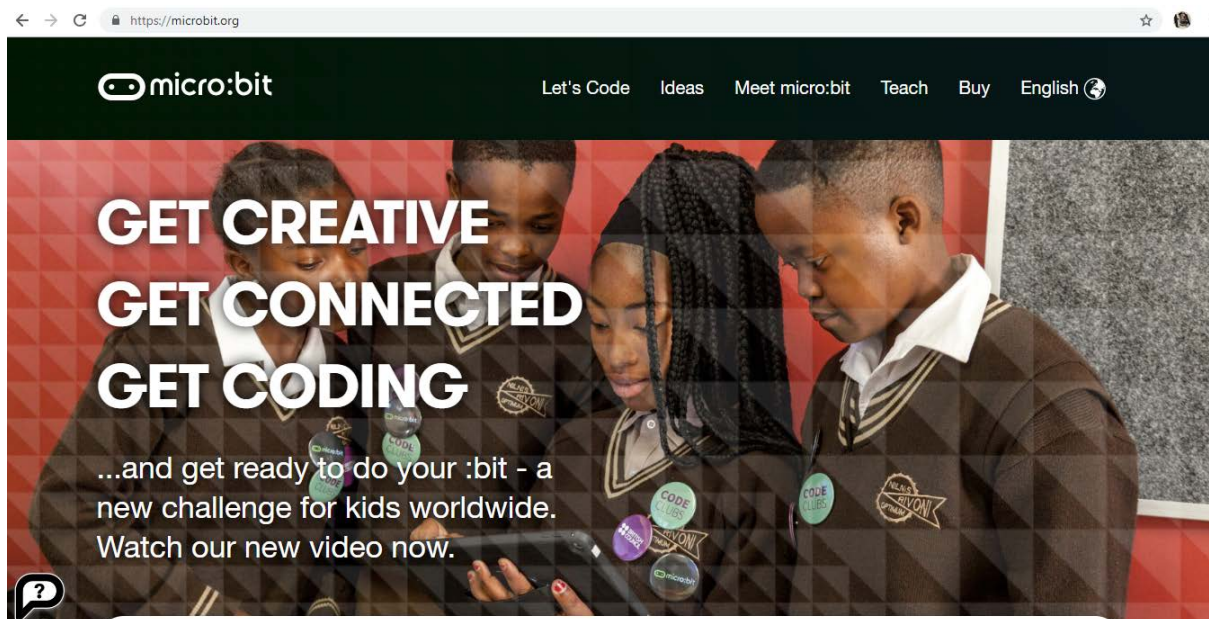


Fall Detection using Microbit with Android Mobile App

Devices & Software:

1. Thunkable
2. Firebase (database)
3. Microbit
4. Android Phone

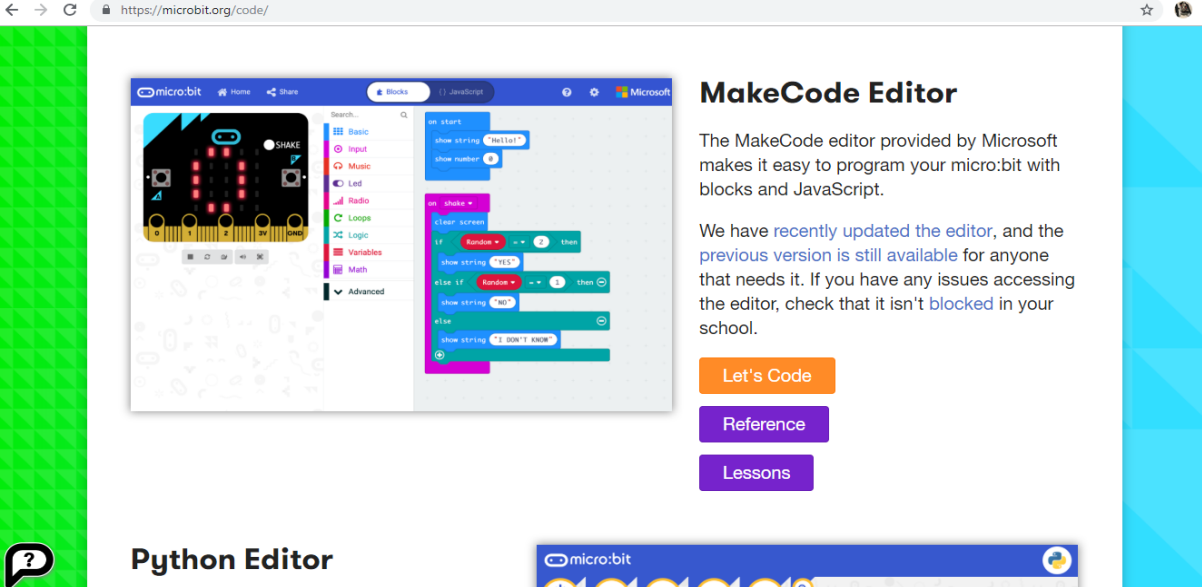
Chapter 1: MICROBIT



Read 5 easy steps to use **Microbit** - <https://microbit.org/guide/quick/>

A screenshot of the 'Step 1: Connect it to your Windows PC' page on the micro:bit website. The browser address bar shows 'https://microbit.org/guide/quick/'. At the top, there are four platform selection buttons: 'Windows' (purple), 'macOS' (yellow), 'Linux' (yellow), and 'Mobile / Tablet' (yellow). The main heading is 'Step 1: Connect it to your Windows PC' with a 'Change platform' link. Below the heading, the text reads: 'Connect the micro:bit to your computer using a micro USB cable. Your micro:bit will show up on your computer as a drive called 'MICROBIT'. Watch out though, it's not a normal USB disk!'. In the center, there is an image of a micro:bit board. At the bottom, the heading for the next step is 'Step 2: Program it on your Windows PC' with a 'Change platform' link.

1. Click "Let's Code"



The screenshot shows the MakeCode Editor interface. On the left, there is a sidebar with a search bar and a list of categories: Basic, Input, Music, Led, Radio, Loops, Logic, Variables, and Math. The main workspace displays a JavaScript code block with the following code:

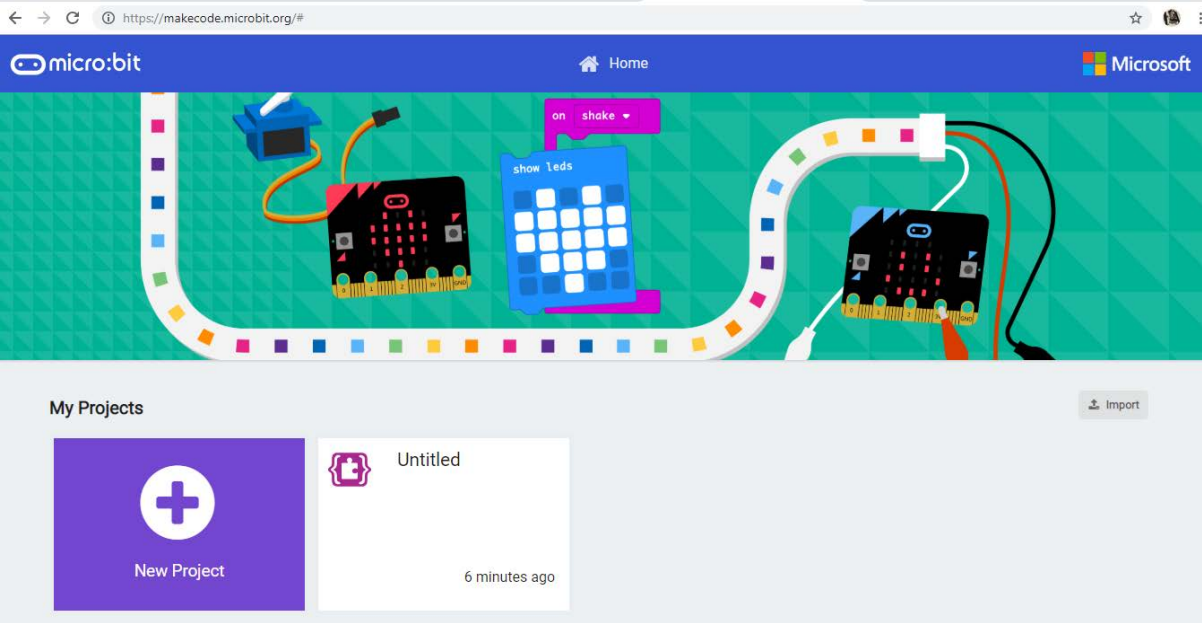
```
on start
  show string "Hello!"
  show number 1

on shake
  clear screen
  if Random - - 2 then
    show string "YES!"
  else if Random - - 2 then
    show string "NO!"
  else
    show string "I DON'T KNOW"
```

Below the code block, there are three buttons: "Let's Code" (orange), "Reference" (purple), and "Lessons" (purple). The URL in the browser is <https://microbit.org/code/>.

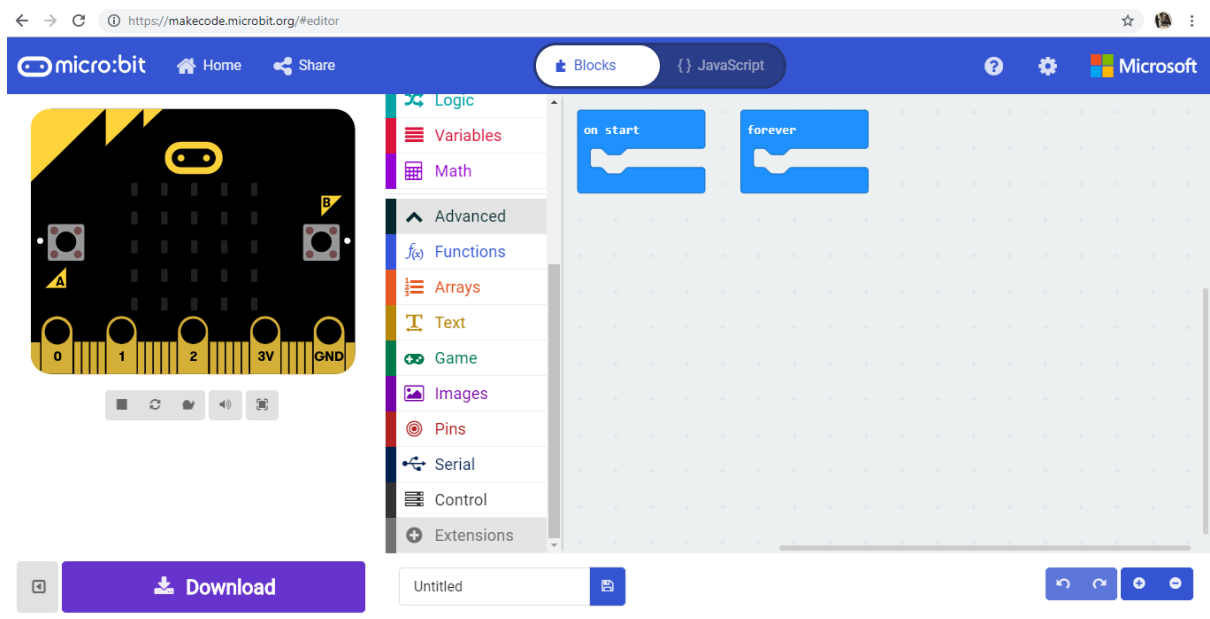
Python Editor

2. Click "New Project"

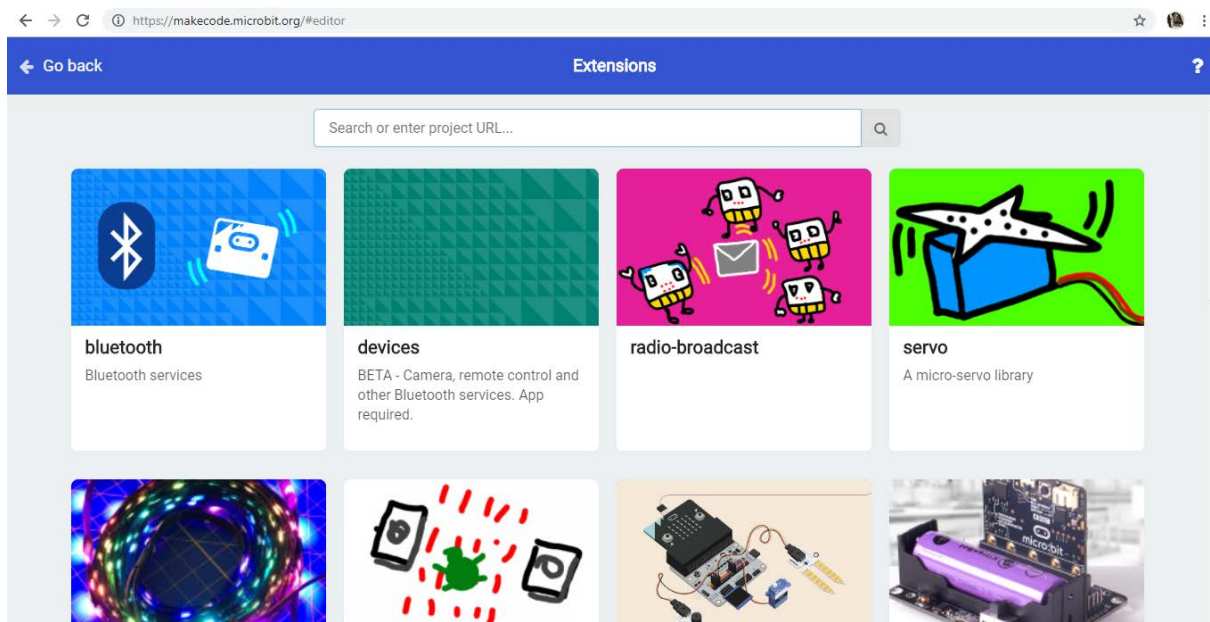


The screenshot shows the 'My Projects' page in the MakeCode Editor. The URL in the browser is <https://makecode.microbit.org/#>. The page features a header with the 'micro:bit' logo, a 'Home' button, and the Microsoft logo. Below the header is a large banner image showing a micro:bit board connected to a USB cable and a keyboard. The main content area is titled 'My Projects' and contains a 'New Project' button (a purple square with a white plus sign) and an 'Untitled' project card (a white square with a purple plus sign icon and the text 'Untitled' and '6 minutes ago'). There is also an 'Import' button in the top right corner of the 'My Projects' section.

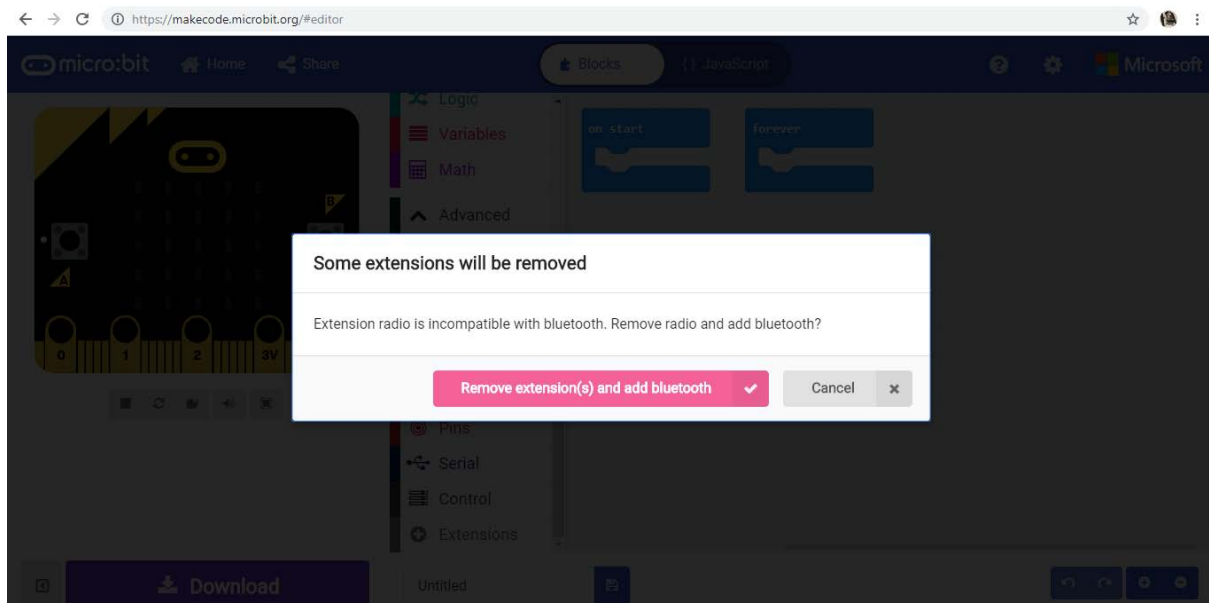
3. Click “Advance then Extension” to add Bluetooth



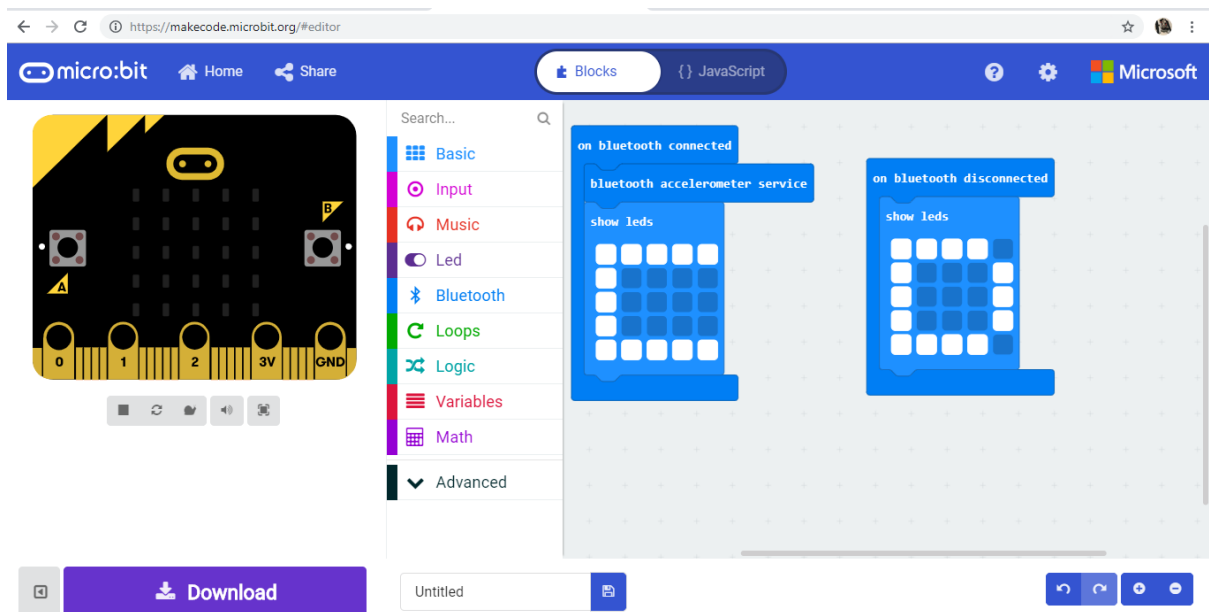
4. Click “Bluetooth”



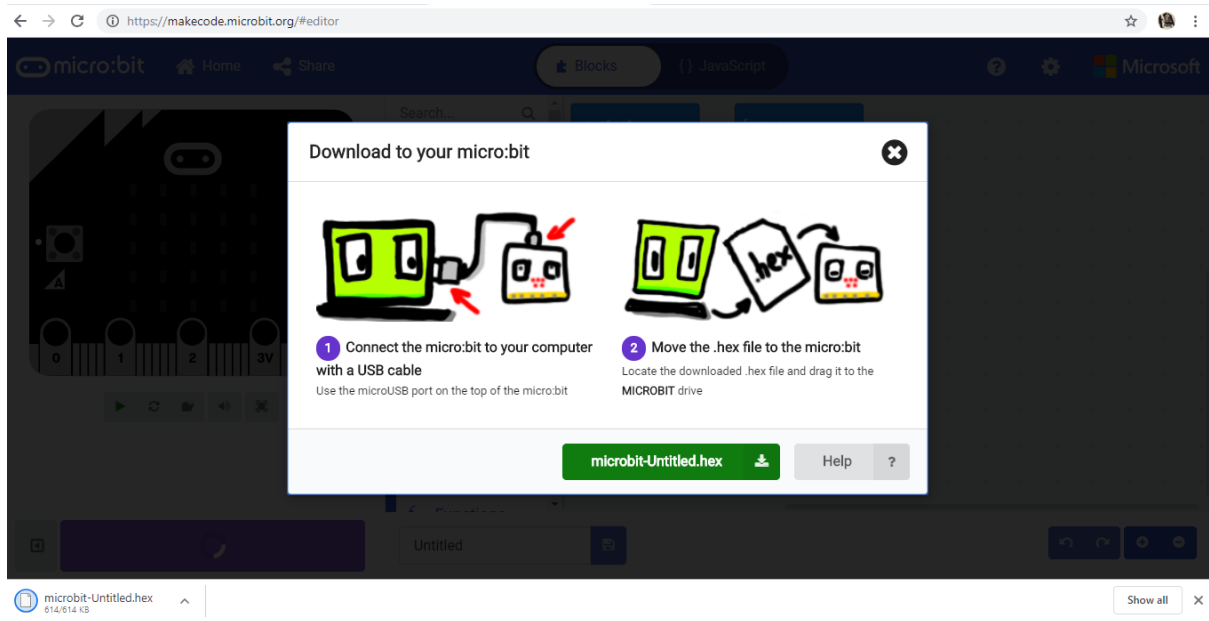
5. Click “Remove Extension”



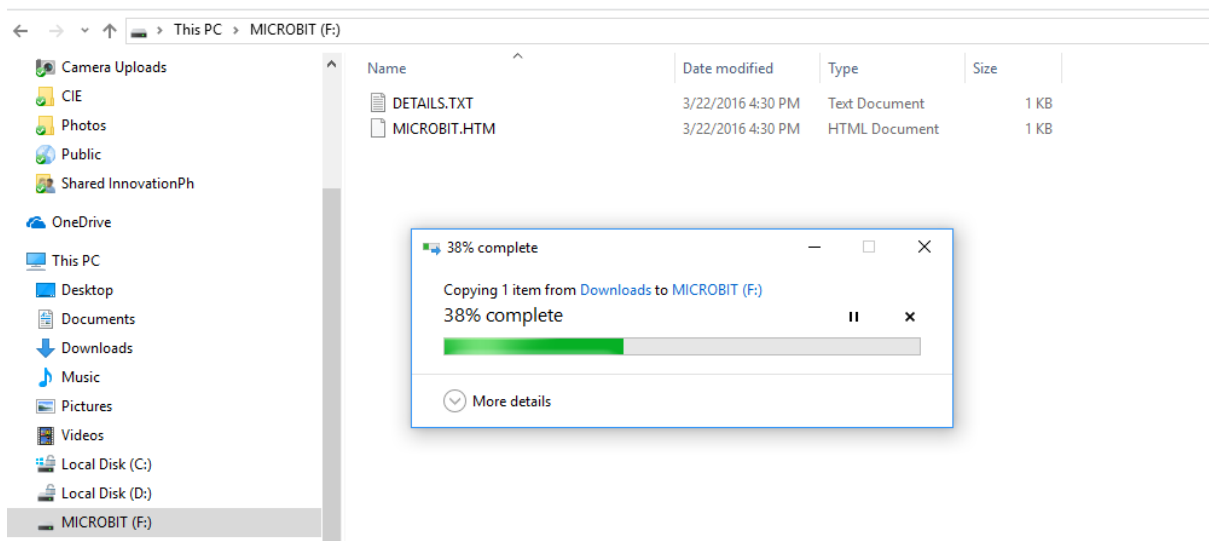
6. Drag & Drag code blocks for Bluetooth



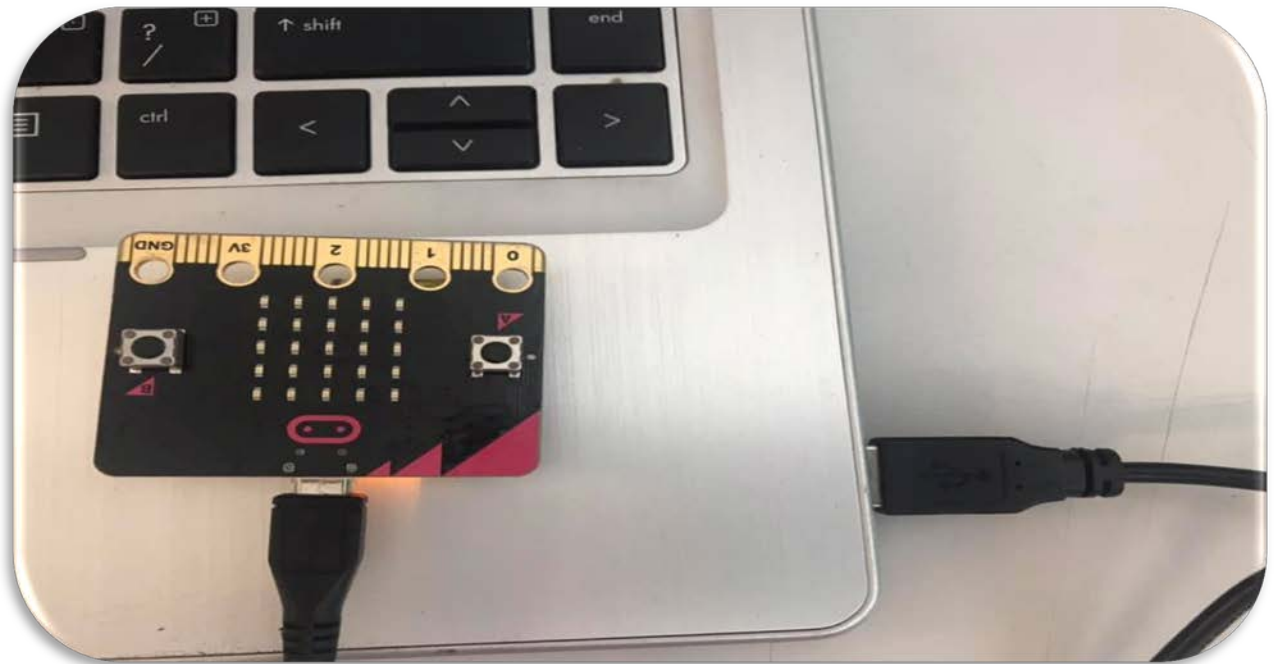
7. Download the hex file



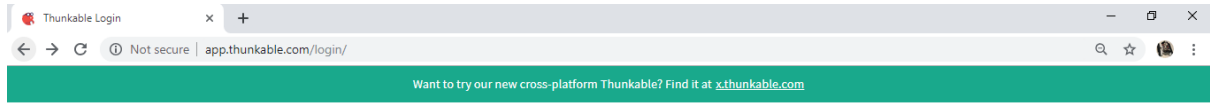
8. Copy & Paste the downloaded Hex file



9. Connection from Computer USB to Microbit

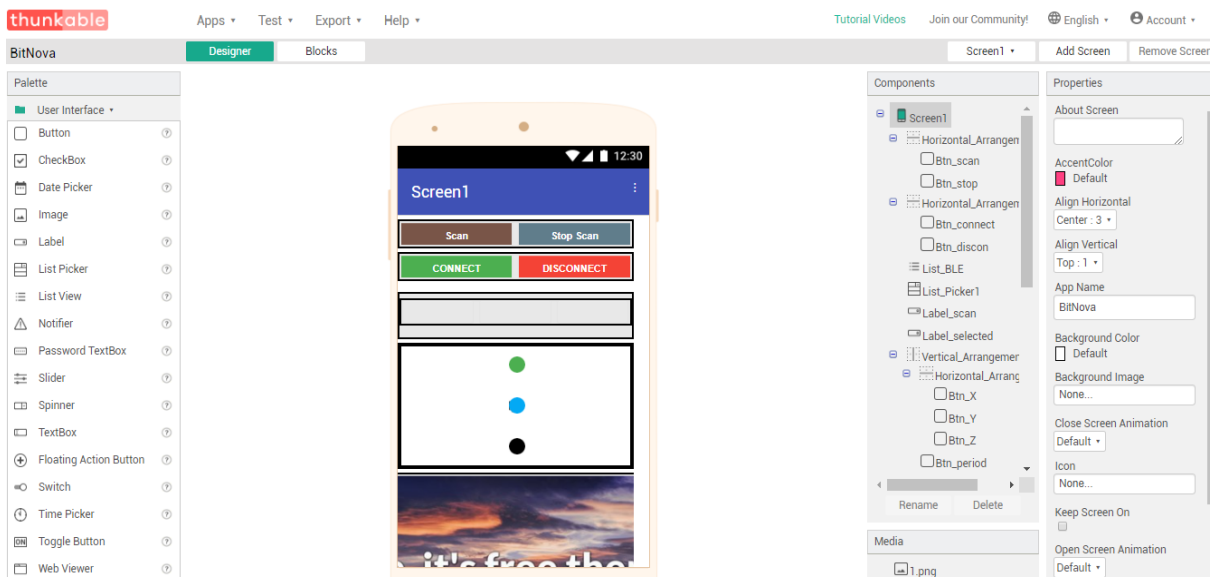


Chapter 2: Thunkable (Classic)

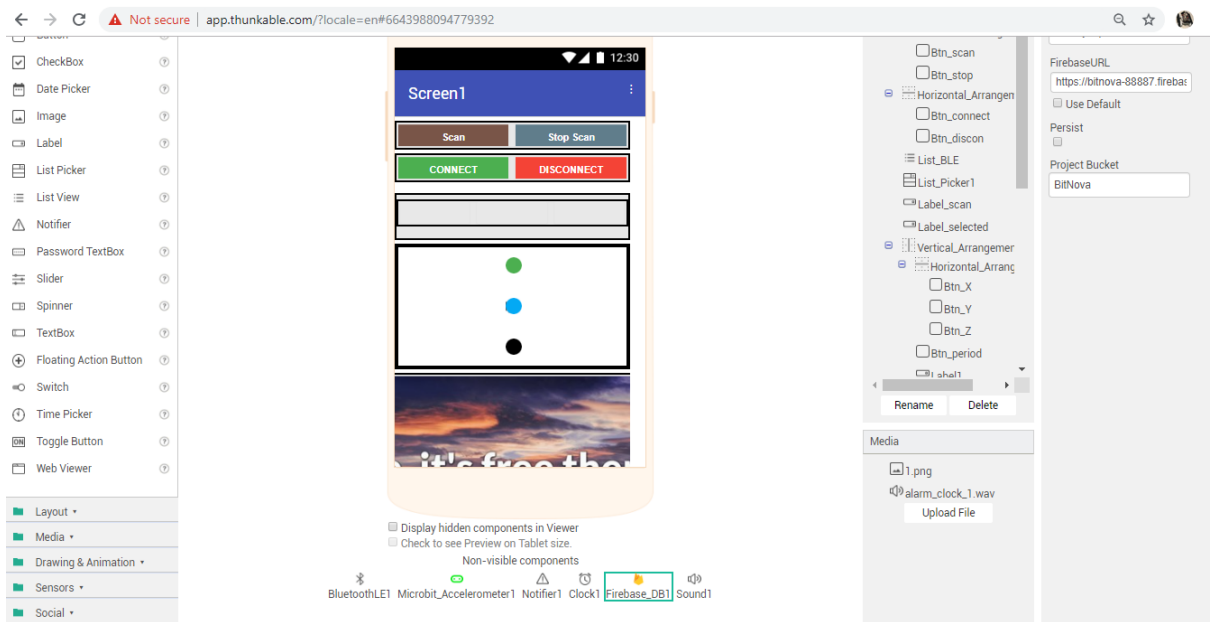


For Fall Detection user

1. Designer View



2. Type Firebase Token & Firebase URL



3. Drag & Drop Thinkable Code



This screenshot shows a block-based programming environment. On the left is a 'Built-in' library with categories like Control, Logic, Math, Text, Lists, Colors, Variables, Procedures, and Screen1. Below it is a 'Media' section with files '1.png' and 'alarm_clock_1.wav'. The main workspace contains the following code:

```

when Microbit_Accelerometer1 AccelerometerDataReceived
  Accelerometer_X Accelerometer_Y Accelerometer_Z
do
  set Btn_X .Text to get Accelerometer_X
  set Btn_Y .Text to get Accelerometer_Y
  set Btn_Z .Text to get Accelerometer_Z
  set Ball1 .X to 130 + (get Accelerometer_X * 130) / 1800
  set Ball2 .X to 130 + (get Accelerometer_Y * 130) / 1800
  set Ball3 .X to 130 + (get Accelerometer_Z * 130) / 1800
  call Firebase_DB1 .Store Value
    tag X axis
    value To Store Btn_X .Text
  call Firebase_DB1 .Store Value
    tag Y axis
    value To Store Btn_Y .Text
  call Firebase_DB1 .Store Value
    tag Z axis
    value To Store Btn_Z .Text
  Show Warnings

```

There is also a separate block for periodic data reception:

```

when Microbit_Accelerometer1 AccelerometerPeriodReceived
  Accelerometer_Period
do
  set Btn_period .Text to get Accelerometer_Period

```

This screenshot is identical to the one above, showing the same block-based programming environment with the same code blocks for processing accelerometer data and storing it to a Firebase database.

This screenshot shows the continuation of the code in the block-based programming environment. It features two main code blocks:

```

then
  call Firebase_DB1 .Store Value
    tag message
    value To Store ALERT!
  call Firebase_DB1 .Store Value
    tag chat
    value To Store NOT OK!
  set Canvas1 .Background Color to red
  call Sound1 .Play

```

```

when send_btn .Click
do
  call Firebase_DB1 .Store Value
    tag chat
    value To Store OK
  call Firebase_DB1 .Store Value
    tag message
    value To Store All Good!
  set Canvas1 .Background Color to white
  call Sound1 .Stop
  Show Warnings

```

On the right side, there is a trigger for data changes in the database:

```

when Firebase_DB1 .Data Changed
  tag value
do
  if get tag == X axis
  then set Btn_X .Text to get value
  if get tag == Y axis
  then set Btn_Y .Text to get value
  if get tag == Z axis
  then set Btn_Z .Text to get value
  if get tag == chat
  then set send_btn .Text to get value

```

- Control
- Logic
- Math
- Text
- Lists
- Colors
- Variables
- Procedures

Screen1

- Horizontal_Arrangen
 - Btn_scan
 - Btn_stop
- Horizontal_Arrangen
 - Btn_connect
 - Btn_discon
- List_BLE

1.png

alarm_clock_1.wav

Upload File

```

when Screen1 Error Occurred
  component functionName errorNumber message
do
  if get errorNumber == 9002
  then call Notifier1 Show Text Dialog
      message Bluetooth is not enabled!
      title No Bluetooth!
      cancelable true
  else if get errorNumber == 9005
  then call Notifier1 Show Text Dialog
      message Bluetooth is not connected!
      title No Connection!
      cancelable true
  end if
end do

when Clock1 Timer
do
  set Canvas1 Background Color to #
  set Clock1 Timer Enabled to false
end do

```

Show Warnings

```

when Microbit Accelerometer1 AccelerometerDataReceived
  Accelerometer_X Accelerometer_Y Accelerometer_Z
do
  set Btn_X Text to get Accelerometer_X
  set Btn_Y Text to get Accelerometer_Y
  set Btn_Z Text to get Accelerometer_Z
  set Ball1 X to 130 + get Accelerometer_X
  set Ball2 X to 130 + get Accelerometer_Y
  set Ball3 X to 130 + get Accelerometer_Z
end do

call Firebase_DB1 Store Value
  tag X axis
  value To Store Btn_X Text
call Firebase_DB1 Store Value
  tag Y axis
  value To Store Btn_Y Text
call Firebase_DB1 Store Value
  tag Z axis

```