

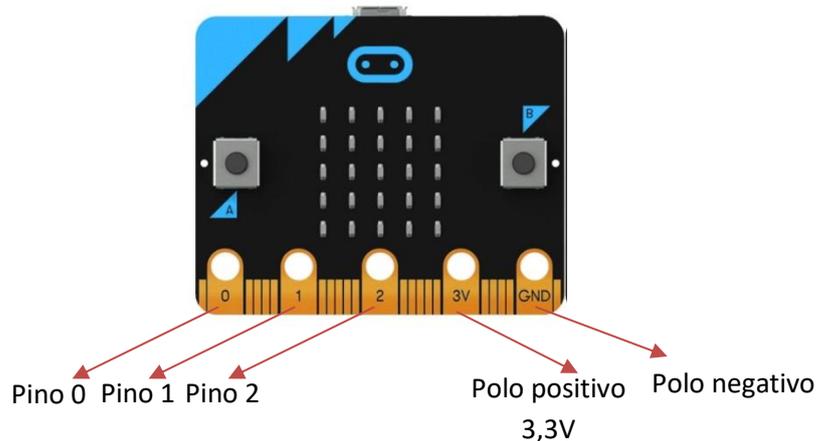


☺ Começa por **aceder** a <https://makecode.microbit.org/>

☺ Clica em  para **começares** a **programar** o **micro:bit**.

☺ Cria um **novo projeto** de nome **Servo motor**.

☺ De **seguida** recorda-se o **PINOUT** do **micro:bit**:



Os **pinos 0, 1 e 2** podem ser **programados** como **saída** (escrita) ou **entrada** (leitura). Podem ainda ter comportamento **analógico** ou **digital**.

☺ Neste guião vais ainda utilizar um **Servo motor SG90**.



Este **componente** é um **motor de ângulo** que opera entre **0º** e **180º**. Para já **não te preocupes** que ele vai **aparecer** junto ao **Micro:bit** quando o **começares** a **programar**.

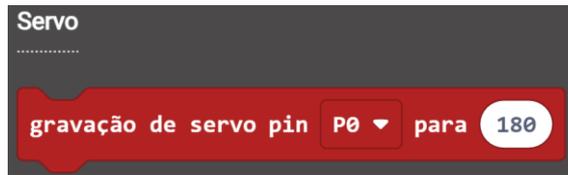
☺ Vamos então, **programar** o **micro:bit** para **controlar** o **motor servo**. Para isso **adiciona** os **seguintes** blocos ao **micro:bit**:



Acede agora à categoria **Avançado**  **Avançado** dos **blocos** e procura por **PINS** 



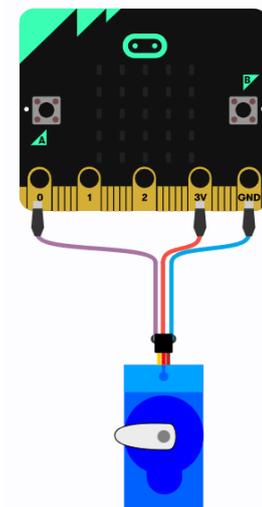
Faz **scroll** até ao **final** e adiciona o **bloco** da **secção Servo** seguinte:



Encaixa dentro do bloco **Botão A** e ficarás com algo do **género**:



- Se **reparares**, na imagem do **lado esquerdo** apareceu automaticamente o **motor servo** e já com as **ligações estabelecidas**.



- Altera o **código** para o **seguinte**:



Simula e verifica que ao clicar no **botão A e B** o **motor mexe-se** com uma **diferença de 90º**.

Sabes o que faz **lembrar** este **movimento**? Uma **cancela** de um **parque de estacionamento**!

E se **tentássemos fazer** uma na **vida real**?



- ☺ Começa por **estabelecer as ligações** entre o teu **Micro:bit físico** e o **motor servo** que o teu professor te forneceu. Deixo-te o **pinout do motor servo**:

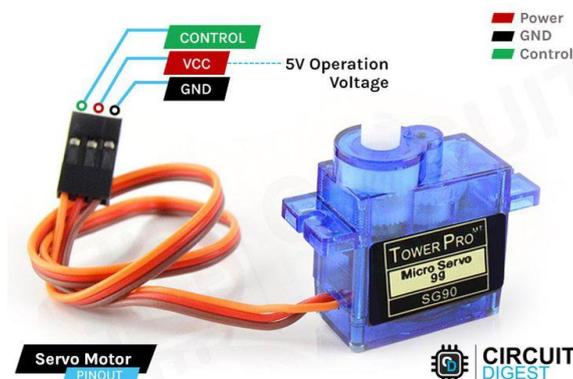
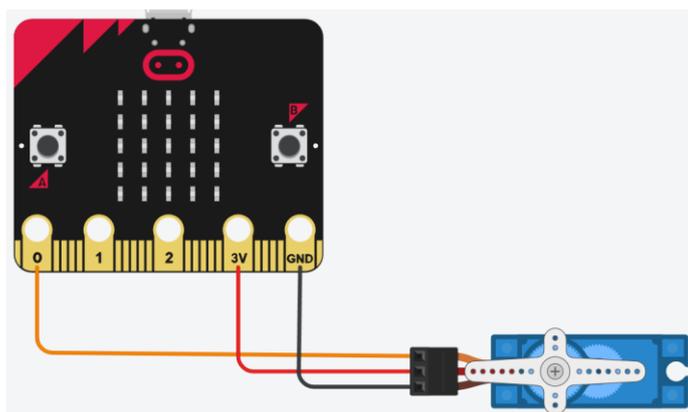


Imagem retirada de circuitdigest.com

Apesar de **indicar** que o **motor** opera a **5V** ele **funciona** também com **3V** pelo que **podes** **ligar diretamente** ao **Arduino** (esta forma simplifica o circuito). **Estabelece** as **seguintes ligações físicas** entre os **Micro:bit** e o **motor Servo**:



- ☺ **Transfere** o ficheiro  e **vê o que acontece** quando **pressionas** os **botões A** ou **B**.
- ☺ Neste **guião** pretende-se **fazer** um **parque** de **estacionamento**. Começa por **preparar** o teu **código** para que seja um **parque** de **estacionamento** com **5 lugares**. Terás, por isso de ter uma **variável** que **conte** as **vezes** que a **cancela abre**. Cria uma **variável** de nome **contador**:

Nome da nova variável: 

Ok 



- ☐ Ao arrancar o **Micro:bit** vamos **assumir** que **não há carros** dentro do **parque** pelo que a **variável deve** estar a **zero**. Garante que o **contador** está a **zero** no **início** do **programa**:



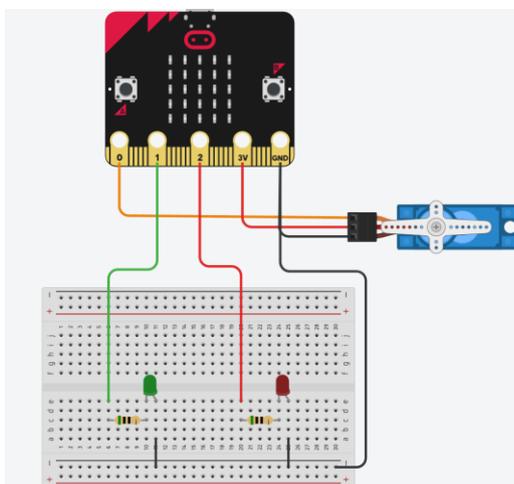
- ☐ Garante agora, que sempre que o **botão B** é **pressionado** (a cancela abre e entra um carro) o **contador incrementa** uma **unidade**.

Cria uma **condição** (bloco se) que **teste** se a **variável** é **inferior** a **5** antes de **abrir** a **cancela**. Apenas no **caso** do **contador** ser **inferior** ou **igual** a **5**, é que **permite** **abrir** cancela. Caso **contrário** a **cancela** **deixa** de **abrir** (como se o botão deixasse de funcionar).

Utiliza os seguintes blocos **corretamente** para **conseguir** a **condição** pretendida:



- ☐ Vamos ainda **aproveitar** os **pinos 1** e **2** para **indicar** se o **parque** está **livre** ou **cheio**. Faz **gravação** no **pino digital 1** e **2** onde irás **ligar** um **LED verde** e **vermelho**, **respetivamente**. **Exemplo** de onde serão **ligados** **fisicamente** os **LEDs verde** e **vermelho**.

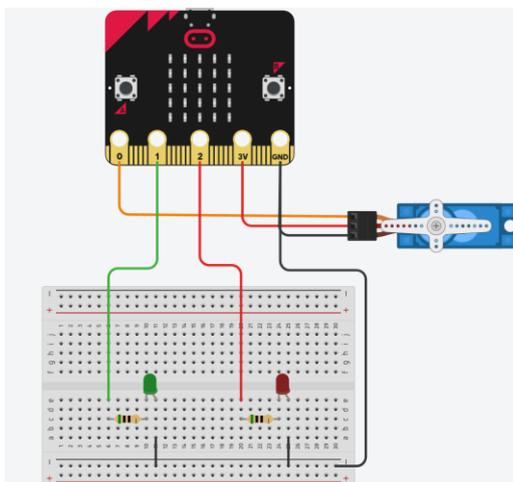


- ☐ **Adiciona** ao **início** forma de **ligar** o **LED verde** e **desligar** o **vermelho**, uma vez que consideramos o **parque vazio** ao **arrancar** o **Micro:bit**.



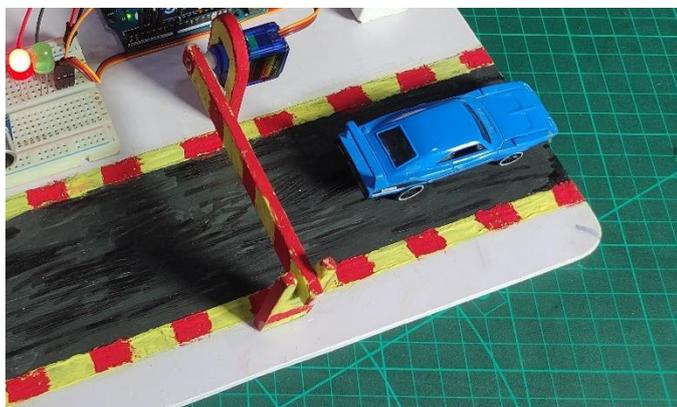


- ☺ O resto é contigo. Faz o código para que o **parque funcione com o LED verde a manter-se ligado até aos 4 carros** e quando abrir para o **quinto carro o LED verde apague e liga-se o vermelho**. No final estabelece as **ligações físicas**, como mostra a figura seguinte e **envia o programa para o Micro:bit físico**.



Atenção: devido a dispormos apenas de 3 pinos de controlo e já estarmos a utilizar os 3 pinos, não vamos poder, para já, retirar carros do parque (necessitávamos de mais um servo motor e, por isso, outra ligação ao micro:bit)

- ☺ Cria **fisicamente** uma **cancela** e um **semáforo** para este **projeto**. Deixo-te um exemplo:



- ☺ **PRO:** faz com que apenas **um pino** do **Microbit** permita **ligar o LED vermelho** ou **verde** (dica: **troca de polaridade**). Desta forma, **libertas um pino** para poderes **controlar outro servo motor** e assim fazer a **cancela de saída**. Também **terás de alterar o código**, uma vez que os **botões A e B** apenas servirão para **abrir as cancelas de entrada e saída**. Assim, **terás de ter um contador de tempo**, que **após abrir a cancela** (entrada ou saída) ao fim de **5 segundos** a fecha automaticamente.
- ☺ **Chama o teu professor para avaliar.**