

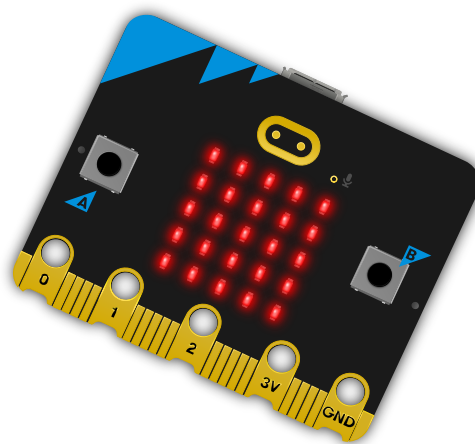
Guia de Consulta Rápida para aprender pensamento computacional e robótica com o BBC micro:bit

Coordenadores:

Breno Ferraz de Oliveira
Eduardo de Amorim Neves

Autores:

Andrey Karvat
Arão Victor Rocha Dos Santos
João Pedro Podolan
José Leonardo Oler
Luciane Guarnieri Brodbeck
Rafaela Lavagnoli
Thainá Gomes Pereira Zampieri



TRUEM

v1 - Setembro de 2023



Sumário

1	Apresentação e Agradecimento	1
2	Primeira letra do nome	2
3	Coração Pulsante - versão 1	3
4	Coração Pulsante - versão 2	4
5	Crachá	5
6	Jogo - Regiões do Brasil	6
7	Alô, eu sou o micro:bit!	7
8	Dado - versão 1	8
9	Dado - versão 2	9
10	Dado - versão 3	10
11	Temperatura	11
12	Mágica	12
13	Antecessor e Sucessor	13
14	Soma - versão 1	14
15	Soma - versão 2	15
16	Subtração - versão 1	16
17	Subtração - versão 2	17
18	Subtração - versão 3	18
19	Multiplicação	19
20	Múltiplos	20
21	Música com o micro:bit - versão 1	21
22	Música com o micro:bit - versão 2	22
23	Música com o micro:bit - versão 3	23
24	Letras	25
25	Snake - versão 1	26



26 Snake - versão 2	27
27 Pedra, Papel, Tesoura - versão 1	29
28 Pedra, Papel, Tesoura - versão 2	31
29 Jogo do Reflexo	32
30 Vela de Aniversário	34
31 Jogo da Chuva - versão 1	35
32 Jogo da Chuva - versão 2	38
33 Carro colorido	41
34 Carro discoteca	42
35 Semáforo-Carro	43
36 Carro com Sensor Ultrassônico	44
37 Carro Segue Faixa	45
38 Controle do carro	46
39 Carro guiado por controle remoto	47
Apêndice A Posição matricial dos LEDs	49
Apêndice B Letras Maiúsculas	50
Apêndice C Letras Minúsculas	51
Apêndice D Ícones disponíveis no micro:bit	52
Apêndice E Música	54
Apêndice F Jogo da cobra e da escada	55



1 Apresentação e Agradecimento

Este é um Guia de Consulta Rápida para ser usado como apoio no ensino de pensamento computacional e robótica. As atividades deste guia foram construídas com base nos quatro pilares do pensamento computacional: a decomposição, o reconhecimento de padrões, a abstração de um problema e os algoritmos.

Os capítulos deste guia possuem QR Codes e hiperlinks dos projetos desenvolvidos. Para acessar uma atividade pelo QR Code, basta apontar a câmera de seu celular para a imagem que aparece no canto superior direito de cada capítulo. O código mostrará um hiperlink e, ao clicar nele, a atividade será aberta online.

Este material foi elaborado pelo **trUEM** que é um projeto de extensão desenvolvido na Universidade Estadual de Maringá (UEM) desde setembro de 2019, fruto de uma parceria de Guilherme Rocha, Denise Adorno Lopes e Breno Ferraz de Oliveira, sendo este o atual coordenador do projeto. O **trUEM** atende pessoas de todas as idades, em especial crianças, que têm interesse em robótica.

As atividades deste Guia de Consulta Rápida, como o próprio nome diz, são apresentadas de forma resumida com o código inteiro já pronto. Caso sejam necessárias informações mais detalhadas, sugerimos a leitura do livro **Aprendendo com o micro:bit BBC**. Disponível em: http://www.truem.com.br/micro_bit.htm.

Agradecemos a Prefeitura Municipal de Maringá que, por meio da Secretaria de Educação e equipe, realizou um convênio junto à Universidade Estadual de Maringá para a inserção do pensamento computacional e da robótica em todas as escolas municipais através do projeto trUEM.

Equipe trUEM.

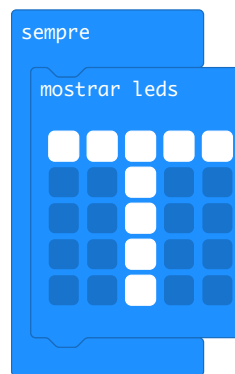


2 Primeira letra do nome

Neste projeto, o aluno deve clicar nos quadradinhos de cor azul escuro no bloco **mostrar leds**, que se encontra na aba **Básico**, para deixá-los branco e, assim, formar a letra inicial do seu nome.

Uma sugestão é fazer a atividade no papel primeiro, antes do aluno realizá-la no computador. Para isso, peça que desenhe 25 bolinhas formando um quadrado de lados 5 por 5 (como no bloco **mostrar leds**) e, usando lápis de cor, preencha as bolinhas que formam a letra inicial do seu nome.

Nos [Apêndice B](#) e [Apêndice C](#) deste Guia de Consulta Rápida há uma referência para desenhar letras maiúsculas e minúsculas nos LEDs do micro:bit.

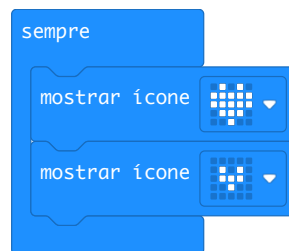


3 Coração Pulsante - versão 1



Nesta atividade, os LEDs do micro:bit se acendem alternando entre dois ícones de coração, dando a impressão que o coração está pulsando. O tempo padrão de troca de um ícone para outro é de cem milésimos de segundo, ou seja, um décimo de segundo.

É interessante utilizar a atividade para conversar com os alunos sobre qual é a média de batimentos do nosso coração por minuto e, ainda, aproveitar para fazer um experimento de contagem de batimentos. Diga para contarem o número de batimentos enquanto marca o tempo de um minuto.





4 Coração Pulsante - versão 2

Na versão 2 da atividade é inserido um intervalo de tempo, isto é, uma pausa de 1000 milissegundos entre um ícone e outro. Note que são necessários dois blocos de pausa, uma para o coração grande e outra para o pequeno. Sem a pausa depois do coração pequeno, o ícone seria alterado para o seguinte no tempo padrão de um décimo de segundo. O tempo de 1000 milissegundos é o mesmo que 1 segundo.

```
sempre
  mostrar ícone [ícone grande]
  pausa (ms) 1000
  mostrar ícone [ícone pequeno]
  pausa (ms) 1000
```

5 Crachá



O projeto do crachá consiste em mostrar nos LEDs do micro:bit informações do aluno, como nome, idade e um ícone que representa seu estado de humor.

O desafio deste projeto é fazer a montagem sem os blocos de **limpar tela**, num primeiro momento, e pedir uma solução para que o ícone mostrado por último não seja empurrado para o lado esquerdo quando um botão é pressionado.

```
no botão A pressionado
  limpar tela
  mostrar string "trUEM"
```

```
no botão B pressionado
  limpar tela
  mostrar string "4"
```

Trocar o nome trUEM pelo nome do aluno.

```
no botão A+B pressionado
  limpar tela
  mostrar ícone [ícone de humor]
```

Trocar 4, que é a idade do projeto trUEM, pela idade do aluno.

O aluno deve colocar o ícone que representa o humor dele.



6 Jogo - Regiões do Brasil

Neste jogo, o professor deve falar o nome de um estado brasileiro e os alunos devem usar o micro:bit para indicar a região do Brasil em que o estado está localizado. Ganha o aluno que tiver o maior número de acertos. Pode-se incluir a regra de ponto extra para o aluno que disser a capital do estado mencionado.

Para aumentar o nível de dificuldade, os alunos devem memorizar os comandos que mostram cada seta para responderem as perguntas sem consultar o código montado.



Observação: Foi utilizada a seta **Oeste** para a região Centro-Oeste porque no micro:bit não existe esta opção.



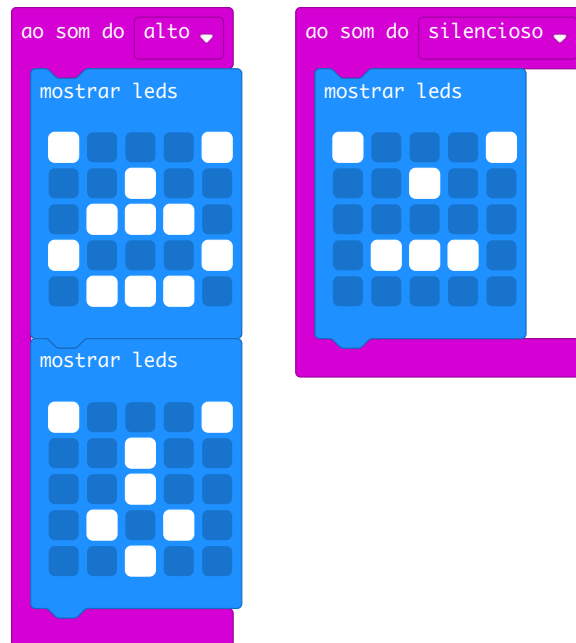
Disponível em: <<https://pt.wikipedia.org/wiki/Brasil>>. Acesso em 29 de julho de 2023.



7 Alô, eu sou o micro:bit!

Esta atividade consiste em programar o micro:bit para que os LEDs se acendam quando falamos próximo ao microfone da placa, simulando uma dublagem.

Os alunos devem desenhar nos LEDs carinhas com bocas que representam os movimentos de fala e silêncio. Assim, quando o microfone receber um som, os ícones se alternarão para mostrar a boca se mexendo e, quando não houver som, a boca aparecerá fechada.





8 Dado - versão 1

Este projeto é a versão digital de um dado para jogos de tabuleiro. No código apresentado, quando o botão **A** do micro:bit é pressionado, um número entre 1 e 6 é escolhido aleatoriamente e mostrado nos LEDs.

Para termos a certeza que o botão foi pressionado, no caso de um mesmo número ser escolhido em sequência, incluímos três ícones que são mostrados antes do resultado.

No [Apêndice F](#) deste Guia de Consulta Rápida há o jogo da cobra e da escada para ser usado na brincadeira com o dado criado neste projeto.

no botão **A** pressionado

mostrar ícone

mostrar ícone

mostrar leds

mostrar número **escolher aleatório** 1 até 6

Pode ser incluído o bloco "no botão B pressionado" para colocar um "limpar tela".

O "no botão A pressionado" pode ser trocado por "em agitar".

Para mostrar que o botão A foi pressionado, foram colocados dois "mostrar ícone" e um "mostrar leds", criando um efeito divertido.



9 Dado - versão 2

Na versão 2 do dado, quando o botão **A** do micro:bit é pressionado, um número entre 0 e 6 é escolhido aleatoriamente e mostrado nos LEDs. O número zero indicará que o jogador perdeu a vez, pois moverá 0 casas no tabuleiro.

Utilize o jogo da cobra e da escada, disponível no [Apêndice F](#), para brincar com o dado.

no botão **A** pressionado

mostrar ícone [ícone de dado]

mostrar ícone [ícone de dado]

mostrar leds

mostrar número [escolher aleatório 0 até 6]

Uma sugestão é incluir o bloco "no botão B pressionado" e colocar um "limpar tela".

O "no botão A pressionado" pode ser trocado por "em agitar".

Para mostrar que o botão A foi pressionado, foi colocado o comando para "mostrar ícone" duas vezes e um "mostrar leds", criando um efeito divertido.

10 Dado - versão 3



A versão 3 do dado utiliza conceitos de variáveis, condicionais e comparação.

Quando o botão **A** do micro:bit é pressionado, um número entre 0 e 6 é escolhido aleatoriamente e armazenado em uma variável. Se o número escolhido for zero, uma caveira aparecerá nos LEDs e a melodia funeral será tocada, indicando que o jogador perdeu a vez. Se o número for diferente de zero, ele será mostrado no micro:bit.

Para brincar com o dado, jogue o jogo da cobra e da escada disponível no [Apêndice F](#).

```
no botão A pressionado
  mostrar ícone [skull]
  mostrar ícone [skull]
  mostrar leds
  definir dado para escolher aleatório 0 até 6
  se dado = 0 então
    mostrar ícone [skull]
    play melody funeral until done
  senão
    mostrar número dado
```

É possível incluir "no botão B pressionado" e colocar um "limpar tela".

O "no botão A pressionado" pode ser trocado por "em agitar".

Para mostrar que o botão A foi pressionado, foi colocado "mostrar ícone" duas vezes e um "mostrar leds", criando um efeito divertido.



11 Temperatura

Nesta atividade, o micro:bit funciona como um termômetro que mede a temperatura do ambiente. A unidade de medida de temperatura padrão do micro:bit é graus Celsius mas, montando uma operação matemática para a conversão, pode ser mostrada também em graus Fahrenheit.

Ao pressionar o botão **A** do micro:bit, a temperatura em graus Celsius será mostrada nos LEDs. Quando o botão **B** for pressionado, a temperatura em graus Fahrenheit aparecerá nos LEDs. Lembrando que Fahrenheit é a unidade de medida de temperatura utilizada nos Estados Unidos.

Peça aos alunos para colocarem o micro:bit entre as duas mãos e aguardar um tempinho. Ao abri-las, perceberão que a temperatura subiu. Ou ainda, colocar a placa na testa para medir a temperatura corporal.

Outra sugestão de atividade é pedir para os alunos pesquisarem diversas temperaturas, como por exemplo: 1 - a menor temperatura ambiente já registrada no Brasil; 2 - a menor temperatura ambiente registrada na Terra; 3 - a temperatura da superfície do Sol; 4 - a temperatura da chama do fogão; e 5 - a menor temperatura possível.

```
no botão A pressionado
  mostrar número temperatura (°C)
  mostrar string "C"
```

```
no botão B pressionado
  mostrar número 9 x temperatura (°C) / 5 + 32
  mostrar string "F"
```

No comando "mostrar número", do bloco "no botão B pressionado", é necessário primeiro inserir a soma, depois a divisão e, por fim, a multiplicação.

Importante: Sempre insira as operações no lugar do zero mais à esquerda.

12 Mágica



Este projeto brinca com a mágica do teletransporte, pois faz um ícone desaparecer de um micro:bit e aparecer em outro apenas com um simples agitar da placa.

Para que isso aconteça, os micro:bits devem estar definidos no mesmo grupo de rádio e enviar a mesma informação, que neste projeto é um número, para mostrar um ícone. Desta forma, o mesmo código deve ser baixado para o par de micro:bits da brincadeira.



13 Antecessor e Sucessor



A atividade deste projeto consiste em programar o micro:bit para mostrar nos LEDs números aleatórios quando os botões A e B são pressionados, simultaneamente, e depois mostrar os números antecessores e sucessores do escolhido. Ao pressionar o botão A, o número antecessor será mostrado nos LEDs e, quando o botão B for pressionado, o número sucessor aparecerá no micro:bit.

```
no botão A+B pressionado
  definir NUMERO para escolher aleatório 1 até 8
  mostrar número NUMERO
```

```
no botão A pressionado
  definir ANTECESSOR para NUMERO - 1
  mostrar número ANTECESSOR
```

```
no botão B pressionado
  definir SUCESSOR para NUMERO + 1
  mostrar número SUCESSOR
```

Importante: Quando for fazer uma variável, **JAMAIS** utilize acentuação e espaços em brancos.

14 Soma - versão 1



A atividade da soma é um dos exemplos do uso da programação para criar um jogo que auxilia na construção do raciocínio matemático dos alunos.

Nesta atividade, além do uso de um comando de adição da aba **Matemática**, há também a utilização de variáveis para a operação. Quando o botão **A** for pressionado, dois números aleatórios serão escolhidos para aparecerem nos LEDs do micro:bit com um sinal de adição entre eles. Depois, serão mostrados um sinal de igualdade e um ponto de interrogação.

O aluno deverá efetuar a soma e dizer a resposta. Só então, pressionará o botão **B** para conferir se acertou.

```
no botão A pressionado
  definir X para escolher aleatório 0 até 9
  definir Y para escolher aleatório 0 até 9
  mostrar número X
  mostrar leds
  mostrar número Y
  mostrar string "="
  mostrar string "?"

no botão B pressionado
  limpar tela
  mostrar número X + Y
```

15 Soma - versão 2



Na versão 2 da atividade da soma, há a inclusão de mais um número na operação. Então, quando o botão **A** for pressionado, três números aleatórios serão escolhidos para serem mostrados nos LEDs do micro:bit e aparecerá um sinal de adição entre cada um deles. Depois, serão mostrados um sinal de igualdade e um ponto de interrogação.

Da mesma forma que na versão 1, o aluno deverá efetuar a soma e apertar o botão **B** para conferir sua resposta.

no botão **A** pressionado

- definir **X** para escolher aleatório 0 até 9
- definir **Y** para escolher aleatório 0 até 9
- definir **Z** para escolher aleatório 0 até 9
- mostrar número **X**
- mostrar leds
- mostrar número **Y**
- mostrar leds
- mostrar número **Z**
- mostrar string "="
- mostrar string "?"

no botão **B** pressionado

- limpar tela
- mostrar número **X + Y + Z**

Para somar 3 variáveis é necessário colocar uma soma dentro de outra soma. A operação de soma está na aba matemática.

16 Subtração - versão 1



Assim como na atividade da soma, podemos criar um projeto com a operação matemática de subtração.

Quando o botão **A** for pressionado, dois números aleatórios serão escolhidos para serem mostrados nos LEDs do micro:bit e um sinal de subtração aparecerá entre eles. Depois, serão mostrados um sinal de igualdade e um ponto de interrogação.

Após efetuar a operação de subtração, o aluno deverá apertar o botão **B** para conferir se calculou corretamente.

Neste projeto, definimos que o segundo número aleatório seja igual ou menor que o primeiro para que a operação não resulte em um número negativo.

```
no botão A pressionado
  definir X para escolher aleatório 0 até 9
  definir Y para escolher aleatório 0 até X
  mostrar número X
  mostrar leds
  mostrar número Y
  mostrar string "="
  mostrar string "?"

no botão B pressionado
  limpar tela
  mostrar número X - Y
```

17 Subtração - versão 2



Na versão 2 do projeto de subtração, há uma condição para mostrar o maior número primeiro. Então, quando o botão **A** for pressionado, dois números aleatórios serão escolhidos e o maior será mostrado nos LEDs do micro:bit e, em seguida, o menor. Entre os números aparecerá um sinal de subtração. Depois, serão mostrados um sinal de igualdade e um ponto de interrogação.

Quando o botão **B** for pressionado, o programa executará mais uma condição para efetuar a subtração do maior número pelo menor para que o resultado não seja um número negativo.

```
no botão A pressionado
  definir X para escolher aleatório 0 até 9
  definir Y para escolher aleatório 0 até 9
  se X > Y então
    mostrar número X
    mostrar leds
    mostrar número Y
  senão
    mostrar número Y
    mostrar leds
    mostrar número X
  +
  mostrar string "="
  mostrar string "?"

no botão B pressionado
  limpar tela
  se X > Y então
    mostrar número X - Y
  senão
    mostrar número Y - X
  +
```

18 Subtração - versão 3



Para a versão 3 do projeto de subtração não há a condição para evitar que o resultado seja um número negativo.

Assim, quando o botão **A** for pressionado, dois números aleatórios serão escolhidos e mostrados nos LEDs do micro:bit e, entre eles, aparecerá um sinal de subtração. Depois, um sinal de igualdade e um ponto de interrogação.

Quando o botão **B** for pressionado, o programa executará a subtração dos dois números e mostrará o resultado.

```
no botão A pressionado
  definir X para escolher aleatório 0 até 9
  definir Y para escolher aleatório 0 até 9
  mostrar número X
  mostrar leds
  mostrar número Y
  mostrar string "="
  mostrar string "?"

no botão B pressionado
  limpar tela
  mostrar número X - Y
```

19 Multiplicação



Esta atividade de multiplicação, como a soma e a subtração, também auxilia no desenvolvimento do raciocínio matemático dos alunos.

Quando o botão **A** for pressionado, dois números aleatórios serão escolhidos para serem mostrados nos LEDs do micro:bit e um sinal de multiplicação aparecerá entre eles. Depois, serão mostrados um sinal de igualdade e um ponto de interrogação.

Após efetuar a operação de multiplicação, o aluno deverá apertar o botão **B** para conferir sua resposta.

Nesta atividade, definimos que o primeiro número aleatório será de 1 até 9 para que não apareça a multiplicação de zero por zero.

De acordo com cada ano escolar, pode-se alterar os números aleatórios para praticar a tabuada que os alunos estão aprendendo naquele momento.

```
no botão A pressionado
  definir X para escolher aleatório 1 até 9
  definir Y para escolher aleatório 0 até 9
  mostrar número X
  mostrar leds
  mostrar número Y
  mostrar string "="
  mostrar string "?"

no botão B pressionado
  limpar tela
  mostrar número X x Y
```

Podem ser inseridos comandos de "pausa (ms)" entre os blocos "mostrar número X" e "mostrar leds", "mostrar número Y" e "mostrar string \"=\" para mostrar mais lentamente a operação.

20 Múltiplos



Outra forma de estimular o raciocínio matemático é a brincadeira de somar ou subtrair um mesmo número por determinadas vezes.

O projeto de múltiplos define o início do jogo no número zero e, pressionando os botões **A** e **B**, respectivamente, o número dois pode ser adicionado ou subtraído ao último resultado.

No botão **B** foi criada uma condição para que não seja efetuada a subtração se o último resultado obtido for o número zero.

Além do número dois, o projeto pode ser alterado por outros números maiores para tornar o jogo mais difícil.

```
no iniciar
  definir numero para 0
  mostrar número numero
```

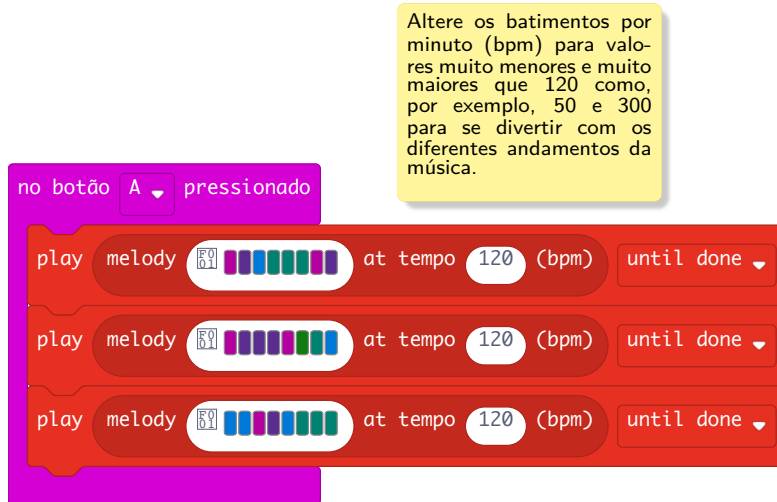
```
no botão A pressionado
  alterar numero por 2
  mostrar número numero
```

```
no botão B pressionado
  se numero > 0 então
    alterar numero por -2
    mostrar número numero
  +
```

21 Música com o micro:bit - versão 1



A atividade de música com o micro:bit é bastante apropriada para ensinar as notas musicais aos alunos.



A música da figura abaixo é a Dó Ré Mi Fá. No editor de notas, cada coluna pode ter apenas uma nota e cada linha representa uma nota musical.

Na primeira linha, de baixo para cima, e na primeira coluna está a nota Dó. Na sequência, subindo as linhas, temos as notas Ré, Mi, Fá, Sol, Lá, Si. A sequência termina com a nota Dó, na última linha, em um tom mais agudo.

Uma sugestão para esta atividade é explicar o que significa o tempo de 120 bpm e alterá-lo por números maiores e menores para que os alunos escutem as diferenças de tempo de uma melodia.



22 Música com o micro:bit - versão 2



Nesta versão 2 da música, incluímos uma condição para que sejam mostrados ícones de carinhas enquanto a música está sendo tocada. As bocas aberta e fechada simulam um cantor.

Os alunos podem ser estimulados a criar outros formatos de boca para deixar a simulação mais completa e divertida.

```
no botão A pressionado
  definir canta para verdadeiro
  play melody at tempo 120 (bpm) until done
  play melody at tempo 120 (bpm) until done
  play melody at tempo 120 (bpm) until done
  definir canta para falso

sempre
  se canta então
    mostrar leds
  senão
    limpar tela
```

23 Música com o micro:bit - versão 3



A música da versão 3 é a Asa Branca, composta por Luiz Gonzaga e Humberto Teixeira.

Para a montagem, utilizamos um outro bloco da aba **Música** que permite selecionar as notas no teclado de um piano. Nesta montagem, também definimos o tempo de batidas por minuto, o tempo de duração de cada nota e o tempo de intervalo entre os versos da música.

```
no botão A pressionado
  definir canta para verdadeiro
  definir o tempo em (bpm) 40
  play tone C Médio for 1/8 batida until done
  play tone D Médio for 1/8 batida until done
  play tone E Médio for 1/4 batida until done
  play tone G Médio for 1/4 batida until done
  play tone G Médio for 1/4 batida until done
  play tone E Médio for 1/4 batida until done
  play tone F Médio for 1/4 batida until done
  play tone F Médio for 1/4 batida until done
  rest for 1/4 batida
  play tone C Médio for 1/8 batida until done
  play tone D Médio for 1/8 batida until done
  play tone E Médio for 1/4 batida until done
  play tone G Médio for 1/4 batida until done
  play tone G Médio for 1/4 batida until done
  play tone F Médio for 1/4 batida until done
  play tone E Médio for 1/2 batida until done
  rest for 1/8 batida

sempre
  se canta então
    mostrar leds
  mostrar leds
  senão
    limpar tela
```

Continua na próxima página.



Continuação da página anterior.

```
play tone C Médio for 1/8 batida until done
play tone C Médio for 1/8 batida until done
play tone D Médio for 1/8 batida until done
play tone E Médio for 1/4 batida until done
play tone G Médio for 1/4 batida until done
rest for 1/8 batida
play tone G Médio for 1/8 batida until done
play tone F Médio for 1/8 batida until done
play tone E Médio for 1/8 batida until done
play tone C Médio for 1/4 batida until done
play tone F Médio for 1/4 batida until done
rest for 1/8 batida
play tone F Médio for 1/8 batida until done
play tone E Médio for 1/8 batida until done
play tone D Médio for 1/8 batida until done
play tone D Médio for 1/4 batida until done
play tone E Médio for 1/4 batida until done
rest for 1/8 batida
play tone D Médio for 1/8 batida until done
play tone D Médio for 1/8 batida until done
play tone C Médio for 1/8 batida until done
play tone C Médio for 1/2 batida until done
definir canta para falso
```

No [Apêndice E](#) deste **Guia de Consulta Rápida**, está a partitura da música. Também há uma sugestão de atividade de adivinhação em conjunto com uma explicação sobre as notas musicais.

24 Letras



Este projeto consiste em mostrar letras maiúsculas e minúsculas nos LEDs do micro:bit acendendo cada luz por meio do par ordenado de coluna (x) e linha (y).

É importante desenhar a matriz de LEDs na lousa e, junto com os alunos, escrever o par (x,y) correspondente a cada luz. Vale lembrar que tanto as colunas como as linhas começam no número zero - Veja o [Apêndice A](#).

no botão **A** pressionado

limpar tela

plotar x 1 y 0

plotar x 2 y 0

plotar x 0 y 1

plotar x 0 y 2

plotar x 0 y 3

plotar x 0 y 4

plotar x 3 y 1

plotar x 3 y 2

plotar x 3 y 3

plotar x 3 y 4

plotar x 1 y 2

plotar x 2 y 2

no botão **B** pressionado

limpar tela

plotar x 1 y 1

plotar x 2 y 1

plotar x 3 y 1

plotar x 0 y 2

plotar x 0 y 3

plotar x 3 y 2

plotar x 3 y 3

plotar x 1 y 4

plotar x 2 y 4

plotar x 3 y 4

plotar x 4 y 4

No botão A pressionado o micro:bit apagará todos os LEDs e, na sequência, acenderá a letra "A" maiúscula.

No botão B pressionado o micro:bit apagará todos os LEDs e, na sequência, acenderá a letra "a" minúscula.

25 Snake - versão 1



Esta atividade é o jogo de acender os LEDs para formar um caminho.

Quando o botão **A** é pressionado, o LED da direita é aceso e, no botão **B**, o LED da linha de baixo se acende. O objetivo é apertar os botões **A** e **B** para reproduzir o caminho definido pelo professor.

É importante observar que foi criada uma condição para que os LEDs continuem sendo acesos mesmo que o aluno pressione os botões mais do que cinco vezes para a direita ou para baixo. Do contrário, o jogo deixaria de funcionar.

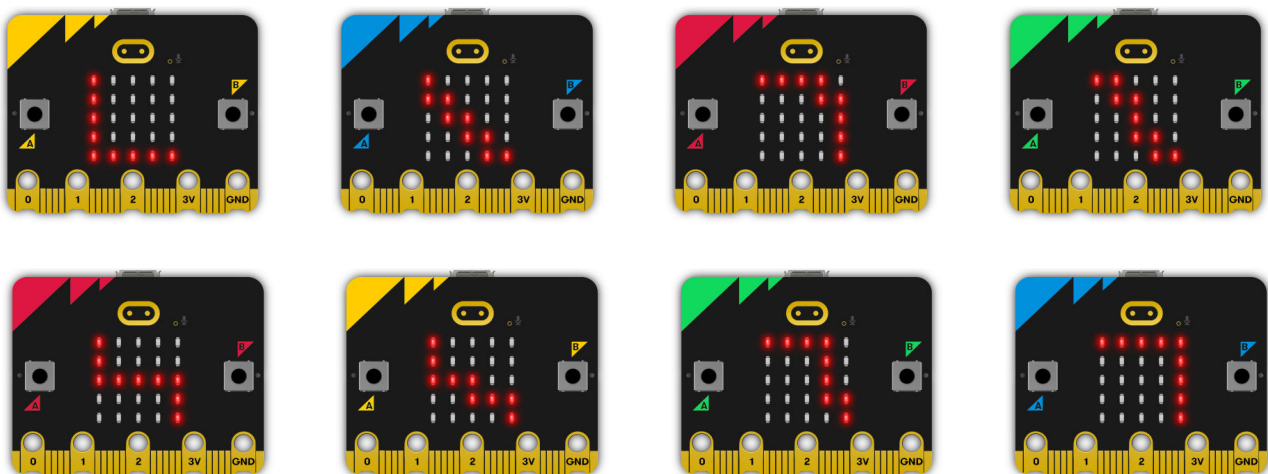
```
no iniciar
  definir X para 0
  definir Y para 0
  plotar x X y Y

no botão A pressionado
  se X < 4 então
    alterar X por 1
    plotar x X y Y

no botão B pressionado
  se Y < 4 então
    alterar Y por 1
    plotar x X y Y
```

Para reiniciar o jogo é necessário apertar o botão de reset que fica na parte detrás da placa.

Seguem alguns exemplos de caminhos que podem ser usados. Lembre-se que só é possível mover o LED para a direita e para baixo. Se o aluno errar o caminho no meio do jogo, ele deve apertar o botão de reset, que fica atrás do micro:bit, para reiniciar o caminho.



26 Snake - versão 2



Nesta versão 2 do jogo Snake é possível saber o tempo que o jogador levou para cumprir sua missão.

Enquanto o jogador estiver construindo o caminho, o micro:bit estará marcando o tempo e, quando o último LED do canto inferior direito for aceso, a contagem do tempo para. Para saber o tempo total de jogo, basta pressionar o logotipo do micro:bit.

```
no iniciar
  definir tempo_inicial para 0
  definir tempo_final para 0
  definir inicio para verdadeiro
  definir X para 0
  definir Y para 0
  plotar x X y Y

no botão A pressionado
  se inicio então
    definir inicio para falso
    definir tempo_inicial para tempo de execução (ms)
    se X < 4 então
      alterar X por 1
      plotar x X y Y
    se X = 4 e Y = 4 então
      definir tempo_final para tempo de execução (ms)
```

O botão A move o LED para direita e o botão B move o LED para baixo.

Para reiniciar o jogo é necessário apertar o botão de reset que fica na parte detrás da placa.

O primeiro SE nos botões A e B serve para acionar a marcação do tempo. O último SE para o tempo quando o LED do canto inferior direito for acionado.

Este condicional SE é necessário para mostrar o tempo somente ao final do jogo, isto é, quando o LED do canto inferior direito é acionado.

O tempo é medido em milissegundos, por este motivo o resultado deve ser dividido por 1000 para ser mostrado em segundos.

Continua na próxima página.



Continuação da página anterior.

```
no botão B pressionado
se inicio então
  definir inicio para falso
  definir tempo_inicial para tempo de execução (ms)
+
se Y < 4 então
  alterar Y por 1
  plotar x X y Y
+
se X = 4 e Y = 4 então
  definir tempo_final para tempo de execução (ms)
+
```

```
no logotipo pressionado
se tempo_inicial ≠ 0 e tempo_final ≠ 0 então
  limpar tela
  mostrar número (tempo_final - tempo_inicial) / 1000
  mostrar string "SEGUNDOS"
+
```

27 Pedra, Papel, Tesoura - versão 1



Neste projeto reproduzimos o famoso jogo jokenpô ou pedra, papel e tesoura. As regras do jogo são: a pedra ganha da tesoura, amassando-a ou quebrando-a; a tesoura ganha do papel, cortando-o; e o papel ganha da pedra, embrulhando-a.

Quando o jogador apertar o botão **A**, os LEDs serão acesos, como na figura abaixo, e um número aleatório de 1 a 3 será escolhido e armazenado na variável **sorteio**. Para cada número há uma condição para mostrar os ícones de tesoura, pedra e papel.

Além de usar o micro:bit para jogar o jokenpô, a diversão está também em desenhar os ícones pedra, papel e tesoura no bloco **mostrar leds**. Deixe os alunos criarem suas próprias representações e utilize nossa sugestão apenas se necessário.

```
no botão A pressionado
limpar tela
mostrar leds
mostrar leds
mostrar leds
definir sorteio para escolher aleatório 1 até 3
```

Continua na próxima página.



Continuação da página anterior.

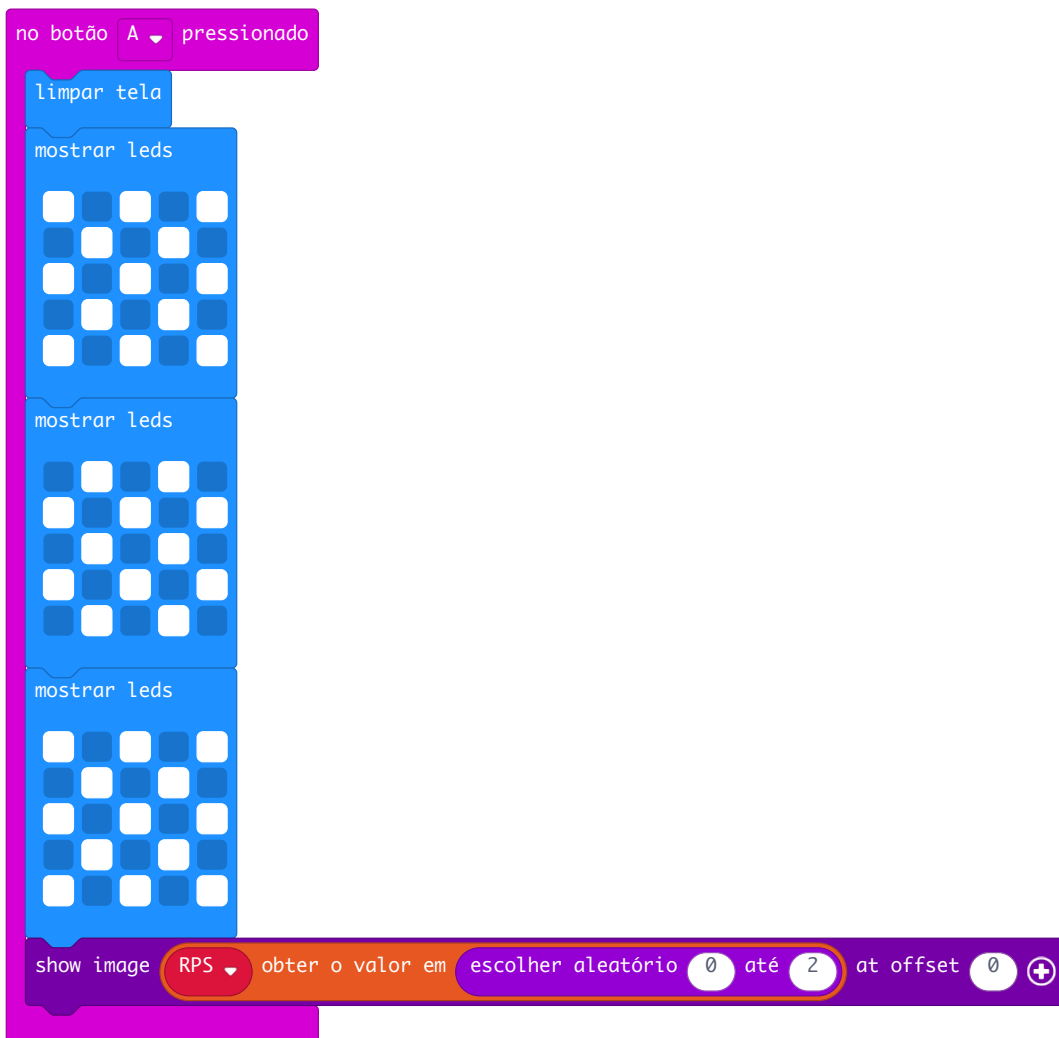
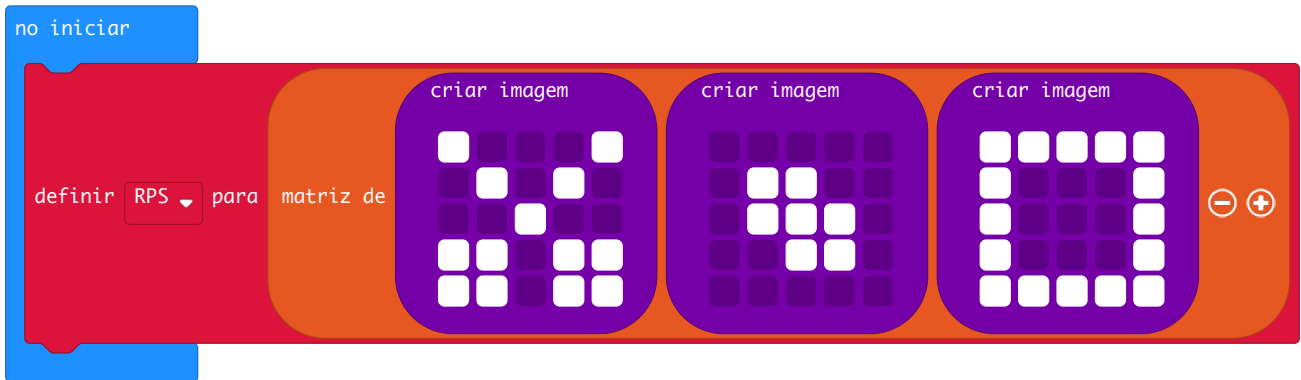
```
se < sorteio > = < 1 > então
  mostrar leds
  [ 4x4 grid of LEDs ]
+
se < sorteio > = < 2 > então
  mostrar leds
  [ 4x4 grid of LEDs ]
+
se < sorteio > = < 3 > então
  mostrar leds
  [ 4x4 grid of LEDs ]
+
[ empty block ]
```

28 Pedra, Papel, Tesoura - versão 2



Na versão 2 do jogo pedra, papel e tesoura a diferença se dá somente na programação. Os ícones são os mesmos e o acionamento continua sendo pelo botão **A**, como na versão 1.

Para a programação desta versão, clique na aba **Avançado** para que apareçam novas funções. Os novos comandos usados no bloco **no iniciar** estão nas abas **Matriz** e **Imagens**. A variável **RPS** criada é o acrônimo de Rock (pedra), Paper (papel), Scissors (tesoura), nome do jogo em inglês.



29 Jogo do Reflexo



O jogo do reflexo é aquele em que o jogador que executa a ação mais rápido, ganha. No micro:bit ganha quem aperta o botão primeiro quando aparece um ícone de carinha feliz nos LEDs. Se apertar ao aparecer uma carinha triste, perde um ponto. Quando o jogador acerta, aparece uma seta na direção de seu botão e toca a música risadinha. Se apertar errado, a seta indica quem perdeu e toca a música triste. O jogo acaba quando um dos jogadores marca três pontos, então aparece a palavra Ganhou e mostra a letra do botão do ganhador.

Para reiniciar o jogo, basta pressionar o logotipo.

```
sempre
se continuar então
limpar tela
pausa (ms) escolher aleatório 1000 até 5000
definir Feliz para escolher de forma aleatória verdadeiro ou falso
definir jogar para verdadeiro
se Feliz então
mostrar ícone
senão
mostrar ícone
+
pausa (ms) 2000
se Jogador_A = 3 então
definir continuar para falso
limpar tela
mostrar string "Ganhou"
mostrar string "A"
+
se Jogador_B = 3 então
definir continuar para falso
limpar tela
mostrar string "Ganhou"
mostrar string "B"
+
+
+
```



Continua na próxima página.



Continuação da página anterior.

```
no iniciar
  definir jogar para falso
  definir continuar para verdadeiro
  definir Jogador_A para 0
  definir Jogador_B para 0
  mostrar ícone
  pausa (ms) 2000
```

```
no logotipo pressionado
  definir jogar para falso
  definir Jogador_A para 0
  definir Jogador_B para 0
  mostrar ícone
  pausa (ms) 2000
  definir continuar para verdadeiro
```

```
no botão A pressionado
  se jogar então
    definir jogar para falso
    mostrar a seta Oeste
    se Feliz então
      alterar Jogador_A por 1
      play sound risadinha until done
    senão
      alterar Jogador_A por -1
      play sound triste until done
  +
  +
```

```
no botão B pressionado
  se jogar então
    definir jogar para falso
    mostrar a seta Leste
    se Feliz então
      alterar Jogador_B por 1
      play sound risadinha until done
    senão
      alterar Jogador_B por -1
      play sound triste until done
  +
  +
```



30 Vela de Aniversário

O projeto da vela de aniversário no micro:bit simula uma chama acesa, piscando os LEDs da primeira linha, que é apagada com um sopro ou som alto. A melodia aniversário toca enquanto a vela está acesa.

Na programação, foi utilizada a condição da vela estar sempre acesa. Depois que é apagada, basta clicar no botão **A** para acendê-la novamente.

```
no iniciar
  definir aceso para verdadeiro
  play melody aniversário in background

no botão A pressionado
  play melody aniversário in background
  definir aceso para verdadeiro

ao som do alto
  definir aceso para falso

sempre
  se aceso então
    mostrar leds
    definir fogo para escolher aleatório 1 até 3
    plotar x fogo y 0
    apagar x 2 y 0
    pausa (ms) 100
  senão
    mostrar leds
```

31 Jogo da Chuva - versão 1



O jogo da chuva é aquele em que uma gota vai caindo e o jogador deve pegá-la antes que ela atinja o chão.

Neste projeto, foi montado um jogo muito divertido com sons, que fica mais difícil com o passar do tempo e, ainda, marca os pontos do jogador ao final.

Para a montagem dos blocos são utilizadas quase todas as abas da biblioteca de funções. O código é um pouco extenso, mas ainda assim, ele pode ser feito com alunos do segundo ano para desenvolver a atenção em copiar uma atividade e fazer uso de diversas funções.

```
no iniciar
  definir pause2 para 500

no botão A pressionado
  se x > 0 então
    apagar x x y 4
    alterar x por -1
    plotar x x y 4

no botão B pressionado
  se x < 4 então
    apagar x x y 4
    alterar x por 1
    plotar x x y 4
```

Continua na próxima página.



Continuação da página anterior.

```
sempre
  se score = 0 então
    iniciar melodia entretenimento repetindo uma vez
    mostrar ícone
    pausa (ms) 100
    mostrar ícone
    pausa (ms) 100
    mostrar leds
    pausa (ms) 100
    limpar tela
    definir x para 2
    plotar x x y 4
    definir rand para escolher aleatório 0 até 4
```

Continua na próxima página.



Continuação da página anterior.

```
para index de 0 a 4
  executar
    plotar x rand y index
    pausa (ms) pause2
    apagar x rand y index
se rand = x então
  se pause2 > 0 então
    reproduzir tom C Médio por 1/8 batida
    alterar pause2 por -25
    plotar x x y 4
    +
    alterar score por 1
  senão -
    iniciar melodia funeral repetindo uma vez
    limpar tela
    mostrar número score
    mostrar string "Game Over"
    plotar x x y 4
    definir score para 0
    definir pause2 para 500
  +
```


32 Jogo da Chuva - versão 2



Na versão 2 do jogo da chuva, foi incluída a condição de contornar o micro:bit. Ou seja, quando o cursor que representa o jogador está localizado na posição (4,4) e o botão **B** é pressionado, o cursor reaparece no LED de posição (0,4). O mesmo acontece quando o cursor está na posição (0,4) e o botão **A** é pressionado, o LED que se acende é o de posição (4,4).

```
no iniciar
  definir pausa para 500

no botão A pressionado
  se x > 0 então
    apagar x x y 4
    alterar x por -1
    plotar x x y 4
  senão
    apagar x x y 4
    definir x para 4
    plotar x x y 4

no botão B pressionado
  se x < 4 então
    apagar x x y 4
    alterar x por 1
    plotar x x y 4
  senão
    apagar x x y 4
    definir x para 0
    plotar x x y 4
```

Continua na próxima página.



Continuação da página anterior.

```
sempre
se pontos = 0 então
  iniciar melodia entretenimento repetindo uma vez
  mostrar ícone
  pausa (ms) 500
  mostrar ícone
  pausa (ms) 500
  mostrar leds
  pausa (ms) 500
  limpar tela
  definir x para 2
  plotar x x y 4
  definir aleatorio para escolher aleatório 0 até 4
```

Continua na próxima página.



Continuação da página anterior.

```
para índice de 0 a 4
  executar
    plotar x aleatorio y índice
    pausa (ms) pausa
    apagar x aleatorio y índice
se aleatorio = x então
  se pausa > 0 então
    alterar pausa por -25
  +
  reproduzir tom C Médio por 1/8 batida
  alterar pontos por 1
  plotar x x y 4
senão -
  iniciar melodia funeral repetindo uma vez
  limpar tela
  mostrar número pontos
  pausa (ms) 5000
  mostrar ícone
  definir pausa para 500
  definir pontos para 0
  plotar x x y 4
  +
```

33 Carro colorido



Nesta atividade, o micro:bit foi acoplado ao carrinho **Maqueen** e foi adicionada a extensão **Neopixel** para montar os blocos de comando.

Quando o carrinho é ligado, os LEDs que estão na parte debaixo ficam trocando de cores na sequência de vermelho, verde e azul. A troca de cores acontece de forma rotacional nos quatro LEDs e o efeito multicolorido que aparece embaixo do carrinho é muito festivo.

```
no iniciar
  mostrar ícone [LEDs]
  definir strip para NeoPixel at pin P15 with 4 leds as RGB (RGB format)
  strip show rainbow from 1 to 360

sempre
  strip rotate pixels by 1
  strip show
  pausa (ms) 100
```

34 Carro discoteca



No projeto do carro discoteca, a programação é feita para que o micro:bit toque música, o carrinho vire para a direita e para a esquerda, como se estivesse dançando, e acenda os LEDs da parte de baixo rotacionando nas cores do arco-íris.

```
no iniciar
  mostrar ícone [LEDs]
  definir LED para NeoPixel at pin P15 with 4 leds as RGB (GRB format)
  LED show rainbow from 1 to 360

sempre
  play melody perseguição until done
  play melody prelúdio until done

  every 500 ms
    LED rotate pixels by 1
    LED show

every 1000 ms
  motor esquerda direção Relógio velocidade 255
  motor direita direção Contra-relógio velocidade 255
  pausa (ms) 500
  motor esquerda direção Contra-relógio velocidade 255
  motor direita direção Relógio velocidade 255
  pausa (ms) 500
```

35 Semáforo-Carro



O carro semáforo é uma atividade para ensinar aos alunos o que significam as cores de um semáforo. Quando as luzes da parte de baixo do carro se acendem na cor verde, ele anda na velocidade máxima. Ao mudar para a cor amarela, a velocidade diminui e, quando as luzes se acendem na cor vermelha, o carro para de andar.

```
no iniciar
  mostrar ícone
  definir LED para NeoPixel at pin P15 with 4 leds as RGB (GRB format)

sempre
  motor todos direção Relógio velocidade 255
  LED show color green
  pausa (ms) 5000
  motor todos direção Relógio velocidade 25
  LED show color yellow
  pausa (ms) 2000
  motor todos direção Relógio velocidade 0
  LED show color red
  pausa (ms) 3000
```

36 Carro com Sensor Ultrassônico



Neste projeto foram juntados o carro colorido com um carro que se movimenta sozinho e desvia de obstáculos.

Para montar os blocos deste projeto é necessário adicionar as extensões **Neopixel** e **Maqueen**. O sensor ultrassônico do carro funciona da mesma forma que o sonar dos morcegos, emitindo sons imperceptíveis aos ouvidos humanos que, ao atingirem um objeto, são refletidos em forma de ecos. Esses ecos são captados pelo sensor e a distância é medida através do tempo de retorno e velocidade do som.

```
no iniciar
  mostrar ícone
  definir strip para NeoPixel at pin P15 with 4 leds as RGB (GRB format)
  strip show rainbow from 1 to 360

sempre
  strip rotate pixels by 1
  strip show
  pausa (ms) 100

sempre
  definir distancia para distância em unidades cm
  se < distancia < 20 e distancia ≠ 0 então
    se escolher de forma aleatória verdadeiro ou falso então
      motor esquerda direção Relógio velocidade 255
      motor direita direção Relógio velocidade 0
      pausa (ms) 600
    senão
      motor esquerda direção Relógio velocidade 0
      motor direita direção Relógio velocidade 255
      pausa (ms) 600
  senão
    motor todos direção Relógio velocidade 255
```

37 Carro Segue Faixa



O carro segue faixa identifica as cores branca e preta por meio de sensores localizados na parte debaixo do carrinho.

Além da atividade de montagem do código, os alunos também terão que criar um circuito com uma fita isolante ou tinta preta em um fundo branco. Para montar os blocos desta atividade é necessário adicionar a extensão Maqueen.

```
no iniciar
  mostrar ícone [ícone de carrinho]

sempre
  se ler batedor esquerda = 0 e ler batedor direita = 0 então
    motor todos direção Relógio velocidade 150
  senão se ler batedor esquerda = 0 e ler batedor direita = 1 então
    motor direita direção Relógio velocidade 150
    motor esquerda direção Relógio velocidade 0
  senão se ler batedor esquerda = 1 e ler batedor direita = 0 então
    motor direita direção Relógio velocidade 0
    motor esquerda direção Relógio velocidade 150
```


38 Controle do carro



Este projeto é o controle do carro que será guiado de forma remota. O controle se dá por sinais de rádio entre os micro:bits.

Quando o carro for ligado, o micro:bit programado como controle deve estar na posição horizontal nas mãos do condutor. Para que o carro vá para a frente, o aluno deve inclinar a placa para a frente. Ao inclinar o micro:bit para trás, o carro irá para trás. Para que o carro vire, basta inclinar o micro:bit para a esquerda ou para a direita.

Quando os botões **A** e **B** forem pressionados ao mesmo tempo, o carro para.

```
no iniciar
  definir grupo do rádio 3
  mostrar número 3

no botão A+B pressionado
  rádio envia cadeia de caracteres "pare"

sempre
  rádio envia value "x" = aceleração (mg) x
  rádio envia value "y" = aceleração (mg) y
```

39 Carro guiado por controle remoto



Para o projeto do carro, o micro:bit que estará acoplado ao Maqueen, receberá os sinais do controle para se mover. Este micro:bit deve estar no mesmo grupo de rádio do controle para que o carro funcione.

Para facilitar a identificação, o número do rádio será mostrado nos LEDs do controle e do micro:bit encaixado no Maqueen. Também serão acesos os LEDs que estão na parte de baixo do Maqueen, na cor verde.

```
no iniciar
  definir strip para NeoPixel at pin P15 with 4 leds as RGB (GRB format)
  strip show color green
  definir grupo do rádio 3
  mostrar número 3

ao receber rádio receivedString
  se receivedString = "pare" então
    motor todos direção Relógio velocidade 0
```

Continua na próxima página.

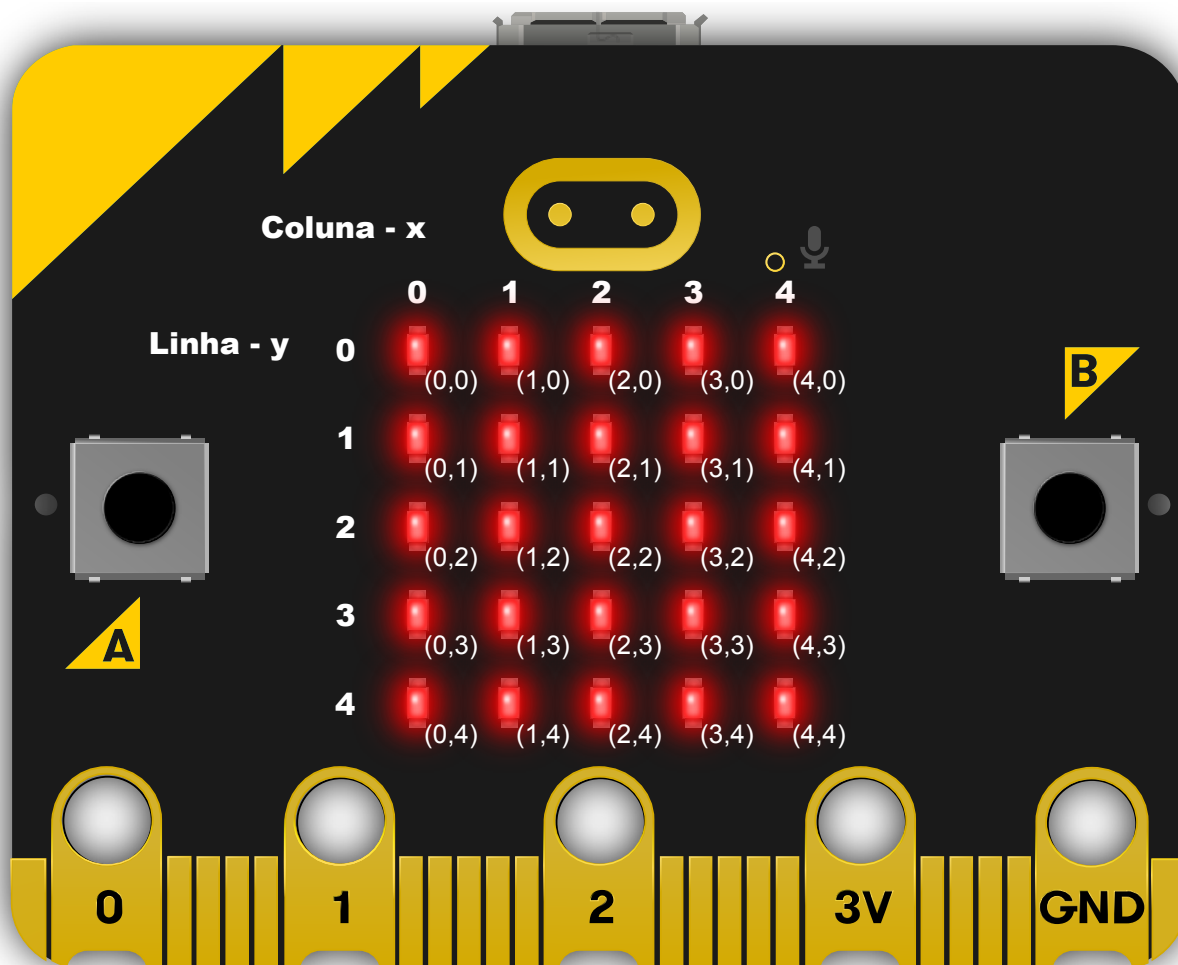


Continuação da página anterior.

```
ao receber rádio name value
se name = "x" então
se value > 255 então
motor esquerda direção Relógio velocidade 255
motor direita direção Relógio velocidade 0
+
se value < -255 então
motor esquerda direção Relógio velocidade 0
motor direita direção Relógio velocidade 255
+
+
se name = "y" então
se value < -255 então
motor todos direção Relógio velocidade 255
+
se value > 255 então
motor todos direção Contra-relógio velocidade 255
+
+
```

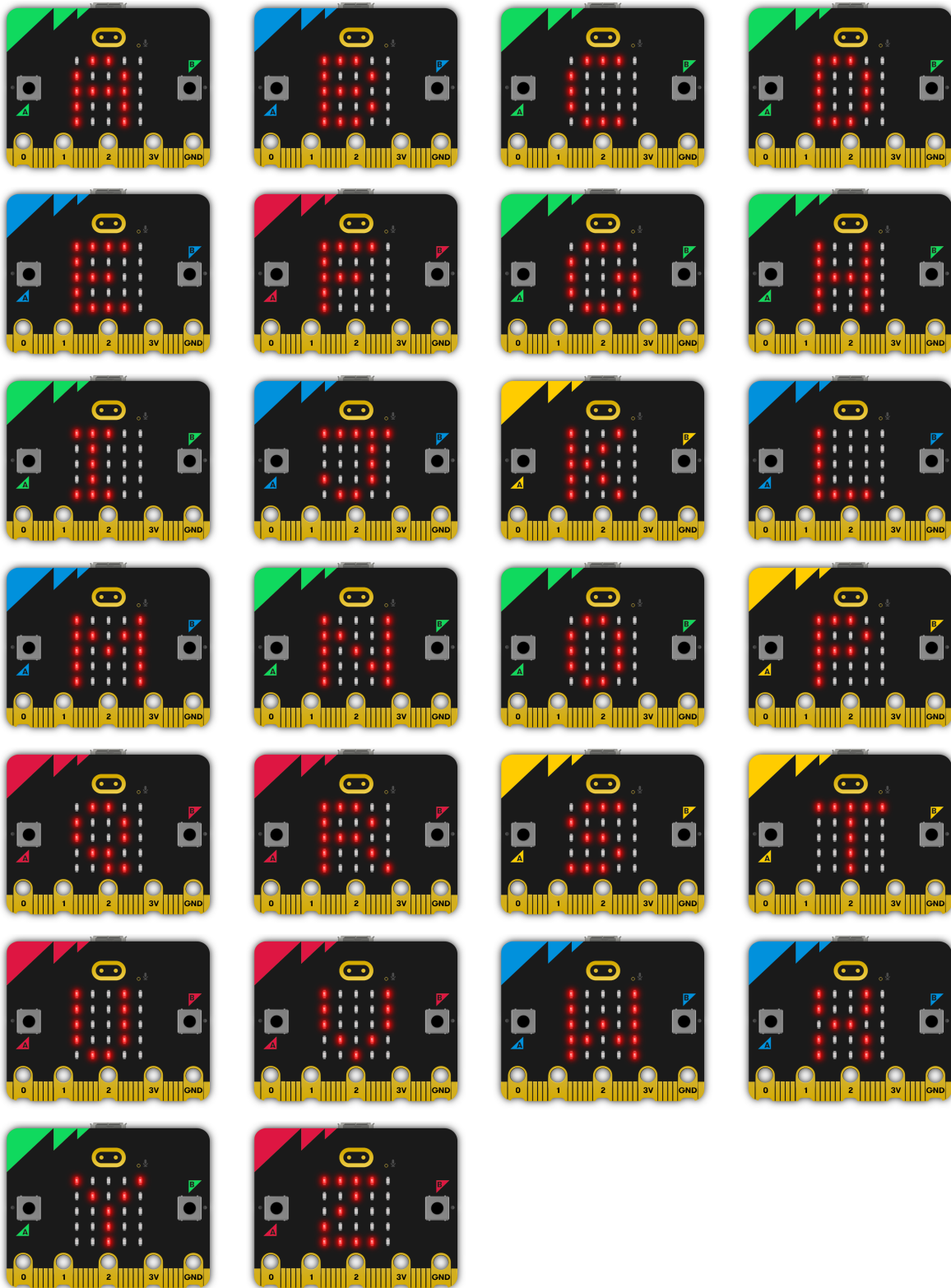
Apêndice A Posição matricial dos LEDs

A figura abaixo é um apoio para o projeto [Letras](#).



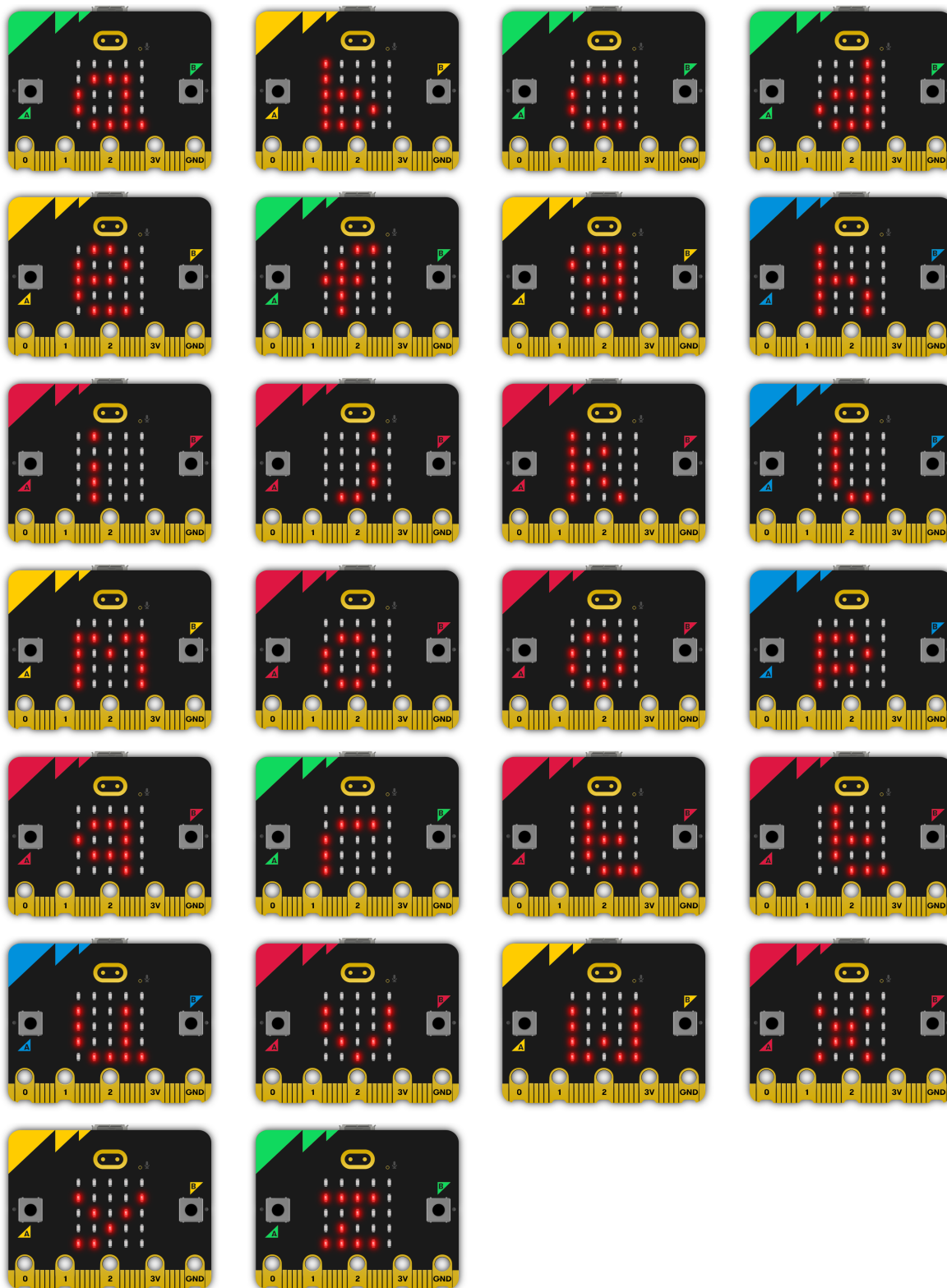
Apêndice B Letras Maiúsculas

As figuras abaixo servem como referência para desenhar os LEDs que serão acesos para formar as letras maiúsculas.



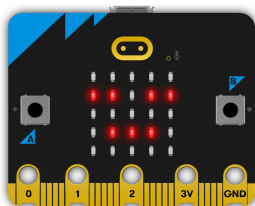
Apêndice C Letras Minúsculas

As figuras abaixo servem como referência para desenhar os LEDs que serão acesos para formar as letras minúsculas.

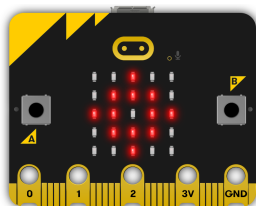


Apêndice D Ícones disponíveis no micro:bit

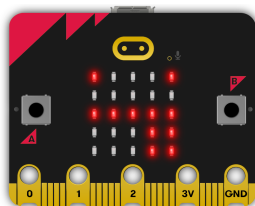
As figuras abaixo servem como referência para desenhar os LEDs que serão acesos para formar ícones.



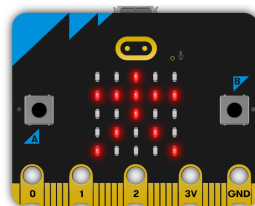
adormecido



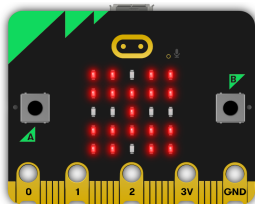
alvo



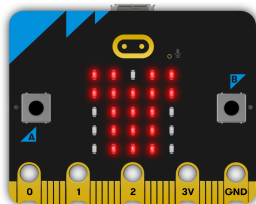
bobo



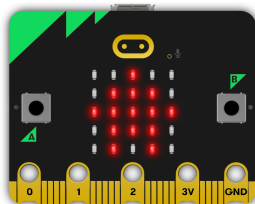
boneco palito



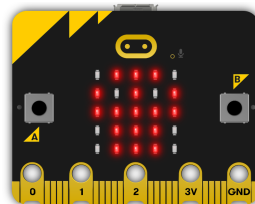
borboleta



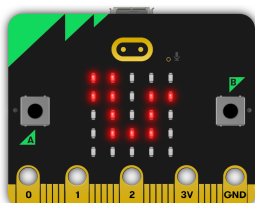
camiseta



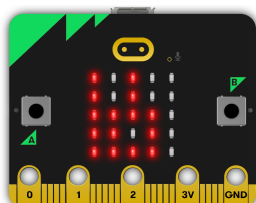
casa



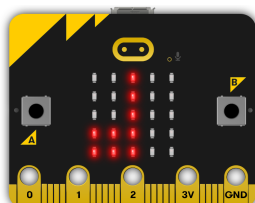
caveira



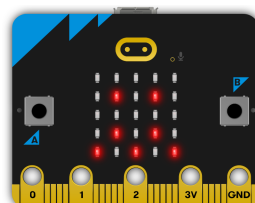
cobra



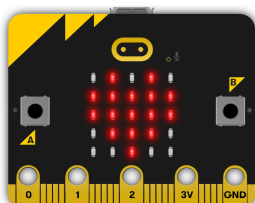
coelho



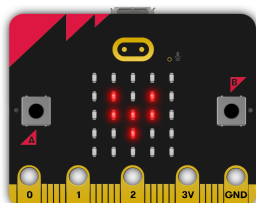
semínima



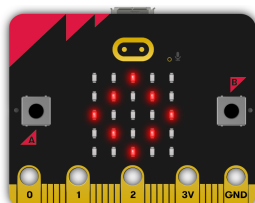
confuso



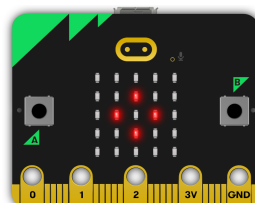
coração



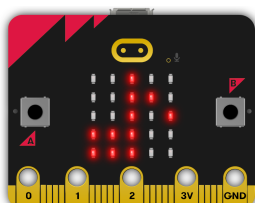
coração pequeno



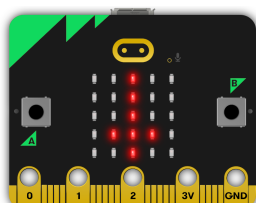
diamante



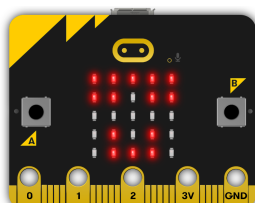
diamante pequeno



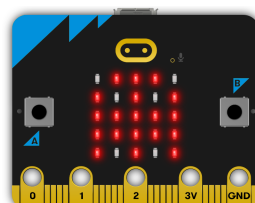
colcheia



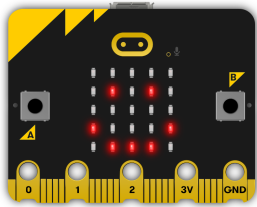
espada



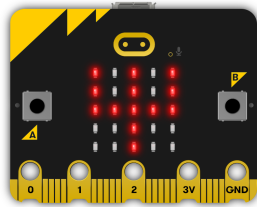
fabuloso



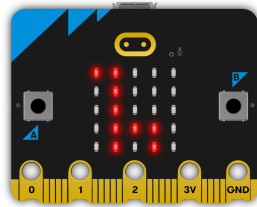
fantasma



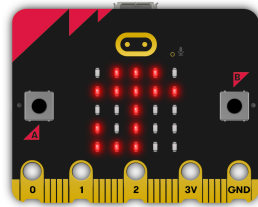
feliz



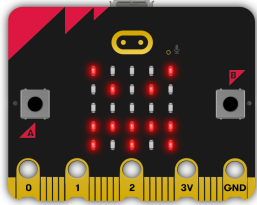
forquilha



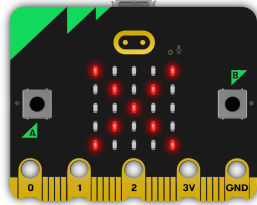
girafa



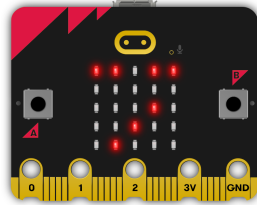
guarda-chuva



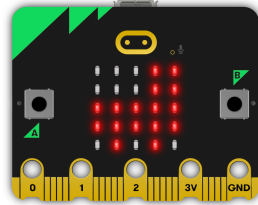
irritado



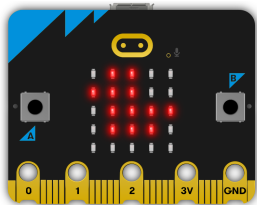
não



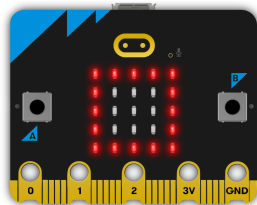
neh



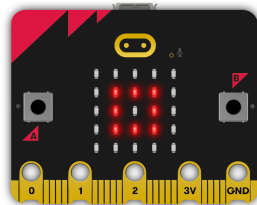
patins



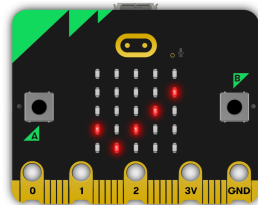
pato



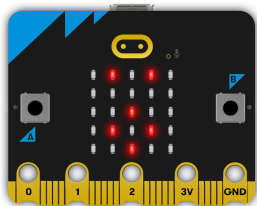
quadrado



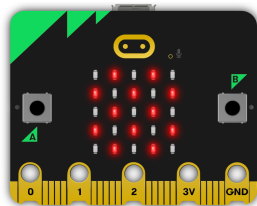
quadrado pequeno



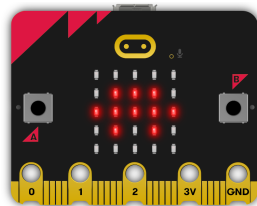
sim



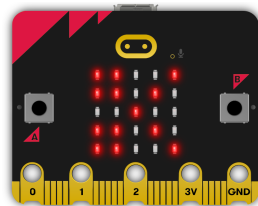
surpreso



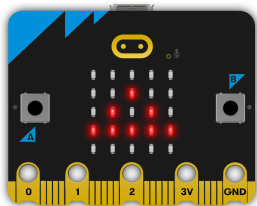
tabuleiro de xadrez



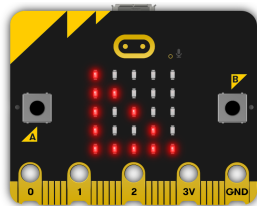
tartaruga



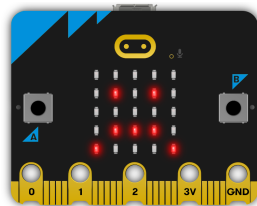
tesoura



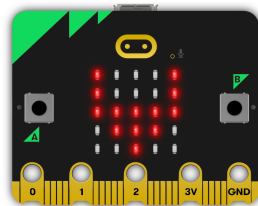
triângulo



triângulo à esquerda



triste



vaca

Apêndice E Música

Qual é a música?

C D E G G E F F C D E G G F E

C C D E G G F E C F F E D D E D D C C

Abaixo estão as notas musicais na clave de Sol.

G Grave A Grave B Grave C Médio D Médio E Médio F Médio G Médio A Médio B Médio C Agudo D Agudo E Agudo

Sol Lá Si Dó Ré Mi Fá Sol Lá Si Dó Ré Mi

Na figura abaixo, temos a representação das figuras rítmicas e o tempo de cada uma delas.

Semibreve Mínima Semínima Colcheia Semicolcheia Fusa

1 - batida 1/2 - batida 1/4 - batida 1/8 - batida 1/16 - batida 1/32 - batida

play tone C Médio for 1 batida until done

Na figura abaixo, vemos a representação das pausas na partitura, isto é, quando nenhuma nota é tocada, e o tempo de cada uma delas.

1 - batida 1/2 - batida 1/4 - batida 1/8 - batida 1/16 - batida 1/32 - batida

rest for 1 batida

Apêndice F Jogo da cobra e da escada

Use o projeto do dado no micro:bit, em qualquer versão, para jogar o jogo da cobra e da escada.

Regras do jogo: quando cair na casa em que há o começo de uma escada, suba-a e avance para a casa ao final da escada; quando cair na casa em que há a cabeça de uma cobra, desça pelo corpo dela até a casa onde termina o rabo. Para ganhar o jogo, o dado precisa sortear exatamente o número de casas que faltam para chegar no mascote do TRUEM. Se o jogador estiver na casa 98 e o número sorteado for 3, ele deve andar até a última casa e voltar uma.

