

10 Experiências para kit A - de iniciação à eletrónica (ID PR.FT3_25)

Guia

V 1.0 – Agosto 2023

Índice

1. Experiência 1 - Diga "Olá" ao KIT.....	3
2. Experiência 2 - Utilização de um LDR e entradas analógicas	10
3. Experiência 3 - Diminuir a intensidade de um LED usando um potenciômetro	16
4. Experiência 4 - Utilização de um transistor para ligar um motor..	22
5. Experiência 5 - Utilização do acelerômetro para controlar a velocidade do motor.....	26
6. Experiência 6 - Fazer soar uma campainha piezoelétrica.....	31
7. Experiência 7 - Iniciar uma experiência de Energia eólica.....	38
8. Experiência 8 - Fazer um jogo usando uma bússola	45
9. Experiência 9 - Criar uma experiência de carga de um condensador	51
10. Experiência 10 - Experiência de utilização do LED RGB, nas suas diversas vertentes.....	60

1. Experiência 1 - Diga "Olá" ao KIT

Introdução

"Olá Mundo!" é o termo usado para definir o primeiro programa que escrevemos numa linguagem de programação ou num novo componente de hardware. Essencialmente, é um simples pedaço de código que permite efetuar o primeiro passo com sucesso. Também dá a oportunidade de se certificar de que tudo está a funcionar e está OK.

Esta experiência requer:

1 micro:bit

1 Cabo USB Micro B

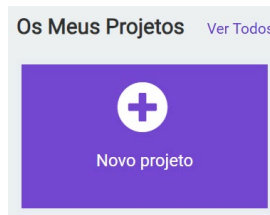
Para o primeiro "Hello World", vamos criar uma animação simples na matriz de LEDs que se repete para sempre. Para ver uma explicação passo a passo de como criámos o programa, continue a ler!

Construindo 'Olá Mundo!'

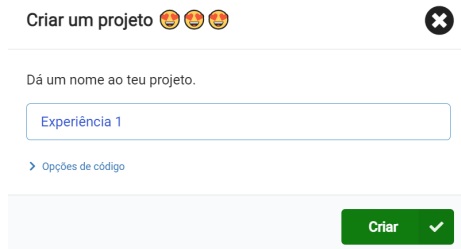
Um "Olá Mundo!" no micro:bit é um pouco diferente do que num microcontrolador normal. O micro:bit não tem um único LED para piscar sozinho, como se encontra no Arduino ou em placas semelhantes. O que o

micro:bit tem é um conjunto de LEDs! Assim, o "Olá Mundo !" para o micro:bit é desenhar algo usando a matriz de LEDs!

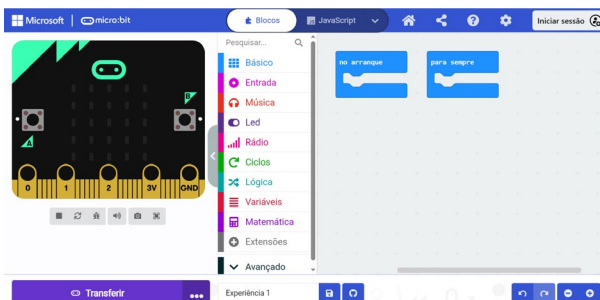
Quando abre o MakeCode (<https://makecode.microbit.org>) é tem acesso a Os Meus Projetos onde pode seleccionar um Novo projeto:



Ao clicar em Novo projeto aparecerá uma janela Criar um projeto, onde é convidado a dar um nome ao projeto :



Posteriormente, quando a aplicação abre, é recebido com dois blocos: o bloco no arranque e o bloco para sempre.



O bloco no arranque é todo o código que será executado no início do programa e só será executado uma vez.



O bloco para sempre é o código que vai repetir-se vezes sem conta... para sempre. Vamos usar o bloco para sempre para desenvolver este "Olá Mundo". Por agora precisamos de começar a adicionar blocos ao para sempre.



Primeiro, clique na categoria Básico. Estes blocos são os blocos de construção básicos de um programa BuildCode. Ele expandirá para uma série de opções.



Clique e arraste o bloco mostrar leds e coloque-o dentro do seu bloco para sempre. Repare que o bloco está programado para caber dentro do bloco para sempre e, se tiver o volume alto do computador, ouvirás um ruído de "clique" satisfatório quando soltar o bloco.

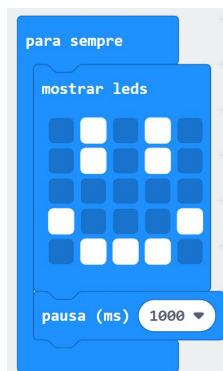


O bloco mostrar leds tem um conjunto de quadrados que simbolizam o conjunto de LEDs. Se clicar num quadrado, este fica vermelho, o que significa que está ligado. Desenhe uma forma simples de pixel art ligando ou desligando diferentes LEDs; deve poder ver o resultado no simulador, no lado esquerdo da sua janela.



Para transformar esta imagem estática numa animação, precisamos de outro bloco mostrar leds para colocar mesmo por baixo do primeiro bloco. Pode então fazer um segundo desenho com este conjunto de retângulos. No simulador, verás as imagens a mudar muito, muito depressa. Precisamos de abrandar isto!

Para abrandar a animação, utiliza-se o bloco de pausa , que se encontra no conjunto de blocos básicos. O bloco de pausa é exatamente o que diz; diz ao micro:bit para fazer uma pausa e esperar durante um certo período de tempo. Coloque um bloco de pausa no programa, como se mostra:



A razão pela qual estamos a colocar um bloco no fim é que este programa é um ciclo. Sem o bloco no fim, a imagem na animação vai mudar muito depressa.



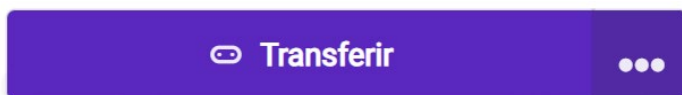
A seguir podem incluir mostrar string e escrever “Olá Mundo!” ou outro conjunto de caracteres. Deverá incluir um bloco pausa para criar um intervalo de tempo de transição no fim do ciclo.

Se quiser brincar com o código e fazer algumas alterações, clica no botão Editar, e ele abrirá um editor MakeCode para começar a alterar o "Olá Mundo!".

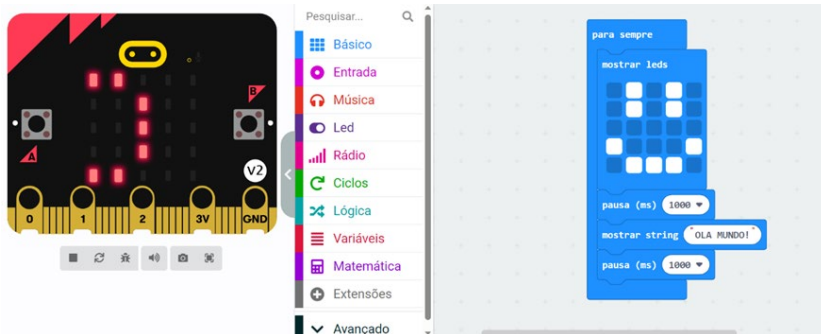
Divirta-se!

Descarregar o programa

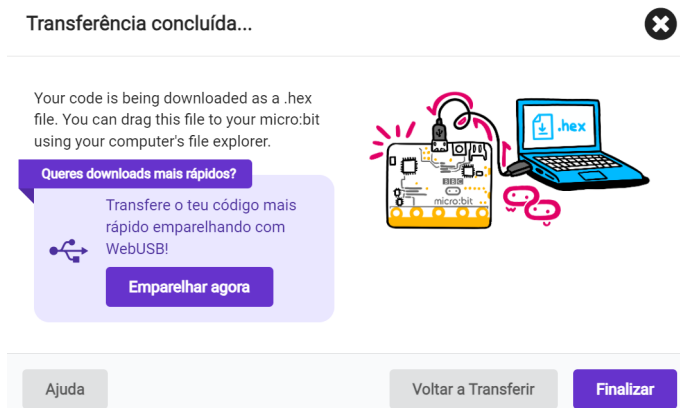
Copie e cole ou recrie o seguinte código no seu próprio editor MakeCode. Também pode descarregar este exemplo clicando no botão de descarregamento no canto superior direito da janela de código.



Irá descarregar o ficheiro do programa para o seu local de descarregamento padrão, provavelmente a pasta Downloads no seu computador, ou qualquer outro local que tenha definido nas suas preferências de descarregamento.



Depois basta clicar e arrastar o ficheiro do programa do local de download para a unidade micro:bit, que aparece como um dispositivo externo.



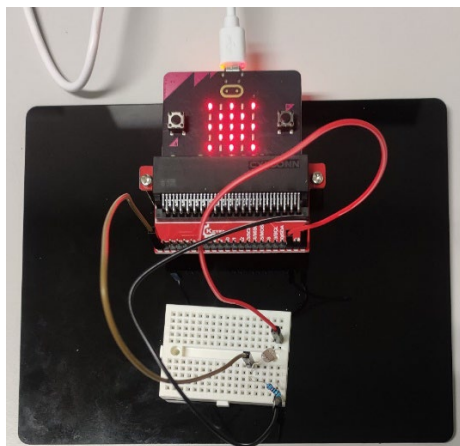
Se ainda não o fez, insira o cabo USB no micro:bit. Assim que aparecer como uma unidade, arraste e largue o ficheiro no micro:bit, e pronto!

O micro:bit vai piscar durante alguns segundos e depois o programa vai começar automaticamente.

Nota: Poderá ser necessário desativar o bloqueador de anúncios/pop-up para ver o circuito simulado!

2. Experiência 2 - Utilização de um LDR e entradas analógicas

A seguir, encontrará a experiência “Utilização de um LDR e entradas analógicas” para o microbit BBC. Incluímos toda a experiência como um exemplo onde aprende a construir o circuito e a programar o micro:bit para controlar o circuito.



Esta experiência requer:

1 BBC micro:bit

1 LDR miniatura.

1 Resistência de 10kΩ.

3 Fios de Jumper M/F. (se usar placa de ensaio)

3 Cabos de crocodilo. (não usando placa de ensaio)

Os objetivos desta experiência são:

Utilizar uma resistência dependente da luz como sensor.

Efectuar uma leitura analógica da Resistência Dependente da Luz através do pino de entrada P0.

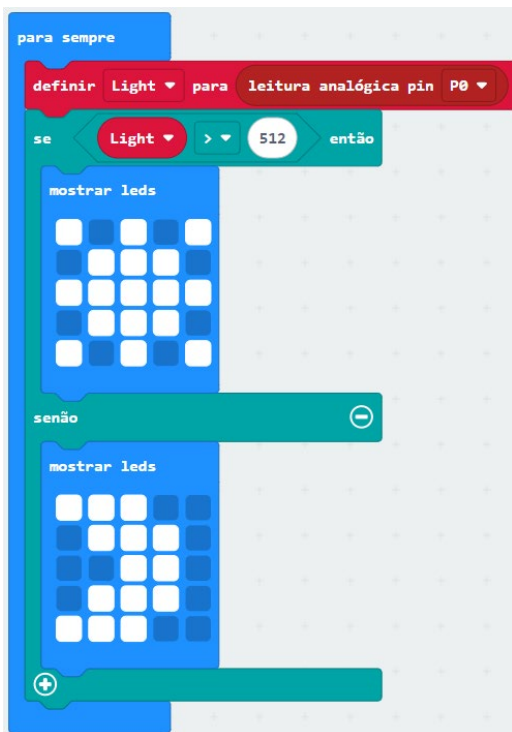
Definir um limiar de luz para decidir se é apresentado um Sol ou uma Lua na matriz de LED.

Objetivo a atingir

Um LDR (Light Dependent Resistor) é um componente elétrico com propriedades únicas. Como o nome sugere, é um tipo de resistência que tem a sua resistência determinada pela quantidade de luz que incide sobre ele. Quanto mais brilhante for a luz, menor será a sua resistência. Estas resistências são utilizadas juntamente com resistências normais para formar divisores de potencial. Quando utilizadas nesta configuração, produzem uma tensão que muda consoante o nível de luz. Um microcontrolador como o BBC micro:bit pode ler esta tensão (analógica), permitindo que um programa reaja a diferentes níveis de luz. Esta experiência vai explicar como usar o LDR e fazer uma leitura analógica.

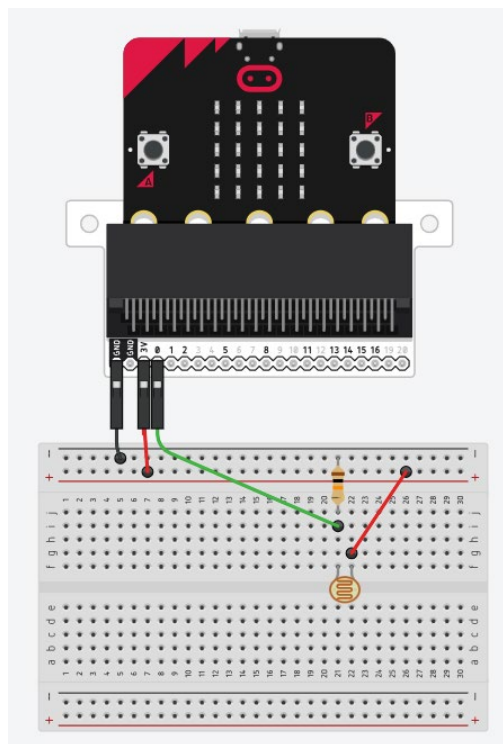
Criar o código usando o editor Microsoft MakeCode

Segue-se um exemplo de como escrever o código para a experiência no Microsoft MakeCode Editor (<https://makecode.microbit.org/>)



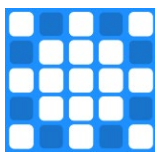
Construir este circuito na placa de ensaios

É possível construir este circuito usando a placa de ensaios e os componentes listados acima, pode seguir o diagrama abaixo.

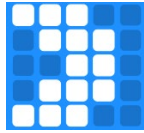


O que vai acontecer:

O BBC micro:bit deve mostrar uma imagem do Sol. Se tapar o LDR com o dedo, deverá mostrar uma imagem da Lua. Quando o LDR está à luz, o BBC micro:bit deve mostrar a imagem do "Sol".



Quando o LDR está no escuro (tapando-o), o BBC micro:bit deve mostrar a imagem da "Lua".



Construir este circuito com fios de crocodilo:

Se não tiveres o Kit para o BBC micro:bit ou placa de ensaios, pode construir este circuito usando o diagrama de circuito como referência e os cabos de crocodilo para unir os componentes. Pode ver um exemplo disso na imagem abaixo.

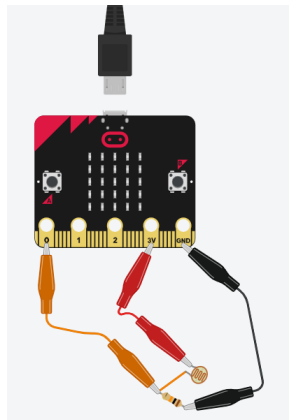
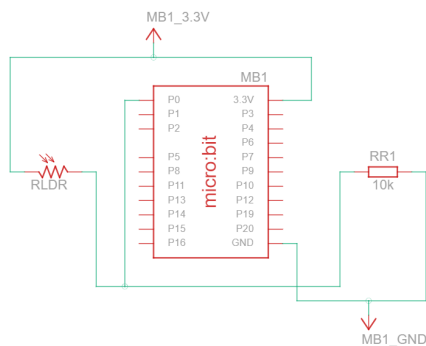


Diagrama de circuito:



O que é que se passa (Como é que funciona?):

A resistência R1 e o LDR estão em série, pelo que dividem os 3V. A proporção de 3V em cada uma delas depende da sua resistência relativa. Se ambas tiverem a mesma resistência, a tensão no meio será metade. Se a resistência representar 75% da resistência total, então a tensão no meio será 75% dos 3V e assim por diante. À medida que mais luz incide sobre o LDR, a sua resistência diminui, o que significa que a resistência R1 representa uma maior percentagem da resistência total. Isto aumenta a tensão em P0. A função de leitura analógica mede a tensão no pino selecionado e converte-a num valor entre 0 e 1023. Se a tensão em P0 for 0V, então este valor será 0. Depois, se a tensão neste pino for o máximo de 3V, então o valor será 1023. Se a tensão em P0 for 1,5V, então o valor será aproximadamente 512. O pino P0 é lido e o valor é armazenado na variável "luz". A instrução 'if' verifica então se o valor armazenado em 'light' é superior a 512. Se for, os LEDs com padrão de um Sol são apresentados na matriz de LEDs; caso contrário, é apresentado o padrão de uma Lua.

3. Experiência 3 - Diminuir a intensidade de um LED usando um potenciômetro

Introdução

Neste circuito, aprenderá a utilizar um potenciômetro para controlar o brilho de um LED, lendo um sensor e armazenando o seu valor entre 0 e 1023 como uma variável, utilizando-o depois como nível de brilho para o LED.

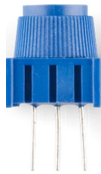
Esta experiência requer:

- 1 micro:bit
- 1 Cabo USB Micro B
- 1 Breakout micro:bit
- 1 Placa de ensaios
- 8 Fios de Jumper
- 1 Potenciômetro de 10k Ω
- 1 LED
- 1 Resistência de 100 Ω

Apresentação do potenciômetro

Um potenciômetro é um sensor analógico baseado em resistência que altera a sua resistência interna com base na rotação do seu botão. O potenciômetro

tem um divisor de tensão interno que lhe permite ler a alteração da tensão no pino central com um microcontrolador (i.e. micro:bit).

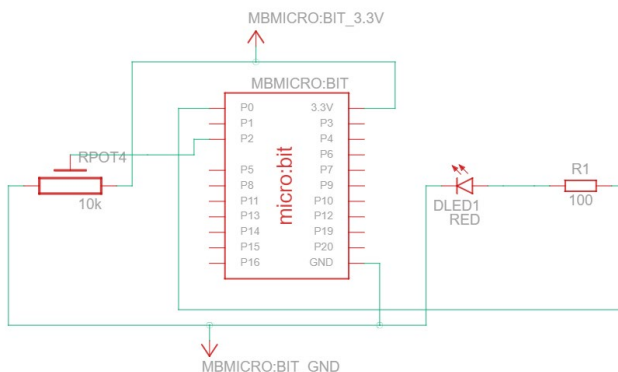


Para ligar o potenciômetro, ligue os dois pinos exteriores a uma tensão de alimentação (3,3 V neste circuito) e a terra. Não importa qual está ligado onde, desde que um esteja ligado à alimentação e o outro à terra. O pino central é ligado a um pino de entrada analógica para que o micro:bit possa medir a mudança de tensão. Quando rodar o botão, a leitura do sensor vai mudar!

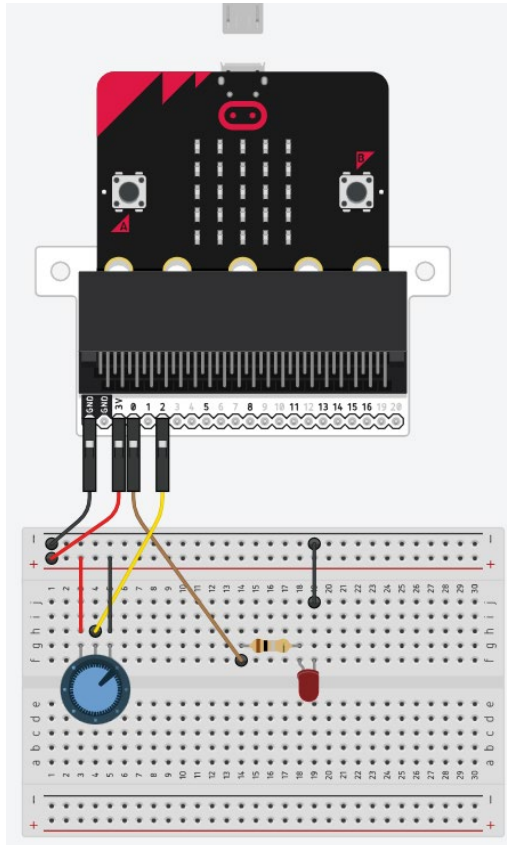
Ligação de hardware

Pronto para começar a ligar tudo? Consulte o esquema elétrico e as ligações na placa de ensaio abaixo para ver como tudo está ligado.

Esquema de ligação da experiência



Esquema do circuito



Exemplo de montagem placa de ensaio

Código a observar

Uma "variável" é um espaço reservado para valores que podem mudar no seu código. Pode criar uma variável utilizando a opção Criar variável no grupo Variáveis. Pode então dar-lhe um nome, o que cria um bloco para a sua variável.

≡ Variáveis

Para armazenar um valor dentro da sua variável recém-criada, utiliza o bloco definir para. O bloco "definir para" permite-lhe selecionar a partir de uma lista de variáveis existentes no seu programa e, em seguida, adicionar um valor que pretende armazenar ou definir para essas variáveis.



Vejamos os blocos de código desta experiência:



Leitura analógica

Neste programa está a ler a tensão do potenciômetro que é de 0 a 3,3 volts. O micro:bit lê esse valor como um número de 10 bits que é um intervalo de

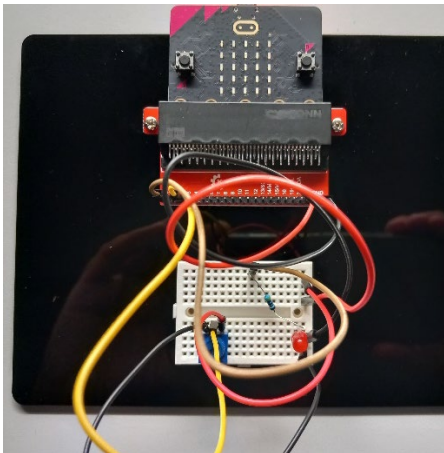
valores de 0 a 1023 usando o bloco de leitura analógica. O bloco de leitura analógica é um bloco baseado em valores, o que significa que tens de o inserir num bloco com uma forma correspondente. Inserimo-lo no bloco set to para armazenar o seu valor como uma variável.

Escrita analógica

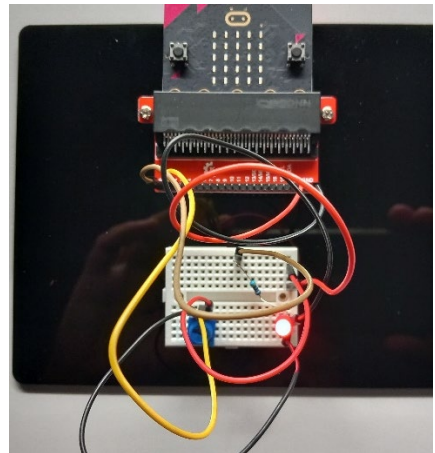
Tal como o bloco de leitura analógica, o bloco de escrita analógica lida com uma gama de valores, mas em vez de ler um pino como entrada, o bloco de escrita analógica produz um valor analógico para um pino. Vemos isto como uma gama de brilho com este led, mas poderia ser um tom de uma campainha, a velocidade de um motor, etc. Definimos a nossa saída analógica para a variável em que guardámos o valor do potenciómetro.

O que deve ver

Deve rodar o potenciómetro. Verá que o LED fica mais claro ou mais escuro consoante a posição do potenciómetro. Se rodar o potenciómetro até ao fim numa direção, ele ficará totalmente ligado e a outra extremidade ficará totalmente desligada.



Potenciómetro na posição inicial :
LED sem brilho



Potenciómetro rodado: LED com
brilho máximo

Resolução de problemas

Funcionamento esporádico

É muito provável que isto se deva a uma ligação instável com os pinos do potenciómetro. Normalmente, isto pode ser resolvido segurando o potenciómetro ou movendo o circuito do potenciómetro para outro local, na sua placa de ensaio.

Não está a funcionar

Certifique-se de que não ligou acidentalmente o pino central, o elemento resistivo do potenciómetro a um pino errado!

O LED não está a acender

Os LEDs só funcionam numa direção. Verifique novamente as suas ligações.

4. Experiência 4 - Utilização de um transistor para ligar um motor

Esta experiência requer:

- 1 BBC micro:bit
- 1 Transistor NPN
- 1 Resistência de 2.2KΩ
- 1 Conector de terminal
- 1 Mini motor DC
- 3 Fios de Jumper M/F
- 1 Ventoinha

Os objectivos desta experiência são:

- Utilizar um transistor para acionar um motor com ventoinha.
- Controlar a velocidade do motor utilizando a modulação por largura de impulsos (PWM).

Introdução

Os pinos de saída da maioria dos microprocessadores só podem fornecer uma pequena quantidade de corrente, não suficiente para um dispositivo que consome muita energia, como um motor. O BBC micro:bit não é exceção a isto. Um transistor pode ser usado para resolver este problema. Um transistor é como um portão para a eletricidade, uma pequena quantidade de corrente

pode ser usada para abrir o portão e deixar passar muita corrente para os componentes que necessitam de muita energia.

Criar o código usando o editor de blocos

De seguida mostra-se um exemplo de como programar a experiência no Editor de Blocos (<https://makecode.microbit.org>)

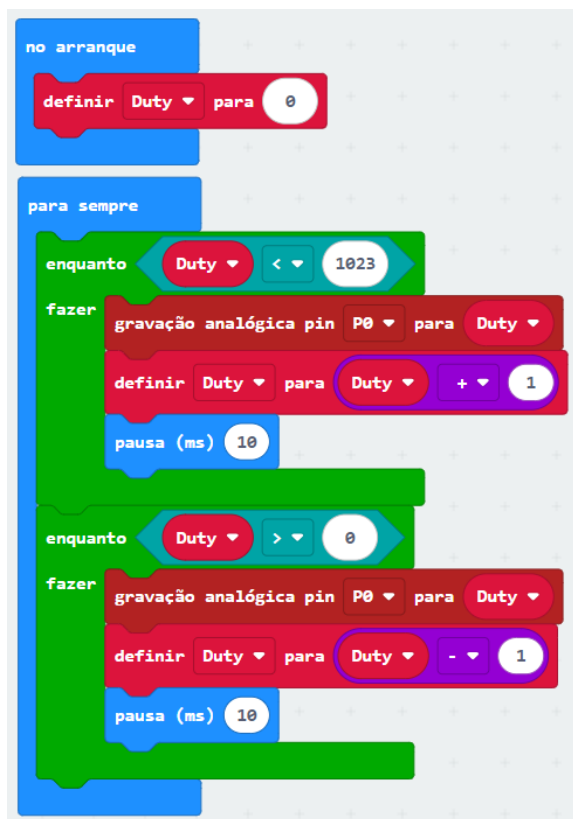
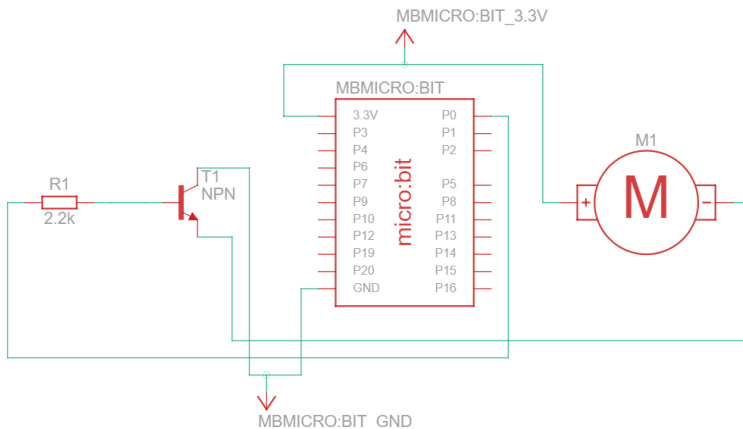
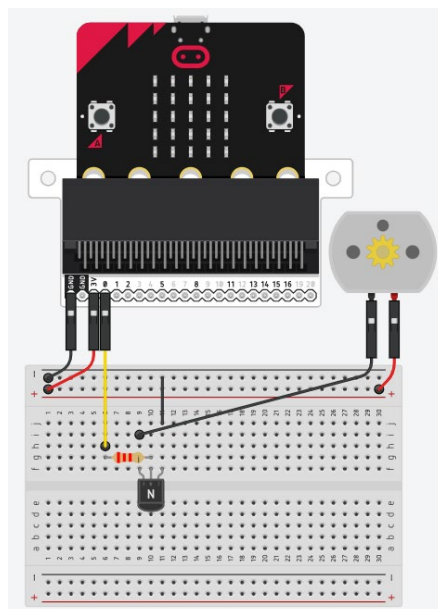


Diagrama do circuito:

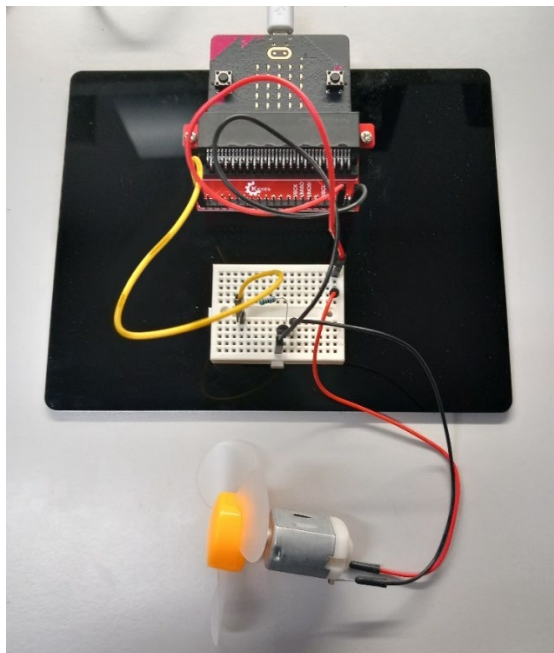


Construir este circuito na placa de ensaio:

É possível construir este circuito usando a placa de ensaio para o BBC micro:bit e os componentes listados acima, pode seguir o diagrama abaixo.



Exemplo de montagem do circuito na placa de ensaio :



O que vai acontecer:

O motor deve, após alguns segundos, começar a rodar lentamente, e depois mais depressa até atingir a velocidade máxima. Nesta altura, abrandará e parará e o ciclo recomeçará.

O que se está a passar (como funciona)?

Se o transístor for ligado e desligado muito rapidamente e passar metade do tempo ligado e metade do tempo desligado, o motor da ventoinha girará a meia velocidade. Ao alterar a percentagem do tempo que o transístor passa ligado (conhecido como ciclo de funcionamento), a velocidade do motor pode ser controlada com precisão. Como os impulsos de energia são ligados e

desligados e a largura dos impulsos é controlada, chamamos a este processo Modulação por Largura de Impulso (PWM). O código funciona em duas fases. O primeiro loop escreve o valor da taxa no Pino P0, depois aumenta o valor da taxa em um e escreve o valor em P0 novamente. Isto repete-se até o valor atingir o máximo de 1023 (velocidade máxima). O segundo ciclo entra então em ação e reduz a taxa em 1 e escreve-a em P0 até o valor atingir 0 (parado). Todo este ciclo está dentro de um loop infinito, pelo que o motor irá acelerar e abrandar para sempre. A saída PWM varia o ciclo de trabalho da tensão de saída, como se mostra abaixo, para variar a velocidade do motor da ventoinha.

5. Experiência 5 - Utilização do acelerómetro para controlar a velocidade do motor

Introdução

Nesta experiência, vai ver como combinar a utilização do acelerómetro no micro:bit para medir a sua orientação e utilizá-lo para controlar o motor.

Componentes necessários

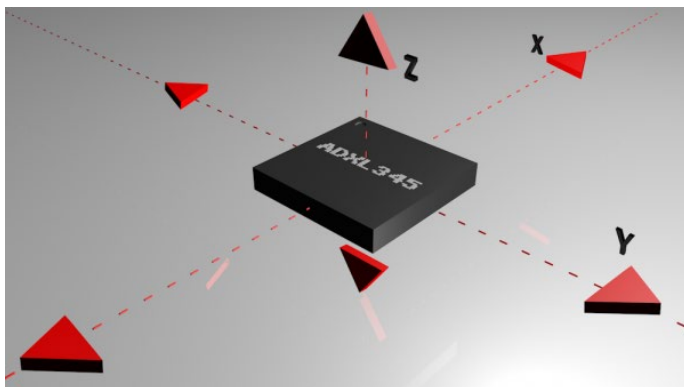
- 1 micro:bit
- 1 Cabo USB Micro B
- 1 Breakout micro:bit
- 1 Placa de ensaios
- 5 Fios de Jumper
- 1 motor com ventoinha

Os objectivos desta experiência são:

- Utilizar o acelerómetro do micro :bit para controlar a velocidade do motor

Apresentação do Acelerómetro

Um acelerómetro é um sensor que mede as forças gravitacionais que o puxam nas três dimensões dos eixos X, Y e Z do chip.

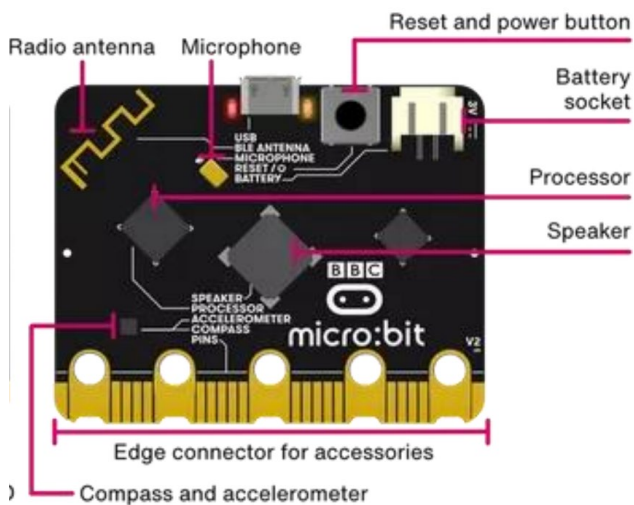


Visualização de um acelerómetro comum (ADXL345) com três eixos

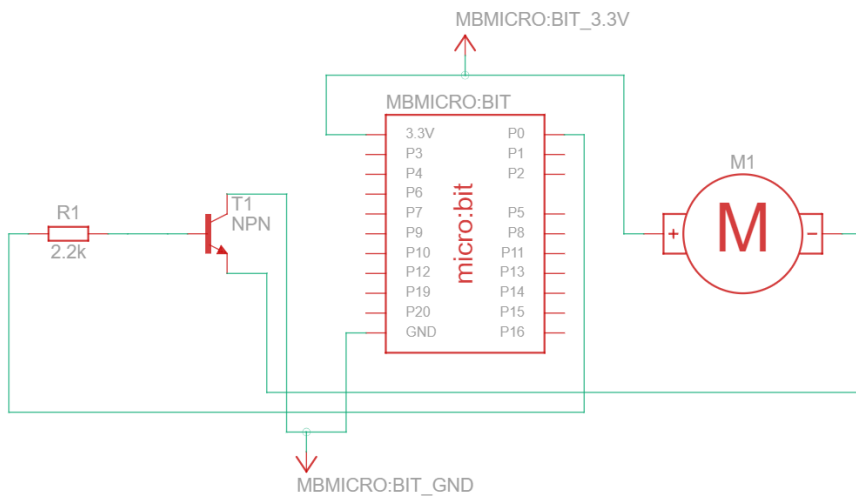
O BBC micro:bit tem um acelerómetro incorporado. O acelerómetro deteta se o micro:bit está a acelerar ou a abrandar em 3 direcções, identificadas abaixo como X, Y e Z. Cada eixo pode medir entre -4096 e 4096, o que corresponde a -4 e 4Gs. Um G é uma aceleração de 9,81 metros por segundo. Esta é a aceleração que os objetos sob a influência da gravidade da Terra estão sujeitos.

Um acelerómetro não só pode medir as forças brutas que puxam o chip e o objeto sobre o qual o chip está assente, como também pode detetar passos, abanões e outros movimentos que tenham um padrão específico. Além disso, é possível utilizar um acelerómetro para detetar simplesmente a orientação em que o dispositivo se encontra. Alguma vez se perguntou como é que o seu telemóvel sabe quando o vira da vertical para a horizontal? É tudo graças ao acelerómetro do telemóvel!

O acelerómetro é um componente que não se encontra no saco de peças do kit. Porquê? Porque está no próprio micro:bit! Na parte de trás do micro:bit pode ver uma série de pequenos chips. Um deles é o acelerómetro. O micro:bit tem um acelerómetro integrado que mede a força gravitacional. Dependendo da versão que tiver, o acelerómetro e a bússola podem estar em circuitos integrados ou combinados num único circuito integrado.

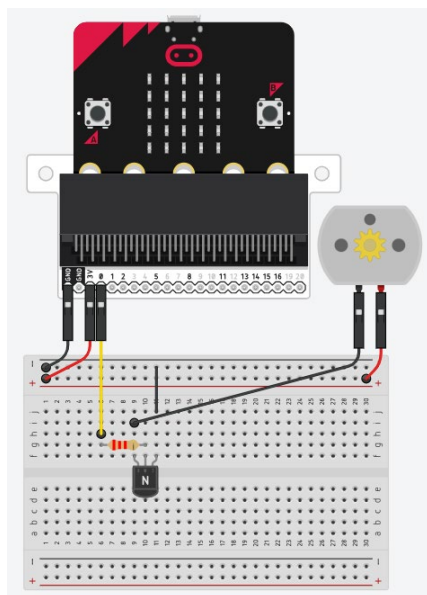


Exemplo de diagrama do circuito:

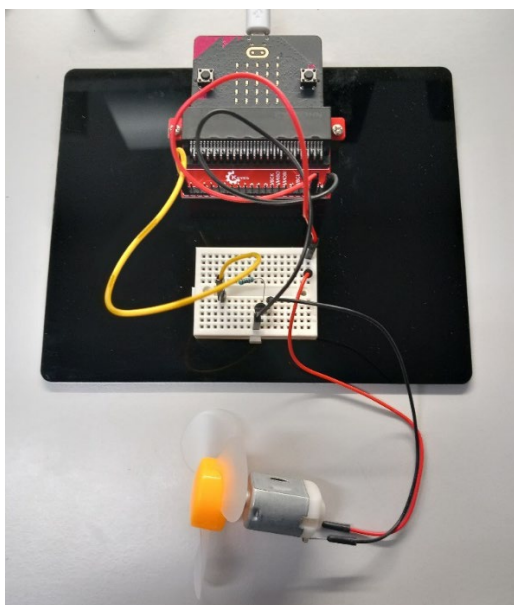


Construir este circuito na placa de ensaio:

É possível construir este circuito usando uma placa de ensaio para o BBC micro:bit e os componentes listados acima. Pode seguir o diagrama abaixo:

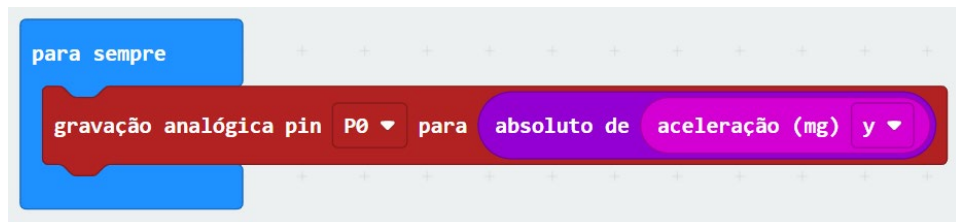


Exemplo de montagem do circuito na placa de ensaio



Criar o código usando o editor de blocos

De seguida mostra-se um exemplo de como programar a experiência no Editor de Blocos (<https://makecode.microbit.org>)



O que vai acontecer:

O motor deverá - segundo a orientação do acelerómetro do micro:bit - começar a rodar ou abrandar consoante a orientação do movimento efetuado na placa micro:bit.

6. Experiência 6 - Fazer soar uma campainha piezoelétrica

Introdução

Nesta experiência, vamos fazer a ponte entre o mundo digital e o mundo analógico. Vamos utilizar um sinal sonoro piezoelétrico que faz um pequeno "clique" quando lhe aplicamos tensão. Por si só, isso não é muito excitante, mas se ligarmos e desligarmos a tensão centenas de vezes por segundo, a campainha piezoelétrica produzirá um tom. E se juntar uma série de tons, temos música! Este circuito e este conjunto de blocos de código vão criar uma máquina de som simples.

Nota: O micro:bit v2 revisto inclui também um buzzer incorporado. Iremos abordar a forma de ligar um buzzer externo para os utilizadores que pretendem ligar o componente distante da placa.

Componentes necessários

Necessitará das seguintes peças:

- 1 micro:bit
- 1 Cabo USB Micro B
- 1 Breakout micro:bit
- 1 Placa de ensaio
- 8 Fios de Jumper
- 1 Buzzer Piezo
- 2 Botões de pressão
- 2 Resistências de 10k Ω

Os objectivos desta experiência são:

- Fazer soar uma campainha piezoelétrica

Apresentação da campainha Piezo

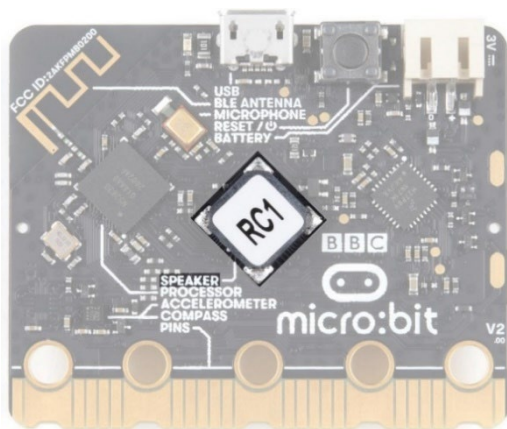
O buzzer é um pequeno componente com um pedaço de metal que se move quando se aplica uma tensão através dele. Este movimento provoca um pequeno som.



Ao ligar e desligar a tensão com alguma rapidez, obtemos diferentes bips e buzzes. Vamos usar a modulação PWM para controlar a velocidade de ligar e desligar o piezoelétrico e, por sua vez, a frequência de áudio que sai da campainha. Ajustar o PWM permite-lhe obter notas musicais a partir da campainha.

Na campainha, verá assinalado um pino que tem um (+) na etiqueta. Esse pino é ligado ao sinal do pino P0. O outro pino deve estar ligado à terra.

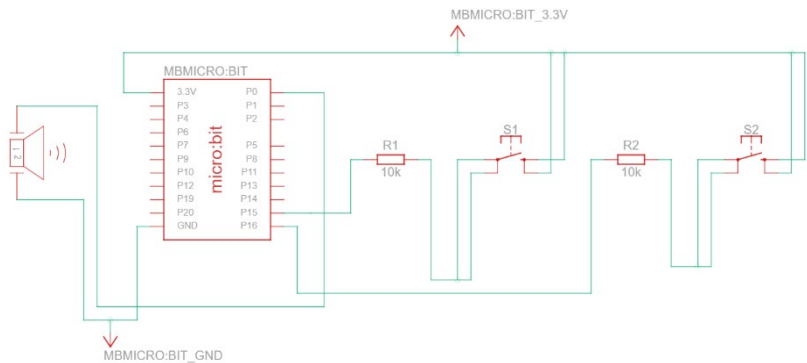
Na versão 2 do micro:bit, existe um sinal sonoro piezoelétrico incorporado na zona central da parte traseira.



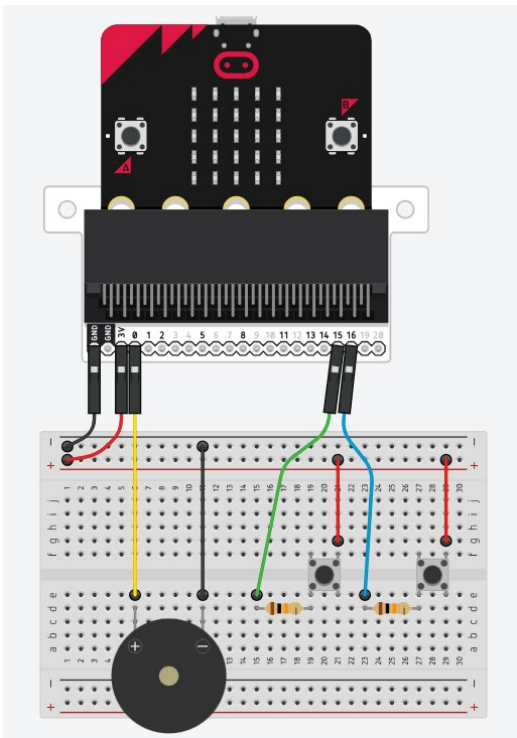
Agora vamos concentrar-nos na ligação de um buzzer piezoelétrico externo.

Pronto para começar a ligar tudo? Consulte o exemplo do esquema elétrico abaixo para conferir como tudo está ligado.

Exemplo de esquema de ligação da experiência



Exemplo de montagem em placa de ensaios



Nota : As calhas de alimentação das placas de ensaio de tamanho normal têm uma quebra a meio. Se acabar por utilizar a metade inferior da calha de alimentação, terá de saltar entre a extremidade superior e a extremidade inferior.

Criar o código usando o editor de blocos

Abrindo a aplicação MakeCode (<https://makecode.microbit.org/>), poderá experimentar a seguinte estrutura de blocos no seu editor. Vejamos os blocos de código desta experiência:



Habilitar altifalante incorporado

O bloco “definir bloco embutido” define a saída para o altifalante interno incorporado no micro:bit. Por defeito, esta saída é feita para o altifalante

integrado no micro:bit v2. Se este bloco não estiver definido, o micro:bit V2 fará a saída para este altifalante integrado em vez do pino 0. Se o definir para OFF, o sinal será enviado para um sinal sonoro piezoelétrico externo no pino 0.

Colocar o “definir o altifalante embutido” na posição Ligado emitirá som para o altifalante incorporado.

Iniciar Repetição de Melodia

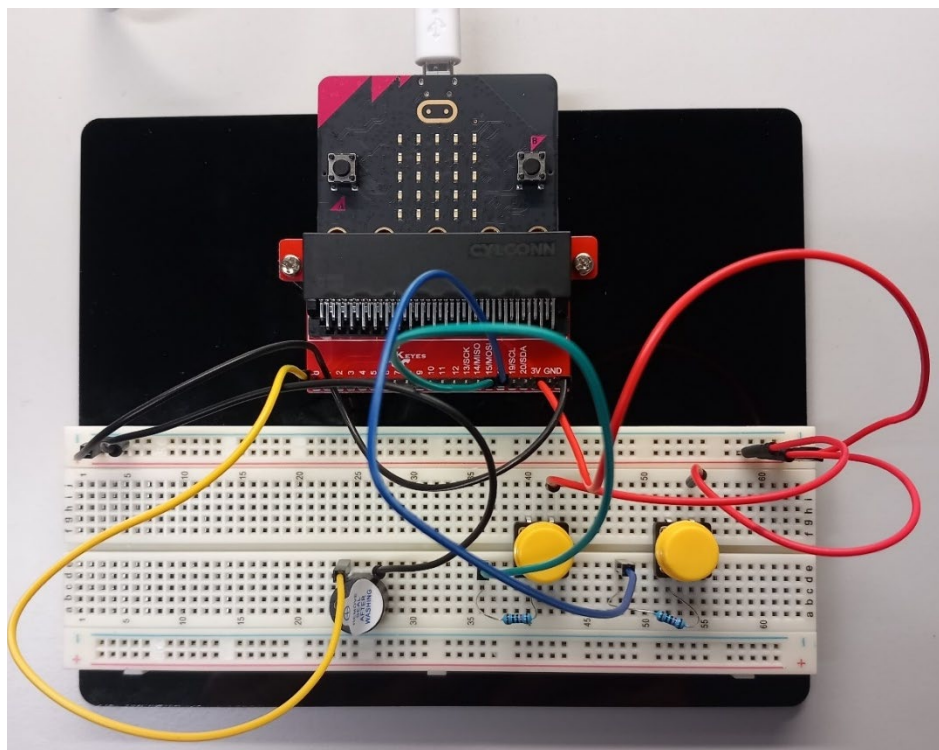
No bloco “para sempre” poderá selecionar uma de várias canções que estão pré-programadas no MakeCode e quantas vezes quer que ela se repita. Note que quando uma melodia está a tocar nenhum outro código pode ser executado, isto chama-se código de "bloqueio" e tem de ser tido em conta no teu programa.

Bloco “Tocar o tom para”

O bloco "Tocar tom para" aceita uma nota que gostaria que a campainha produzisse e o período de tempo, em batidas por segundo, que gostarias que tocasse.

O que deve ouvir:

Deve ouvir uma música que começa assim que o programa iniciar a correr no micro:bit! Quando essa música terminar, pode premir um dos dois botões, e outra música começará. Cada botão tem a sua própria música, e há outra música se premir os dois botões ao mesmo tempo.



Ajustar volume

Caso pretenda, poderá ligar ainda um potenciómetro entre a campainha e o pino 0 (Jumper amarelo). Com a adição de um potenciómetro ajustará o volume.

Resolução de problemas

Sem som

Dado o tamanho e a forma do buzzer piezoelétrico, é fácil não encontrar os orifícios certos na placa de ensaio. Tente verificar novamente a sua colocação.

Verifique também se o botão de pressão está ligado corretamente. Se estiver mal ligado, o circuito nunca será completado, mesmo que carregue ou não no botão.

7. Experiência 7 - Iniciar uma experiência de Energia eólica

Esta experiência requer os seguintes componentes:

- 1 BBC micro:bit
- 1 Conector de terminal (*edge connector breakout*)
- 1 Mini motor DC
- 2 Resistência de $2.2k\Omega$
- 1 Ventoinha
- 2 Fios de ligação M/F
- 1 Fios do jumper M/M

Os objectivos desta experiência são:

- Iniciar uma experiência de Energia eólica
- Gerar uma tensão soprando numa pá de ventoinha para fazer girar um motor.
- Visualizar esta tensão usando um pino de entrada analógica no micro:bit

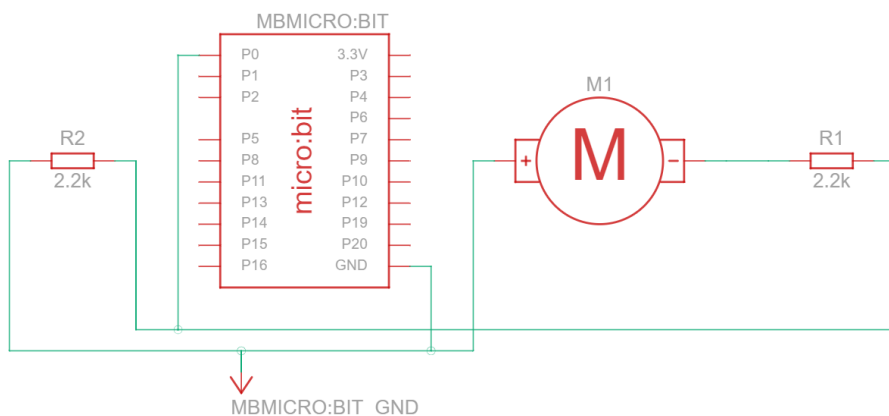
Introdução

As turbinas eólicas convertem a energia cinética do ar em energia eléctrica. Nesta experiência, o motor e a ventoinha são utilizados em sentido inverso para gerar uma tensão que pode ser medida e apresentada pelo micro:bit.

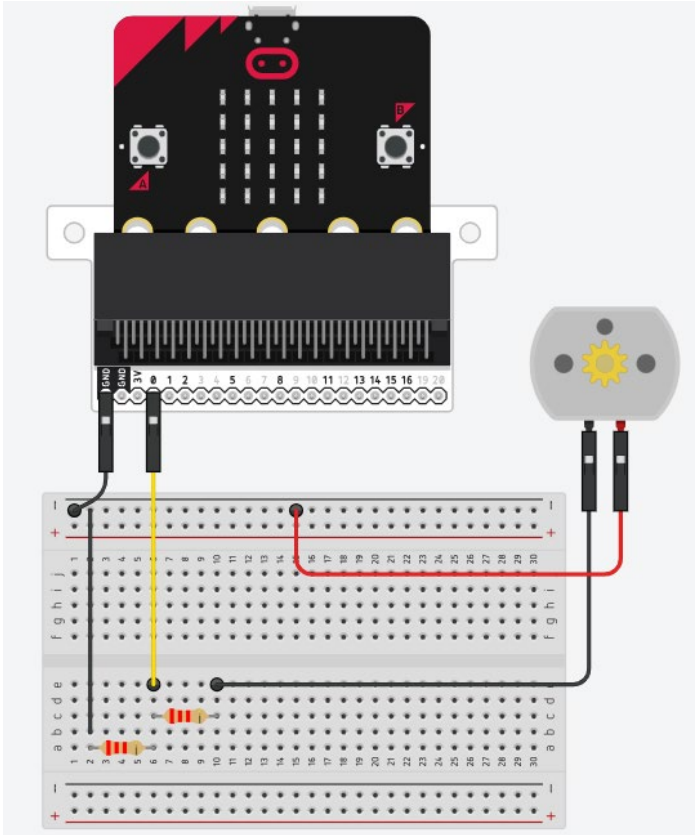
Construindo este circuito na placa de ensaios:

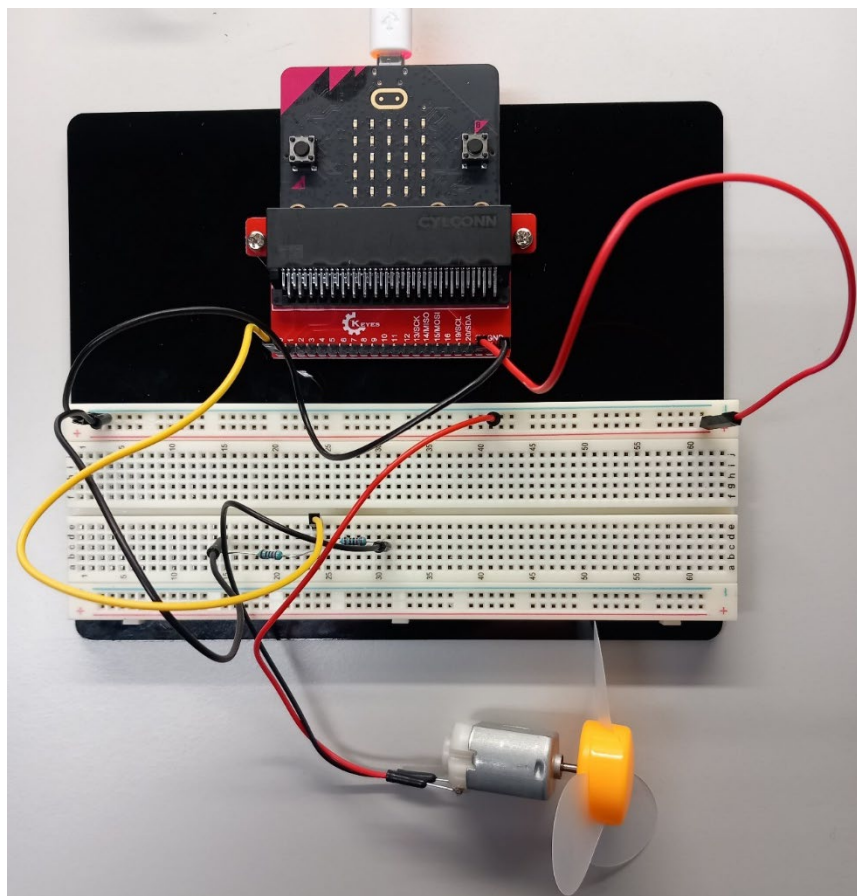
É possível construir este circuito usando a placa de ensaios para o micro:bit e os componentes listados acima.

Exemplo do diagrama de circuito:



Exemplo de montagem em placa de ensaios





Código JavaScript para o editor MakeCode:

Segue-se um exemplo de como escrever o código para a experiência em linguagem JavaScript.

```

1  let maior = 0
2  let valor = 0
3  input.onButtonPressed(Button.A, function () {
4      basic.showNumber(maior)
5  })
6  basic.forever(function () {
7      valor = pins.analogReadPin(AnalogPin.P0)
8      if (valor > maior) {
9          maior = valor
10     }
11 })

```

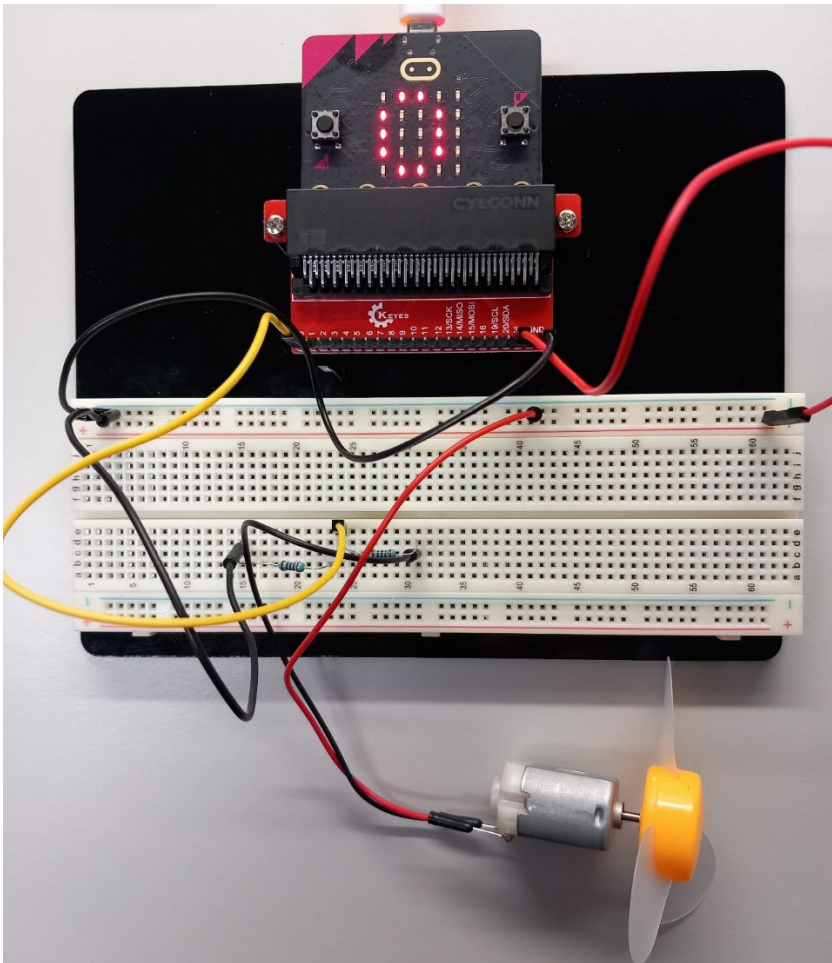
Ou como alternativa podemos usar a programação equivalente em blocos :



O que vai acontecer ?

Assim que o código estiver carregado no micro:bit, prima o botão A e deverá aparecer um número baixo no intervalo de 1 a 5. Esta é a voltagem lida a partir

de P0. A razão pela qual não mostra o zero nesta configuração é que P0 é muito sensível, pelo que é provável que apanhe uma pequena tensão.



Tente soprar suavemente na ventoinha até que esta comece a rodar e, em seguida, prima novamente o botão A. Desta vez, deve apresentar uma leitura entre 200 e 300. Finalmente, tente respirar fundo e soprar com força na ventoinha, o que deverá dar uma leitura próxima de 1000!

O que é que se passa (como é que funciona)?

Quando a corrente eléctrica passa por um fio dentro do campo magnético, produz um binário (força de rotação) que faz girar um motor. O oposto também é verdadeiro: se for aplicado um binário a um motor, é gerada uma corrente. O binário é fornecido pelo sopro da ventoinha. Embora a corrente gerada pelo sopro da ventoinha seja muito pequena, a tensão pode atingir alguns volts, R1 e R2 formam um divisor de potencial que reduz a tensão do motor para metade, mantendo-a maioritariamente num intervalo mensurável. Esta tensão reduzida a metade é lida pelo micro:bit e convertida num número entre 0 e 1023.

Este valor é então comparado com o valor atualmente armazenado na variável 'maior' do código. Se este valor for superior ao valor atualmente armazenado, substitui o valor antigo.

Quando o botão A é premido, o valor mais alto registado é apresentado no ecrã.

Se reiniciar o micro:bit usando o seu botão de reset, este valor será apagado.

8. Experiência 8 - Fazer um jogo usando uma bússola

Introdução

Já alguma vez utilizou uma bússola? Está um pouco confuso quanto à direção a seguir? Vamos construir uma bússola digital que o manterá no caminho certo para o Pólo Norte usando o chip de bússola integrado do micro:bit!

Componentes necessários

- 1 micro:bit
- 1 Cabo USB Micro B
- 1 Conetor breakout do micro:bit
- 1 Placa de ensaios
- 4 Fios de Jumper
- 2 LEDs
- 2 Resistências de 100Ω

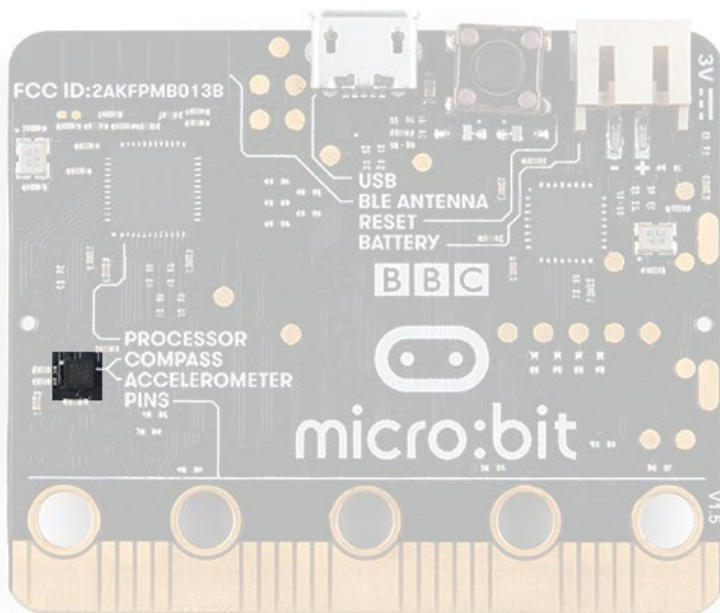
Os objetivos desta experiência são:

- Fazer um jogo usando uma bússola
- Conhecer mais sobre campo magnético

Introdução à Bússola (Magnetómetro)

Numa experiência anterior, aprendeu sobre o acelerómetro, que mede a gravidade. A bússola, ou tecnicamente o magnetómetro, mede um campo

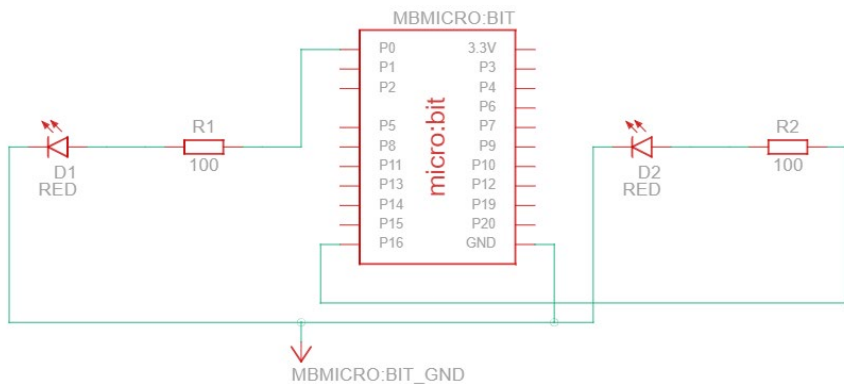
magnético. Os campos magnéticos existem em diferentes tamanhos, mas o maior é o produzido pela própria Terra, e é por isso que as bússolas funcionam. O micro:bit tem uma bússola integrada (também conhecida como magnetómetro) que pode detetar a sua orientação usando o campo magnético da Terra. Dependendo da versão que tiver, o acelerómetro e a bússola podem estar em circuitos integrados separados ou combinados num único circuito.



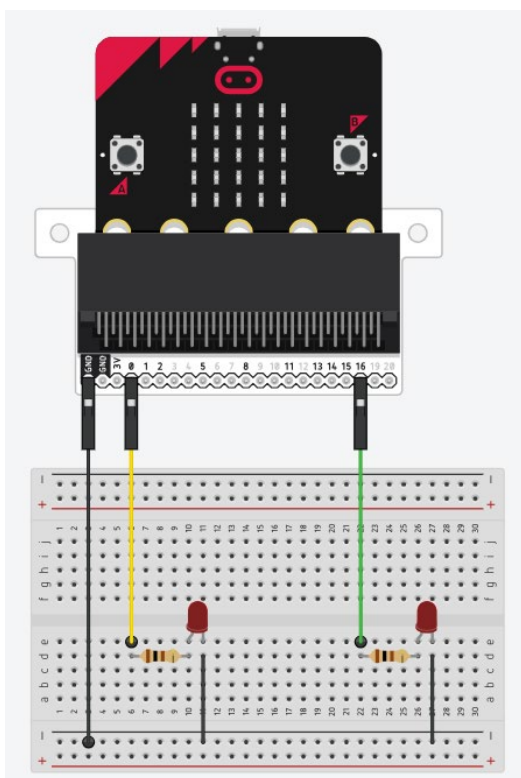
O magnetómetro deteta o norte magnético e representa o seu rumo em graus, sendo o Norte 0 graus, o leste 90 graus, o Sul 180 graus e o Oeste 270 graus.

Ligação de hardware

Consulte o esquema elétrico abaixo para ver como tudo está ligado.

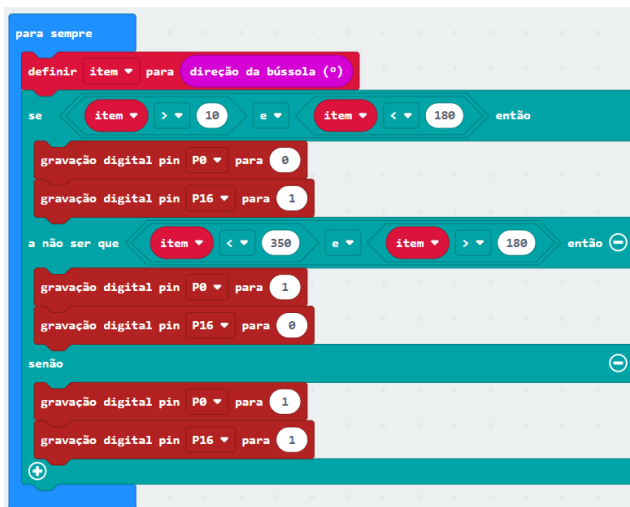


Esquema de ligação da experiência na placa de ensaios



Criar o código usando o editor de blocos

De seguida mostra-se um exemplo de como programar a experiência no Editor de Blocos (<https://makecode.microbit.org>)



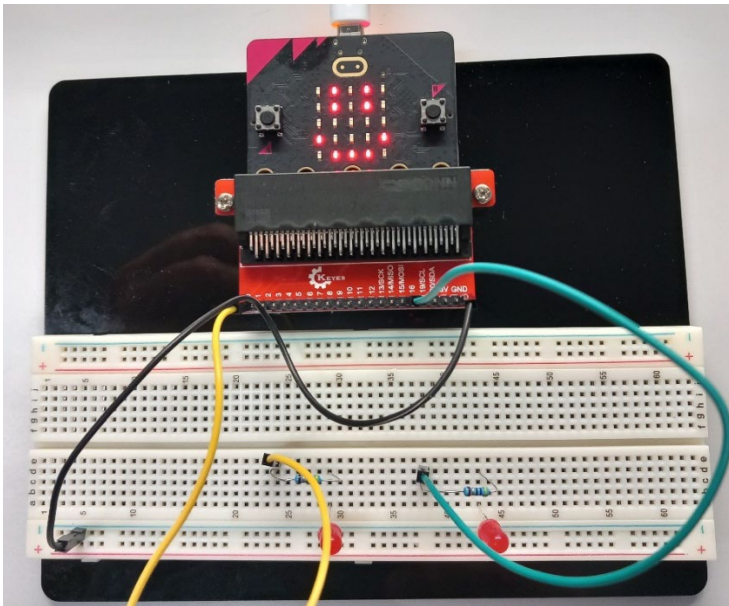
Direção da bússola

O bloco de “direção da bússola” retorna a direção se estiver a segurar o micro:bit numa posição plana com os pinos voltados para si. Zero graus é o Norte. Armazenamos essa direção numa variável chamada “item”.

O restante do código é direto, mas o bloco lógico “and” é usado. Isso combina duas declarações lógicas numa declaração que retorna “true” quando ambas as declarações são verdadeiras e somente verdadeiras.

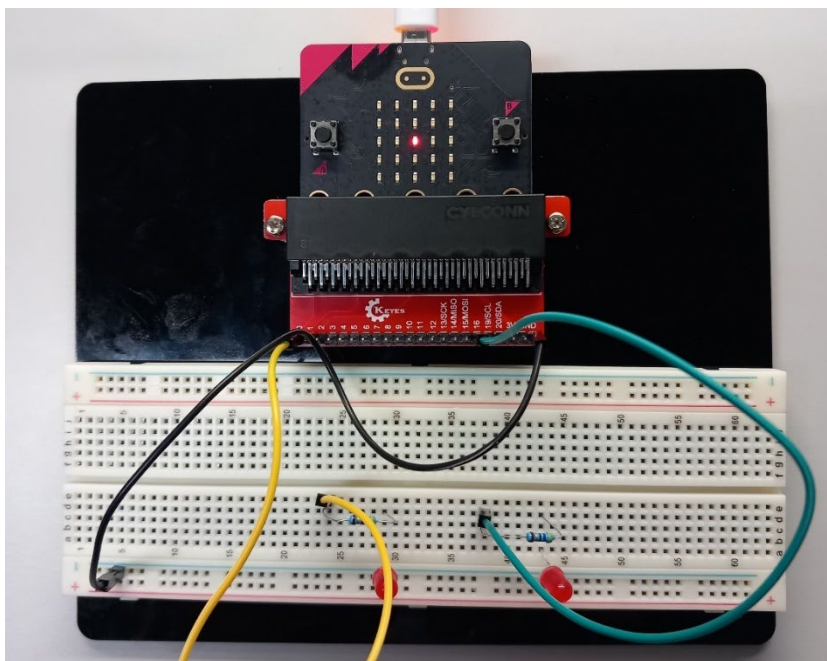
O que você deve ver?

Quando o código for carregado, verá primeiro as instruções na matriz de LEDs micro:bit. As instruções solicitarão que mova o micro:bit até que toda a parte do da matriz seja preenchida com leds ligados. Assim que isso acontece será exibido um rosto sorridente.



Este processo é para calibrar o magnetômetro do micro:bit. Os LEDs começarão a acender. Um ou outro, ou ambos.

Parado, gire na direção do LED que está aceso. Se ambos os LEDs estão ligados, ou um dos LEDs a piscar com a maior frequência, geralmente indica que está voltado para Norte! (Se tiver uma bússola que o ajude a encontrar o Norte ou qualquer outra direção pode alterar o código!)



Resolução de problemas

Direção da bússola

Se calibrar o micro:bit enquanto estiveres dentro de casa, perto do computador ou de algo que produza um campo magnético maior, o micro:bit vai perder o sentido de direção.

Não é realmente o Norte?

Pode ter um motor ou um íman por perto! O magnetómetro é sensível a todos os campos magnéticos, incluindo os que podem ser produzidos por outros componentes eletrónicos, metal ou um íman.

9. Experiência 9 - Criar uma experiência de carga de um condensador

Componentes necessários

Necessitará dos seguintes componentes:

- 1 micro:bit
- 1 Cabo USB Micro B
- 1 Breakout micro:bit
- 1 Placa de ensaios
- 9 Fios de Jumper
- 1 Potenciômetro de $10k\Omega$
- 1 Condensador $470\mu F$
- 2 Botões de pressão
- 4 LED (1 vermelho, 1 laranja, 1 amarelo, 1 verde)
- 4 Resistência de 100Ω
- 2 Resistências de $2.2K\Omega$

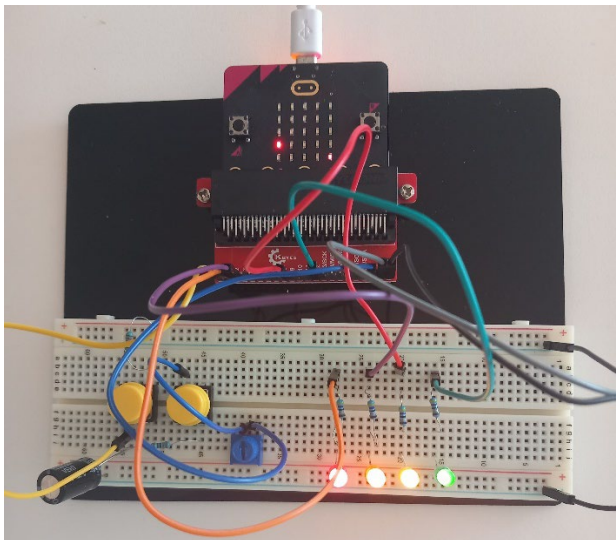
Introdução

Nesta experiência, vamos utilizar a entrada analógica P0 do microbit para medir a tensão no lado positivo de um condensador. Será criado um circuito que permite carregar o condensador através de uma resistência variável (potenciômetro). O micro:bit será programado para mostrar o nível de carga do condensador, tanto numericamente (na matriz do microbit) como

utilizando os LEDs externos para criar uma escala de carga com código de cores.

Os objetivos desta experiência são:

- Criar uma experiência de carga de um condensador
- Visualizar escala de carga do condensador nos LEDs externos
- Visualizar valor numérico da carga na matriz do micro:bit
- Montar um circuito de carga e descarga do condensador
- Programar micro:bit em modo blocos e linguagens de programação JavaScript e Python

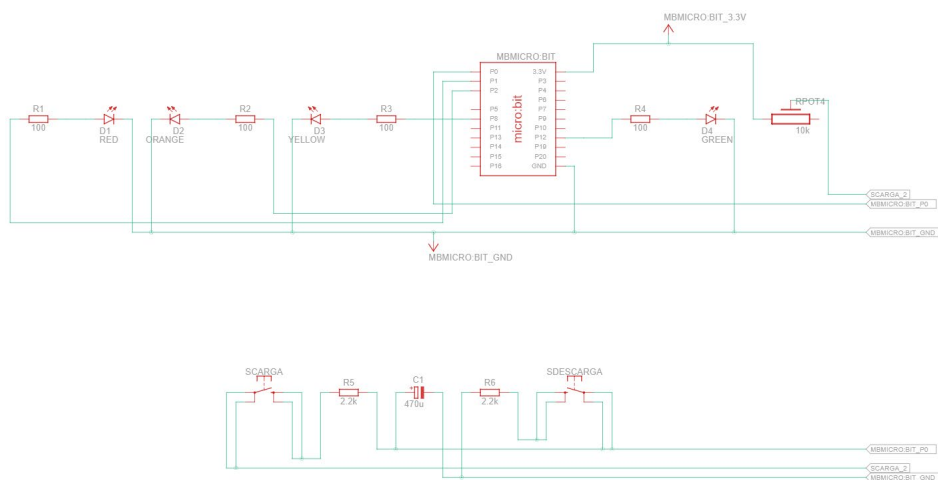


Ligação de hardware

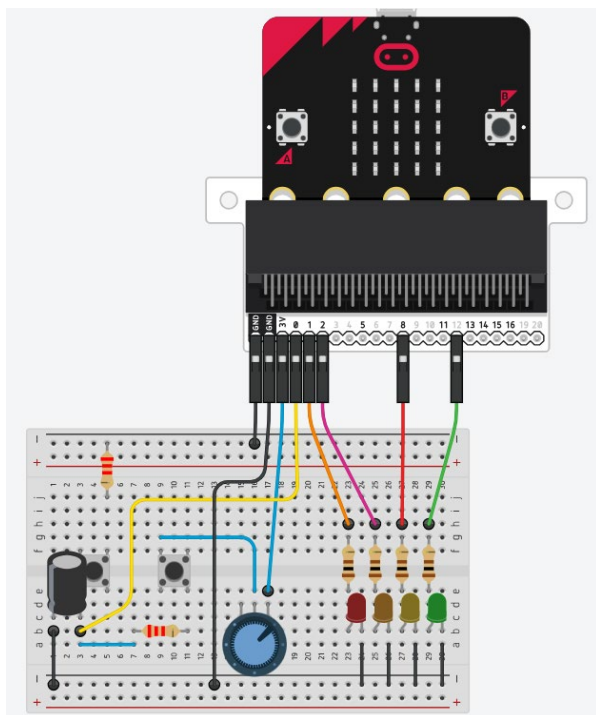
O circuito foi projetado para quando se clicar continuamente no botão da direita, a carga do condensador progredirá de 0 até 100% e a escala de LEDs vai acender sucessivamente até ficarem todos ligados.

Ao carregar no botão da esquerda efetuará o processo inverso e haverá uma descarga total do condensador, os quatro LEDs apagam e a matriz mostrará o valor zero.

Consulte o esquema elétrico abaixo para ver como tudo deverá ficar ligado.



Esquema de ligação da experiência na placa de ensaios



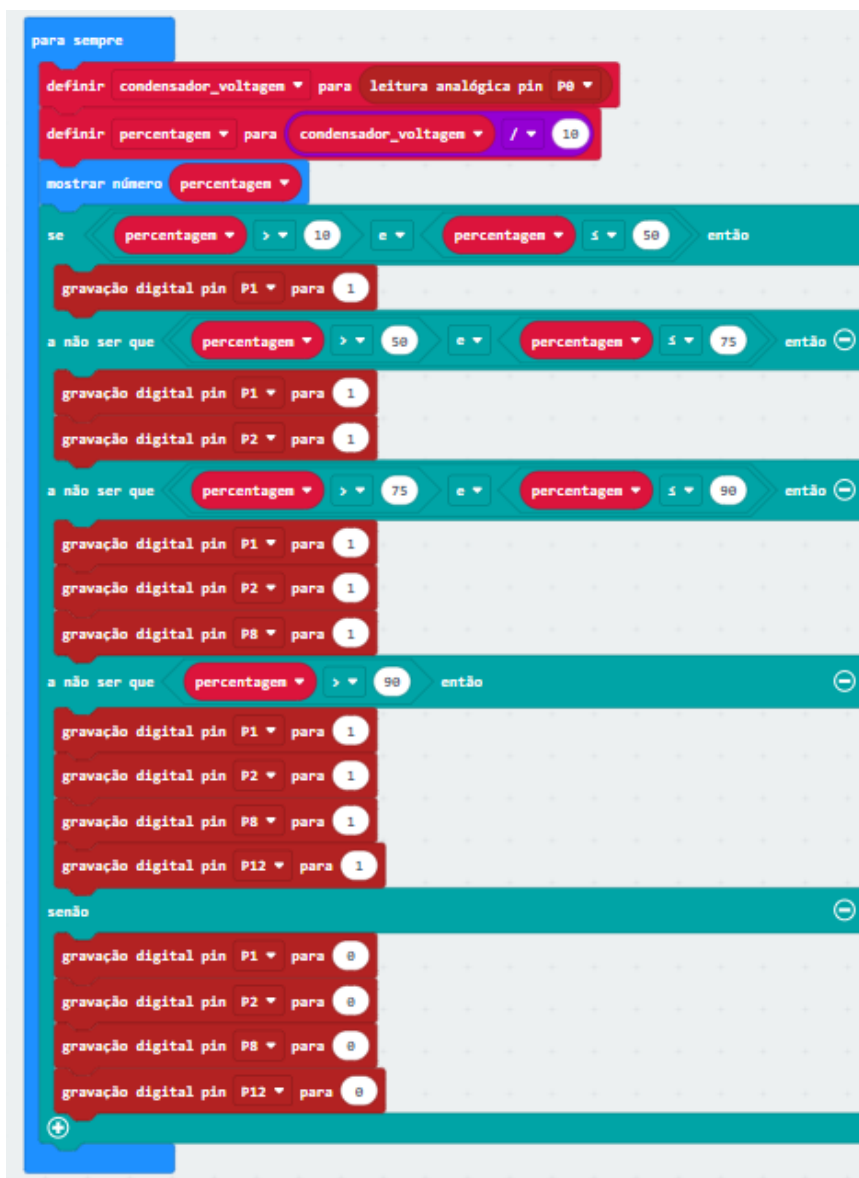
Criar o código usando o editor de blocos

De seguida mostra-se um exemplo de como programar a experiência no Editor de Blocos (<https://makecode.microbit.org>)

No código são criadas duas variáveis: `condensador_voltagem` (para ler a carga no condensador através da entrada analógica P0) e a variável `percentagem` (valor da percentagem de carga no condensador) que será mostrada na matriz do micro:bit.

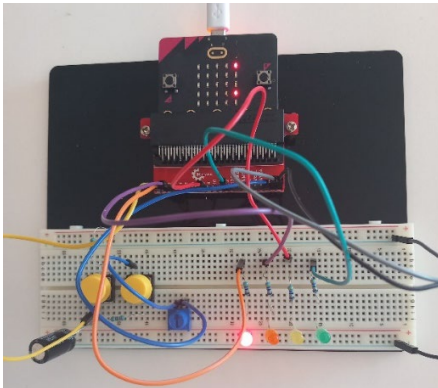
Conforme a percentagem de carga será enviado um sinal para os LEDs numa escala de 100% dividida por quatro etapas de 25%. O LED vermelho está ligado à saída P1 e assinala 25% de carga. No pino P2 encontra-se ligado o LED laranja (50%), no pino P8 o LED amarelo (75%) e finalmente no pino 12 o LED verde que assinala os 100% de carga.

A rotina do código estrutura a ligação aos quatro LEDs descritos anteriormente de forma cumulativa de acordo com os blocos mostrados abaixo ficam a saída a 1 (ligado) ou a 0 (desligado, no caso de clicar no botão de descarga que se encontra na parte esquerda do circuito).

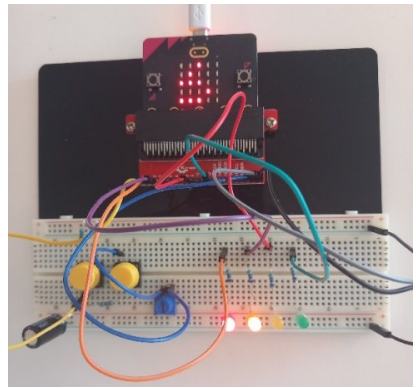


As imagens seguinte mostram a progressão de carga do condensador em etapas de 25% na escala assinalada pelo estado sucessivamente ligado de cada um dos quatro LEDs.

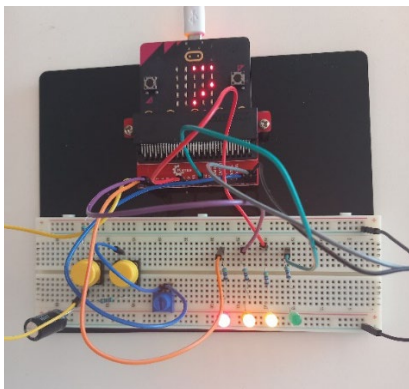
A matriz do micro:bit vai mostrando o correspondente valor numérico dessa progressão de carga.



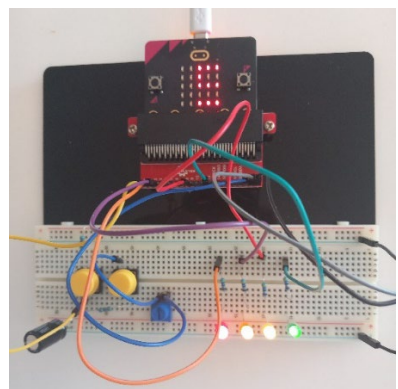
Carga do condensador : 25%



Carga do condensador : 50%

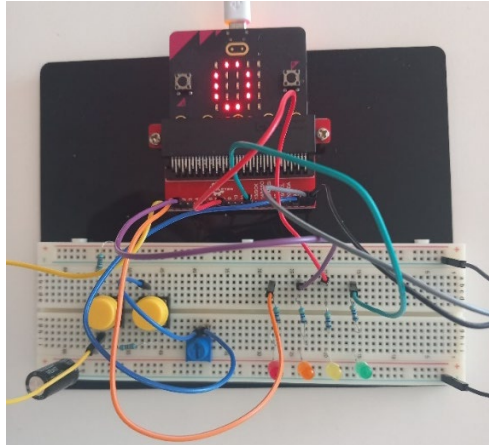


Carga do condensador : 75%



Carga do condensador > 90%

Quando se pressiona o botão da esquerda (efetua a descarga do condensador através da resistência de 2.2k ligada à terra), o estado de descarga é assinalado por um 0 na matrizdo micro :bit e os quatro LEDs da escala exterior aparecem apagados.



Criar o código usando linguagem JavaScript

```
1  let cap_voltage = 0
2  let percentage = 0
3  basic.forever(function () {
4      cap_voltage = pins.analogReadPin(AnalogPin.P0)
5      percentage = cap_voltage / 10
6      basic.showNumber(percentage)
7      if (percentage > 10 && percentage <= 50) {
8          pins.digitalWritePin(DigitalPin.P1, 1)
9      } else if (percentage > 50 && percentage <= 75) {
10         pins.digitalWritePin(DigitalPin.P1, 1)
11         pins.digitalWritePin(DigitalPin.P2, 1)
12     } else if (percentage > 75 && percentage <= 90) {
13         pins.digitalWritePin(DigitalPin.P1, 1)
14         pins.digitalWritePin(DigitalPin.P2, 1)
15         pins.digitalWritePin(DigitalPin.P8, 1)
16     } else if (percentage > 90) {
17         pins.digitalWritePin(DigitalPin.P1, 1)
18         pins.digitalWritePin(DigitalPin.P2, 1)
19         pins.digitalWritePin(DigitalPin.P8, 1)
20         pins.digitalWritePin(DigitalPin.P12, 1)
21     } else {
22         pins.digitalWritePin(DigitalPin.P1, 0)
23         pins.digitalWritePin(DigitalPin.P2, 0)
24         pins.digitalWritePin(DigitalPin.P8, 0)
25         pins.digitalWritePin(DigitalPin.P12, 0)
26     }
27 })
```

Criar o código usando linguagem Python

```
condensador_voltagem = 0
# Percentagem de tensão máxima (até 3V)
percentagem = 0
# Mostra percentagem da tensão (ou carga) na matriz LED do microbit
def show_voltagem(v):
    # Converte percentagem para a matriz LED
    leds = int(v / 4)
    # Mostra a respetiva numeração de acordo com a dimensão x,y da matriz
    for x in range(0, 5):
        for y in range(0, 5):
            check = (x * 5) + y + 1
            if check <= leds:
                led.plot(x, y)
            else:
                led.unplot(x, y)
    return
# Ilumina os 4 LEDs de acordo com a percentagem da tensão (carga no condensador)
def light_leds(v):
    # apaga todos os LEDs
    pins.digital_write_pin(DigitalPin.P1, 0)
    pins.digital_write_pin(DigitalPin.P2, 0)
    pins.digital_write_pin(DigitalPin.P8, 0)
    pins.digital_write_pin(DigitalPin.P12, 0)
    if v > 25:
        # se a carga do condensador atingir 25% liga primeiro LED
        pins.digital_write_pin(DigitalPin.P1, 1)
    if v > 50:
        # se carga do condensador atingir os 50% liga segundo LED
        pins.digital_write_pin(DigitalPin.P2, 1)
    if v > 75:
        # se a carga do condensador atingir 75% liga terceiro LED
        pins.digital_write_pin(DigitalPin.P8, 1)
    if v > 90:
        # o quarto LED ligado assinala carga acima dos 90%
        pins.digital_write_pin(DigitalPin.P12, 1)
    return
#ciclo principal
def on_forever():
    global condensador_voltagem, percentagem
    # lê tensão no condensador através do pino 0 do micro :bit
    condensador_voltagem = pins.analog_read_pin(AnalogPin.P0)
```

```
# Converte valor da carga (max 1023) numa percentagem através da amostragem de 10 bits
da porta analógica P0
percentagem = condensador_voltagem / 10
# Mostra na matriz de LEDs do micro :bit
show_voltagem(percentagem)
# Mostra percentagem de carga no circuito de LEDs
light_leds(percentagem)
basic.forever(on_forever)
```

10. Experiência 10 - Experiência de utilização do LED RGB, nas suas diversas vertentes

Introdução

Nesta experiência vamos utilizar um LED RGB, mudando a cor do mesmo e criando combinações de cores únicas. Dependendo do brilho de cada díodo, quase todas as cores são possíveis!

Componentes necessários

Vais precisar dos seguintes componentes:

- 1 micro:bit
- 1 Cabo USB Micro B
- 1 Breakout do micro:bit
- 1 Placa de ensaios
- 4 Cabos jumper
- 1 LED RGB de cátodo comum
- 3 Resistências de 100Ω

Os objetivos desta experiência são:

- Utilizar o LED RGB, nas suas diversas vertentes
- Configurar diferentes cores no LED RGB
- Associar os botões A e B do micro:bit ao controlo de cor

Funcionamento do LED RGB (vermelho/verde/azul)

O LED RGB (vermelho/verde/azul) é composto por três LEDs num só. O LED RGB tem quatro pinos, sendo que cada um dos três pinos mais curtos controla uma cor individual: vermelho, verde ou azul. O pino mais longo do LED RGB é o pino de terra comum.

Podemos criar um LED de cor personalizada ligando e desligando cores diferentes, fazendo diferentes combinações entre elas. Por exemplo, se ligar o pino vermelho e o pino verde, o LED RGB acender-se-á em cor amarelo



Mas que pino corresponde a que cor? Pegue no LEDRGB de modo a que o pino mais comprido (terra comum) fique alinhado à esquerda, como mostra a

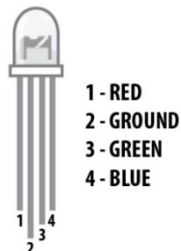
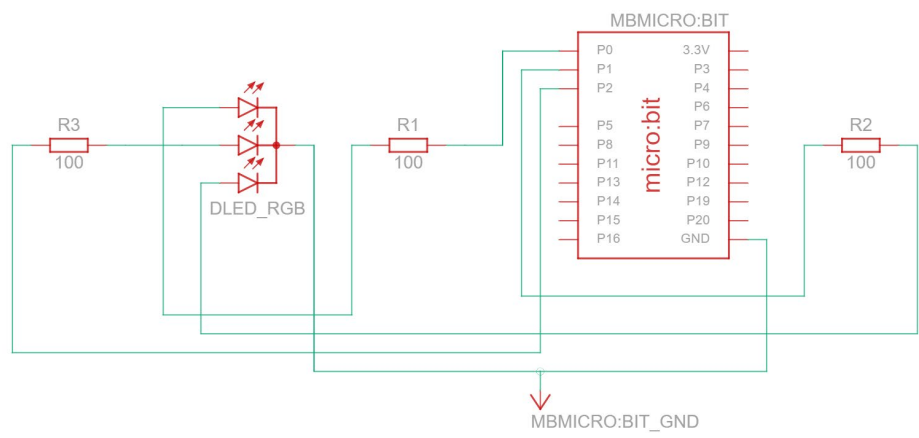


imagem abaixo. Os pinos são (RED)Vermelho, (GROUND)Terra, (GREEN)Verde e (BLUE)Azul, começando pela esquerda.

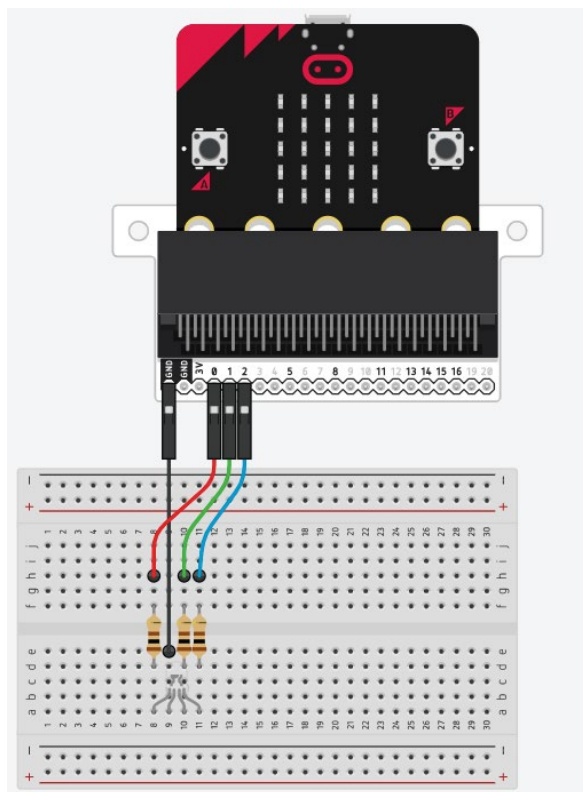
Ao ligar o LED RGB, cada pino colorido precisa de uma resistência limitadora de corrente em série com o pino de E/S do micro:bit que pretende usar para o controlar, como acontece com qualquer LED normal.

Ligação de hardware

Consulte o esquema elétrico abaixo para ver como tudo está ligado:

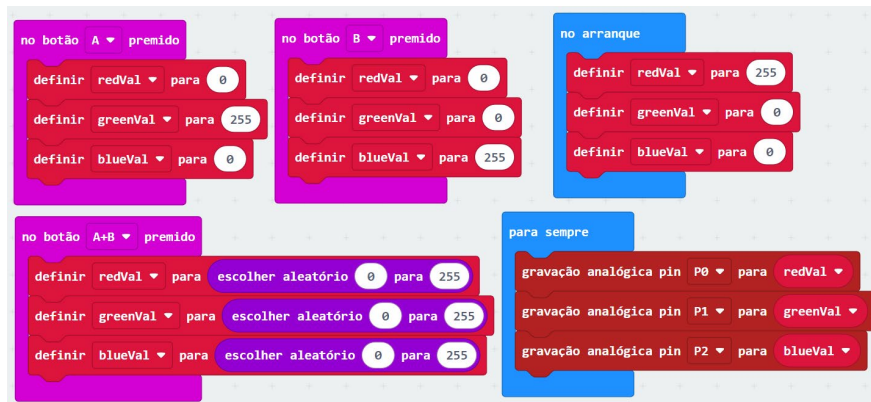


Esquema de ligação da experiência na placa de ensaios



Criar o código usando o editor de blocos

De seguida mostra-se um exemplo de como programar a experiência no Editor de Blocos (<https://makecode.microbit.org>)



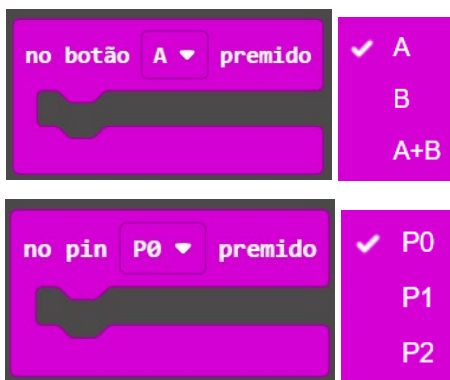
Criar o código usando linguagem JavaScript

```
1  input.onButtonPressed(Button.A, function () {
2      redVal = 0
3      greenVal = 255
4      blueVal = 0
5  })
6  input.onButtonPressed(Button.AB, function () {
7      redVal = randint(0, 255)
8      greenVal = randint(0, 255)
9      blueVal = randint(0, 255)
10 })
11 input.onButtonPressed(Button.B, function () {
12     redVal = 0
13     greenVal = 0
14     blueVal = 255
15 })
16 let blueVal = 0
17 let greenVal = 0
18 let redVal = 0
19 redVal = 255
20 greenVal = 0
21 blueVal = 0
22 basic.forever(function () {
23     pins.analogWritePin(AnalogPin.P0, redVal)
24     pins.analogWritePin(AnalogPin.P1, greenVal)
25     pins.analogWritePin(AnalogPin.P2, blueVal)
26 })
```


Acionar os botões do micro:bit

Encontrará o bloco **no botão premido** na seção de blocos de entrada. É um tipo de bloco diferente daquele a que está habituado. É o que se chama um bloco de evento que executa o código quando algo acontece, e apenas quando isso acontece. Neste caso, é quando os botões integrados (A, B ou A+B) são premidos.

É possível selecionar o botão A, o botão B e ambos os botões (A+B).



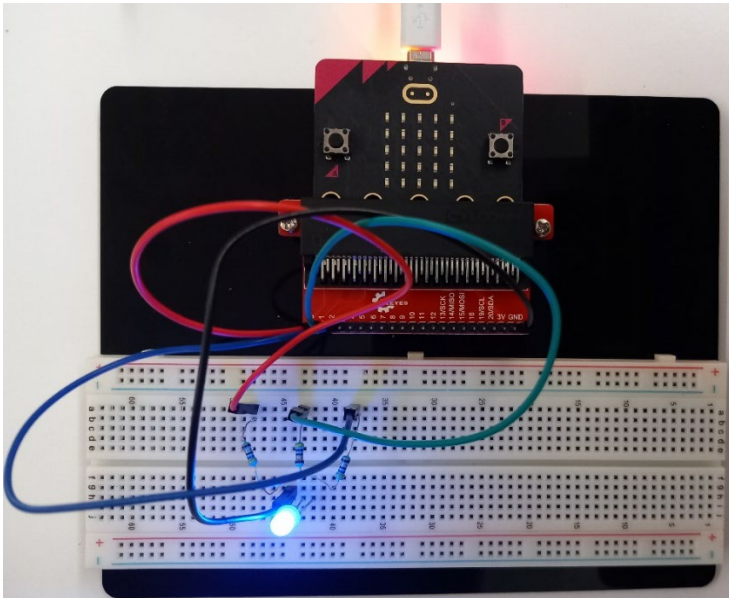
Note que também existe uma função de evento de pino que funciona da mesma forma, e pode utilizá-la com botões externos para criar os seus próprios eventos de hardware externos.

Seleção aleatória

No bloco "no botão A+B premido", repara que foram definidas as variáveis dos pinos de cor para números aleatórios, utilizando o bloco "escolher aleatório". Este bloco pode variar no intervalo de valores entre 0 e outro valor. Neste caso, usamos 255, que é o pico da saída do bloco de escrita analógica.

Funcionamento: o que debes ver

No arranque deve ver o LED acender a vermelho. Se pressionar o botão A do micro:bit, a cor mudará para verde; se pressionar o botão B, a cor mudará para azul; e finalmente, se pressionar o botão A e B em simultâneo, o LED RGB mudará para uma cor aleatória.



Resolução de problemas

O LED permanece escuro ou apresenta uma cor incorrecta: com os quatro pinos do LED RGB tão próximos uns dos outros, por vezes é fácil haver enganos. Verifique novamente se cada pino está ligado corretamente segundo o esquema proposto.

Vermelho mais brilhante que as restantes cores

O díodo vermelho dentro do LED RGB pode ser um pouco mais brilhante do que os outros dois. Para tornar as suas cores mais equilibradas, utilize um valor de resistência mais elevada no pino RED(vermelho).