**Abstract**

Digitalizing Spirit level is a tool for making sure pictures, shelves or work surfaces are level. The new micro:bit's built-in speaker makes it easy to improve your spirit level with audio feedback. The micro:bit's accelerometer can measure angles of tilt in two directions: up and down (pitch) and side to side (roll). This project uses this to show when the micro:bit is level on the LED display and by making a sound, which could be useful when hanging a picture or making a work surface.

**Introduction:**

A Spirit level is a tool that helps you find a line that is perfectly horizontal or vertical, useful if you are hanging a picture or making sure a work surface is level. Many phones also have Spirit level apps which use an accelerometer sensor measuring the Spirit level. Earth's gravity to show when they are level. The micro:bit also has an accelerometer.

**Existing Tool:**



**Proposed Tool:**

|  |  |
| --- | --- |
| Not in Level | In Level |

**Proposed System**

**What is it?**

Make a tool for making sure pictures, shelves or work surfaces are level. The new micro:bit's built-in speaker makes it easy to improve your spirit level with audio feedback.

**What is micro:bit:**

The **Micro Bit** (also referred to as **BBC Micro Bit**, stylized as **micro:bit**) is an [open source hardware](https://en.wikipedia.org/wiki/Open_source_hardware) [ARM](https://en.wikipedia.org/wiki/ARM_architecture)-based [embedded system](https://en.wikipedia.org/wiki/Embedded_system) designed by the [BBC](https://en.wikipedia.org/wiki/BBC) for use in computer education in the [United Kingdom](https://en.wikipedia.org/wiki/United_Kingdom).

The device is described as half the size of a credit card and has an [ARM Cortex-M0](https://en.wikipedia.org/wiki/ARM_Cortex-M0) processor, accelerometer and magnetometer sensors, Bluetooth and USB connectivity, a display consisting of 25 [LEDs](https://en.wikipedia.org/wiki/Light-emitting_diode), two programmable buttons, and can be powered by either USB or an external battery pack. The device inputs and outputs are through five ring connectors that form part of a larger 25-pin edge connector. In October 2020, a physically nearly identical v2 board was released that features a [Cortex-M4](https://en.wikipedia.org/wiki/Cortex-M4)F microcontroller, with more memory and other new features.

Hardware

**v1**

The physical board measures 43 mm × 52 mm and, in its first production run, included:

[](https://en.wikipedia.org/wiki/File:BBC_Micro_Bit_with_original_Packaging.jpg)

micro:bit v1 with its original packaging behind it

* [Nordic nRF51822](https://en.wikipedia.org/wiki/NRF51_series) – 16 [MHz](https://en.wikipedia.org/wiki/Hertz) 32-bit [ARM Cortex-M0](https://en.wikipedia.org/wiki/ARM_Cortex-M0) [microcontroller](https://en.wikipedia.org/wiki/Microcontroller), 256 KB [flash memory](https://en.wikipedia.org/wiki/Flash_memory), 16 KB [static RAM](https://en.wikipedia.org/wiki/Static_random-access_memory), 2.4 [GHz](https://en.wikipedia.org/wiki/Hertz) [Bluetooth low energy](https://en.wikipedia.org/wiki/Bluetooth_low_energy) wireless networking. The ARM core has the capability to switch between 16 MHz or 32.768 kHz.
* NXP/Freescale KL26Z – 48 MHz [ARM Cortex-M0+](https://en.wikipedia.org/wiki/ARM_Cortex-M0%2B) core microcontroller, that includes a full-speed USB 2.0 [On-The-Go](https://en.wikipedia.org/wiki/On-The-Go) (OTG) controller, used as a communication interface between [USB](https://en.wikipedia.org/wiki/USB) and main Nordic microcontroller. This device also performs the voltage regulation from the USB supply (4.5-5.25 V) down to the nominal 3.3 volts used by the rest of the PCB. When running on batteries this regulator is not used.
* NXP/Freescale MMA8652 – 3-axis [accelerometer](https://en.wikipedia.org/wiki/Accelerometer) sensor via [I²C](https://en.wikipedia.org/wiki/I%C2%B2C)-bus.
* NXP/Freescale MAG3110 – 3-axis [magnetometer](https://en.wikipedia.org/wiki/Magnetometer) sensor via I²C-bus (to act as a compass and metal detector).
* [MicroUSB](https://en.wikipedia.org/wiki/USB#Mini_and_micro_connectors) connector, battery connector, 25-pin [edge connector](https://en.wikipedia.org/wiki/Edge_connector).
* Display consisting of 25 [LEDs](https://en.wikipedia.org/wiki/Light-emitting_diode) in a 5×5 array.
* Three tactile pushbuttons (two for applications, one for reset).

I/O includes three ring connectors (plus one power one ground) which accept [crocodile clips](https://en.wikipedia.org/wiki/Crocodile_clip) or 4 mm [banana plugs](https://en.wikipedia.org/wiki/Banana_plug) as well as a 25-pin edge connector with two or three [PWM](https://en.wikipedia.org/wiki/Pulse-width_modulation) outputs, six to 17 [GPIO](https://en.wikipedia.org/wiki/GPIO) pins (depending on configuration), six analog inputs, serial I/O, [SPI](https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus), and [I²C](https://en.wikipedia.org/wiki/I%C2%B2C). Unlike early prototypes, which had an integral battery, an external battery pack (AAA batteries) can be used to power the device as a standalone or wearable product. Health and safety concerns, as well as cost, were given as reasons for the removal of the button battery from early designs.

The available hardware design documentation consist of only the schematic and [BOM](https://en.wikipedia.org/wiki/Bill_of_Materials) distributed under the [Creative Commons By Attribution](https://en.wikipedia.org/wiki/CC_BY) license, no [PCB layout](https://en.wikipedia.org/wiki/PCB_layout) is available. The compatible reference design by Micro:bit Educational Foundation, however, is fully documented.

**v2**

v2, released on 13 October 2020, includes:

[](https://en.wikipedia.org/wiki/File:Micro-bit_v2.JPG)

micro:bit v2 with its original packaging behind it

* Nordic nRF52833 – 64 [MHz](https://en.wikipedia.org/wiki/Hertz) 32-bit [ARM Cortex-M4](https://en.wikipedia.org/wiki/ARM_Cortex-M4) [microcontroller](https://en.wikipedia.org/wiki/Microcontroller), 512 KB [flash memory](https://en.wikipedia.org/wiki/Flash_memory), 128 KB [static RAM](https://en.wikipedia.org/wiki/Static_random-access_memory), 2.4 [GHz](https://en.wikipedia.org/wiki/Hertz) [Bluetooth low energy](https://en.wikipedia.org/wiki/Bluetooth_low_energy) wireless networking provided by Nordic S113 SoftDevice, integrated temperature sensor.
* NXP/Freescale KL27Z – 48 MHz [ARM Cortex-M0+](https://en.wikipedia.org/wiki/ARM_Cortex-M0%2B) core microcontroller, preprogrammed as a full-speed USB 2.0 controller, used as a communication interface between [USB](https://en.wikipedia.org/wiki/USB) and the CPU.
* Either [ST](https://en.wikipedia.org/wiki/STMicroelectronics) LSM303 or [NXP](https://en.wikipedia.org/wiki/NXP_Semiconductors) FXOS8700 – 3-axis combined [accelerometer](https://en.wikipedia.org/wiki/Accelerometer) and [magnetometer](https://en.wikipedia.org/wiki/Magnetometer) sensor via [I²C](https://en.wikipedia.org/wiki/I%C2%B2C)-bus.
* Knowles [MEMS](https://en.wikipedia.org/wiki/MEMS) microphone with a built-in LED indicator.
* Jiangsu Huaneng MLT-8530 magnetic visual.
* [MicroUSB](https://en.wikipedia.org/wiki/USB#Mini_and_micro_connectors) connector, [JST PH](https://en.wikipedia.org/wiki/JST_connector) battery connector, 25-pin [edge connector](https://en.wikipedia.org/wiki/Edge_connector).
* Display consisting of 25 [LEDs](https://en.wikipedia.org/wiki/Light-emitting_diode) in a 5×5 matrix.
* Three tactile pushbuttons (two for applications, one for reset) and a touch sensor button.

In micro:bit v2, the reset button can be used to turn the board off by holding it for 3 seconds.

Software

There are three official code editors on the [micro:bit foundation web site](https://microbit.org/code/):

* [Microsoft MakeCode](http://makecode.microbit.org/)
* [MicroPython](https://en.wikipedia.org/wiki/MicroPython)
* [Scratch](https://scratch.mit.edu/microbit)

The Python programming experience on the Micro Bit is provided by [MicroPython](https://en.wikipedia.org/wiki/MicroPython" \o "MicroPython). Users are able to write Python scripts in the Micro Bit web editor which are then combined with the MicroPython firmware and uploaded to the device. Users can also access the MicroPython [REPL](https://en.wikipedia.org/wiki/Read-eval-print_loop) running directly on the device via the USB serial connection, which allows them to interact directly with the Micro Bit's peripherals.

The Micro Bit was created using the ARM [mbed](https://en.wikipedia.org/wiki/Mbed" \o "Mbed) development kits. The run-time system and programming interface utilize the mbed cloud compiler service to compile the user's code into a .UF2 file. The compiled code is then flashed onto the device using USB or Bluetooth connections. The device appears as a USB drive when connected to a computer, and code can be flashed dragging and dropping the .UF2 file.

Other editors for the BBC micro:bit include:

* [Mu](http://codewith.mu/), a Python editor
* [Espruino](http://www.espruino.com/MicroBit), a JavaScript interpreter
* [EduBlocks](https://edublocks.org/microbit.html), a block editor for MicroPython

Other programming languages for the BBC micro:bit include:

* [Free Pascal](https://en.wikipedia.org/wiki/Free_Pascal) ([instructions](http://wiki.freepascal.org/micro:bit))
* [Simulink in Matlab](https://en.wikipedia.org/wiki/Simulink) ([Simulink Coder Support Package for BBC micro:bit Board](https://www.mathworks.com/matlabcentral/fileexchange/60273-simulink-coder-support-package-for-bbc-micro-bit-board)) signal logging, parameter tuning, code development from the Simulink block editor.
* [C++](https://en.wikipedia.org/wiki/C%2B%2B) ([instructions](https://lancaster-university.github.io/microbit-docs/))
* [Forth](https://en.wikipedia.org/wiki/Forth_(programming_language)) ([instructions](https://wiki.forth-ev.de/doku.php/en:projects:microbit:start))
* [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)) ([instructions](http://www.ulisp.com/show?2672))
* [Rust](https://en.wikipedia.org/wiki/Rust_(programming_language)) ([instructions](https://docs.rust-embedded.org/discovery/microbit/))
* [Ada](https://en.wikipedia.org/wiki/Ada_(programming_language)) ([instructions](https://blog.adacore.com/ada-on-the-microbit))
* [Swift](https://en.wikipedia.org/wiki/Swift_(programming_language)) ([instructions](https://microbit.org/guide/swift-playgrounds/))
* [BASIC](https://en.wikipedia.org/wiki/BASIC) ([instructions](https://www.coridium.us/coridium/blog/basic-for-microbit))

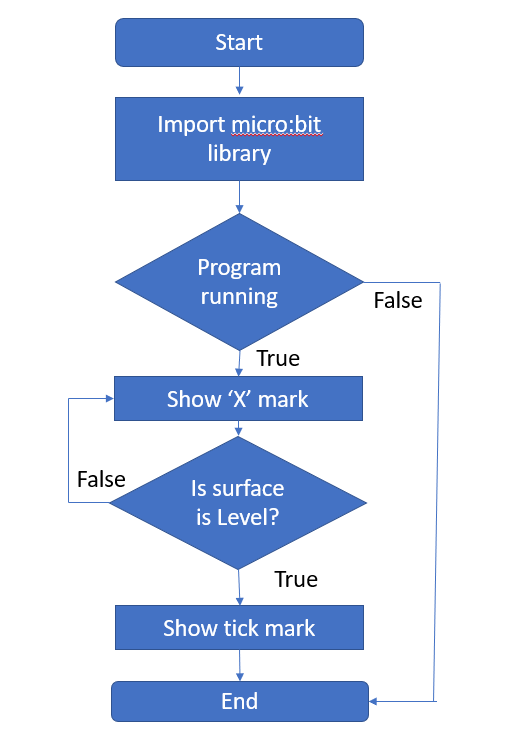
Operating systems which can be built for the BBC micro:bit:

* [Zephyr](https://en.wikipedia.org/wiki/Zephyr_(operating_system)) - the Zephyr lightweight OS comes with the required parameters file to be able to run it on this board.

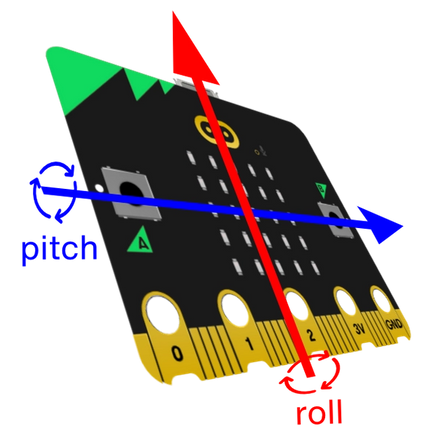
**What we'll learn**

* How to use the accelerometer sensor to measure angles
* What 'pitch' and 'roll' are
* How to combine sensor readings in two dimensions to find a level surface

**Flow diagram:**



**How it works**



* The micro:bit's accelerometer can measure angles of tilt in two directions: up and down (pitch) and side to side (roll). This project uses this to show when the micro:bit is level on the LED display and by making a sound, which could be useful when hanging a picture or making a work surface.
* A loop keeps the program constantly measuring the micro:bit's angle.
* If the accelerometer measures an angle of tilt between +5 **and** -5 degrees of both pitch **and** roll, the micro:bit must be reasonably level. It then shows a tick on the LED display output and plays a musical tone.
* To hear the tone attach headphones or a speaker to pin 0 and GND, or if you have a new micro:bit you will hear it on the built-in speaker.
* If **either** the pitch **or** the roll is outside the range +5 to -5 degrees, it shows a cross on the LED display and stops the sound.

**Import a program in Microsoft MakeCode:**

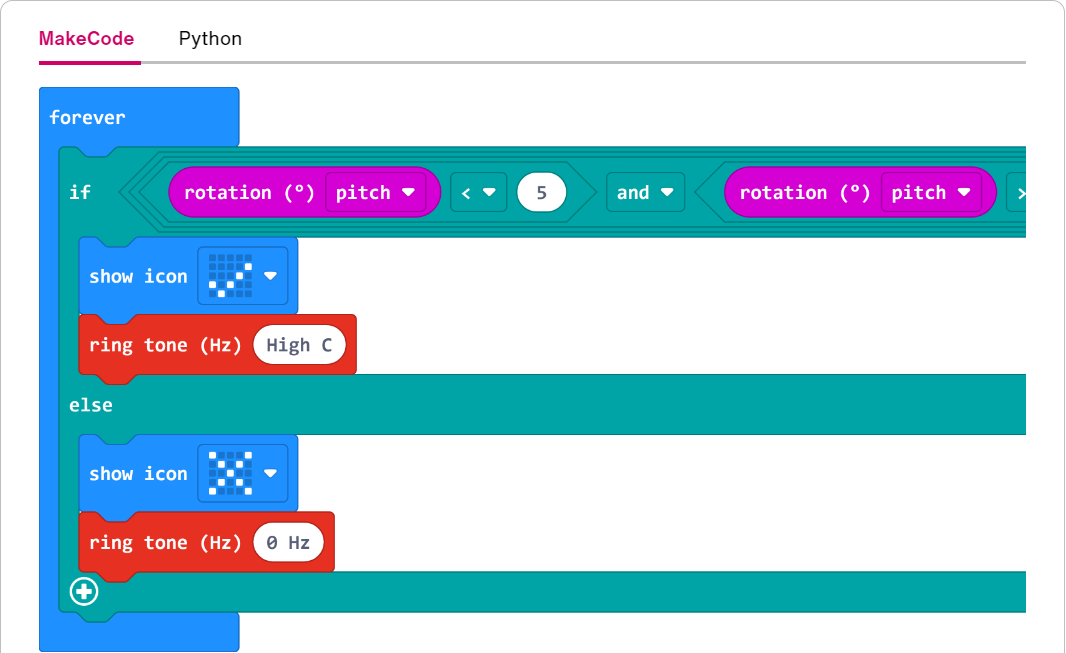
1. In a browser, go to [makecode.microbit.org](https://makecode.microbit.org/) or open the app
2. Enter the URL [makecode.microbit.org/\_amDYa3KdqU5w](https://makecode.microbit.org/_amDYa3KdqU5w)
3. The smile animation project will open

**Download a MakeCode program to micro:bit**

1. Connect the micro:bit to your computer using the micro-USB cable
2. In the open MakeCode project, select Download
3. Move the downloaded .hex file to the MICROBIT drive

**Coding:**

**MakeCode**



Python

[1from microbit import \*](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[2import music](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[3](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[4# Uses accelerometer readings in the x and y axis](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[5# and also allows the micro:bit to be slightly off-level](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[6# to make it work better in practice](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[7while True:](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[8 if accelerometer.get\_x() > -10 and accelerometer.get\_x() < 10 and accelerometer.get\_y() > -10 and accelerometer.get\_y() < 10:](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[9 display.show(Image.YES)](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[10 music.play('C5:1')](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[11 sleep(200)](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[12 else:](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[13 display.show(Image.NO)](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

[14](https://classroom.microbit.org/?id=spirit-level&project=Project%3A%20Spirit%20level&name=Spirit%20level&editors=makecode%2Cpython)

**Software/Hardware Requirement:**

* **Hardware Requirement:**
* A micro:bit
* MakeCode or Python editor
* battery pack (optional)
* **Software Requirement:**
* Code block
* Python

**Output:**

|  |  |
| --- | --- |
| **Not in Level** | **In Level** |

**Future Enhancement**

* We can try to make the musical pitch change depending on the angle.
* We can try to code an LED 'bubble' that moves around the screen like a real spirit level.

**References /Bibliography:**

* [**https://microbit.org/**](https://microbit.org/)
* [**https://makecode.microbit.org/**](https://makecode.microbit.org/)
* [**https://microbit.org/projects/make-it-code-it/spirit-level/?editor=makecode**](https://microbit.org/projects/make-it-code-it/spirit-level/?editor=makecode)