọn danh mục [Functions]. Nhấp vào [Make a Function], đổi tên doSomething thành 'obstacle\_avoidance' rồi nhấp vào nút [Done]. Khối [function obstacle\_avoidance] sẽ được thêm vào không gian làm việc của ban.





Tiếp tục xây dựng mã của bạn cho **chế độ tránh chướng ngại vật** bằng cách thêm các khối vào khối chức năng.



Bạn có thể nhấn vào biểu tượng 🐼 Để thu nhỏ đoạn mã sau khi lập trình xong.

Nhấn vào biểu tượng Nếu bạn cần mở rộng & kiểm tra

Bạn có nhận thấy rằng mã tương tự như những gì bạn đã xây dựng trong Chương 7 không? Tuy nhiên, ở đây, các khối nằm trong khối [function obstacle\_avoidance], thay vì khối [forever].



 $\odot$ 

atter T atter



Lặp lại Bước 2 để tạo một chức năng khác cho **chế độ theo dòng.** 

Thêm các khối vào khối **[function line\_following]** như được hiển thị.

Mã này tương tự như những gì bạn đã xây dựng trong Chương 8. Tuy nhiên, ở đây, các khối nằm trong khối [function line\_following] thay vì khối [mãi mãi].

\*Hãy lưu ý rằng bạn sẽ cần tạo một biến mới [position].

Nếu bạn chưa chắc chắn về cách làm trên thì có thể truy cập https://link.cytron.io/zoombit-tutorial-9 để được hướng dẫn từng bước.







Tiếp theo, chúng ta sẽ thêm "modes" vào chương trình của mình để khi chúng ta thay đổi từ chế độ này sang chế độ khác, ZOOM:BIT sẽ tự động thực hiện tác vụ tương ứng mà chúng ta gán cho chế độ cụ thể đó.



Nhấp vào danh mục **[Variables]** và sau đó chọn **[Make a Variable]**. Đặt tên cho biến của bạn (ví dụ là "mode") và sau đó nhấn nút **[Ok]**.

Hicrosoft   Omicro:bit	🔹 Blocks  JavaScript 🗸	* < 0 *	
	Search Q Basic Input Make a Variable Make a Variable		
•••••••••••••••••••••••••••••••••••••••	© Led all Radio C Loops	New variable name:	0
	→ Logic   ■ Variables   ■ Math	mode	
	ZOOM:BIT REKA:BIT Neopixel Advanced		<sup>ok</sup>
Download •••	Lesson 6 - Angle Correcti		

Thêm các khối được đánh dấu sau vào mã của bạn. Bạn có thể lấy các khối bạn cần từ các ngăn kéo danh mục sau:





#### Tải mã xuống ZOOM:BIT của bạn và bây giờ bạn có thể mang ô tô rô-bốt của mình đi bất cứ đâu và khoe những thủ thuật mà nó có thể làm với bạn bè của bạn ~

Khi bắt đầu	Cho chạy âm thanh (hello), hiển thị mặt cười, và bật cả hai đèn pha với đầu hướng về phía trước. Đặt chế độ thành 0.	
Mãi mãi	Luôn kiểm tra "mode". Nếu Mode = 1, cho chạy Obstacle Avoidance function; Hoặc nếu Mode = 2, sau đó chạy Line Following function.	
<b>Trên logo nhấn</b> (or 'on logo down' if using micro:bit V1)	Thay đổi chế độ bằng 1, dừng di chuyển và hiển thị biểu tượng cho chế độ hiện tại - ô tô (Mode 0), trái tim (Mode1) và hình vuông (Mode 2). Nếu Mode không phải là 1 hoặc 2, thì hãy đặt Mode thành 0.	
Trên nút A được nhấn	Rẽ phải	Le rip! zi
Trên nút B được nhấn	Rẽ trái	
Trên các nút A+B được nhấn	Đi thẳng	0
		ZOOMIBIT

8

101

00

4000 T 4000

### Và đây là một thử thách THÚ VI dành cho bạn!!

Bạn có thể dạy ZOOM:BIT một thủ thuật mới không? Dạy anh ta giải phương trình toán học chẳng hạn? Hãy thử thêm một "Mode" khác vào mã của bạn trước đó.



102

## BONUS CHAPTER (LET'S START!)



https://link.cytron.io /zoombit-bonus-chapter

> Roger, Roger... Can You Hear Me?
**Bạn có biết ?** Micro:bit trên ZOOM:BIT của bạn được trang bị chức năng liên lạc vô tuyến. Nói cách khác, nếu bạn có một micro:bit khác, bạn có thể lập trình nó để sử dụng như một một bộ điều khiển từ xa để điều khiển ZOOM:BIT của bạn. Hãy thử Xem nào!

 $\odot$ 

Basic

1



Tạo đoạn mã sau và tải xuống bảng micro:bit mà bạn sẽ sử dụng làm bộ điều khiển từ xa. Bạn có thể lấy các khối bạn cần từ các ngăn kéo danh mục này:

Radio

Input





Tạo một dự án mới trong MakeCode Editor
của bạn và thêm phần mở rộng ZOOM:BIT
(bạn có thể tham khảo trang 44-45).

Xây dựng mã sau đây để kích hoạt ZOOM:BIT của bạn để nhận hướng dẫn từ bộ điều khiển từ xa.

Chúng ta cần đặt cả micro:bit (bộ điều khiển từ xa) và ZOOM:BIT vào cùng một nhóm radio để chúng truyền và nhận tín hiệu radio từ nhau. Trong ví dụ này, chúng ta đặt cả hai thành nhóm radio 1.

 $\odot$ 

4000 T 4000

3





## <sup>4</sup> Tải mã xuống ZOOM:BIT của bạn. Bật nguồn cho cả micro:bit (bộ điều khiển từ xa) và ZOOM:BIT.





#### Và đây là một thử thách THÚ VI dành cho bạn!!

Sửa đổi mã để thêm nhiều "hướng dẫn" hơn để ZOOM:BIT thực hiện - có thể là để truyền tải một thông điệp bí mật hoặc để gửi một món quà. Tạo bất ngờ cho các thành viên gia đình hoặc bạn bè của bạn – hãy trốn nơi nào đó và sau đó điều khiển từ xa ZOOM:BIT của bạn để tiếp cận họ.



Đây là một mẹo cho bạn. Bạn có thể thêm [trên nút B được nhấn] và [trên nút A+B được nhấn] vào mã điều khiển từ xa (micro:bit) ; và bạn sẽ cần thêm các điều kiện khác-nếu mới vào mã của ZOOM:BIT.





# Nhật ký học của tôi với ZOOM:BIT

Tôi đã hoàn thành việc xây dựng ZOOM:BIT của mình vào ngày \_\_\_\_\_\_; và cùng nhau khám phá những bài học trong cuốn sách này và thử sức với mọi thử thách.



## Nhật ký học tập của tôi với ZOOM:BIT



P/S: Hãy tìm một giáo viên / phụ huynh để xác nhận cho bạn!

# Thông điệp từ team rero EDU@Cytron

Tuyệt vời... CHÚC MỪNG!! Bạn đã chế tạo thành công xe rôbốt của riêng mình; và cùng với ZOOM:BIT, bạn đã học cách viết mã và hoàn thành các thử thách theo nhóm. Bạn đã làm rất tốt! Chúng tôi hy vọng bạn cũng đã có thật nhiều niềm vui trong suốt quá trình này.

Vì thì, tiếp theo là gì ? Bạn có thể truy cập www.cytron.io để khám phá và nhận các cảm biến hoặc bộ phận bổ sung để tùy chỉnh ô tô rô-bốt của mình. Còn việc thêm Màn hình Grove OLED vào cổng I2C thì sao? Hoặc có lẽ, thêm nhiều động cơ servo để tạo thành một cánh tay robot? Các khả năng là vô tận. Chúc các bạn khám phá vui vẻ~



Hãy chia sẻ hành trình ZOOM:BIT với chúng tôi trênTelegram <u>t.me./zoombit\_support</u> Chúng tôi rất muốn lắng nghe từ bạn. Xin chúc mừng~

https://t.me /zoombit\_support

Adam & Anna





