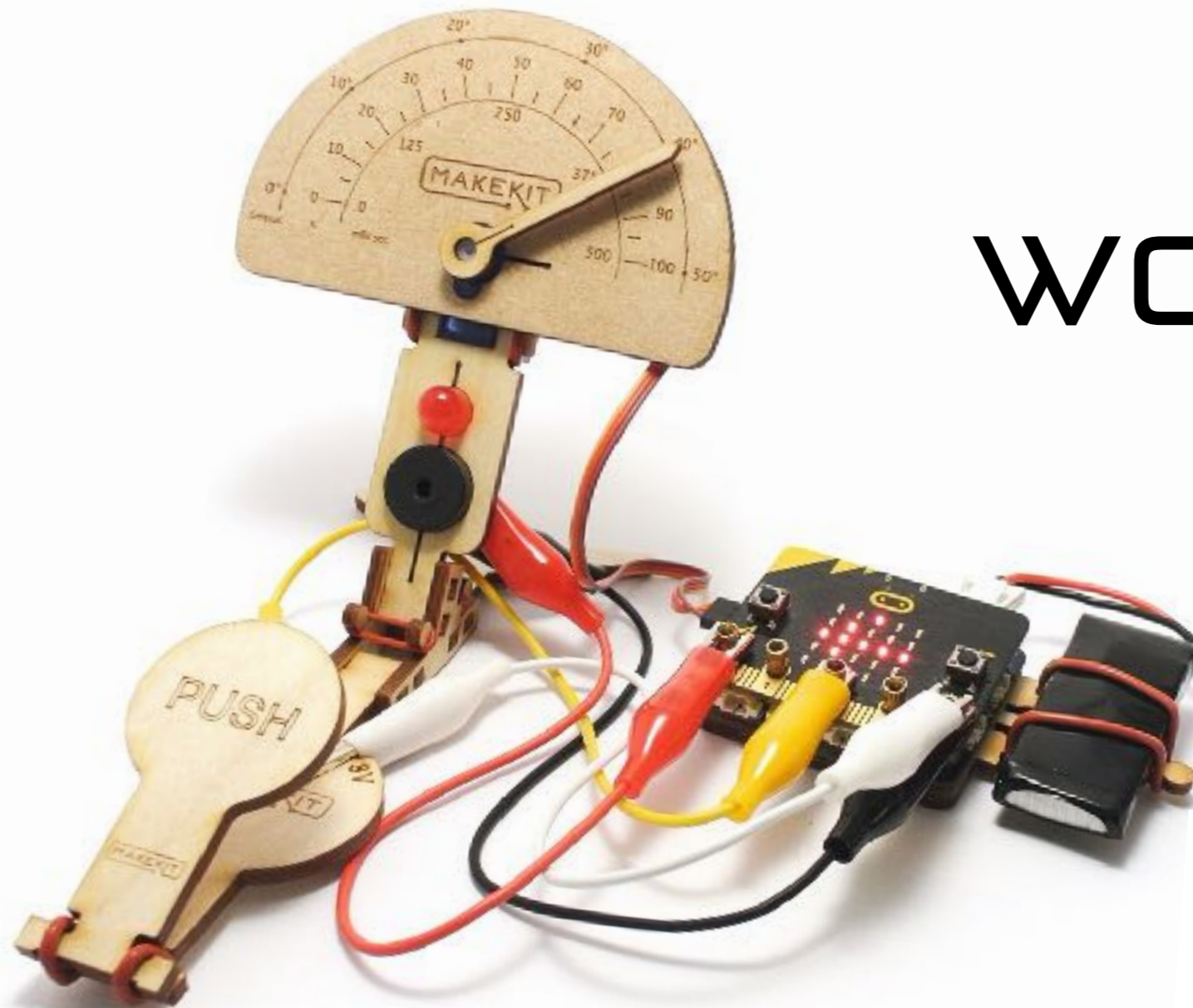




WONDER:KIT

EXPLORER + REACTIONBIT



Introduction

Wonder:kit explorer is the basic package in the wonder: kit concept and will be used together with a micro:bit.

It provides basic experience with physical components such as LED, servo (movable arm) and buzzer (speaker), and connects these to micro:bit sensors.

Reaction:bit is a reaction time tester based on a switch, buzzer and code.

In addition to this guide, there is an inventor concept where students will create their own inventions.

Before you get started, you should know some basic uses of the micro:bit:

- Easy coding with makecode (makecode.microbit.org)
- Simple codes, we recommend "Flashing Heart" and "Rock, paper, scissors" as a minimum
- Connection and transfer of the code to the micro:bit

More about this can be found in our guide **Introduction to micro:bit**



Content

Content	Page	Difficulty
Charging the battery	4	Easy
Explorer: Assembly	6	Easy
Code tasks with LED and buzzer	13	Easy
Servo engine	25	Medium
Reaction:bit - reaction tester	33	Advanced
Extension to reaction tests	49	Advanced

Charging

Plug the battery into the adapter cable. Red light comes on.
When charging is complete after about 1 hour, the connector lights up green.



Charge faster

Set the charger to charge twice as fast (from two to one hour to charge empty battery)

Press **Select**. "Current 0.5A" is displayed.
Press **Setting** 5 times. "Current 1A" is displayed
Hold **Setting** until "Save setting" appears.



Assembly EXPLORER

Parts



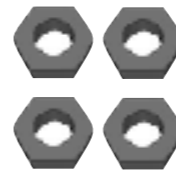
4 pcs nyonscrew m3x12



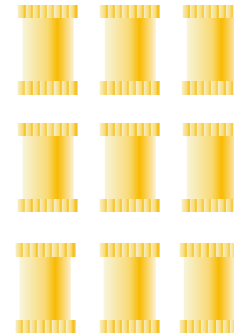
1 pcs
Nyonscrew
m3x8



2 pcs
Nyonscrew
Countersunk
M3



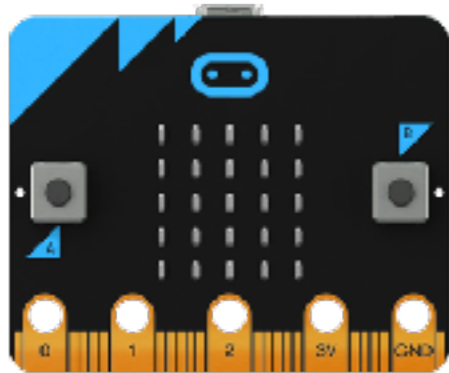
4 pcs nylonnuts
m3



9 pcs
knurled barrel
nuts



2 pcs
crocodile clips
In different colors



micro:bit



Control card



Buzzer



LED

Tools needed:

- Small Phillips screwdriver
- Scissor or wallpaper knife (for cutting aluminum foil for first time installation for the reactionbit)

Screws Part 1

Tool: Phillips screwdriver

Parts:



4 stk nyonscrew m3x12



1 pcs
Control Card



4 pcs
knurled barrel
nuts



Tighten the barrel nuts with the nylon screws.
They must be screwed "finger tight":

- Tight enough to make live contact with the control board,
- Not so hard that the screw or control board is damaged.

NOTE! Check that the nuts are tight so that they have good contact with the control board. They must conduct power to the micro:bit.

Screws Part 2

Tool: Phillips screwdriver , and a socket wrench

Parts:



2 pcs
Nylonscrews
Countersunk



2 pcs nylon
nuts m3



Control Card with screws



Fasten the nylon screws with nuts as shown in the picture.

Screws Part 3

Tool: Phillips screwdriver

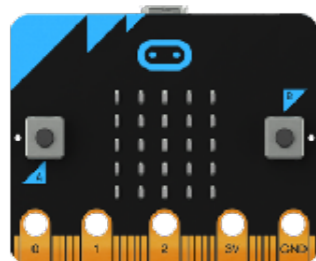
Parts:



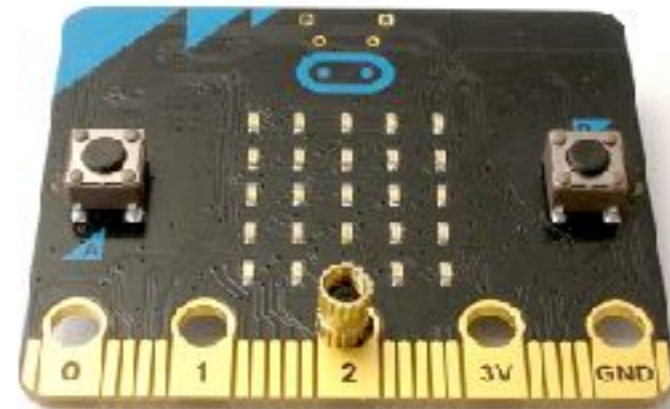
1 pcs
Nylonscrew
m3x8



1 pcs
knurled barrel
nuts



micro:bit

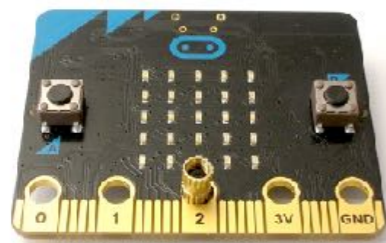


Pull the screw up through P2 and screw the barrel nut on the top

Screws Part 4

Tool:

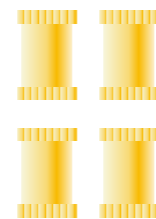
Parts:



micro:bit



Control card



4 pcs
knurled barrel
nuts



- Place the micro:bit over the control card so that the screws go through the holes. The screen should point up.
- Tighten the four barrel nuts
- Make sure they are tight enough.

NOTE! Check that the nuts are tight so that they have good contact with the control board. They must conduct power to micro:bit.

Code tasks

EXPLORER

Light indicator, LED

An LED is a diode that emits light.

Diodes have the property that the current can only go one way. Therefore, we must give them power in the right direction for it to shine.



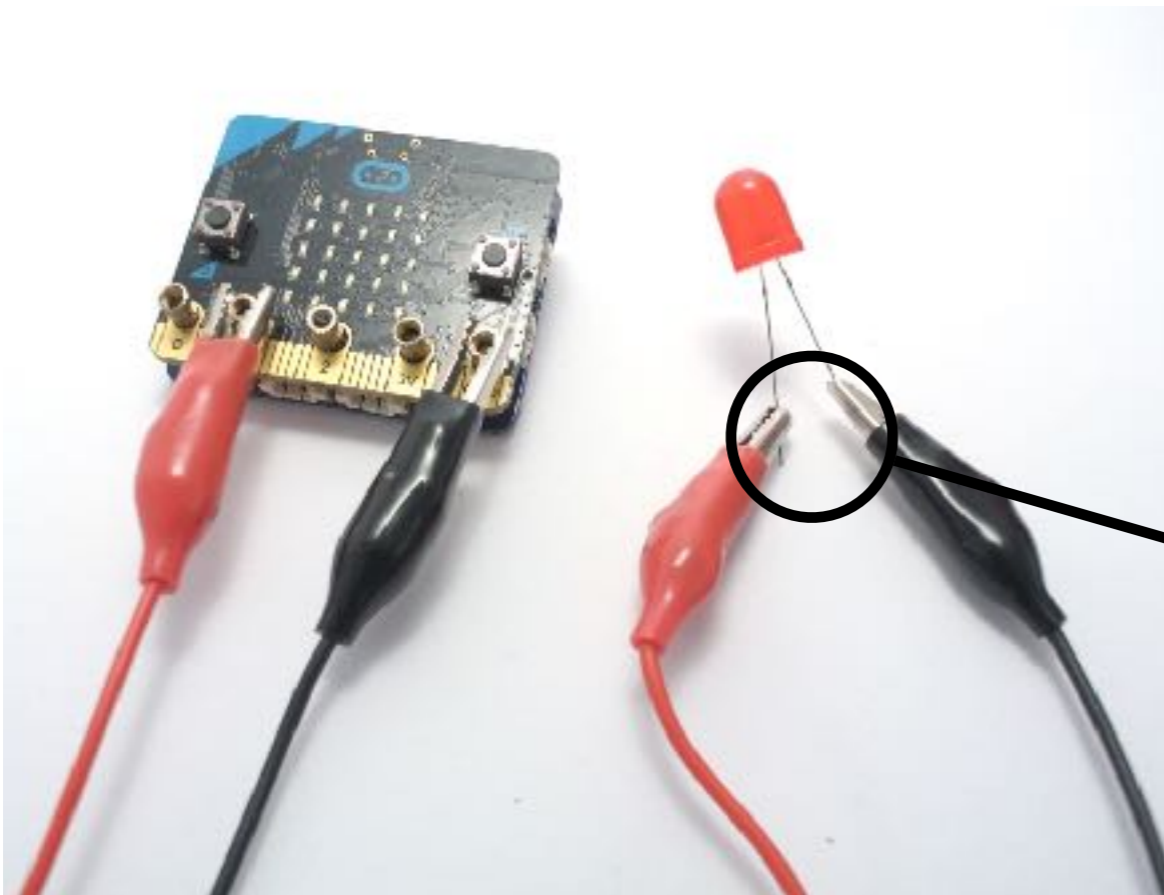
You need:

Two crocodile clips, wonder:bit + micro:bit device, LED, USB cable



Make a light switch

Connect the LED with two crocodile clamps as shown in the picture. The long leg goes to P1 and the short leg to GND (ground). Long leg means positive, and we will send positive current from P1. Make sure that the metal on the two crocodile clips does not come into contact with each other. Then we get a short circuit, which can damage the micro: bit.



"Long legs is a positive thing"

These must not touch each other as it may cause a short circuit



Code the light switch

Grab two “on button A” and change one to “on button B”

Go down to Advanced - Pins and select digital write 0 in “on button A” and digital write 1 in “on button B”

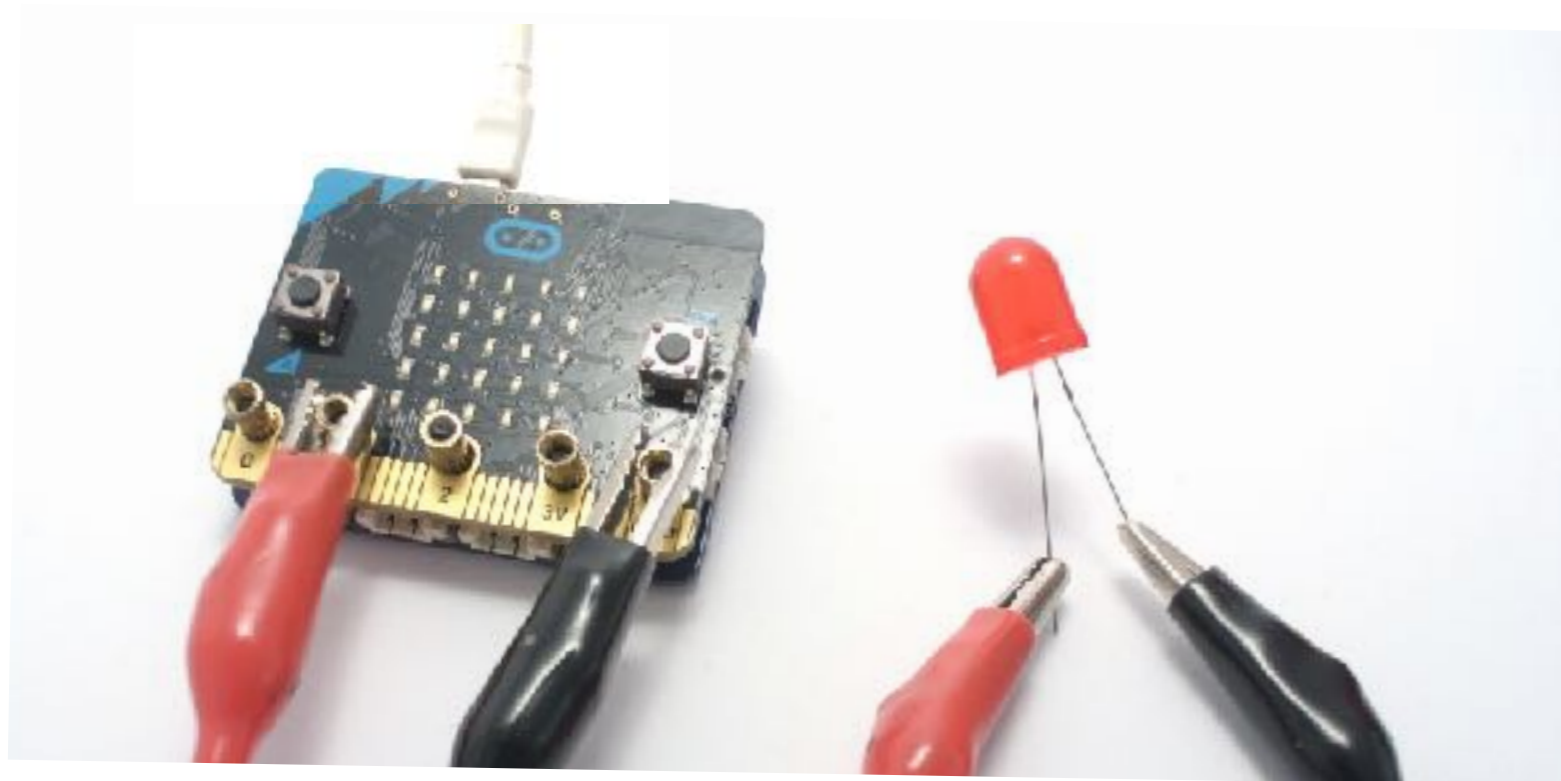
The image shows a block-based programming environment for an Arduino project. The interface includes a top bar with 'Share', 'Blocks', and 'JavaScript' tabs. On the left is a sidebar with a category menu (Loops, Logic, Variables, Math, Advanced, Functions, Arrays, Text, Game, Images, Pins, more) and a visual representation of an Arduino board with pins labeled 'B' and 'GND'. The main workspace contains the following code blocks:

- on start** (blue block)
- forever** (blue loop block)
- on button A pressed** (pink block) containing a **digital write pin P1 to 0** (red block).
- on button B pressed** (pink block) containing a **digital write pin P1 to 1** (red block).

An inset box on the right provides a detailed view of the 'on button A pressed' block. It shows a dropdown menu for 'A' with options 'A', 'B', and 'A+B'. The 'digital write pin P1 to 1' block is also visible within this inset.

Test the light switch

Download the code and press both A and B in turn.



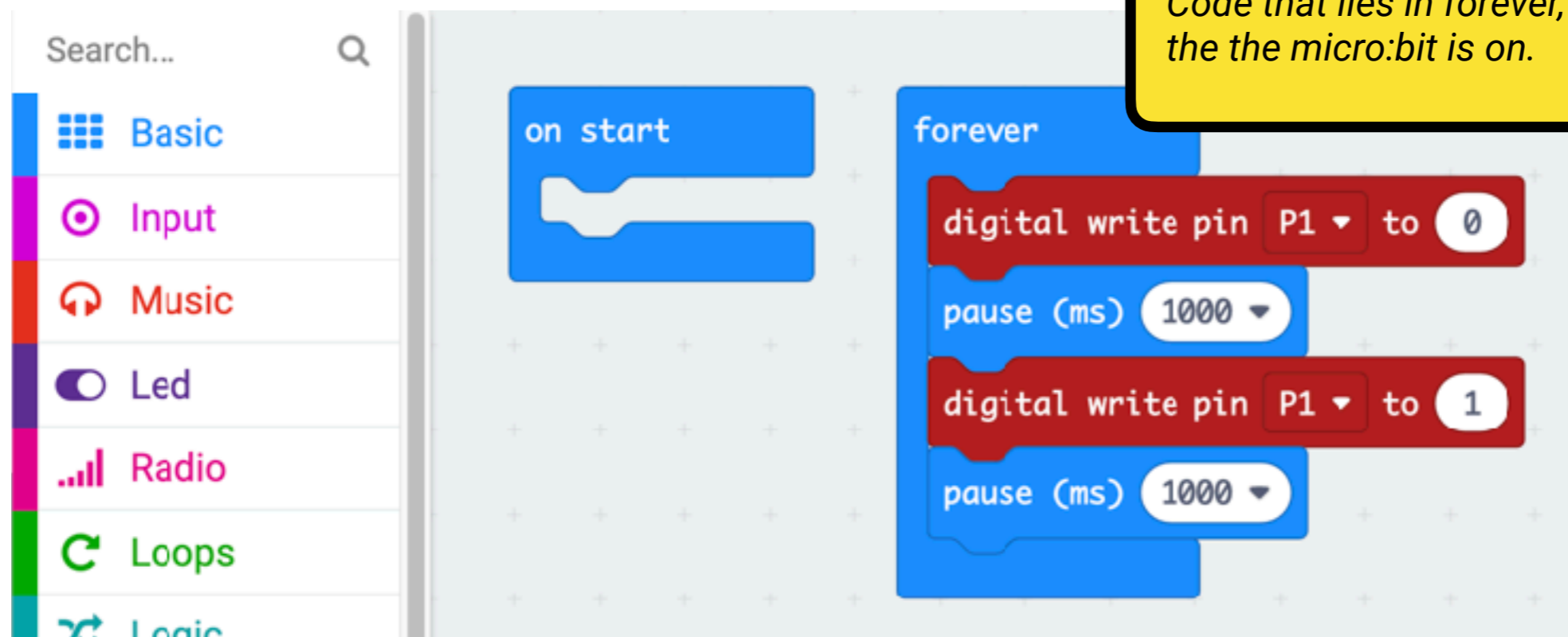
What happens?

When we press the B button, we turn on the power that goes out of P1. This is enough to make the LED light up. When we press A, it goes out again.

Code a turn signal

We can make our LED flash automatically by using the forever loop and pause function. First we turn off the power. Then we wait a second (1000 milliseconds). Then we turn on the power, and wait another second, before the code repeats itself automatically.

Download the code and see what happens.

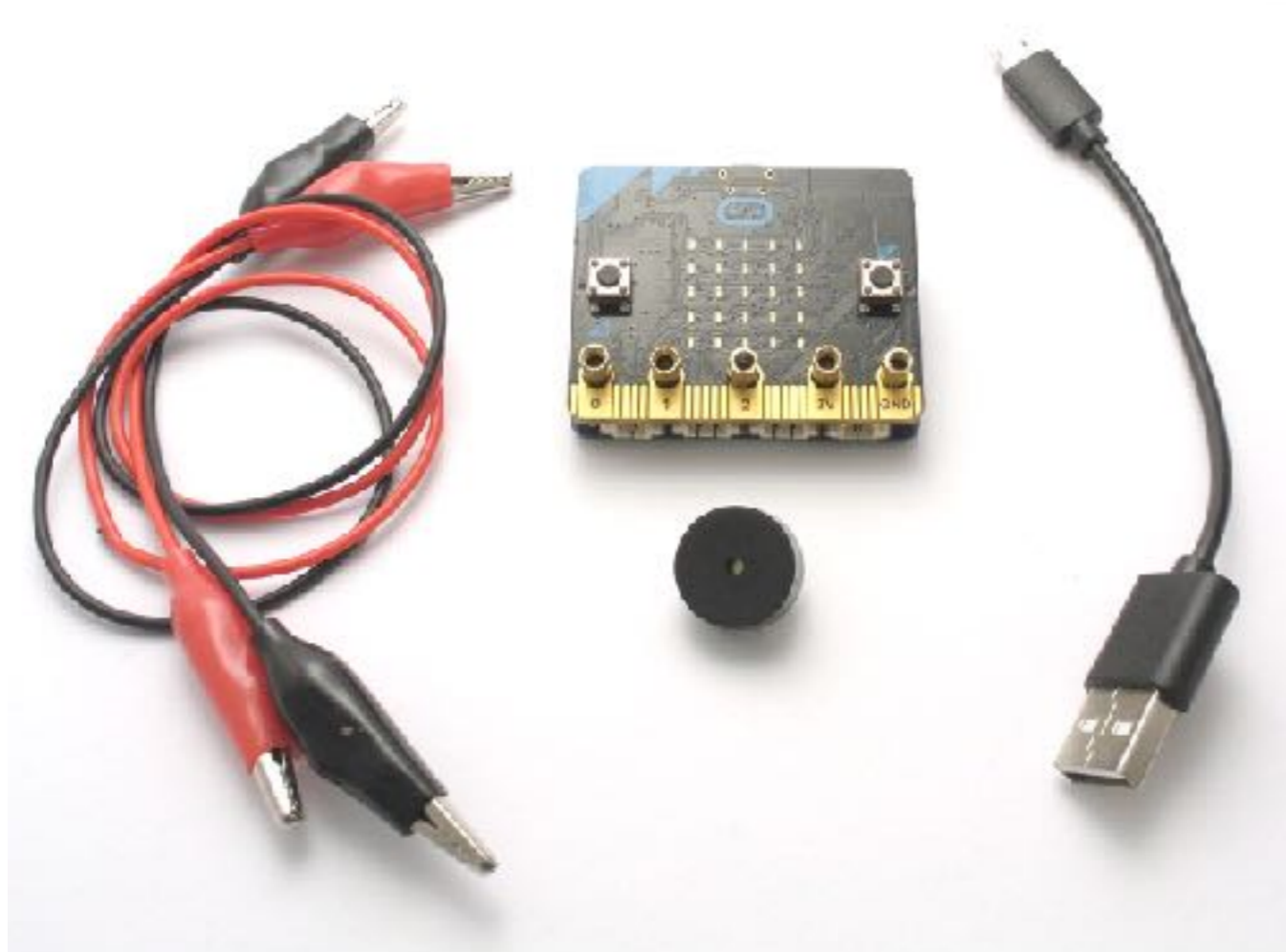


Try this:

Change the pauses to ex. 500 ms or 2000 ms, and see what happens.

Buzzer (høytaler)

En buzzer er en slags høytaler. Den omformer elektriske signaler til hørbare lydbølger.

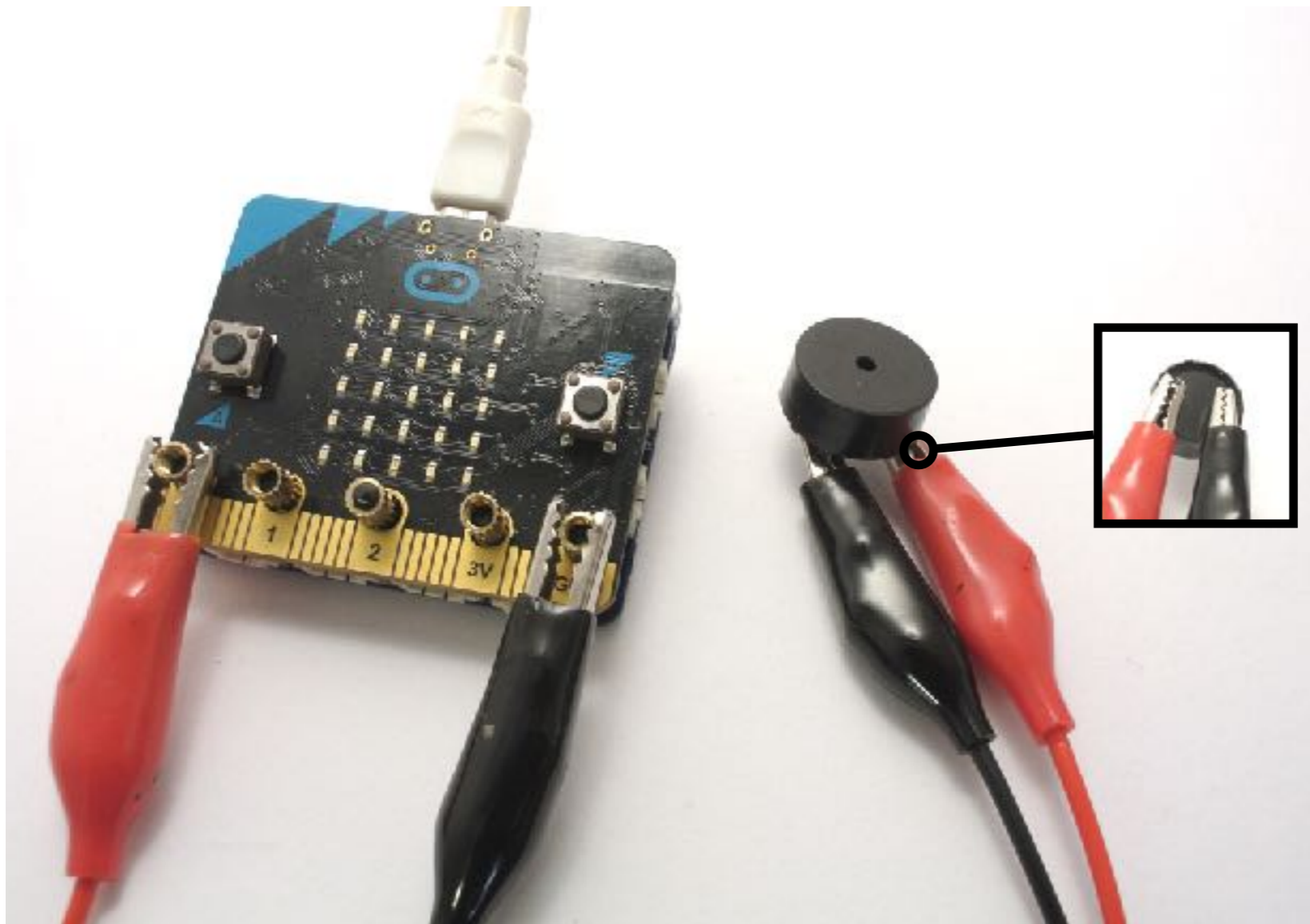


Du trenger:

To krokodilleklemmer, wonder:bit+micro:bit enheten, lysdiode, USB-ledning

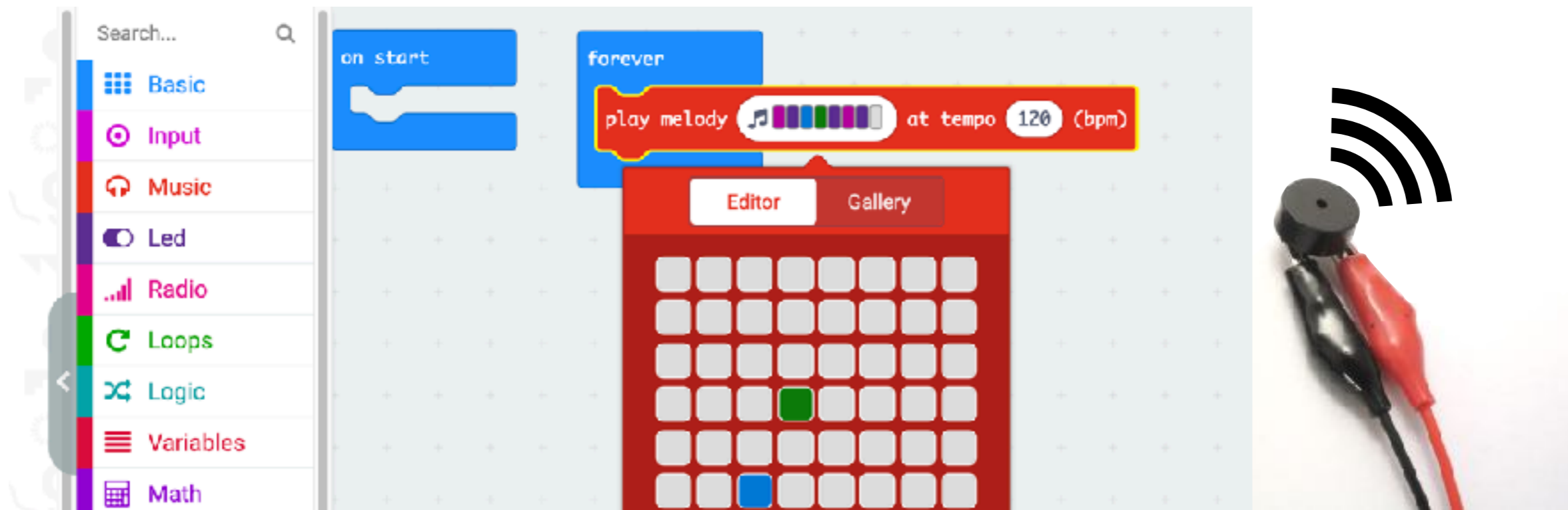
Lyd med buzzeren

Koble buzzeren til P0 og GND (jord) som vist på bildet.



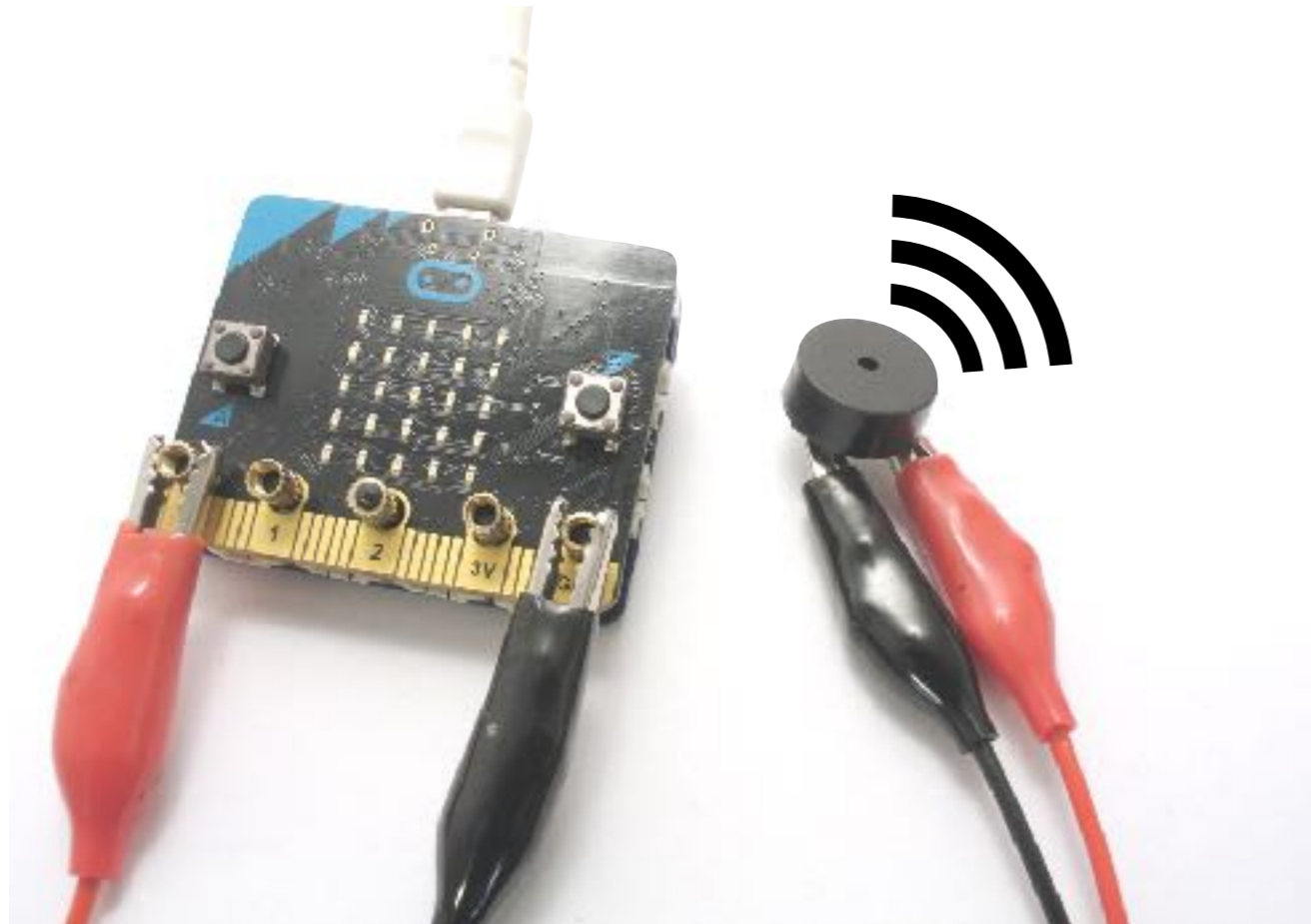
Kode buzzeren

Fra music-kategorien henter du ut en “play melody”. Klikk i melodien og komponer din egen sang, eller velg en ferdig i galleriet. Denne legger du i forever-løkken.



Test buzzeren

Last over koden og se hva som skjer



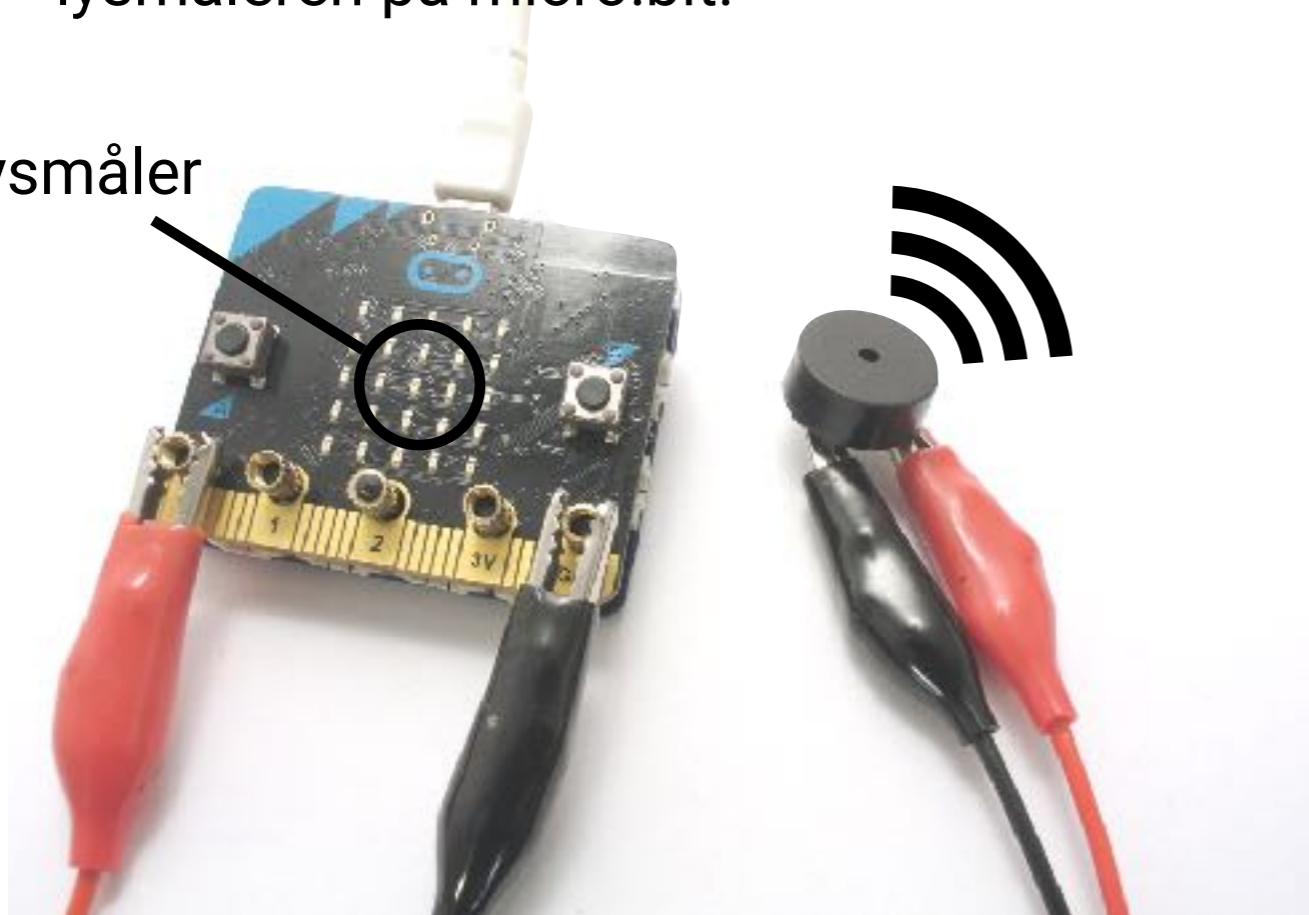
Hva skjer?

Når vi bruker “play melody”, sendes elektriske signaler ut fra gjennom P0 og tilbake til jord. Signalene gjøres om til lydbølger i buzzeren, som oppfattes som ulike toner i øret.

Lag en theremin

En theremin er det eldste elektroniske instrumentet, og kan spilles ved kun ved å bevege hendene over en føler. For å måle håndbevegelser, kan vi bruke lysmåleren på micro:bit.

Lysmåler



Lysmåleren gir oss et tall mellom 0 og 255. Jo mer lys, jo høyere tall.



*Russeren Leon Theremin
oppfant thereminen i 1919
(Wikimedia commons)*

Try the thermine

Move your hand up and down, or from side to side over the micro:bit light meter. What happens to the tone?

Things to try:

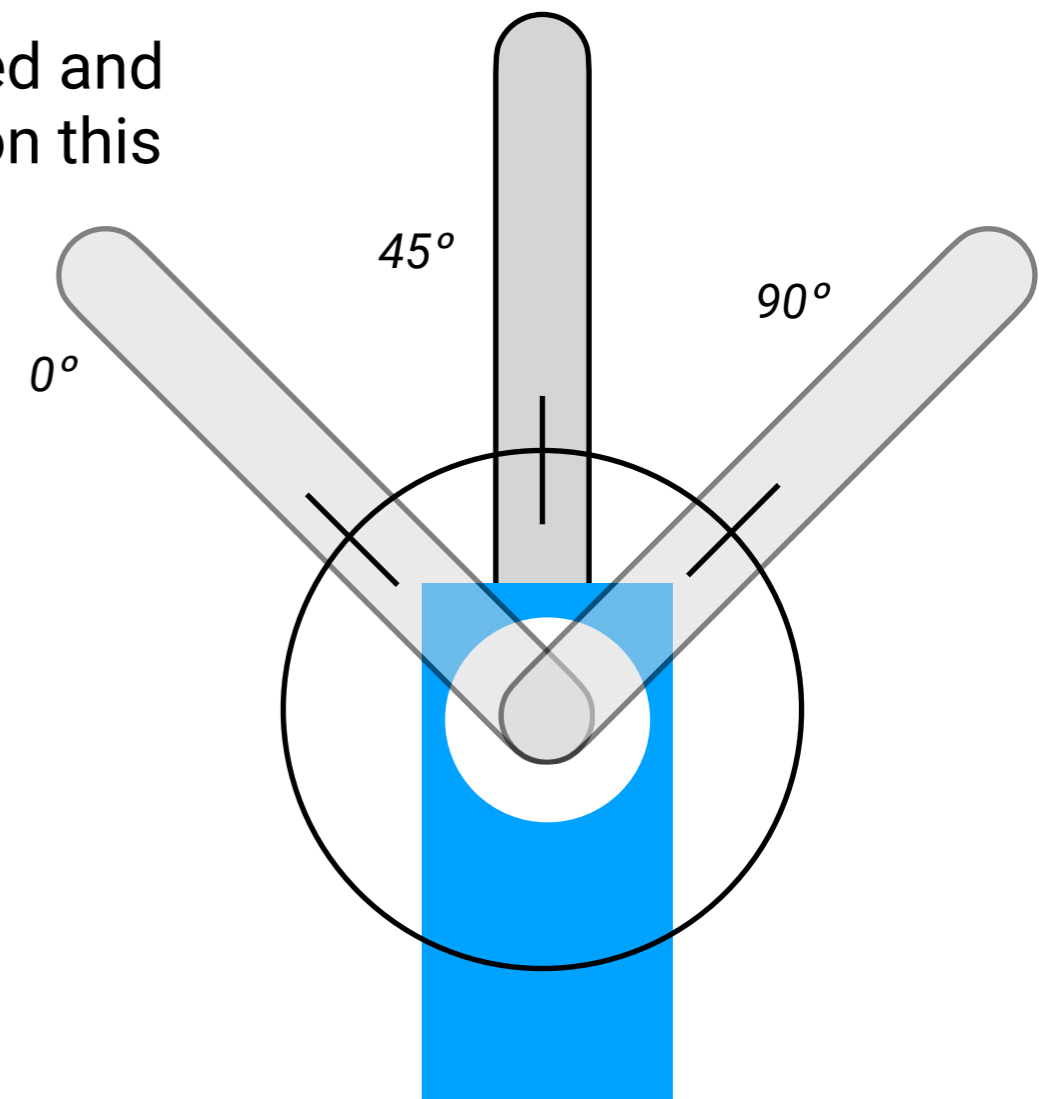
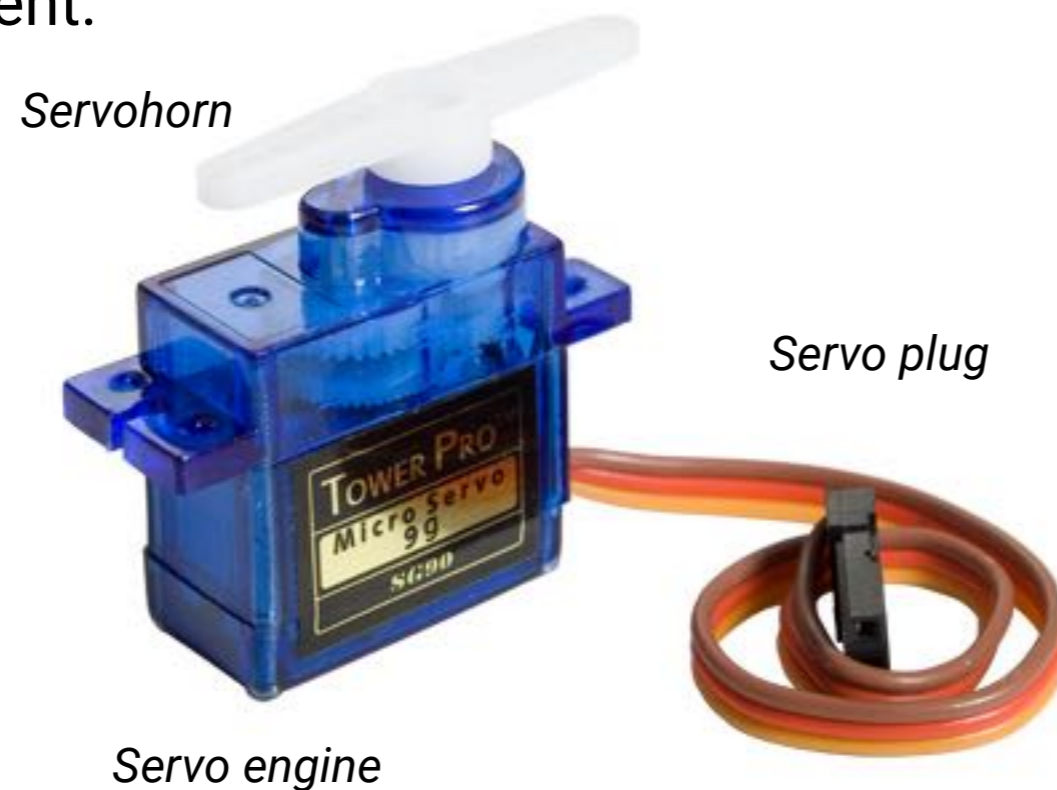
- Light with a flashlight on the sensor
- Turn on or off or dim the ceiling light - do you hear the difference?
- Place your thumb over the opening on the buzzer. What happens to the sound? Can this be used to shape the sound or music you make?
- Can you find more ways to play the instrument?



Servo engine

A servo engine is an electromechanical arm that can move at a given angle, based on a signal.

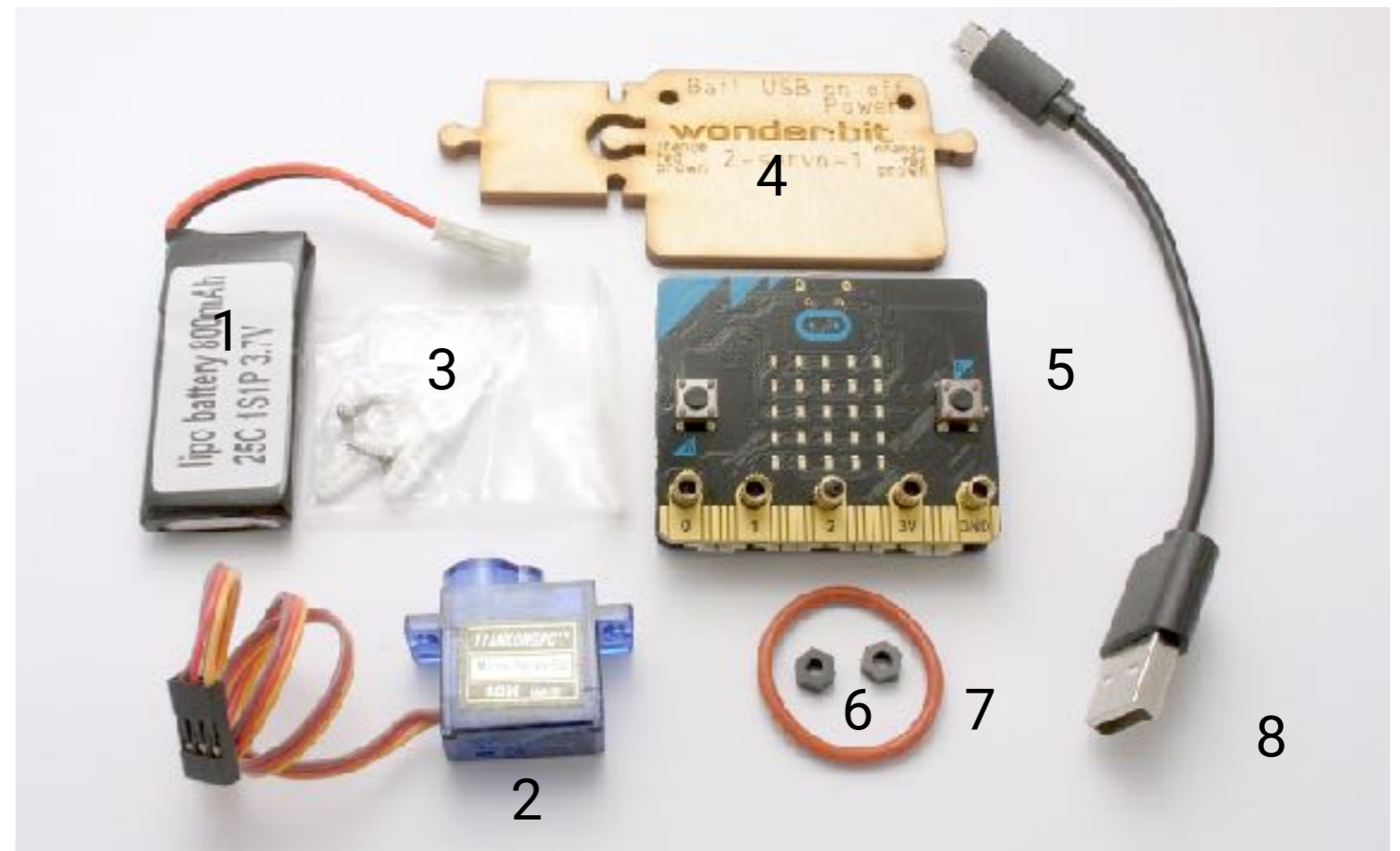
The servo engine needs a little more energy than led and buzzer. Therefore, we recommend lithium battery on this experiment.



Servo engine

You need:

1. Lithium-battery
2. Servo engine
3. Servohorn
4. Wonder:bit bracket
5. Wonder:bit-device
6. Two nuts
7. Large rubber band
8. Micro USB-cable



Bracket

Tool: Socket wrench 5,5mm

Parts:



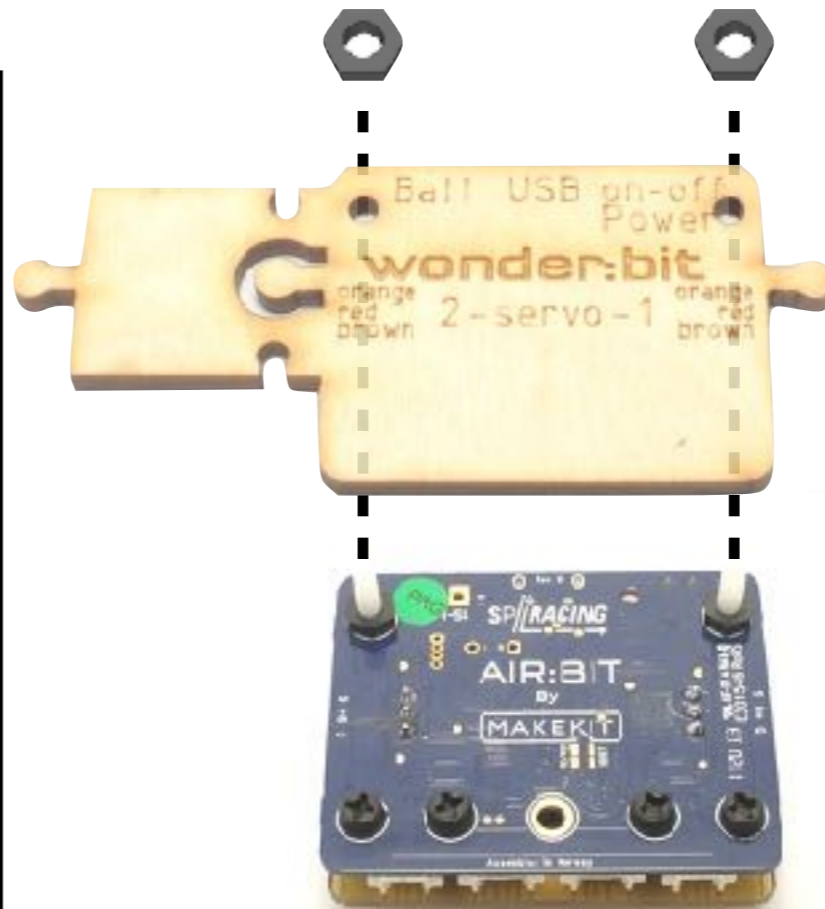
Wonder:bit-
device



Bracket



2 pcs
Nylonnuts m3



Turn the wonder:bit unit upside down and fasten the battery holder (bracket) with two nuts.

Battery

Tool:

Parts:



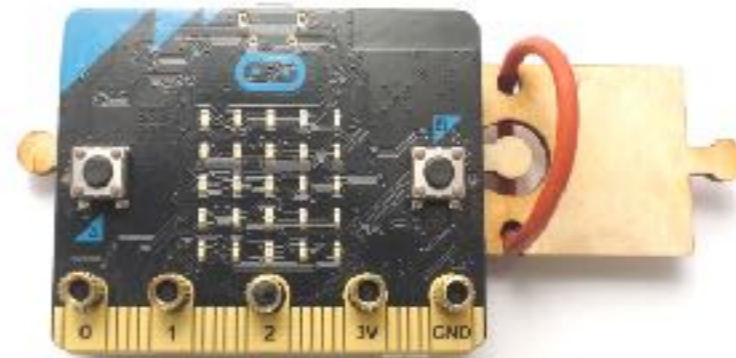
Wonder:bit-
device



Large rubber
band



Lithiumbattery

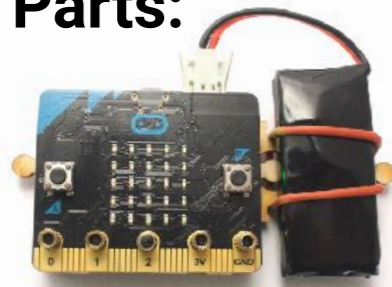


- Turn the wonder:bit device back.
- Insert the rubber band into the slot, pull it over the battery and attach it to the knob on the right side.
- Plug in the battery (gray or white plug)

Servo

Tool:

Parts:



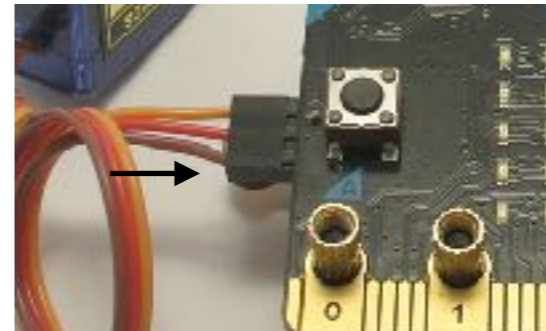
Wonder:bit-
device



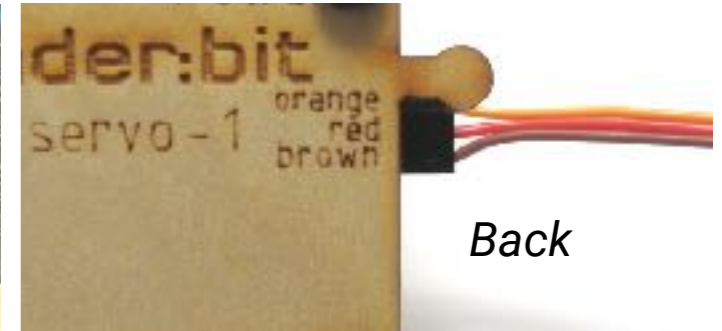
Servo engine



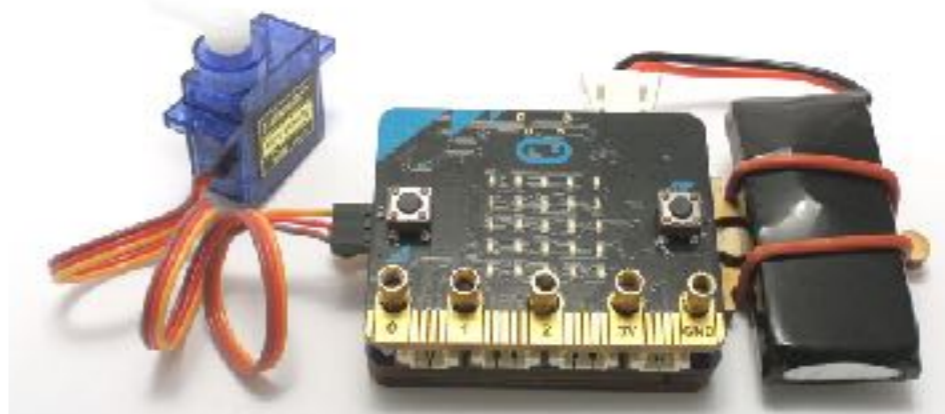
Servohorn



Front



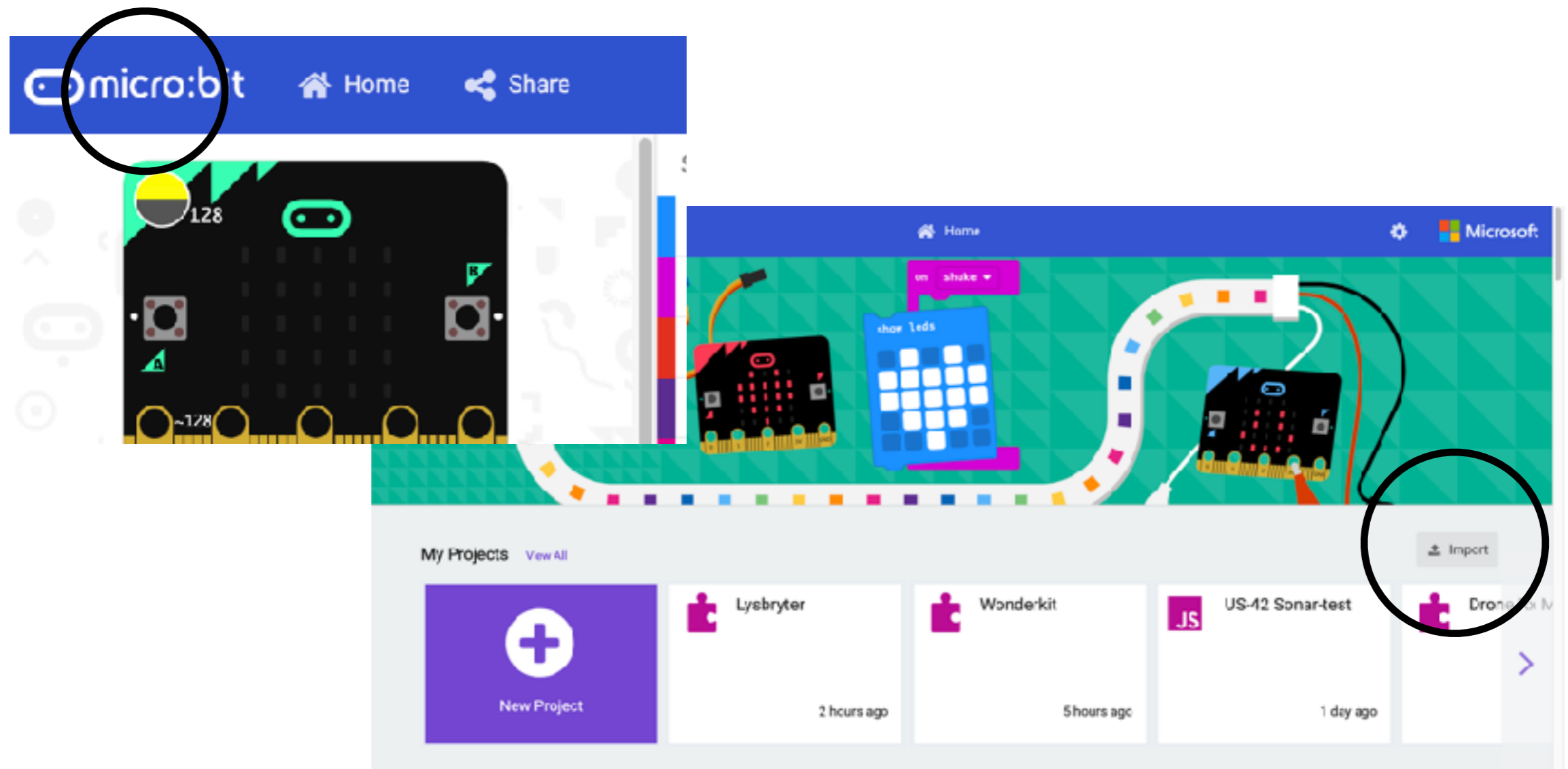
Back



- Press the servo horn into place on the servo engine
- Connect the power plug to the wonder:bit unit
- Notice the colors when you plug in, orange up and brown down.

Code the servo

To control the servo, we need the wonderbit-library. Download wonderbit-servo.hex from where you found this guide. Click "micro:bit" to get to the home page, then "import".



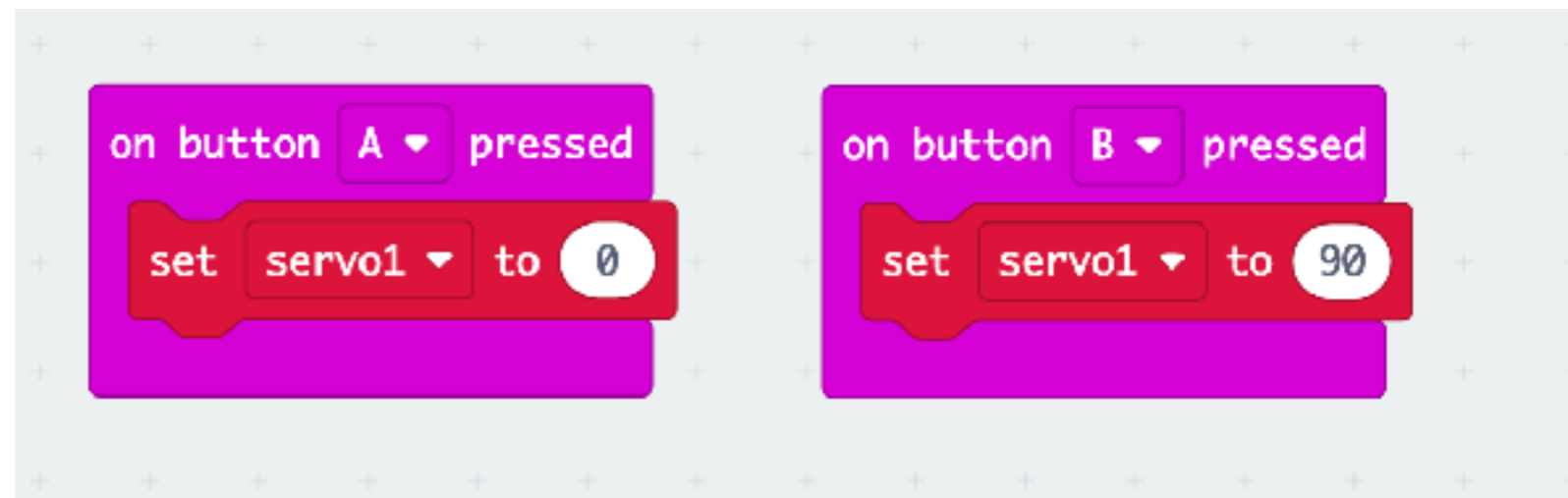
Code the servo

We start with the code below. Servo1 is a variable, which we must change to control our servo.

Here is the code we start with:

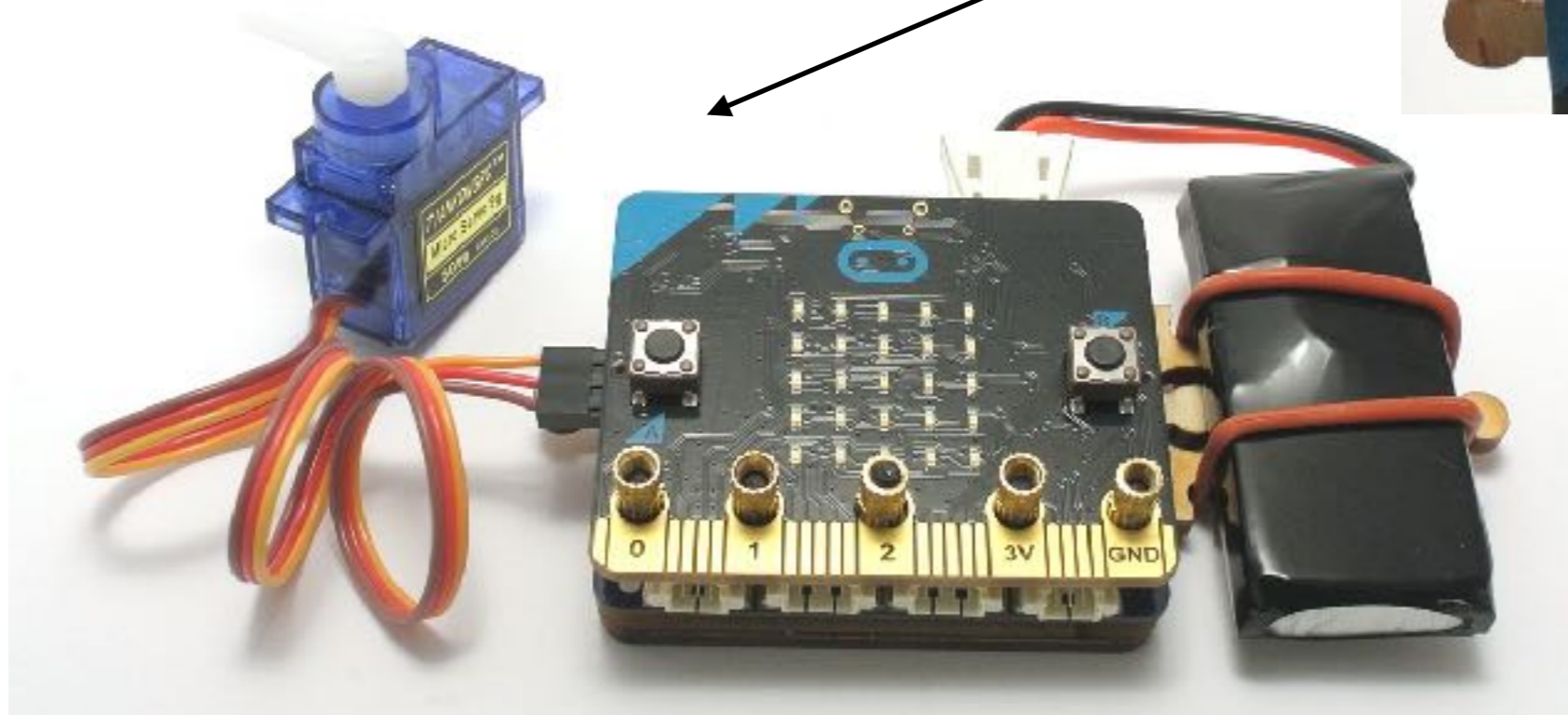


Add this code:

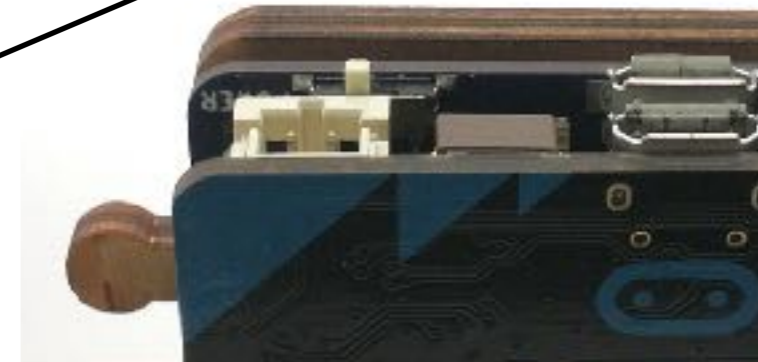


Test the servo

Load the code and turn on the power switch.
Wait 5-10 seconds.
Press the buttons in turn.



Off - On



What happens?

When we press the buttons, we change the angle of the servo1 variable. This is transmitted as a signal to servo engines, which move the horn to the desired position.

If it does not work:

Disconnect the usb and turn the power switch on and off

Check if the servo plug is connected correct, with orange up and brown down.

Check if there is power on the control board (green light)

The orange light below P0 should stop flashing after about 7 seconds. If not, check/tighten the barrel nuts (gold colored)

REACTION:BIT

Parts, reaction:bit



Aluminum foil*
double-sided tape*
Button
(*First time installation)



Servo

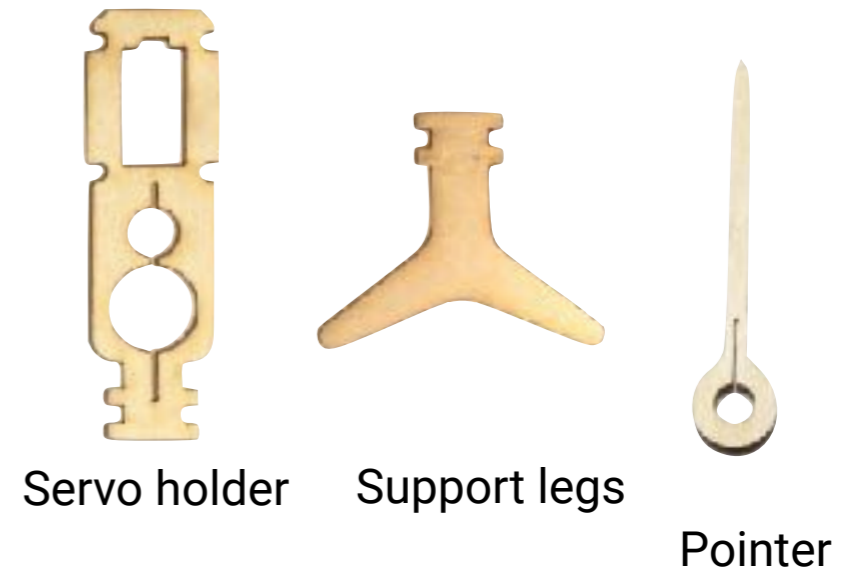


Buzzer



Large rubber band

Button bottom



Servo holder

Support legs

Pointer



2x Angle brackets



Small rubber band



Large rubber band



Measuring disc



4x crocodile clips

Button

Tool: Wallpaper knife or scissor

Parts:



Aluminum
foil

Double
-sided
tape

Button



Attach the double-sided tape to the underside of the button, at the very top.



Cover the tape with aluminum foil and cut off the extra foil.



Result

Button

Tool:

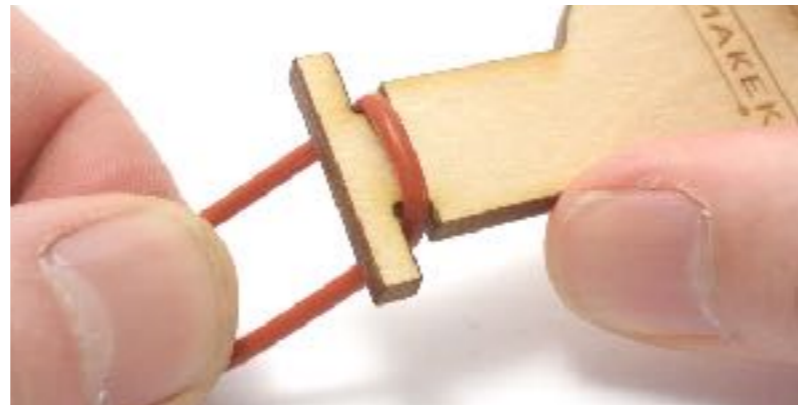
Parts:



Large rubber band



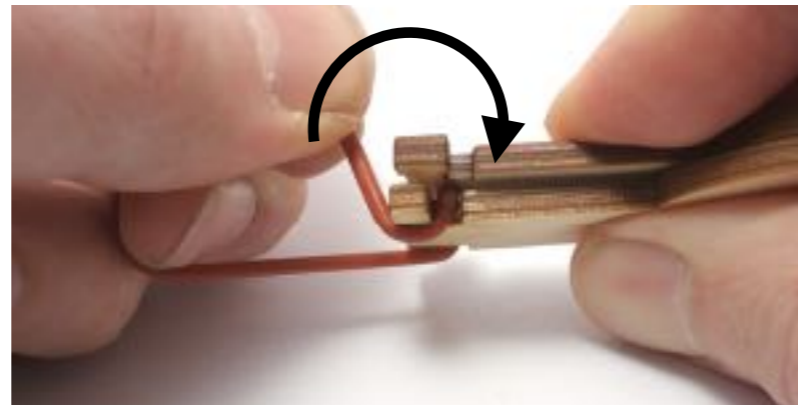
Button bottom



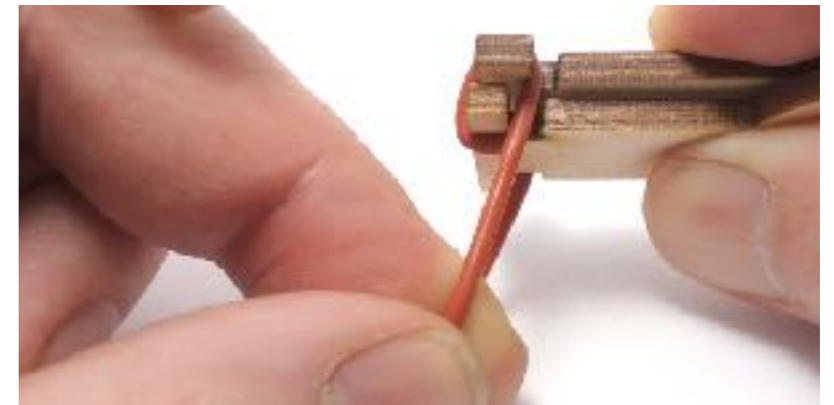
Thread the rubber ring into the button base



Place the push button on the top



Pass the rubber ring around both knobs and back again.



Repeat on the other side so that it is symmetrical.



The back should be like this



The button springs upwards

Install servo holder

Tool:

Parts:



Servo holder



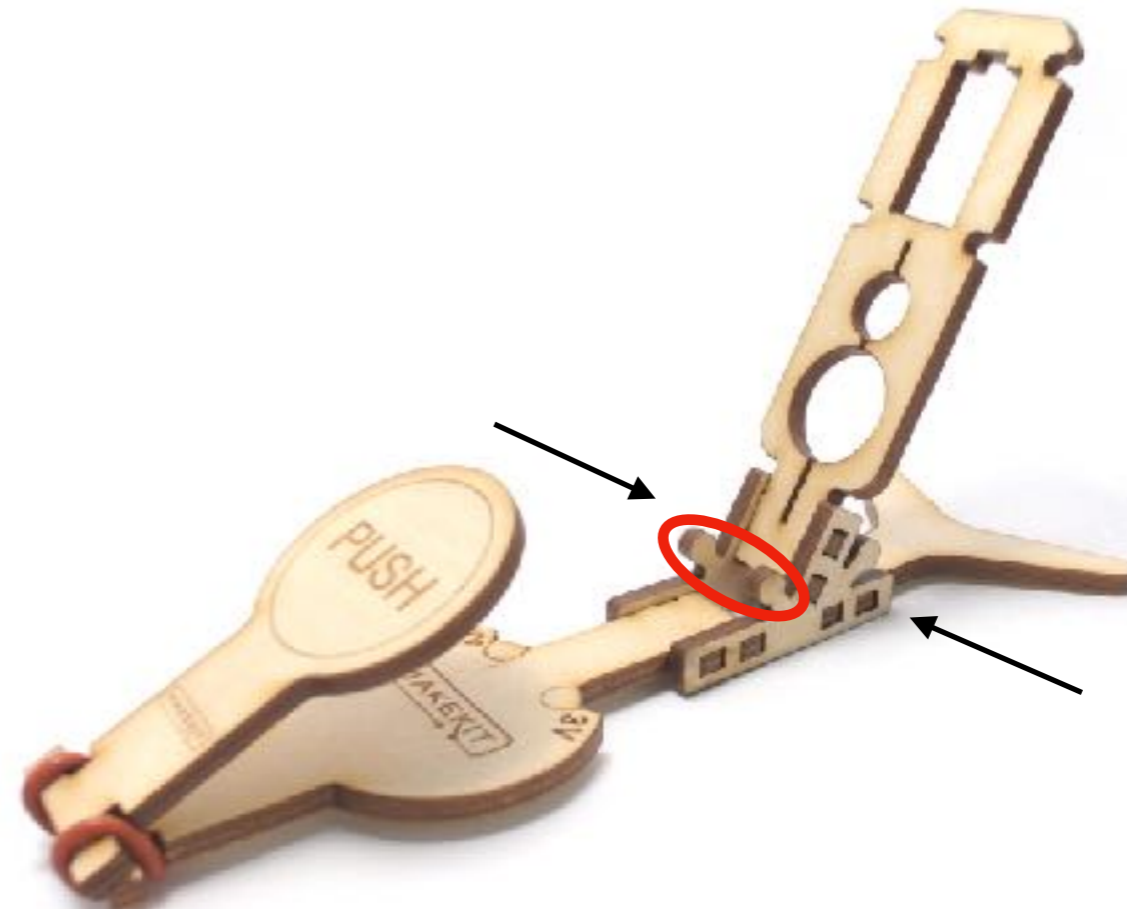
Support legs



2x Angle brackets



Small rubber band



Attach the button to the servo holder and support legs with the angle brackets. If they do not stuck, you can use a small rubber ring between the fastening knobs (red ring)

Install the servo

Tool:

Parts:



Servo



Buzzer



Large rubber band



Measuring disc



Pull the servo cord through a large rubber ring.



Push the servo through the bracket, with the gear up. Pull the rubber ring around it, so that it secures the servo.



Push the Measuring disc in place.



Push the buzzer in place.

Wiring

Tool:



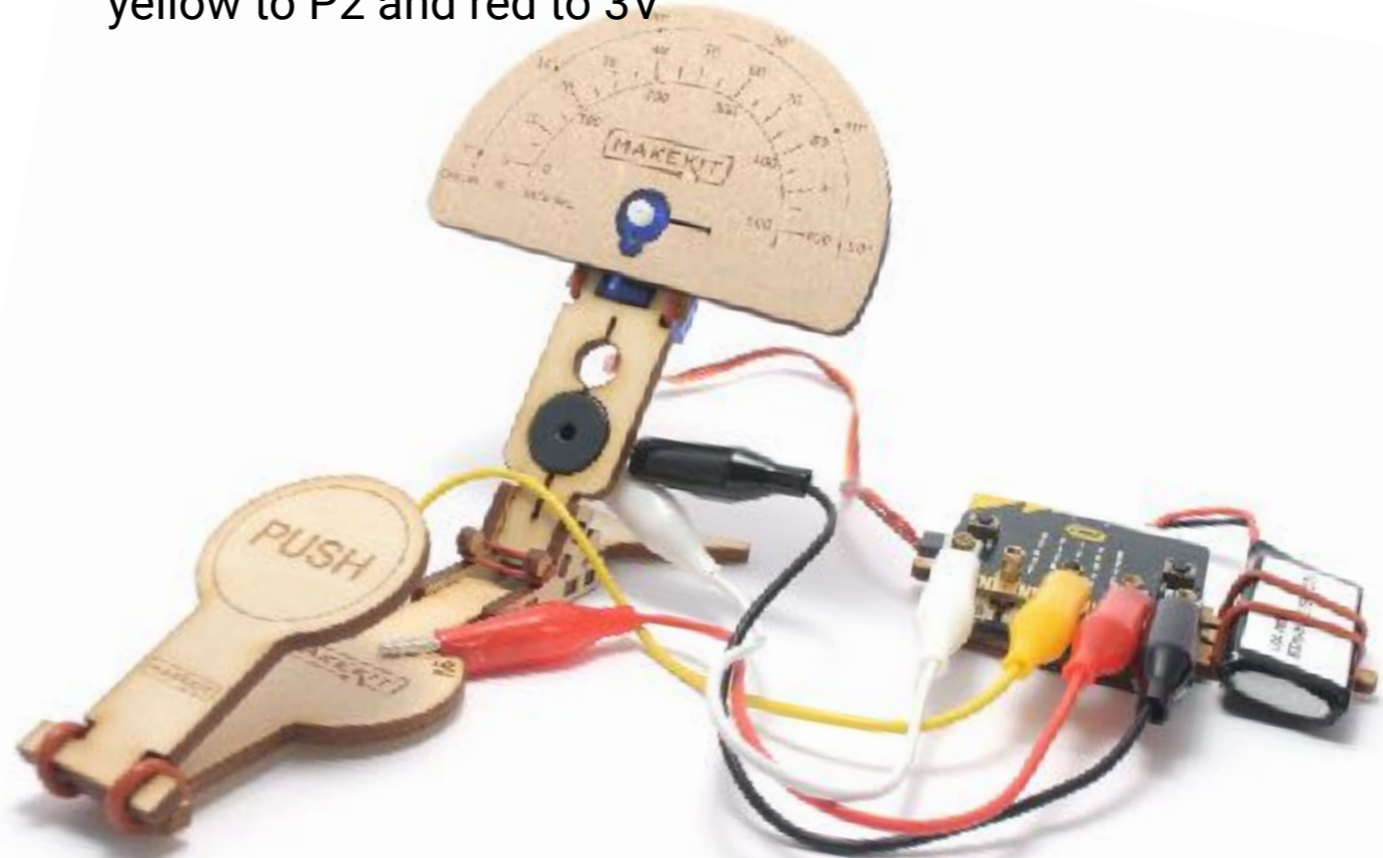
4x crocodile clips



Fasten a yellow and a red crocodile clamp into the slots, yellow to P2 and red to 3V



Connect the crocodile clips to the micro:bit, like this

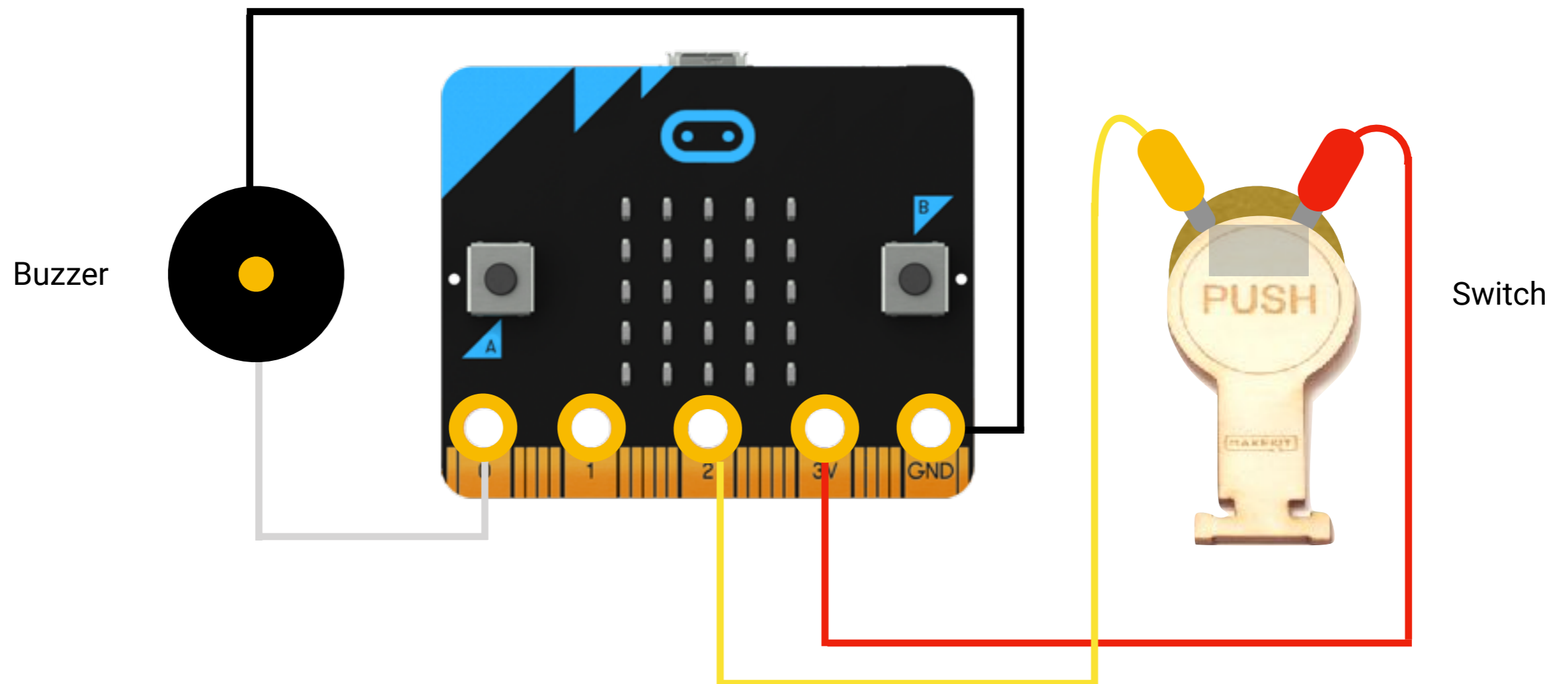


Connect the other end of the yellow and red crocodile clamp to P2 and 3V

Connect the two legs of the buzzer to P0 and GND (ground)

See wiring diagram on the next page

Wiring diagram



Test the connections: Morse code

We can test our connections by making a Morse code device.

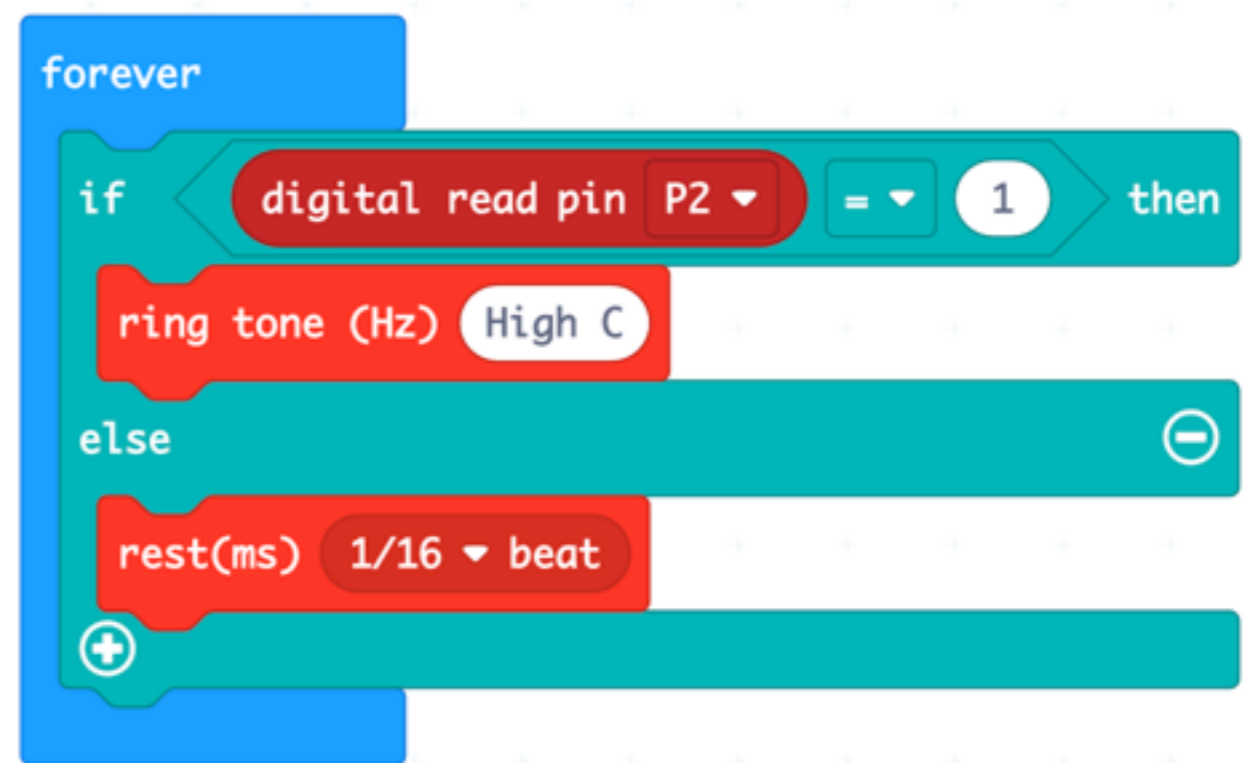
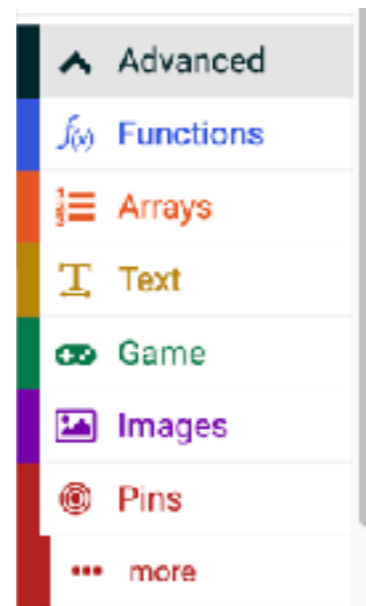
Use if-blocks, pin-blocks and music-blocks to create this code.

Digital read can be found under Advanced and Pins

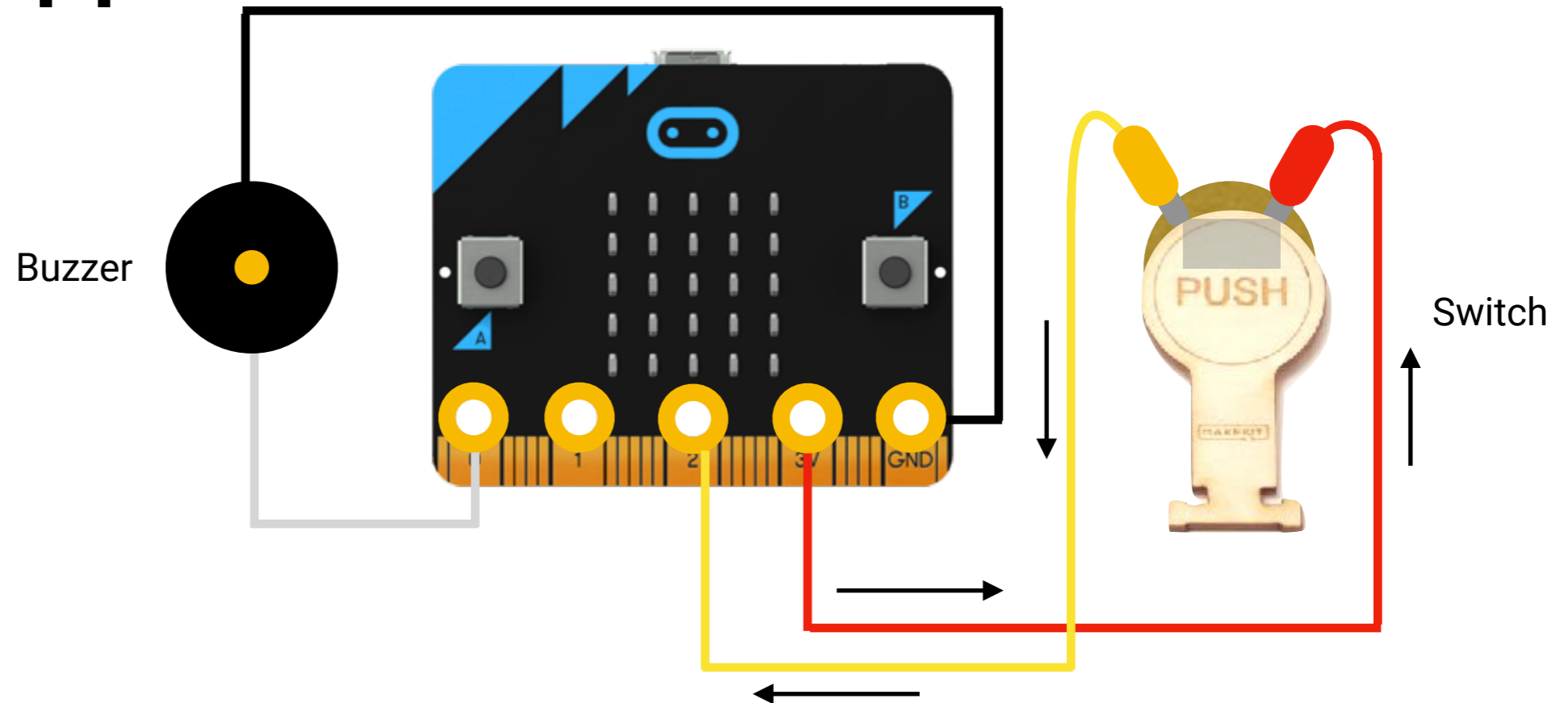
Download the code for micro:bit and test.

When you press the button, you will hear a beep from the speaker.

If not, check the connections with the crocodile clips.



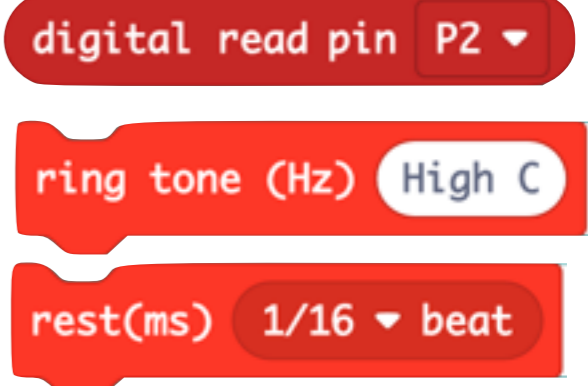
What happens?



When you press the button, contact is made between the crocodile clips via the aluminum foil. The current then enters a circuit from 3V (plus pole), via the switch and back to P2.

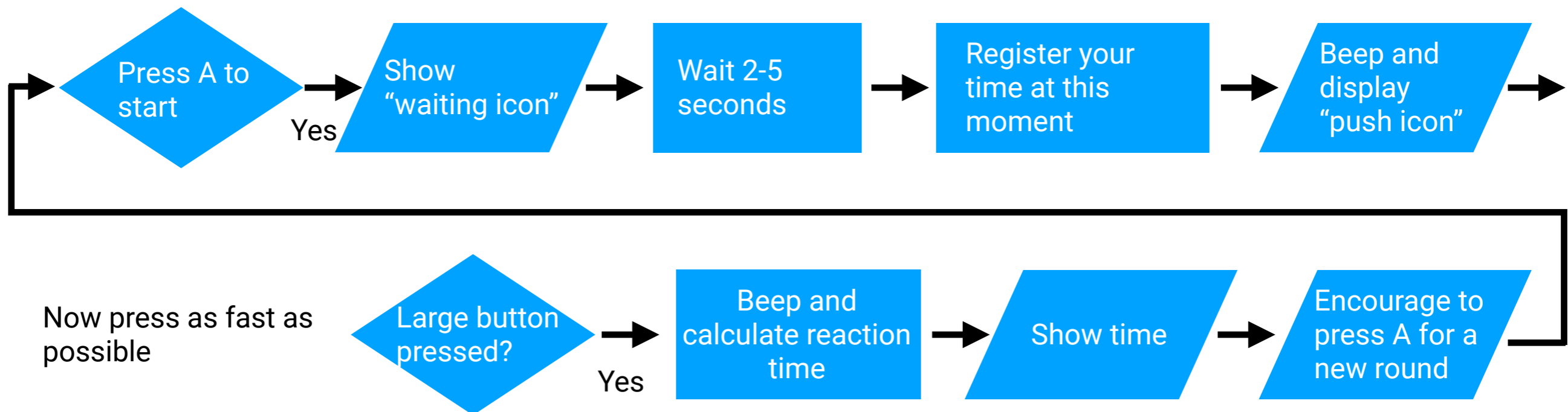
The current in P2 can be measured with digital read P2, and will give the answer 1 if a certain amount of current passes through, and 0 if no current passes through the circuit.

Furthermore, we have coded so that we will make a beep in the speaker if there is current through the circuit. Then an electrical signal is sent out from P0, through the buzzer and back to GND (ground). We stop the beep with the block rest(ms).



Reaction time tester

This flow chart shows how the game should work. When we press A, we display an icon that symbolizes waiting before we press. Then we pause a random time between 2 and 5 seconds. As soon as the pause is over, we register the time from the micro:bit's internal clock, and display a new icon. Now the player must press as fast as possible. As soon as the button is pressed, we calculate the reaction time by looking at the difference between new time and old time.



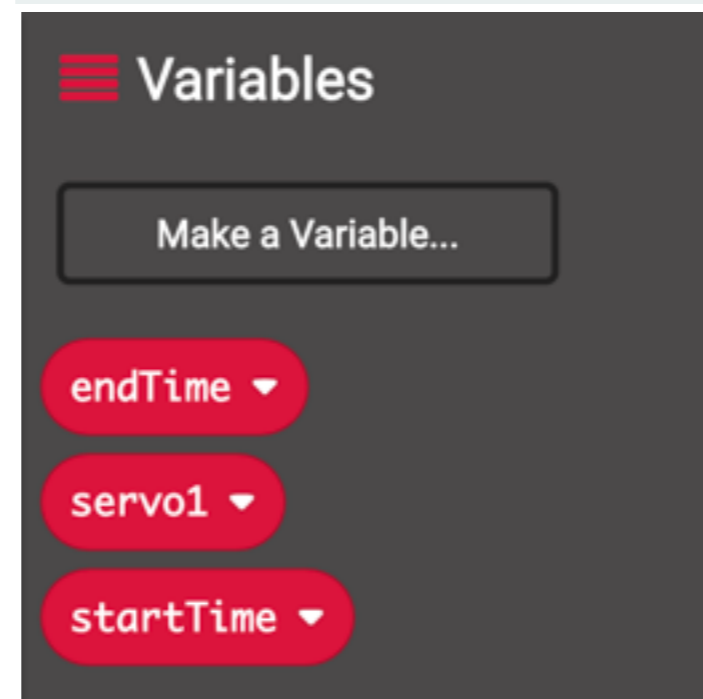
Code

Continue with your code that will contain this (delete other code)

Here is the code we start with:



Create the variables
startTime and endTime



Code - At start

The first time the game starts, we want to encourage you to press A.

Enter a "show arrow west" on start.

Make sure that the wonderkit-block "initialize" is already inside.



Code - Button A

When we press A:

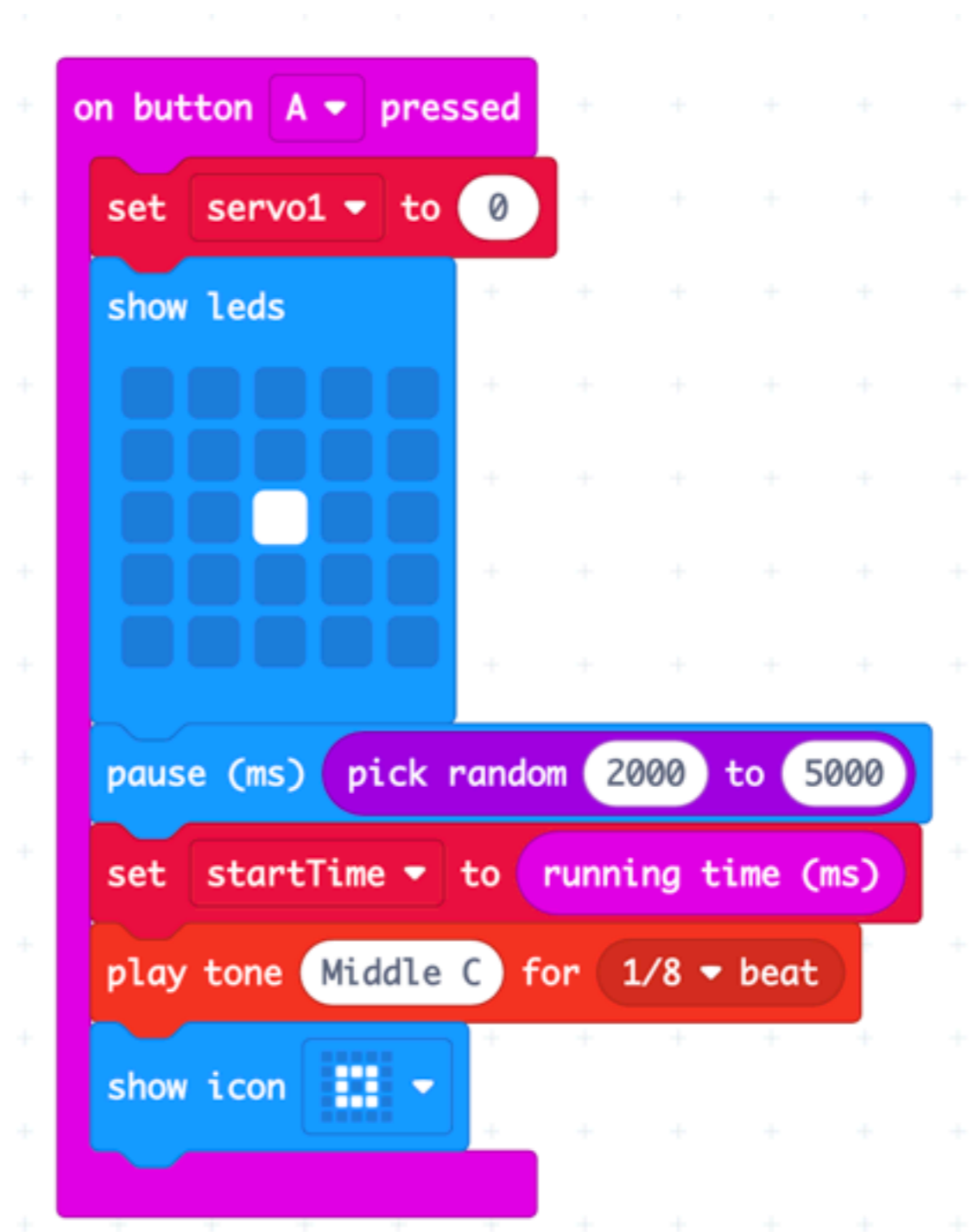
Display an icon that symbolizes pause. Feel free to create your own icon.

Pause at random between 2 and 5 seconds.
(2000-5000 milliseconds)

Record what the internal clock is right now

Play a short note to make it clear that the timing has started

Show an icon that symbolizes timing



Code - Record pressure and time

Repeat forever:

When we register current on P2:

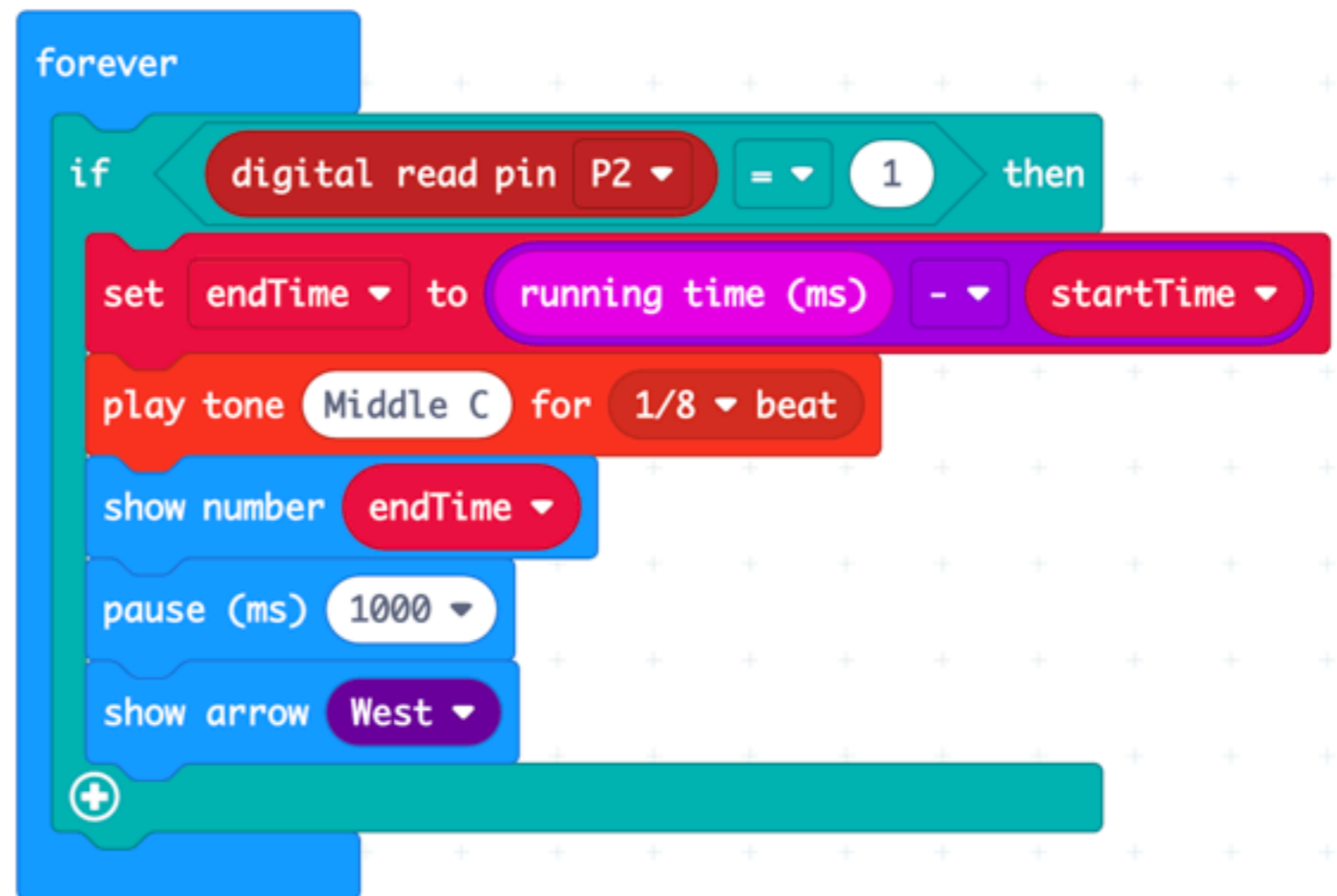
Calculate the end time by taking the current time minus the time when the testing started

Play a tone to say that pressure is registered.
Feel free to choose a lighter tone (click white field to change)

Show the reaction time on the screen

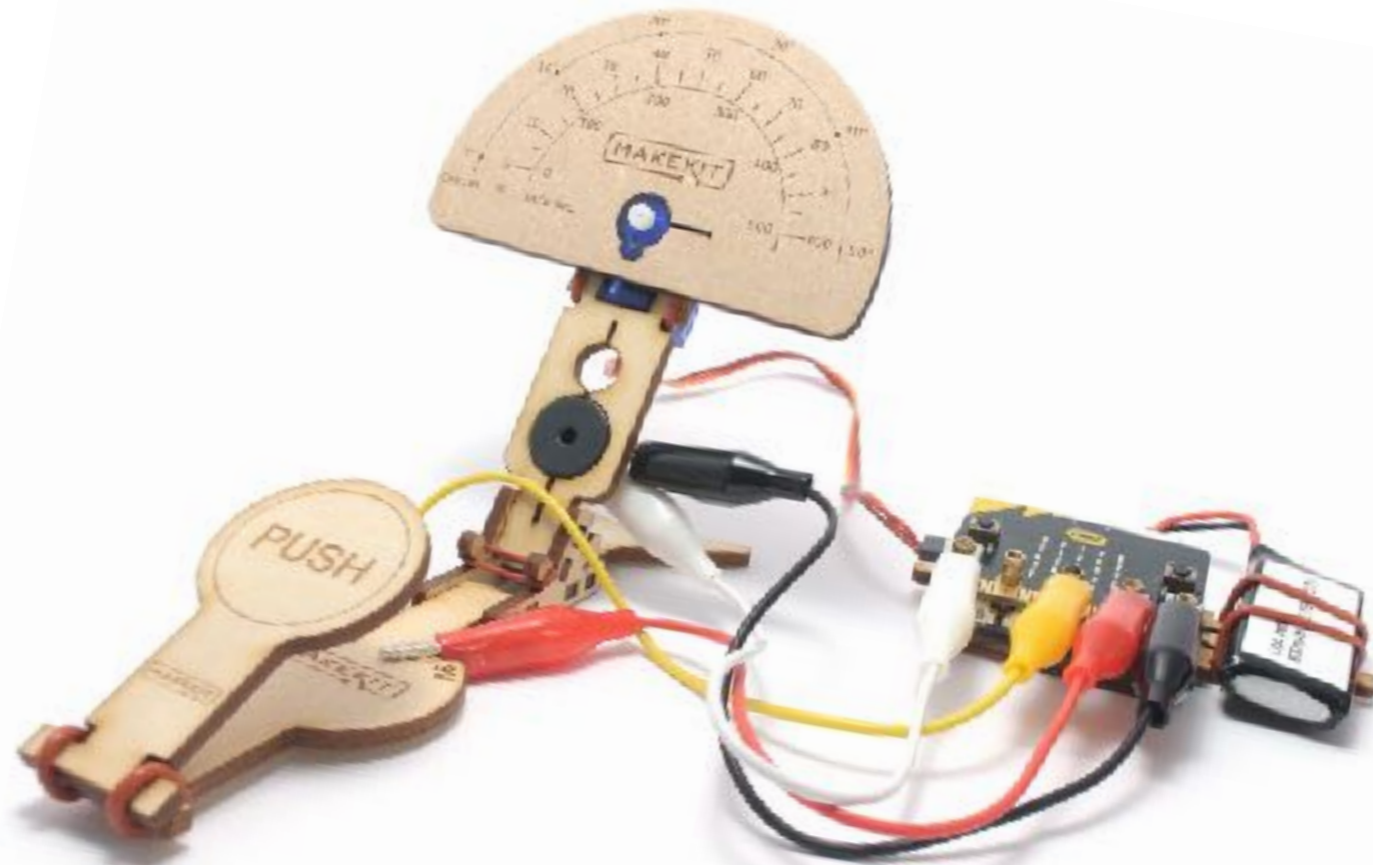
Take a break before a new game

Show arrow to the left, to indicate that you must press button A to start again.



Download the code to the Micro:bit

Test the game



Press A to start. When the beep comes, press the button as soon as you can. Then the time should be displayed on the screen. Who gets the lowest time?

If that doesn't work, do the following:

Look carefully at your code. Are there any details you forgot?

Are the connections correct? Check that everything is connected correctly and that the gold-colored barrel nuts are tightened.

Does the micro:bit have power? It must be powered by USB or battery.

Expand with "speedometer"

Parts:



Pointer

Add the following to the code:
At start: set servo to 45 degrees (Center position)

Download the code to the Micro:bit

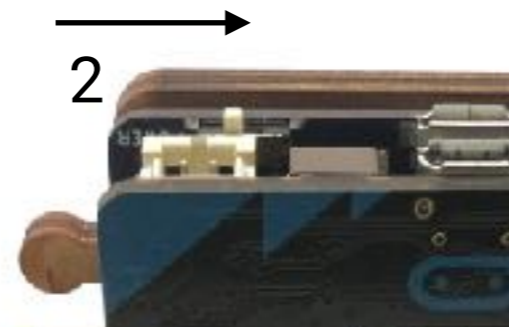


Expand with "speedometer"

Plug in battery (1)

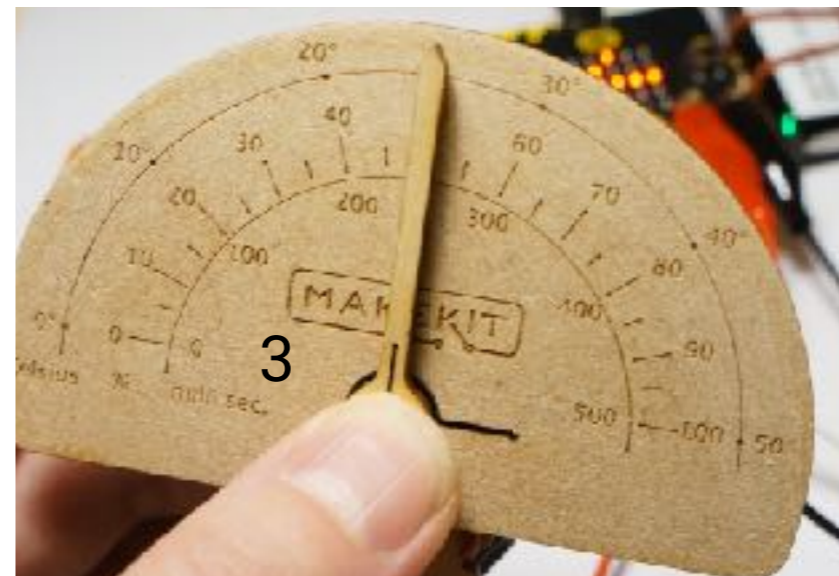


Turn on the power switch on the back of the control card (2)



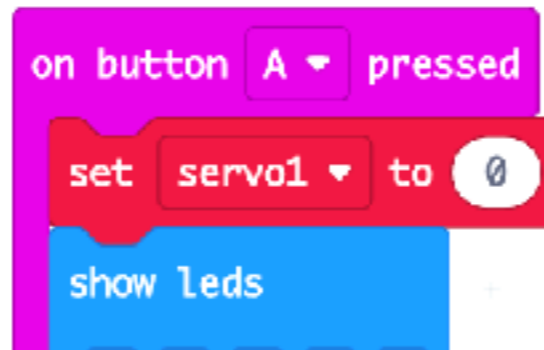
Press the pointer in place (3)

FYI! Do not force the servo to change the angle. Just push the pointer straight in without rotating.



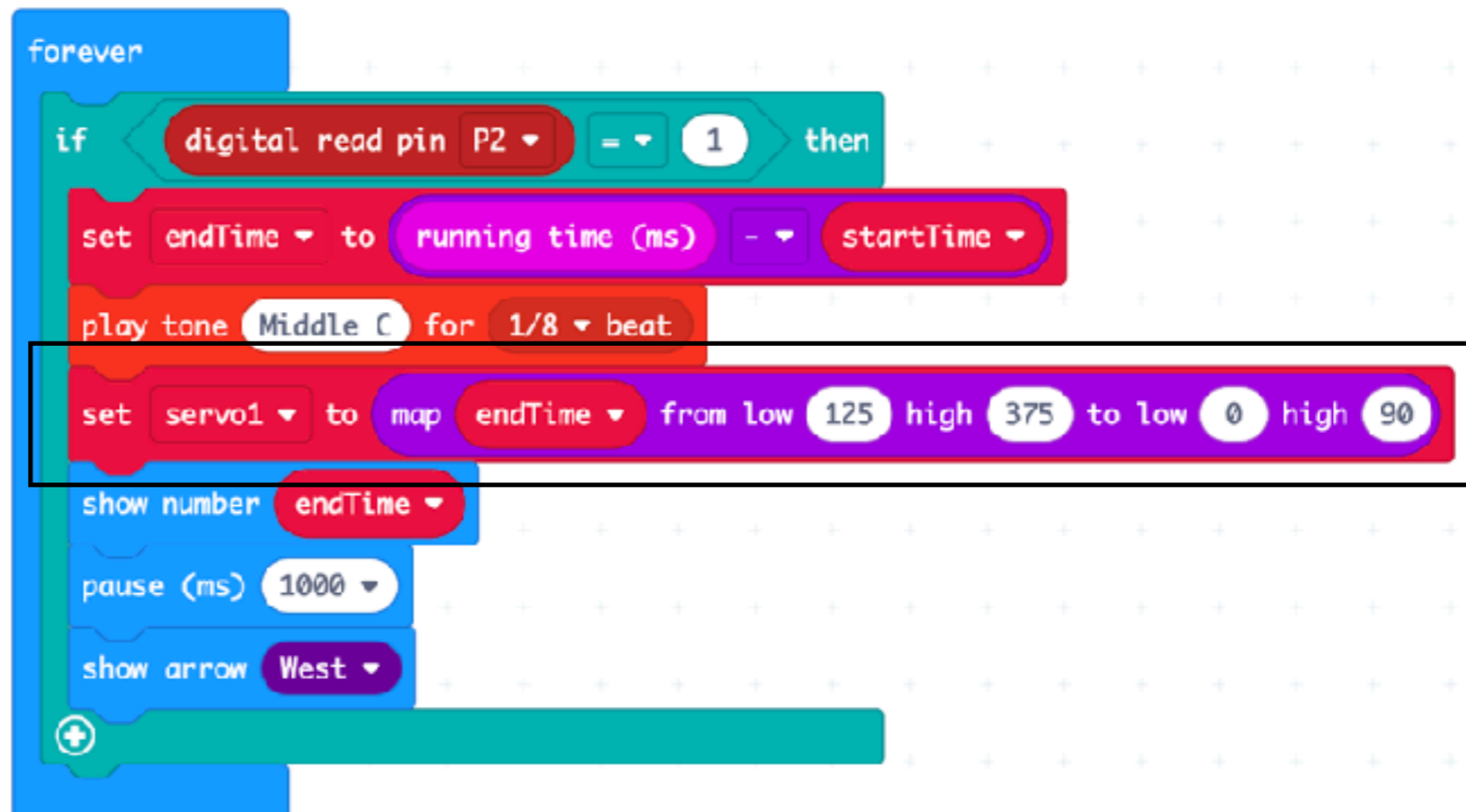
Code - Display reaction time with servo

Top of on button A:
Set servo1 to 0



```
on button A pressed
  set servo1 to 0
  show leds
```

Under play tone:
Set servo1 to the following
Map can be found under
"math" (math blocks)



```
forever
  if digital read pin P2 = 1 then
    set endTime to running time (ms) - startTime
    play tone Middle C for 1/8 beat
    set servo1 to map endTime from low 125 high 375 to low 0 high 90
    show number endTime
    pause (ms) 1000
    show arrow West
```

**Download the code to the
Micro:bit**

What happens?

When the code starts, the arrow is set to the middle position, because the angle in on start is set to 45.



When you press A, the arrow goes to the left, because the angle is set to 0.



When the time is measured, the arrow goes to the time in milliseconds that you have used.

The arrow shows an approximate reaction time between 125 and 375 milliseconds.



Contact:

Any questions, comments or feedback:

support@makekit.no

Documents main folder:

makekit.no/docs



www.makekit.no



support@makekit.no



[makekit](#)



[gomakekit](#) (also twitter)