# Team H.A.S.S

# BTEC Robotics Assignment 3

# Project Micro:bit Pedometer

# The Team + Roles:

| Team Member Name: | Role: | Skills: | Entails/Details: |
|---|---|---|---|
| Harry Fitzgerald-Smith | Project Manager/ Programmer | • Communication<br>• Time management<br>• Leadership<br>• Java-Script + Python Experience<br>• Management skills | • Creating plans for Project, Assigning Roles<br>• Conducting research for project<br>• Time management, Resource admin and overseeing<br>• Creating the code for the project |
| Sara Khan | Guide Creator - Writing and Structuring | • Written communication<br>• Collaboration<br>• Concision<br>• Communication skills<br>• Editing + Formatting | • Creating the written part of the step-by-step Guide<br>• Be a part of creating the layout for the step-by-step guide<br>• Structuring the guide to be effective and useful |
| Sumaya Hassan | Guide Creator - Drawings + Images | • Collaboration<br>• Creative thinking<br>• Formatting<br>• Design<br>• Graphics Design | • Creating drawings for the step-by-step guide<br>• Making the guide be visibly intriguing<br>• Creating visuals for the step-by-step guide |
| Angel Smith | Guide Creator (Supporting) + Assembling/Designing | • Collaboration<br>• Written communication<br>• Logic<br>• Creative Thinking<br>• Graphics design | • Supporting Sumaya and Sara in creating the guide<br>• Creating visuals and writing |

| | | | |
|---|---|---|---|
| | | | • Designing how the Pedometer should look. |

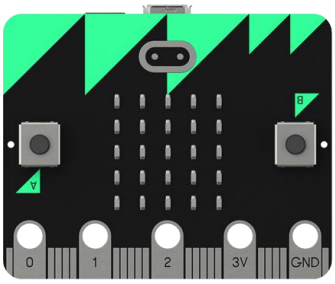
TEAM H.A.S.S

# How To Micro:bit - Creating a pedometer


*In this how to guide, you will be taken through a series of steps on how to make a pedometer using a micro:bit. From this you will learn a variety of functions that a microbit can do, and learn to interact with the microbit software in order to program the microbit*

# Preface:

## Introduction to Micro:Bits:




Microbits are miniature Arduino devices, developed by the BBC, that has a 5x5 LED Light system embedded into it - which users are able to control each LED light by performing algorithms.
The base of a Micro:bit acts as a motherboard, which allows it to act as a miniature computer that users can code. The specs of the Micro:Bit include 16k of RAM, and a 32-Bit CPU, which allow it to serve as a dedicated system to run a single program which the user can code.
The Micro:Bit itself is a very small device in comparison to the Arduino, compacting breadboards, the motherboard and LEDs into a single device; which allows it to be carried around and be used in more practical situations.


The device bases itself mainly on Java-script, which, for learning purposes, is generalised into a block form; but can also be coded using pure Javascript or Python by importing from the Micro:Bit Repository.
On the front of the device, two buttons, A and B, can be coded to conduct specific algorithms in the program - which allows for additional user input and functionality in the programs to make them more practical.

## Introduction to the Project:

Built on the board of the Micro:bit is an accelerometer. These are small devices that can detect motion with the device in any direction; meaning that they can be used to detect body movement in a practical

situation; including steps.

Accelerometers are used in most conventional devices, such as fitbits and even modern smartphones, to count the steps that the user takes in a day, based upon the movement occurring when the device is on. Due to the Micro:bits being compact and portable with a power supply, possessing an accelerometer on the board, this would mean that users could code their own custom version of a pedometer that can be carried around throughout the day.

Based upon statistics provided by the NHS, people are more likely to live a healthier life if they walk at least 10'000 steps a day. Using a pedometer can be used to evaluate the lifestyle you have, and try to achieve these targets. Because of the customisable capability of a Micro:bit, and there being a few pieces of equipment it could be more valuable for usability in opposition to commercial Fitbits. Commercial Fitbits are generally non customisable, and generally only count the number of steps with little user interaction.

## Estimated Timings:

**Set Up: 10 Mins**

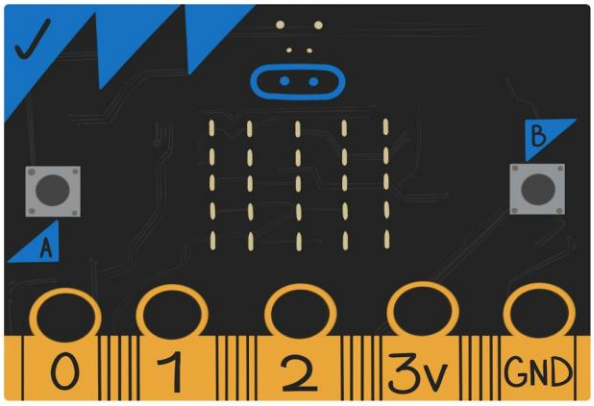
**Coding: 30 Mins**

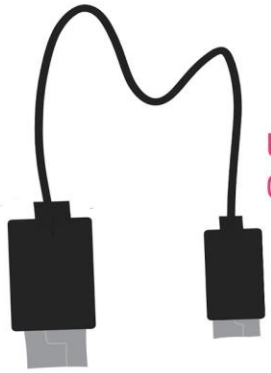
**Sewing/Refining: 20 mins**

**Total Estimated Time for Project: 60 Mins**

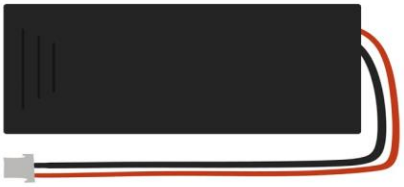
# Equipment Needed:

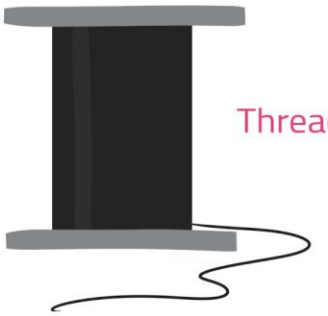
Micro:Bit



USB Cable





Thread

Carbon Zinc / Alkaline batteries

A belt (to attach everything to!)

Velcro

# Method:

## Step 1:

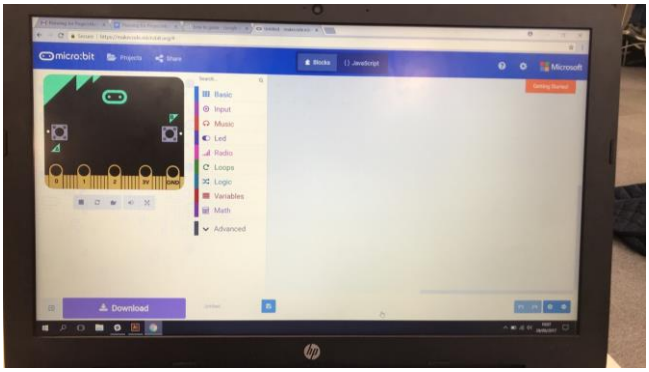
Plug the USB to the microbit in one of the USB ports and ensure it is connected and secure and place the microbit in a reachable and safe place.

## Step 2:

Plug the receiving end into the laptop, the microbit should light up to indicate that it is connected with a yellow light.

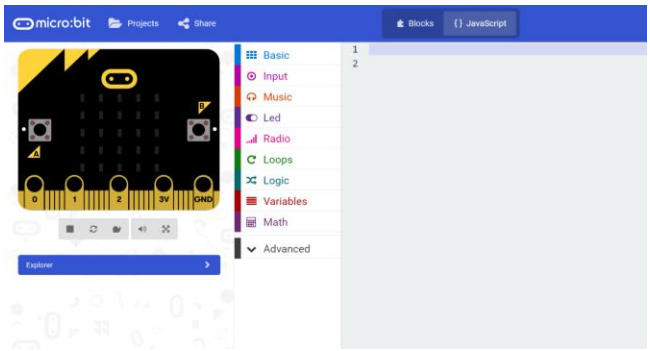
## Step 3:

Go to https://makecode.microbit.org/ , click on the appropriate programming language to program the microbit. You can choose to either code using a block-based method or using Java-Script.



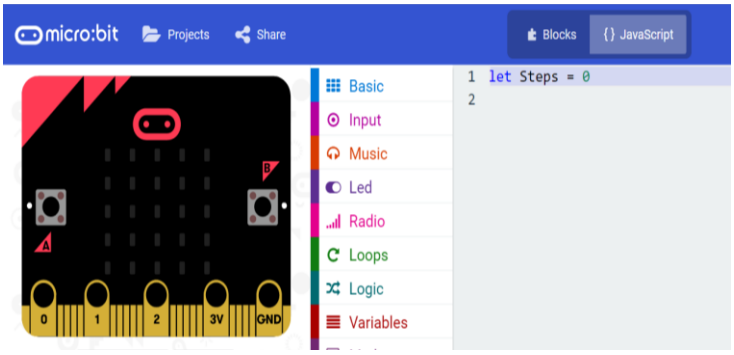
## Step 4:

For this example, select Javascript for an option of programming as it is a suitable program for beginners and is easy to code for a microbit (As a high-level language)



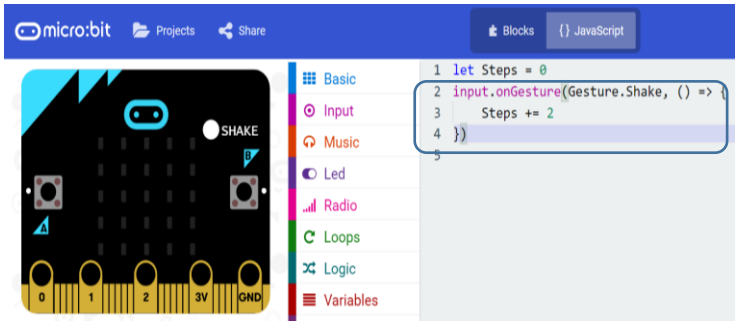

## Step 5A:

Create a variable for the steps (“Steps”) and set the value to zero. This will mean at the start of the program, the number of steps will be set at none.




## Step 5B:

Add 2 to the step variable each time the gesture, “shake”, is detected by the Micro:bit as an input. When the device is attached, the walking motion will resultantly shake the device on the leg it is parallel to. As humans make motions with two legs, we have to add an additional step so that motion in both legs is detected.

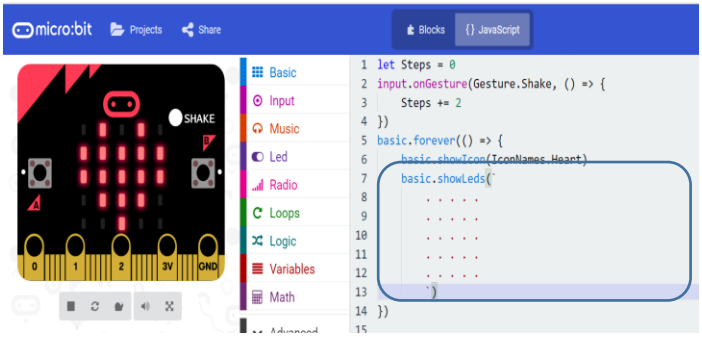


# Step 5C:

i) Time to be creative! To identify that the device is operating correctly, use the LED lights to form an image that will identify if the device is on and active; which will remain on the screen in a forever loop until certain conditions are met. One example of this a heart, which is already stored as an icon in the console, which may be logical in the health context of this device

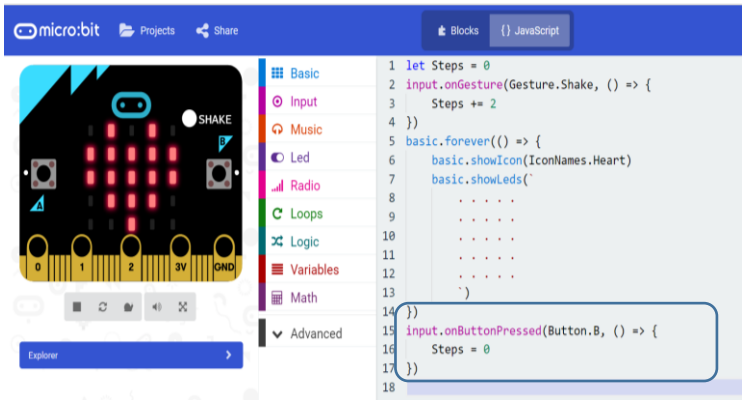

.

ii) To add clear identification that the Micro-bit is running the program - in the forever loop, we can clear the LED screen to make the heart flash when none of the buttons are pressed, so that we know that the program is operating. To do this, we must identify the LED lights we wish to turn off by representing a 5x5 grid with dots in the function.



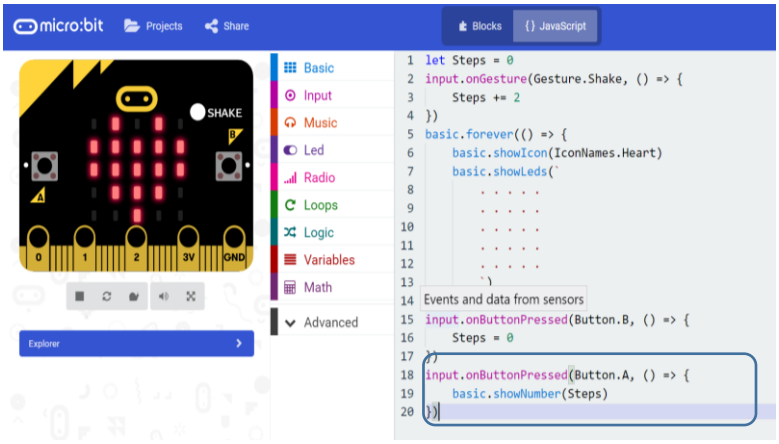

# Step 5D:

Now we need to program a control for the user to reset their steps. For this, we must assign the B button, on press, to set the step variable to become zero once again. This will allow the user a manual form of being able to count their steps over the duration of a few days.



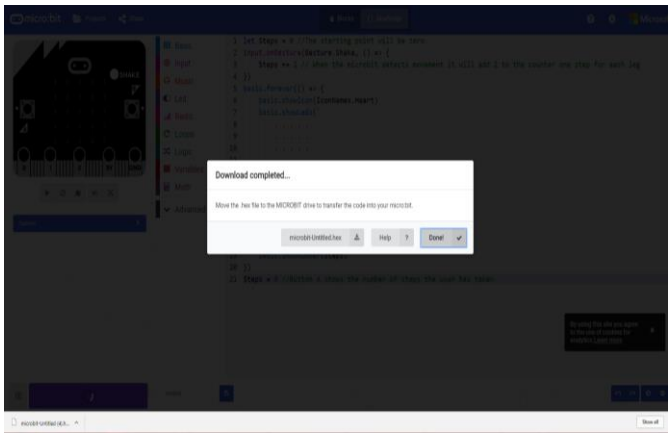
## Step 5E:

Finally, for the user to see the number of steps they have done, they must be able to see the contents of the step variable. For this, we must assign the A button, on press, to show the contents of the step variable on the LED screen.




## Step 5G:

Download the hex file onto your computer and then locate your Micro:bit portable device folder on your computer. Once this folder has been found, drag and drop the hex file into this folder, and then wait for the file to copy. Once the file is in the Micro:Bit Device, the LED image should form on the screen of your Micro:Bit.

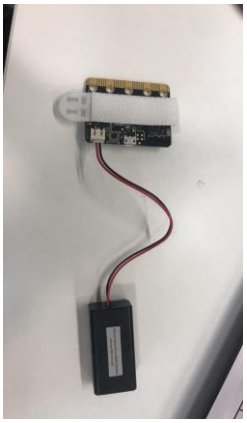


## Step 6:

Detach the microbit from the computer and attach the battery to the microbit. We use the battery to power the microbit so we can make the microbit portable. Attach the microbit to the strap using a sticky back Velcro as we will use this to secure it onto the the belt

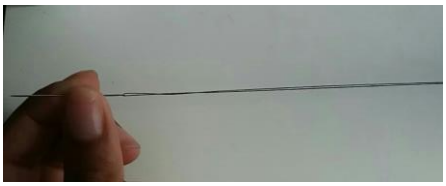
## Step 7:

Use the Velcro to attach the microbit onto the belt, which will enable the user to move with the microbit. To make the microbit more secure sewing the microbit onto the belt as it will reduce the chances of the microbit falling off the belt.



## Step 8:

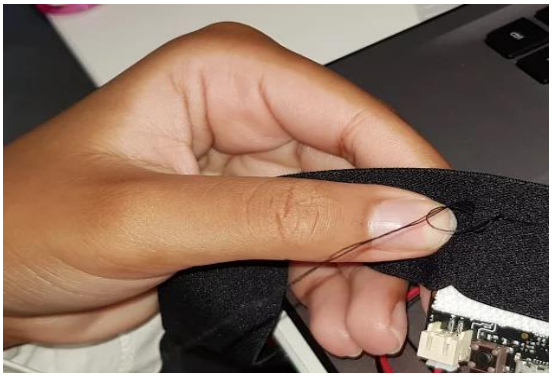
Thread the string through the eye of the needle. Pull it through and keep pulling until the length of the thread is at an appropriate length for stitching the microbit on the belt. Cut the thread and make the thread equal length (make the ends of the thread meet). Tie a knot at the end of the thread.



## Step 9:

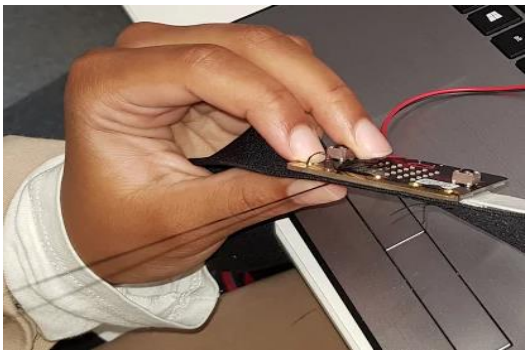
Place the needle on the bottom of the belt. Push the needle just outside the microbit. Keep pulling through until the knot prevent you from pulling the thread further. Place the needle at the first hole, push the needle though. Keep pulling the needle until the thread snags. Go back outside the microbit (do not place the needle where the knot is) push the needle through the belt. Pull once again until the thread snags. Go to first hole push the needle through. Pull unit thread snags.  The first hole of the microbit is secured onto the belt

## Step 10:



Place the needle at the second hole of the microbit. Push the needle through and pull the needle until the thread snags. Go to the first hole and push the needle through again. Keep pulling until the thread snags again. Go to the second hole push needle through. Pull until thread snags. Make sure that you are not stitching the thread together, each stitch needs to be separate.  The second hole is secured onto the belt.

## Step 11:



Move onto the third hole and push the needle through. Pull until the thread snags. Place the needle at the second hole of the microbit. Push the needle through the second hole. Pull until the thread snags. Go back to the third hole push the needle through. Pull until the thread snags. The third hole is secured onto the belt.

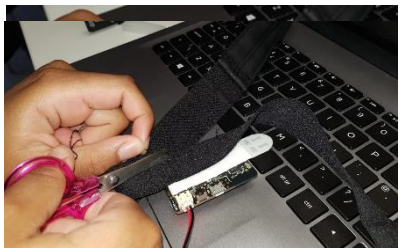
## Step 12:

Repeat steps for all the holes on the microbit to ensure that all the holes are secure onto the belt. Once you get to the end of the microbit we have to make knot at the end to keep the stitching in its place.

## Step 13:

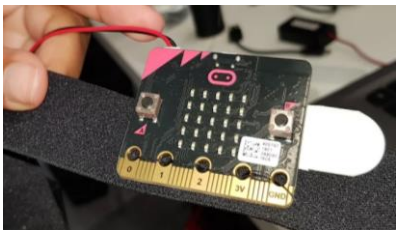
To make a knot at the end of the stitching we slide the needle through the last last stitch we made at the bottom of the belt. Pull the needle around the stitch but keep the thread lose we do not need it to snag yet. Slide the needle underneath the stitch again this will make loop around the stitch. Pull the needle through the loop and pull the needle until the thread snags. The knot has been made, the microbit is secured to the belt.

## Step 14:



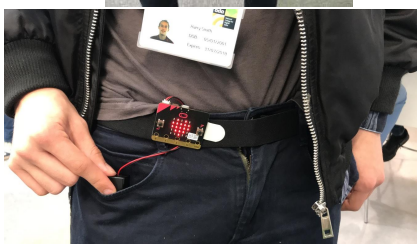
If there is excess thread just cut it off with scissors.

The end product should be similar to this image. To test that the microbit is secure just shake the strap and turn it upside down to see if the microbit falls of, if not then well done, you have successfully sew the microbit firmly. If it is not secure, just add an extra layer of thread to ensure that the microbit does not fall of



## Step 15:

**Well done!** Now is time to put the pedometer in action. Put on the strap on your belt or wrist and tighten it to prevent it from falling



Once the strap is on, set the microbit at a place that is comfortable and gripped and place the battery either inside your pocket or behind the strap.

## Step 16:

Once the microbit is on, make sure you are comfortable with the microbit and ensure that the microbit is secure and not at risk of falling. Because the pedometer records the steps by detecting a ‘shake’, make sure that the microbit is against you so it can record the step accurately.

There you go! Now you have your own microbit pedometer. Test the pedometer by going for a jog and see how much steps you took.

## Step 17:

**Start moving!** Once you ensured the pedometer is on and is secure, you can start moving and it will count your steps automatically.

From here you can start recording the amount of steps you take and see if you are leading a healthy lifestyle.