

Lesson 2 - Accelerometer & Music

By the end of the lesson, students should be able to:

Accelerometer

- Get the current acceleration in the x, y and z axes
- Recall and use `while` loop
- Recall and use conditional statements

Music

- Detects the current gesture of the microbit
- Play melodies using the microbit and buzzer
- Unblock the `music.play()` function








Version

Date: January 2020

Format: 8 lessons x 2 hours

Important! View speaker notes for details

Things to note

-  **Unplugged** = Activities not involving technology (Videos, Kinaesthetic activities etc.)
-  **Discussion** = Get the students to think and respond about a question
-  **Guided** = Demonstration → Instructor does the activity while the student mimics)
-  **Unguided** = Instructor will give the students the task and show what the final result should look like and give the students a certain amount of time to do it by themselves before moving on to “Check for Understanding”
-  **Check for Understanding** = Instructor will go through the solution with them or get a student to share the solution
-  **Sandbox** = Free-Play (Students recap what they learnt from the entire day by creating a project)
-  **Bonus** = This is given to students who are fast-paced

Materials Needed

Per student:

- 1x microbit set
 - 1x microbit
 - 1x usb
 - 1x battery pack
 - 2x AAA batteries
- 1x breakout board set
 - 1x breakout board
 - 1x buzzer
- 1x Chromebook/Laptop

Frequently Asked Questions

“What if I can't finish the activities for that particular day? ”

- In the event that you can't finish all of the activities in the given time, DO NOT rush to finish the concepts and just continue where you left off the next week.
- The bonus activities are for the faster students that have completed the general task that was given to the whole class. You do not need to cover this with everyone.

“How do you know you’ve been teaching the right way?”

- When students are able to create their sandbox with minimal to no help from you.

Frequently Asked Questions

“What is the purpose of this course?”

- For students to practice applying Python knowledge learnt previously on a micro-controller and build structures with Strawbees to present their creations to tasks given.
- Students also learn more basic coding concepts through Python and use them via computational thinking.

“What is computational thinking?”

- Computational thinking allows us to take a complex problem, understand what the problem is and develop possible solutions. We can then present these solutions in a way that a computer, a human, or both, can understand.

Frequently Asked Questions

“Why must I follow the speaker notes and teach in a certain way? I prefer to freestyle.”

- For follow-up purposes as there will be cases where you might be unable to teach your class on a particular day and another instructor will need to cover you.

Frequently Asked Questions

“ Can students bring home the Strawbees structure?”

- No, but they can take pictures of their structures before dismantling them.

“ Can students bring home the microbit set?”

- No, but they can take videos of their projects/ pictures of their structures before dismantling them.

“Can students buy the Strawbees or microbit set?”

- No.



PYTHON

LEVEL 2

< 8) =) ; D :) / >
CODE IN THE COMMUNITY

Attendance Taking

Please ensure that your attendance has been taken at the start of every lesson.

You will need to attain 80% attendance in order to graduate from this course.

Lesson 1	Lesson 2	Lesson 3	Lesson 4	Lesson 5	...
✓	✓	✓	✓	✓	...



Recap & Agenda

Today's Lesson

- Introduction to Accelerometer
- X Spirit Level
- Introduction to Breakout Board
- Accelerometer Music Player
- Movement Alarm



Accelerometer

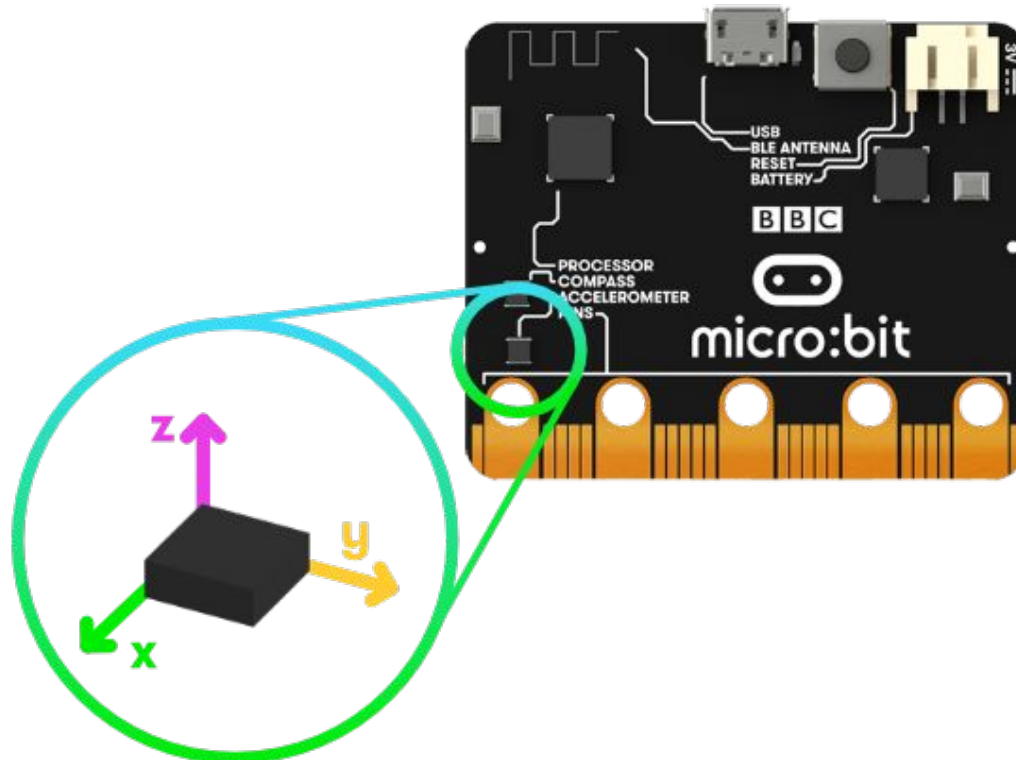


Introduction to Accelerometer

Mobile

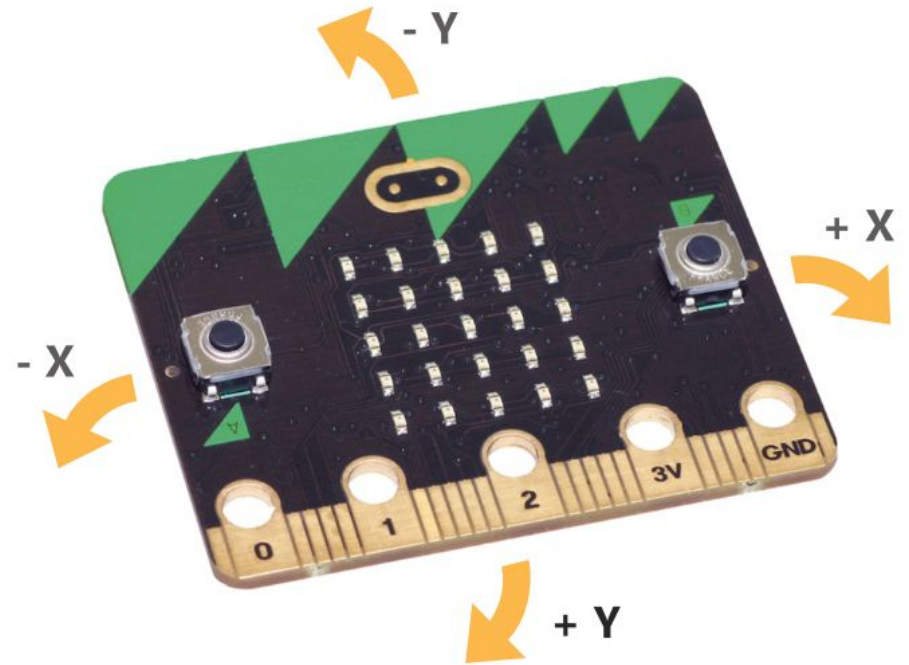
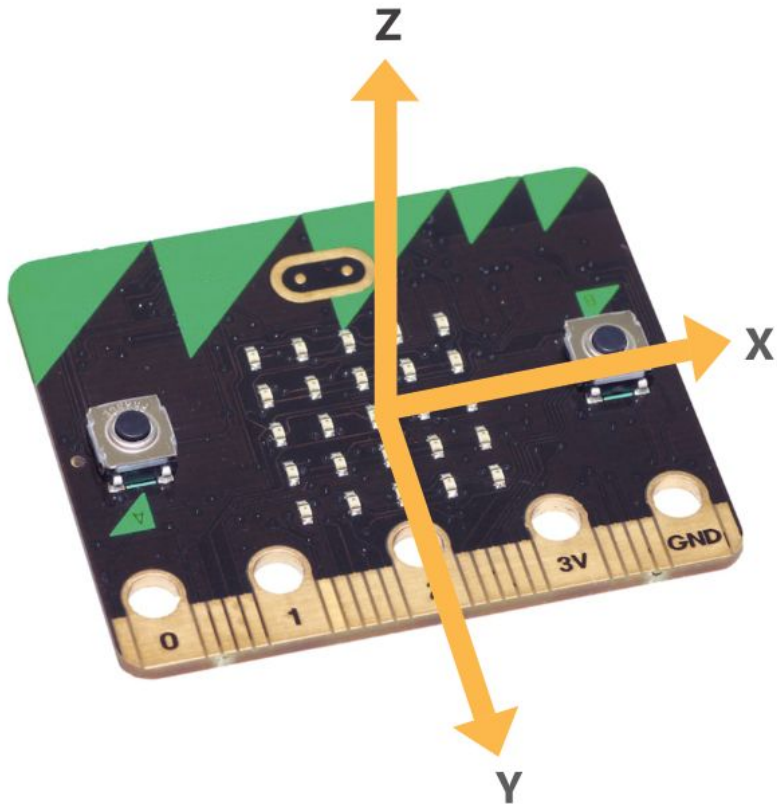


Accelerometers are devices that measure acceleration across multiple axes



 Unplugged

X Y Z axis



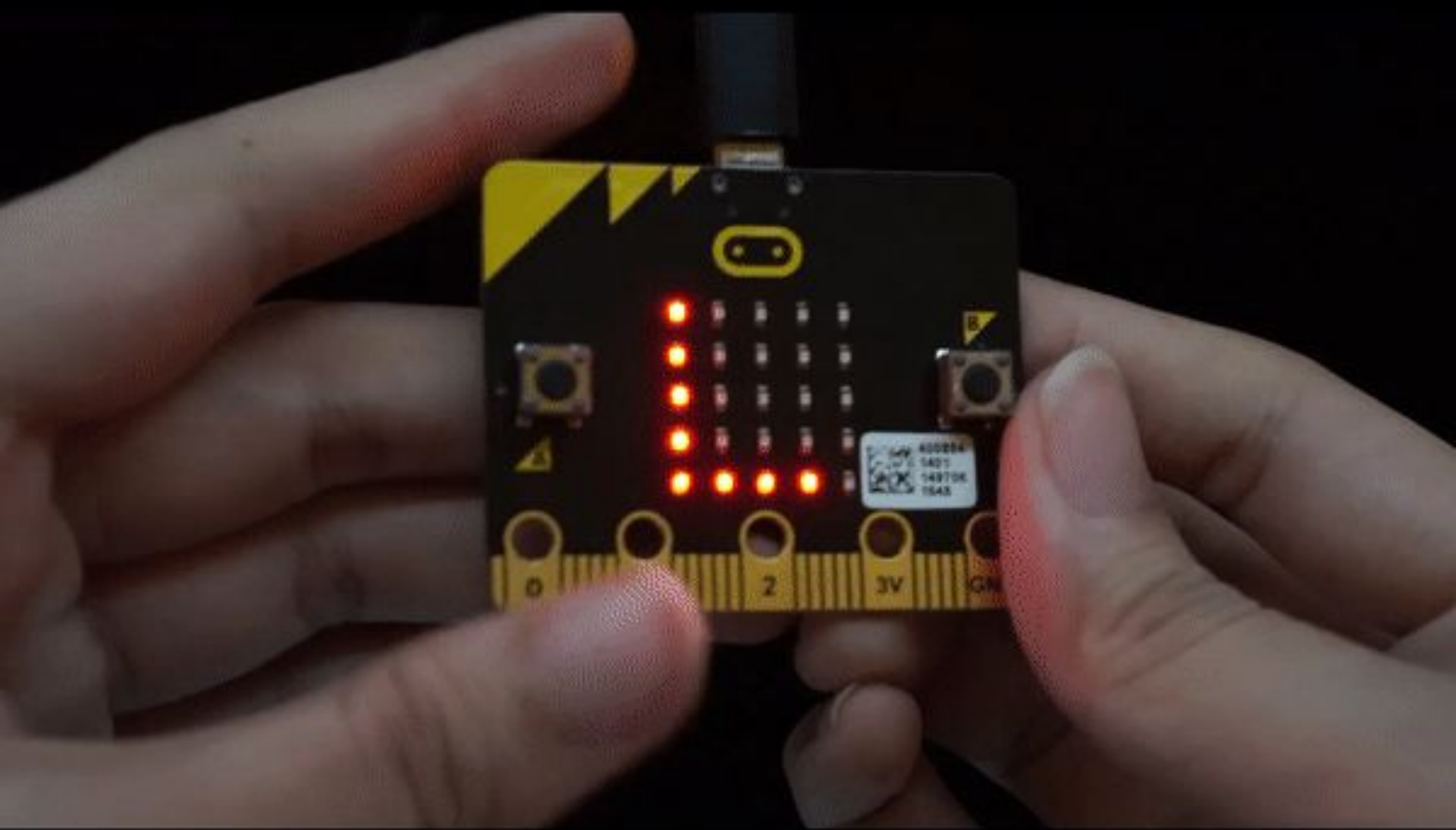
What kind of devices have accelerometers?

Unplugged





Balance Game



Programme the micro:bit

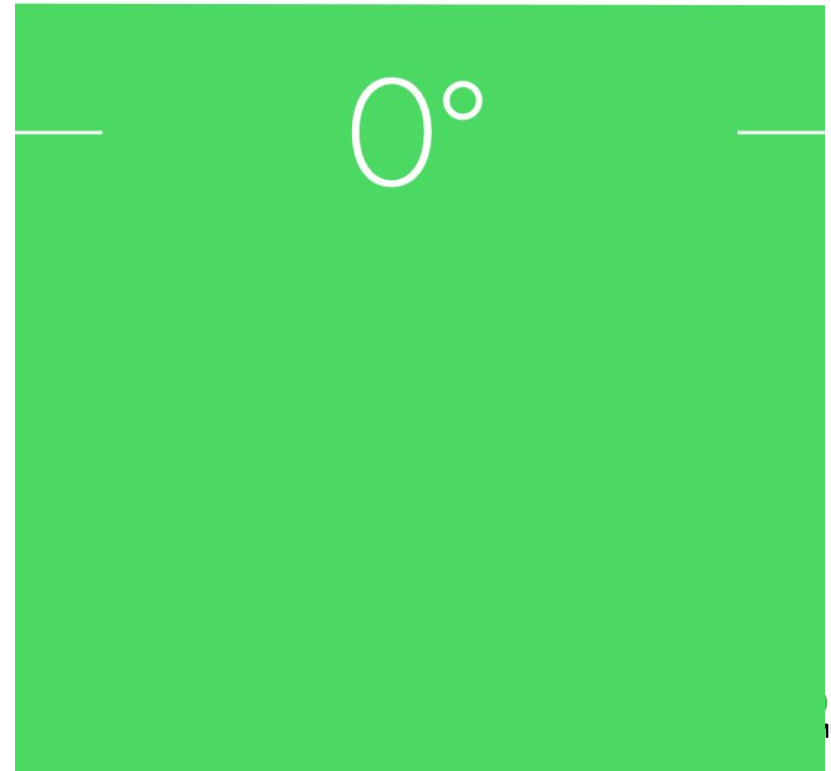
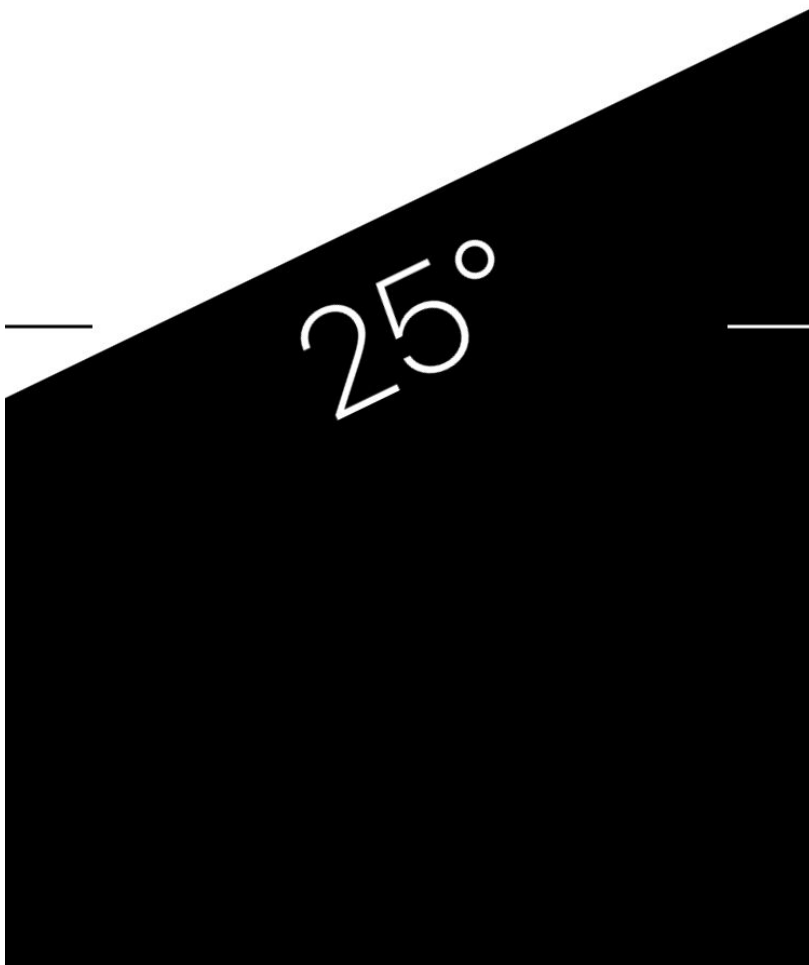
Create a spirit level that tells the user to tilt the micro:bit left or right to achieve balance on the x axis.



Guided

What is a Spirit Level?

What is a Spirit Level?





How To
Turn Your
iPhone Into
A
Spirit Level



Guided

Create a `while True` loop

Why?



Guided



**Find the Function that
would return the X value of
the Accelerometer**

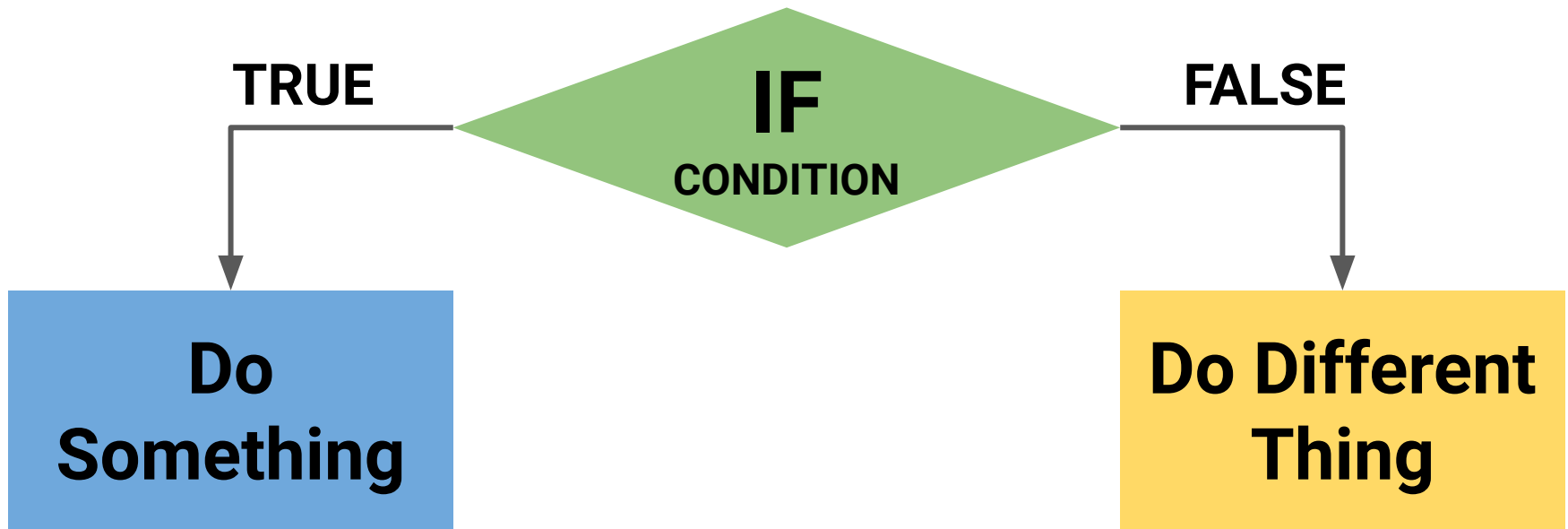


X value of Accelerometer

```
microbit.accelerometer.get_x()
```

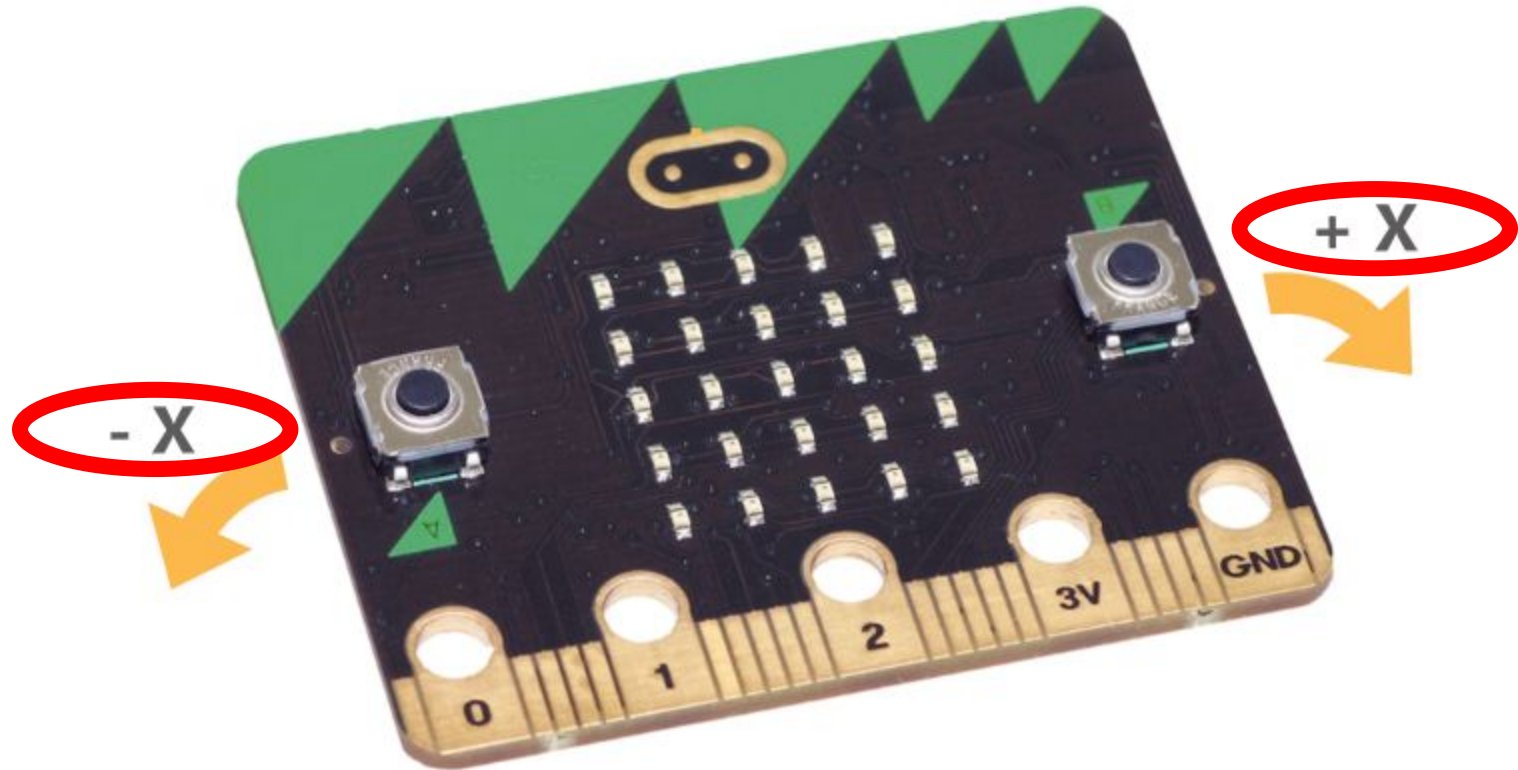
Get the acceleration measurement in the x axis, as a positive or negative integer, depending on the direction. The measurement is given in milli-g. By default the accelerometer is configured with a range of +/- 2g, and so this method will return within the range of +/- 2000mg.

Conditional



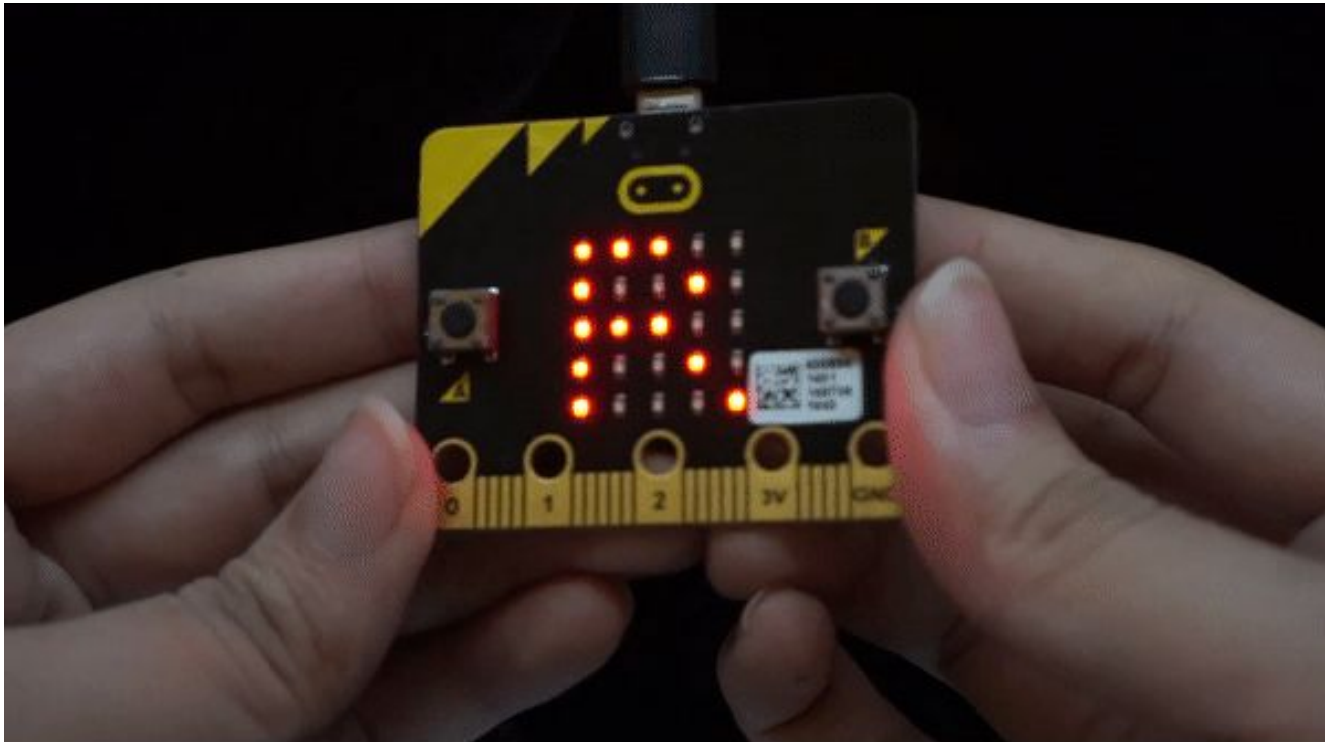


Guided



Conditionals

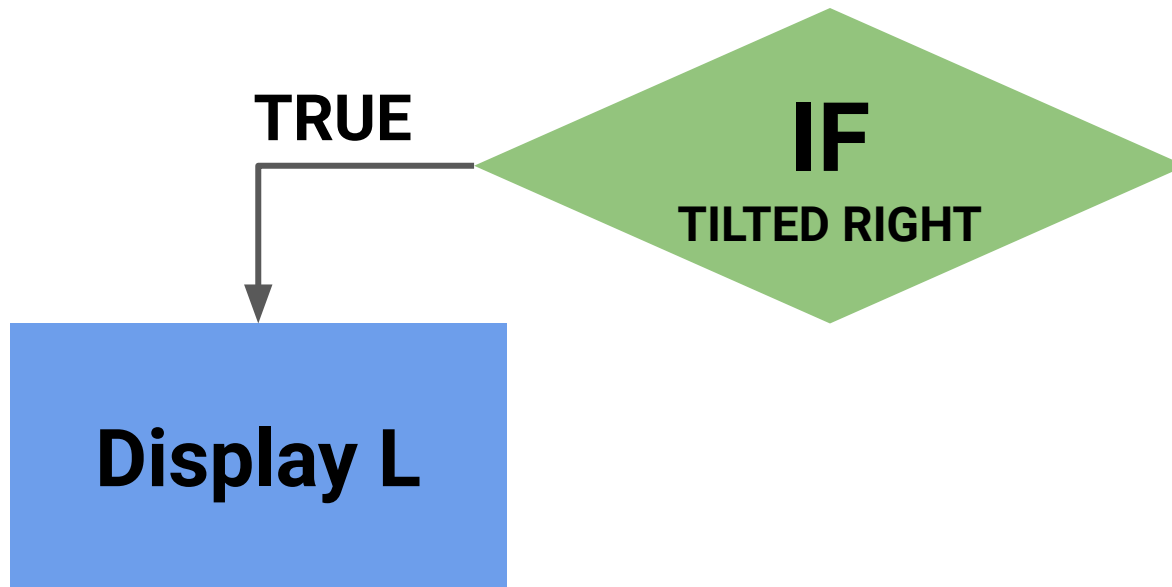
If micro:bit tilted right, display L





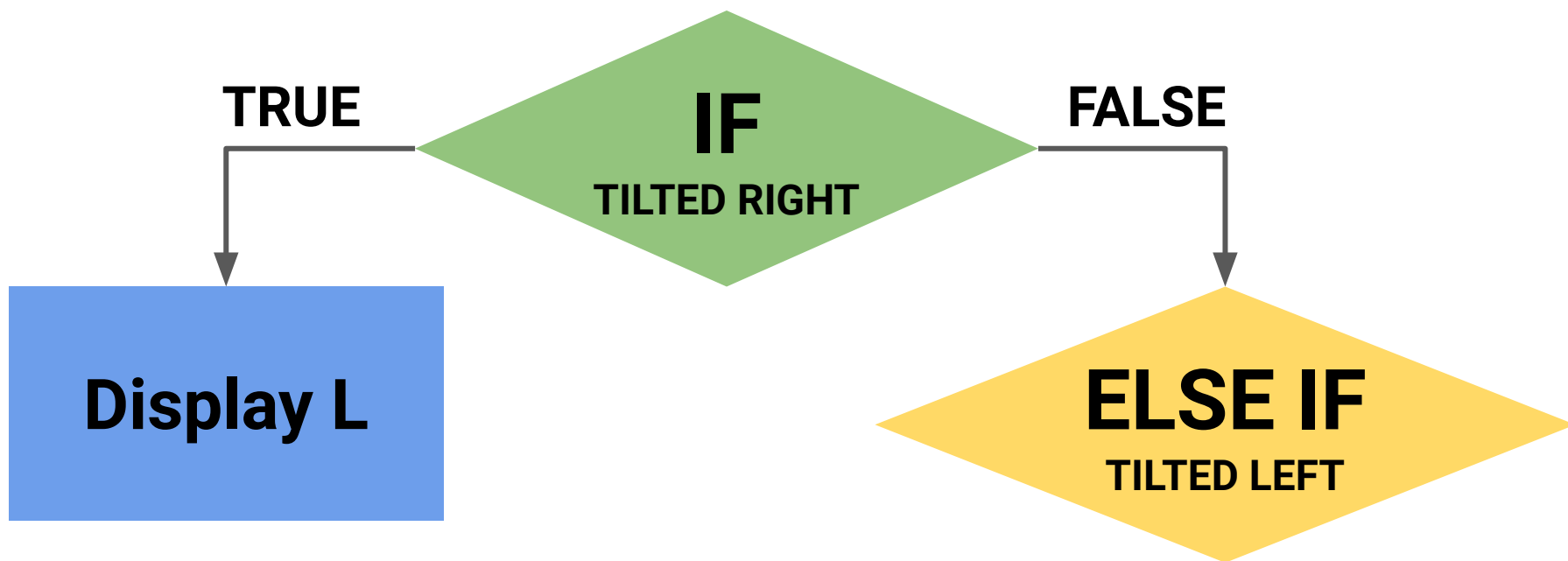
Guided

Conditional



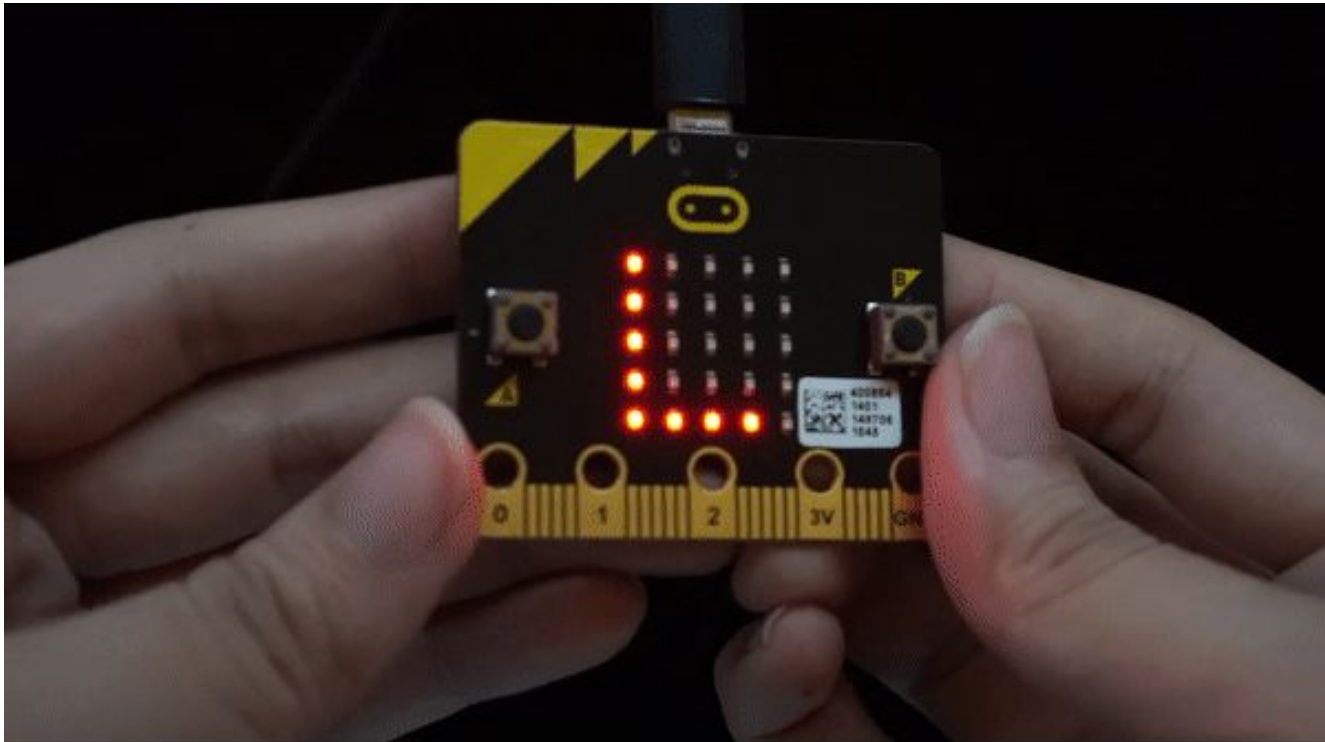


Conditional



Conditionals

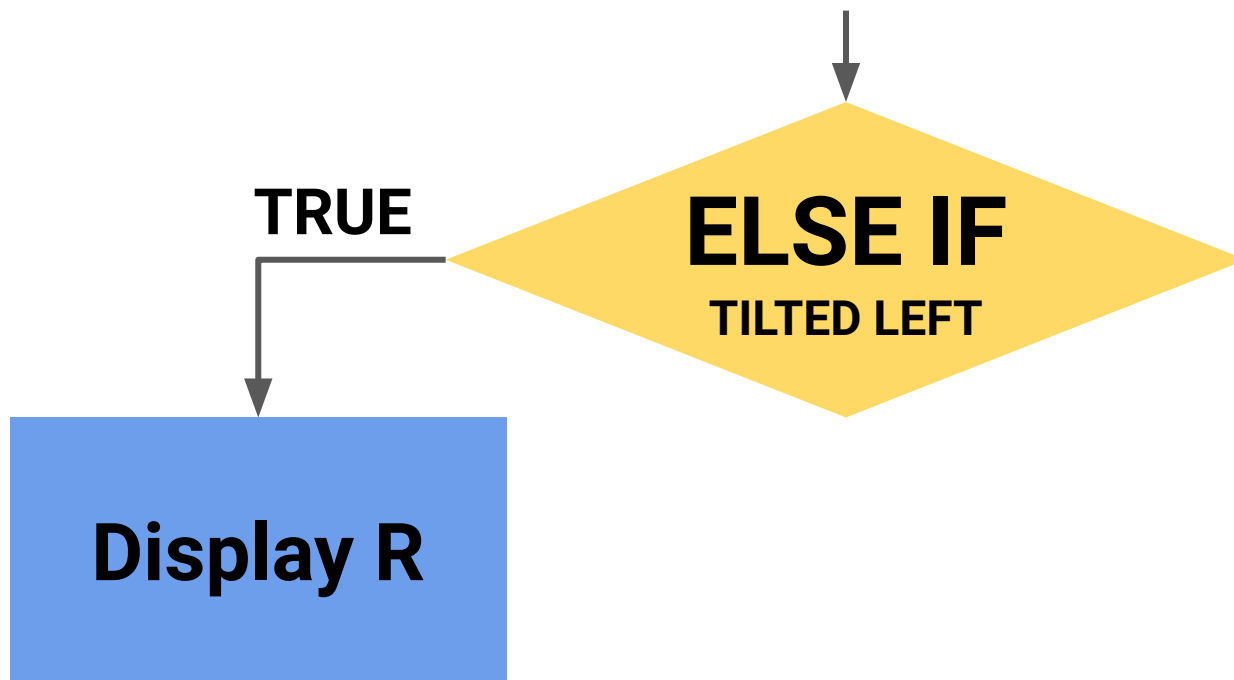
Else if micro:bit tilted left, display R





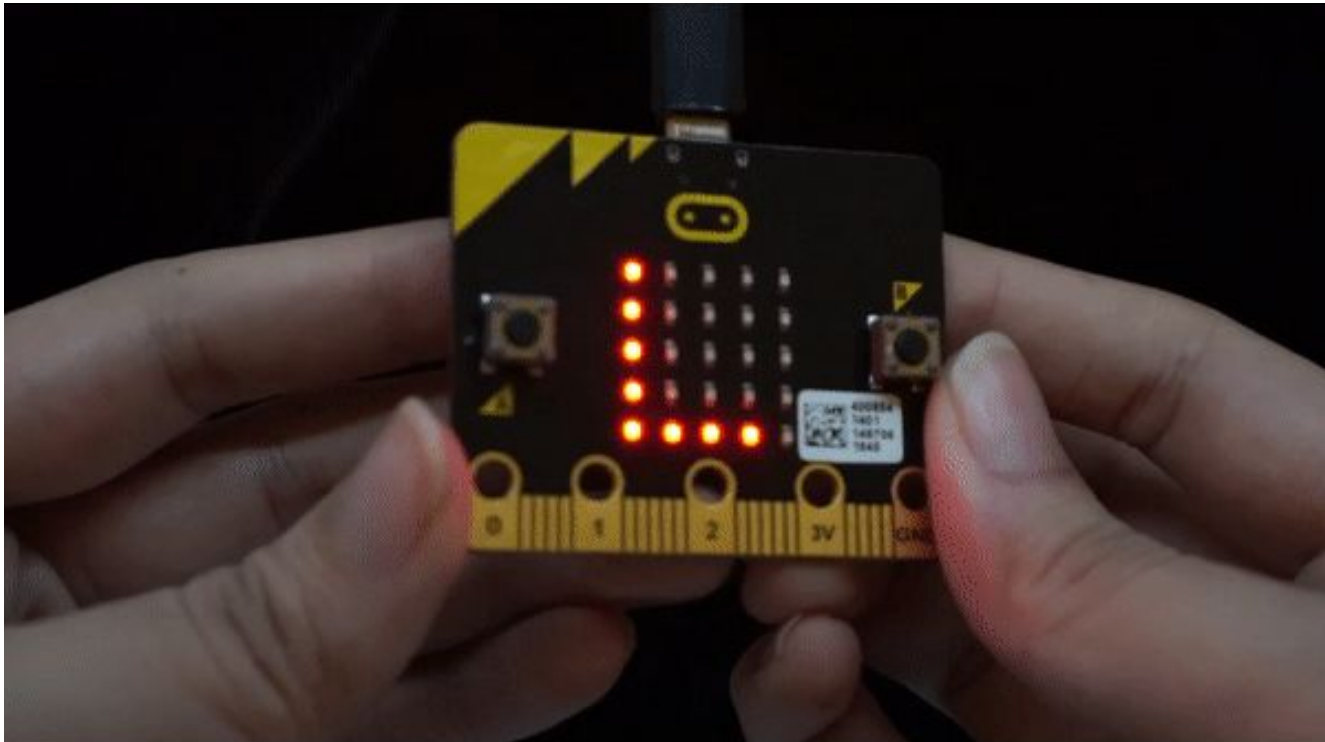
Guided

Conditional



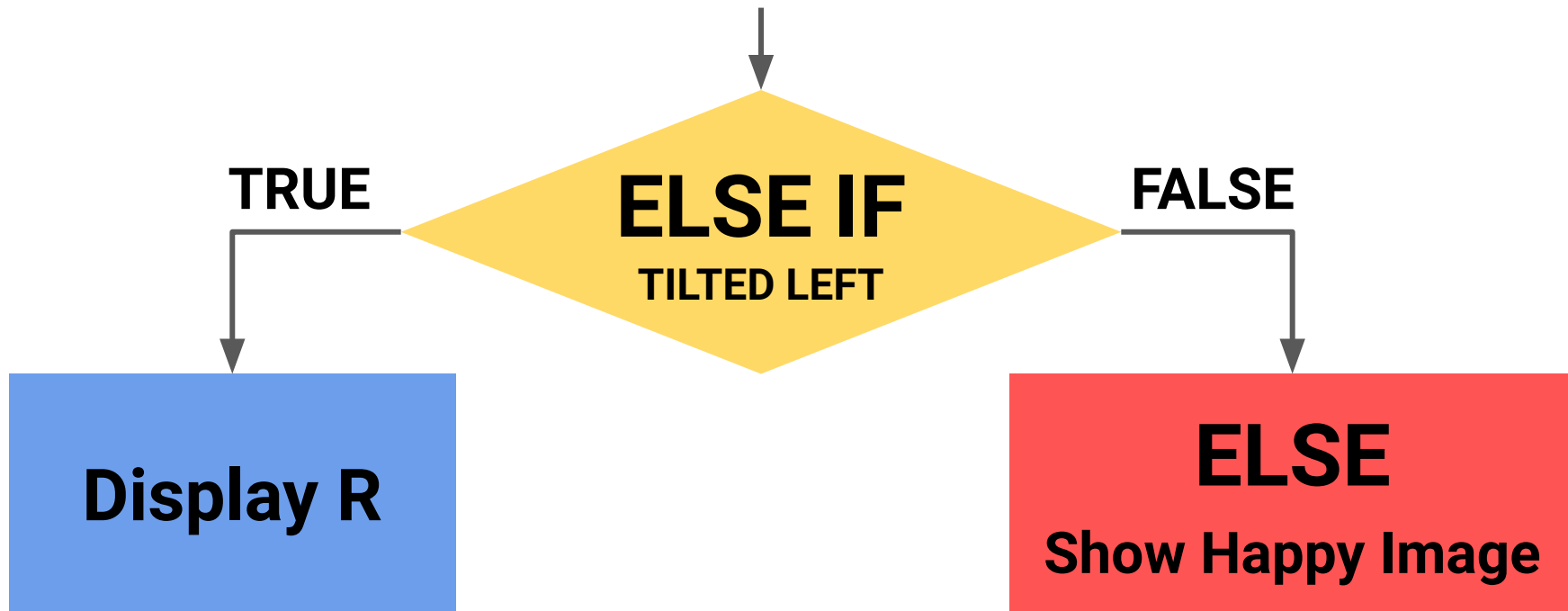
Conditionals

Else show a happy image





Conditional





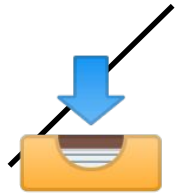
Write the code

- Else if micro:bit is tilted left
- Else micro:bit is balanced



Indentation

```
from microbit import *  
  
while True:  
    reading = accelerometer.get_x  
    if reading > 20:  
        display.show("L")  
    elif reading < -20:  
        display.show("R")  
    else:  
        display.show(Image.HAPPY)
```



Download the Project onto the micro:bit



TypeError

Line 6 TypeError unsupported types ...



Guided

How to Correct the Code

```
accelerometer.get_x()
```





**Give your project
a name and save it!**



Break

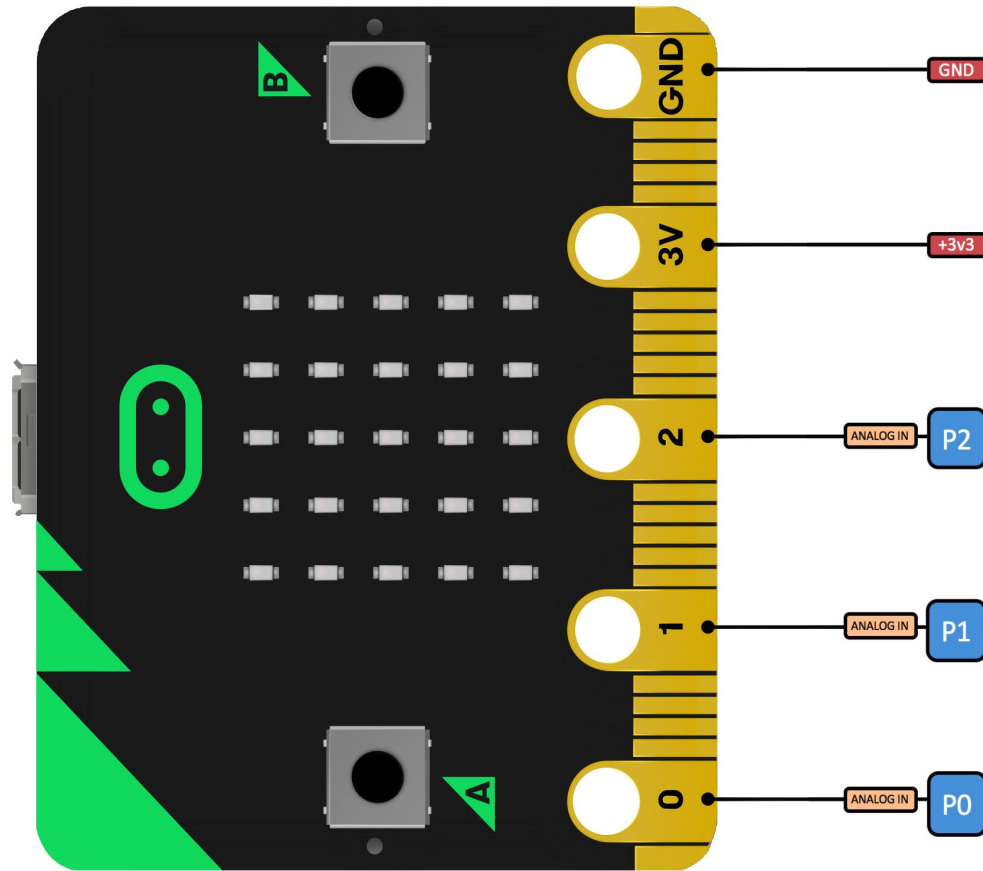


Music

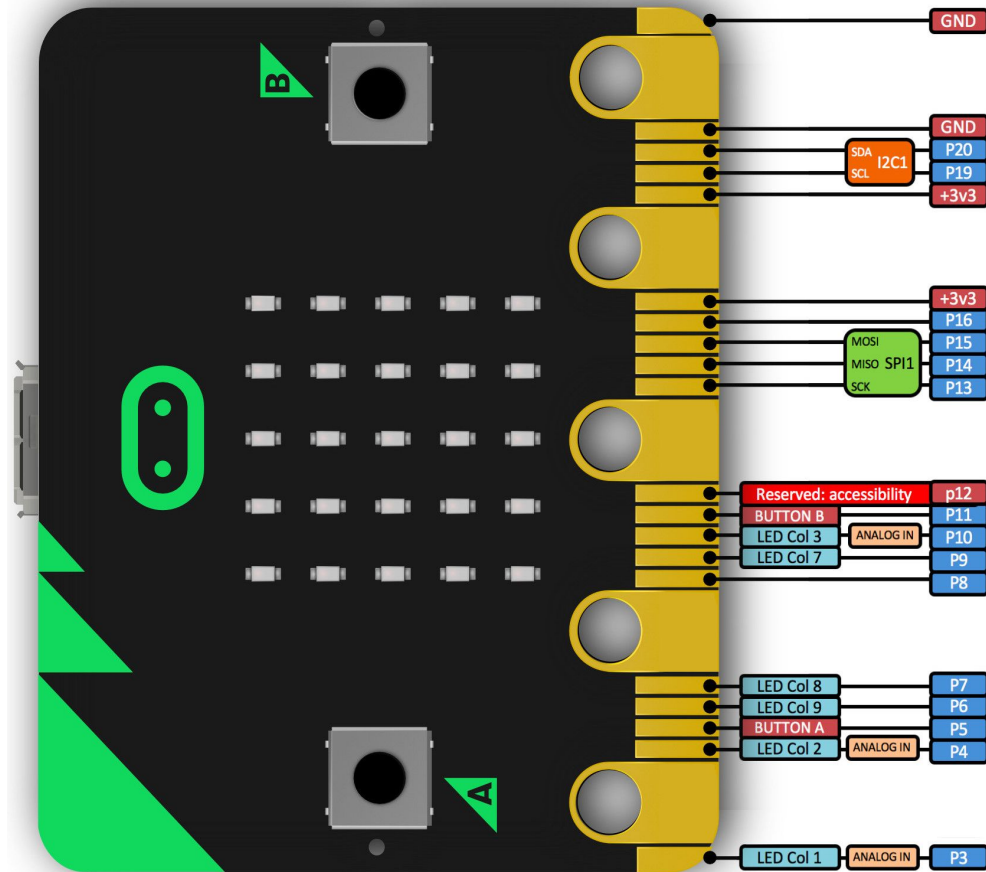


Introduction to Breakout Board

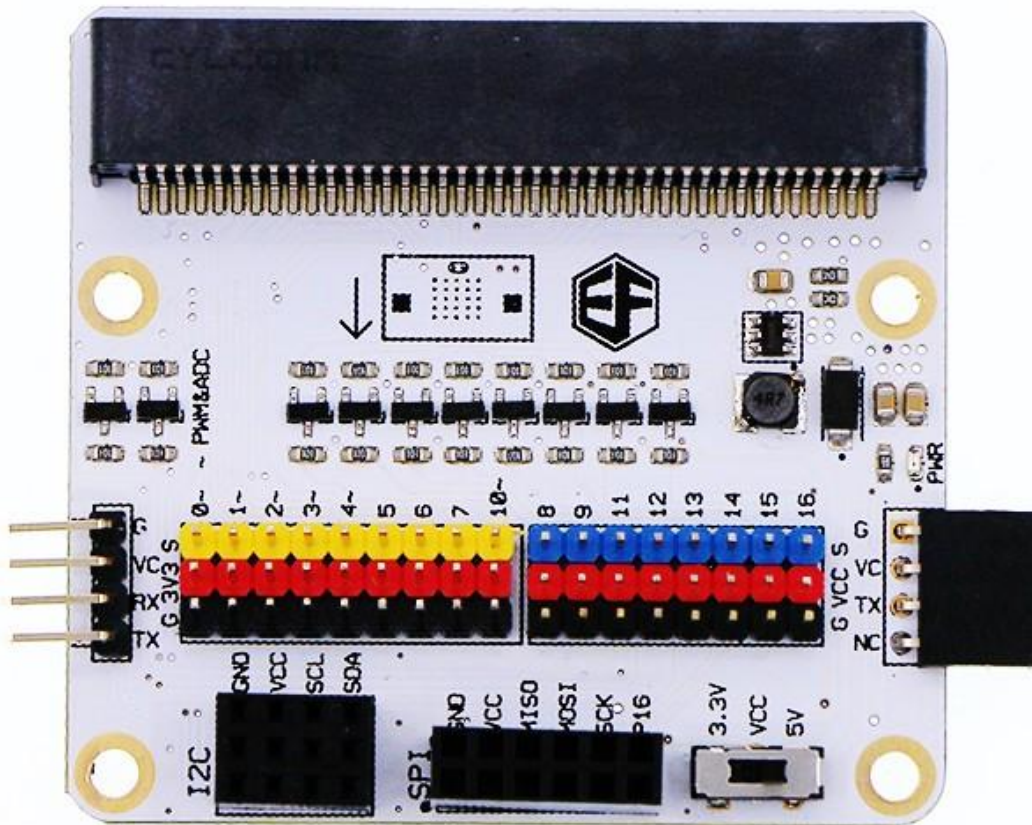
Large Pins

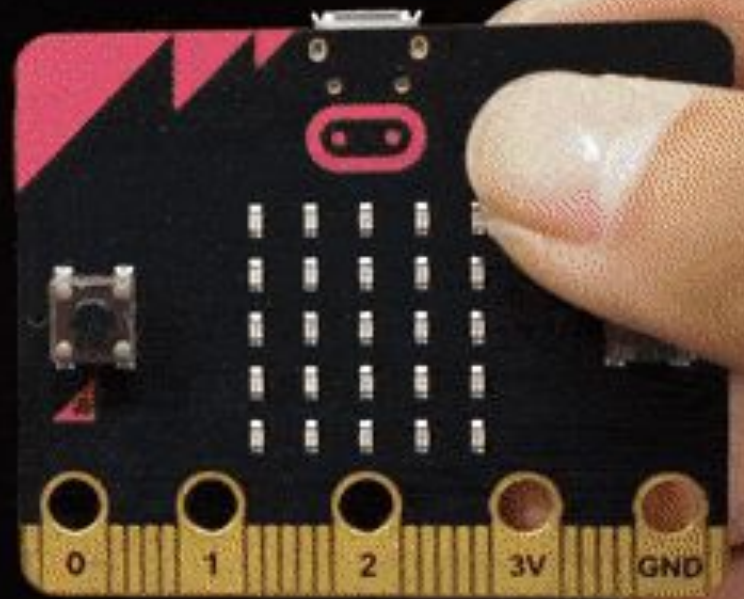


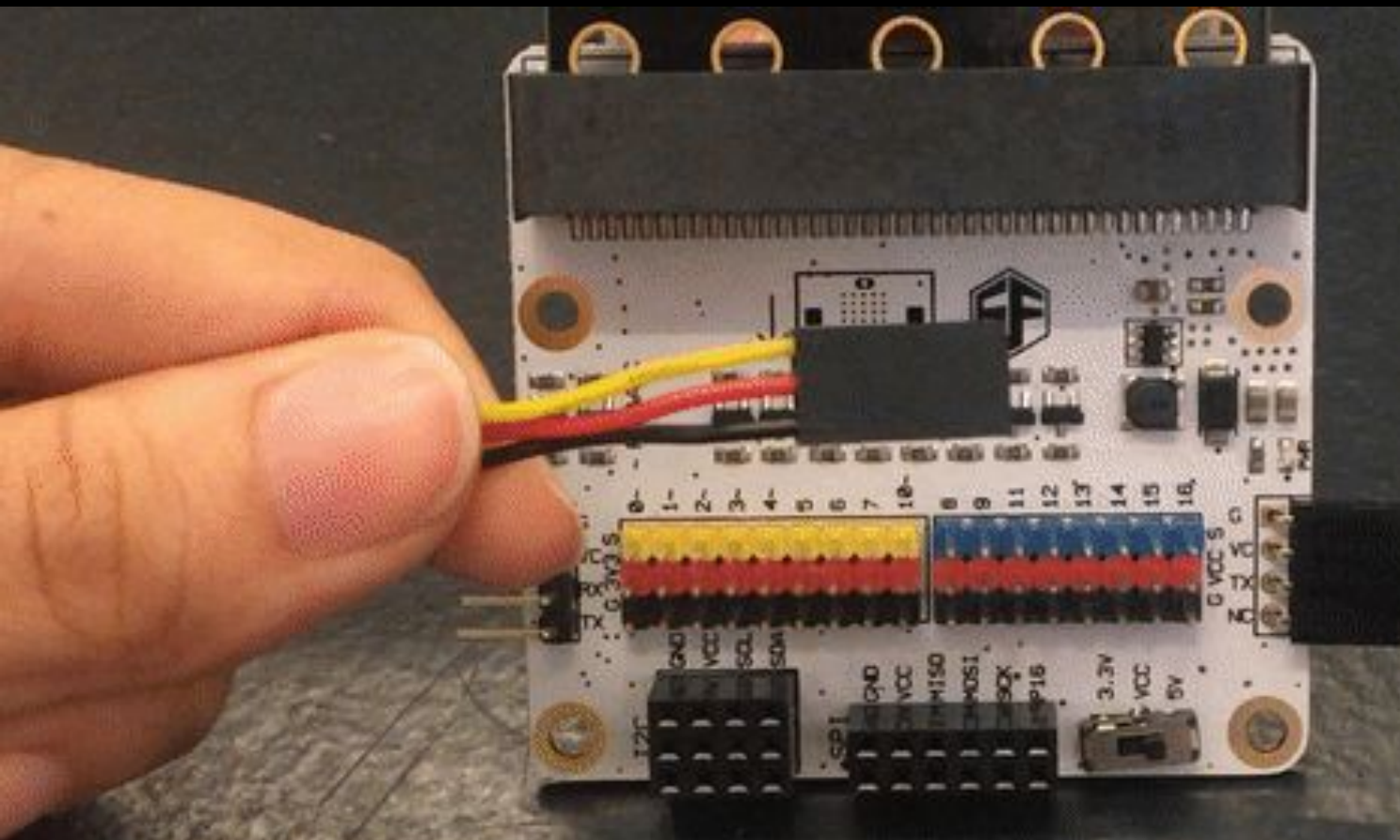
Small Pins



Breakout Board

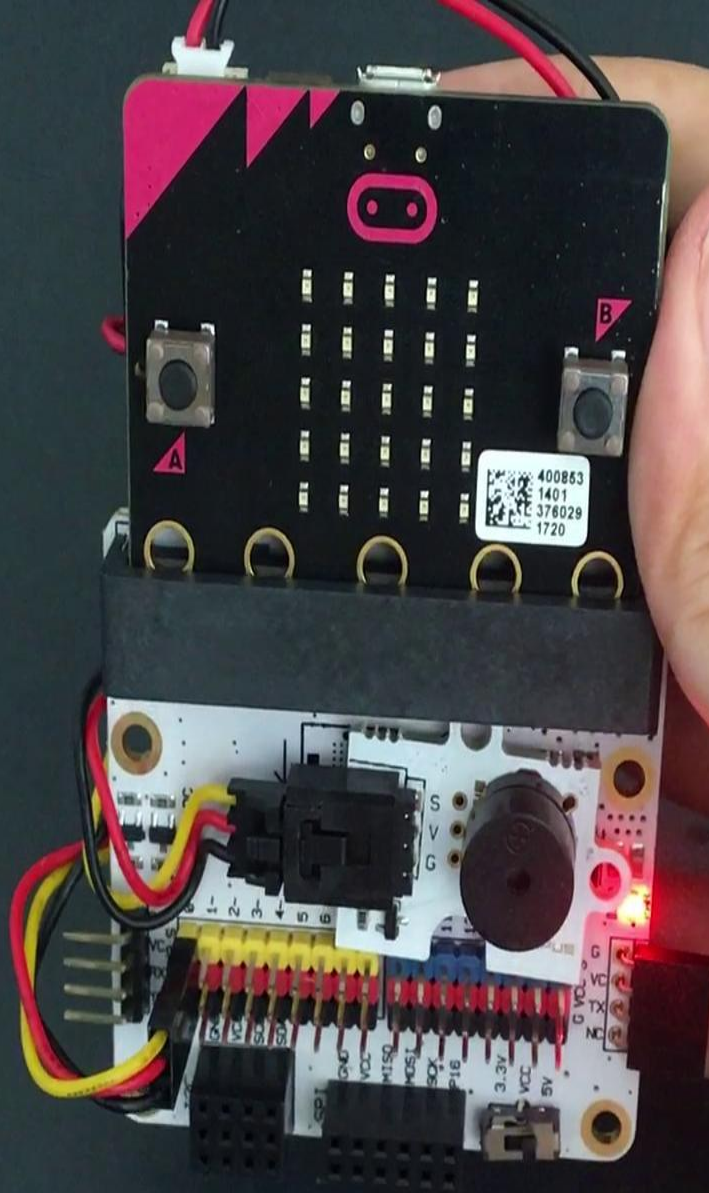








Accelerometer Music Player



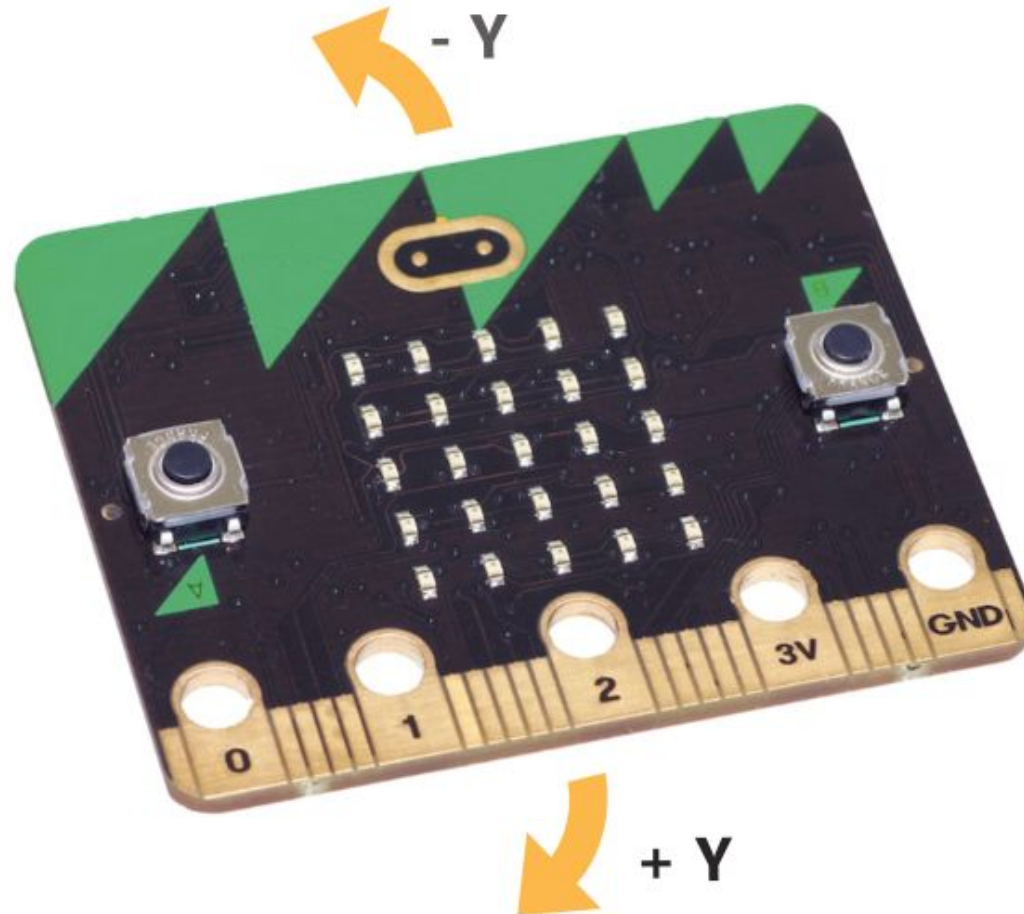
Programme the micro:bit

Play a music note and control the note by tilting the microbit along the y axis.



Guided

Y value of Accelerometer





Guided

Create a `while True` loop

Why?



Guided



**Find the Function that
would return the Y value of
the Accelerometer**



Y value of Accelerometer

```
microbit.accelerometer.get_y()
```

Get the acceleration measurement in the **y** axis, as a positive or negative integer, depending on the direction. The measurement is given in milli-g. By default the accelerometer is configured with a range of +/- 2g, and so this method will return within the range of +/- 2000mg.



Guided



Find the Function to play a music pitch



Guided

Music Pitch

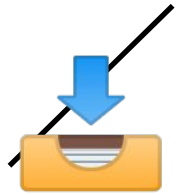
```
music.pitch(frequency, duration=-1, pin=microbit.pin0, wait=True)
```

Plays a pitch at the integer frequency given for the specified number of milliseconds. For example, if the frequency is set to 440 and the length to 1000 then we hear a standard concert A for one second.

Note that you can only play one pitch on one pin at any one time.

If `wait` is set to `True`, this function is blocking.

If `duration` is negative the pitch is played continuously until either the blocking call is interrupted or, in the case of a background call, a new frequency is set or `stop` is called (see below).



Download the Project onto the micro:bit

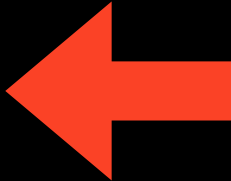
ValueError

Line 6 ValueError invalid pitch



How to Correct the Code

```
while True:
    freq = accelerometer.get_y()
    if freq < 0:
        freq = freq * -1
    music.pitch(freq, 100)
```



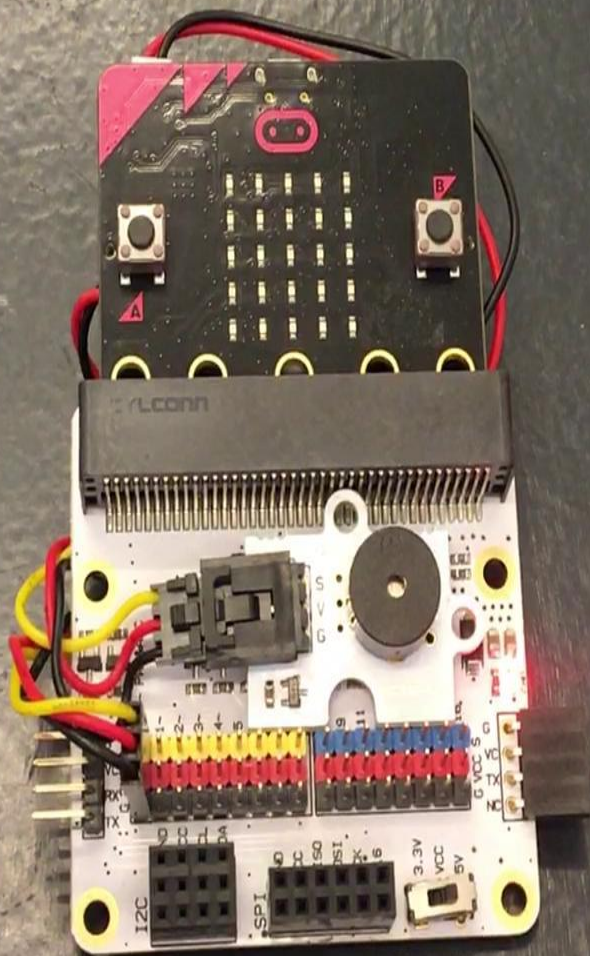


**Give your project
a name and save it!**

👐 Unguided



Movement Alarm





Programme the micro:bit

When the microbit is moved, display an X and repeatedly play music. If button A is pressed, stop the music.

 Unguided

 Hints

**Find accelerometer gestures,
code to play music and the list
of melodies in the microbit
documentation**



Check the microbit's Movement

! Note

MicroPython understands the following gesture names: "up", "down", "left", "right", "face up", "face down", "freefall", "3g", "6g", "8g", "shake". Gestures are always represented as strings.

```
microbit.accelerometer.is_gesture(name)
```

Return True or False to indicate if the named gesture is currently active.



Guided

Play a Melody

```
music.play(music, pin=microbit.pin0, wait=True, loop=False) 🔗
```

Plays `music` containing the musical DSL defined above.

If `music` is a string it is expected to be a single note such as, `'c1:4'`.

If `music` is specified as a list of notes (as defined in the section on the musical DSL, above) then they are played one after the other to perform a melody.

In both cases, the `duration` and `octave` values are reset to their defaults before the music (whatever it may be) is played.

An optional argument to specify the output pin can be used to override the default of `microbit.pin0`.

If `wait` is set to `True`, this function is blocking.

If `loop` is set to `True`, the tune repeats until `stop` is called (see below) or the blocking call is interrupted.



Some Available Melodies

- **DADADADUM** - the opening to Beethoven's 5th Symphony in C minor.
- **ENTERTAINER** - the opening fragment of Scott Joplin's Ragtime classic "The Entertainer".
- **PRELUDE** - the opening of the first Prelude in C Major of J.S.Bach's 48 Preludes and Fugues.
- **ODE** - the "Ode to Joy" theme from Beethoven's 9th Symphony in D minor.
- **NYAN** - the Nyan Cat theme (<http://www.nyan.cat/>). The composer is unknown. This is fair use for educational porpoises (as they say in New York).
- **RINGTONE** - something that sounds like a mobile phone ringtone. To be used to indicate an incoming message.
- **FUNK** - a funky bass line for secret agents and criminal masterminds.
- **BLUES** - a boogie-woogie 12-bar blues walking bass.
- **BIRTHDAY** - "Happy Birthday to You..." for copyright status see: <http://www.bbc.co.uk/news/world-us-canada-34332853>



Stopping the music

```
music.stop(pin=microbit.pin0)
```

Stops all music playback on a given pin, eg. `music.stop(pin1)`. If no pin is given, eg. `music.stop()` pin0 is assumed.



**Give your project
a name and save it!**