



Coventry and Warwickshire



# Hackpack Anthology

## Volume II

### V.0.2



You can access Hack pack anthology volume 1 here:

<https://github.com/ncscomputing/Hackpack/raw/master/Hackpack%20Anthology%20V1.1.pdf>

In version 0.1 of anthology volume 2 there were two hack. The original got corrupted. You can access it here:

<https://github.com/ncscomputing/Hackpack/raw/master/Hackpack%20Anthology%20V2%200.1.pdf>

You can also access the collection of Micro:bit resources that were made for CPC here:

[http://warksjammy.blogspot.co.uk/2017/04/blog-8-what-can-ucreate-with-microbit\\_17.html](http://warksjammy.blogspot.co.uk/2017/04/blog-8-what-can-ucreate-with-microbit_17.html)

**Here are the current tutorials/ hacks:**

1. Liverpool Make Fest Minecraft Pixel Art bot.
2. Edu Blocks graphing Sense hat emulator data in Minecraft
3. BitIO tutorial 1: Getting Started with BitIO
4. BitIO tutorial 2: Extending the Accelerometer demo: Walking Rainbow Blocks
5. BitIO tutorial 3: Extending the accelerometer 'Tilting Rainbow Road'
6. BitIO tutorial 4: 'Tilt me around the World'

### **Hack 1: Liverpool Make Fest Minecraft Pixel Art bot. @ncscomputing**

For Liverpool MakeFest I created a Minecraft twitter bot that tweeted a screen capture of pixelart coded using Python and EduBlocks.

I have used the following Raspberry Pi tutorials as part of this build when dealing with Twitter:

<https://www.raspberrypi.org/learning/microbit-selfies/worksheet/>

Here is what I wanted it to do:

1. Provide a quickish way(5mins approx) of people being able to create some Minecraft Pixelart without being able to forced to sit and code for 20 mins.
2. Furthermore share their creation with the world via Twitter. Ideally use a Microbit to trigger the tweet. (This is still hit and miss)

Current review: It does one and two is still in the mixer.

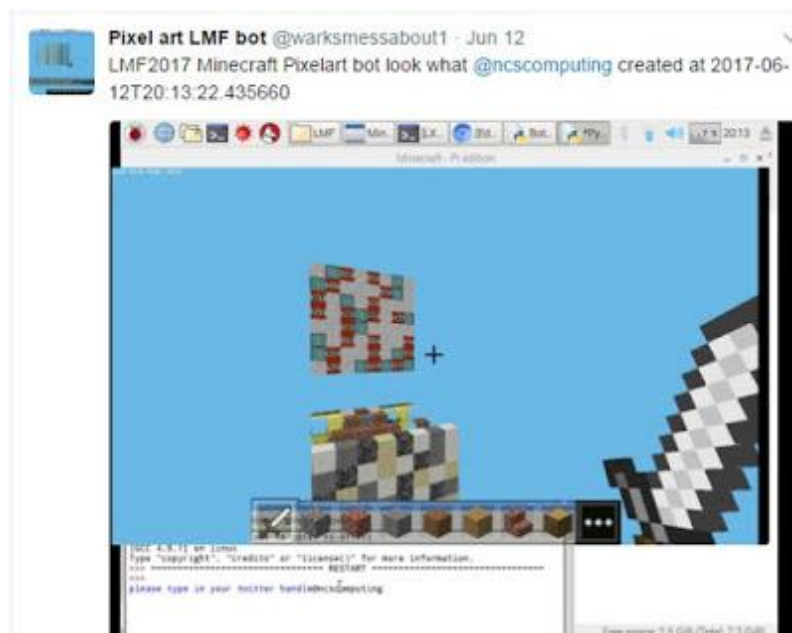
Here is version one which worked perfectly:



Here is version 2, which doesn't work so reliably when it comes to the Microbit part.



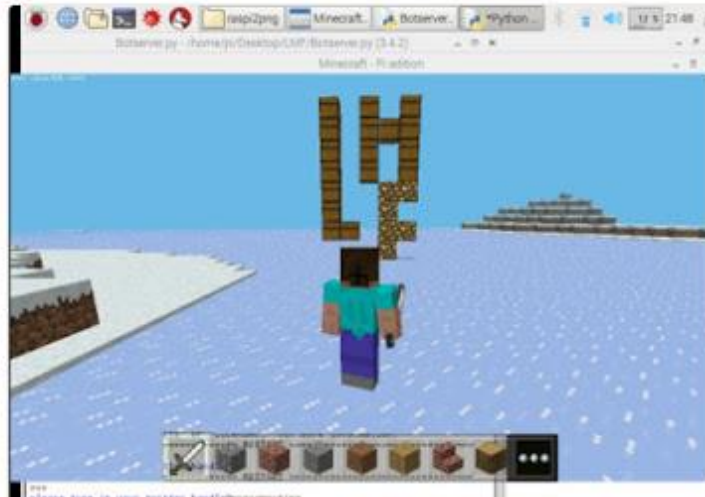
**Here are examples of what it produced:**





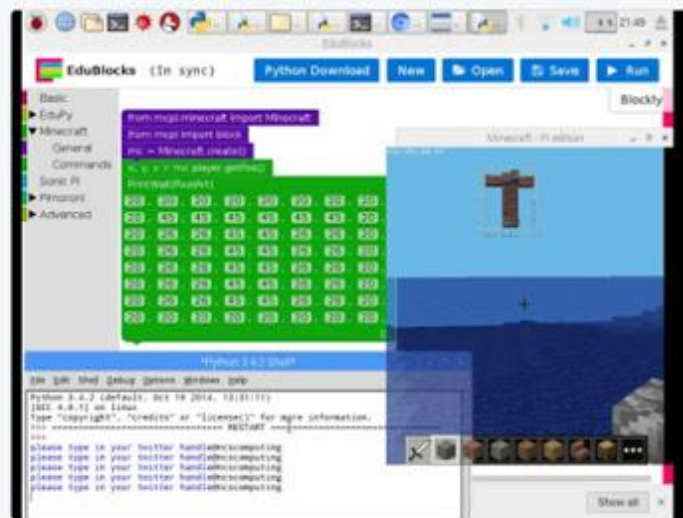
Pixel art LMF bot @warksmessabout1 · Jun 2

LMF2017 Minecraft Pixelart bot look what @ncscomputing created



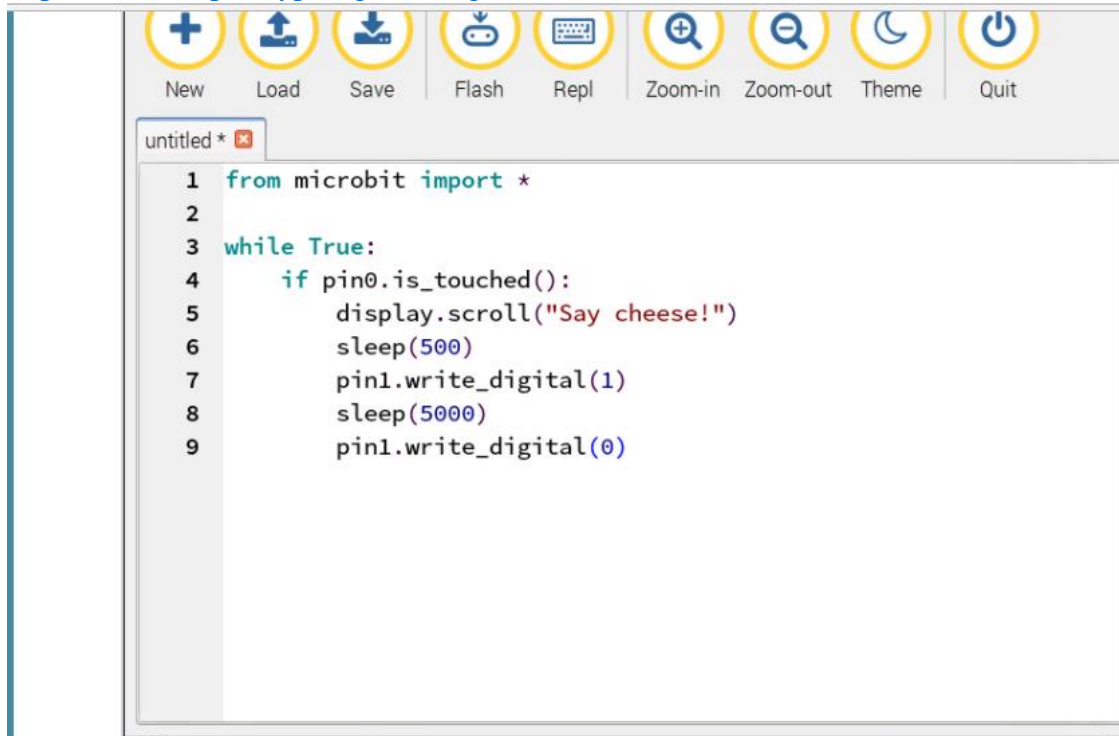
Pixel art LMF bot @warksmessabout1 · Jun 7

LMF2017 Minecraft Pixelart bot look what @ncscomputing created at 2017-06-07T21:49:36.780573



### **Code for micro:bit taken from RPi tutorial :**

<https://www.raspberrypi.org/learning/microbit-selfies/worksheet/>



### **Code for Twitter bot:**

```
from gpiozero import Button
from time import sleep
from signal import pause
```

```
from datetime import datetime
```

```
import subprocess
#from mcpi.minecraft import Minecraft
from mcpi import minecraft as minecraft
from time import sleep, time
import sys
```

```
from twython import Twython
#once you have created your own Twitter app put in your info below
```

```
consumer_key = "
consumer_secret = "
access_token = "
access_token_secret = "
```

```
mc = minecraft.Minecraft.create()
```

```
api = Twython(consumer_key,consumer_secret,access_token,access_token_secret)
def Tweet(handle):
```

```
    timestamp = datetime.now().isoformat()
```

```
msg = "#LivMF17 Pixelart bot! look what "+handle+" created at "+timestamp
mc.postToChat(msg)
a=subprocess.check_output('./raspi2png -d 3 -p "myscreenshot.png"',shell=True)
photo = open('myscreenshot.png', 'rb')
api.update_status_with_media(status=msg, media=photo)
```

```
button = Button(4,pull_up = False)
while True:
    button.wait_for_press()
    handle = input("please type in your twitter handle")
    Tweet(handle)
    sleep(4)
```

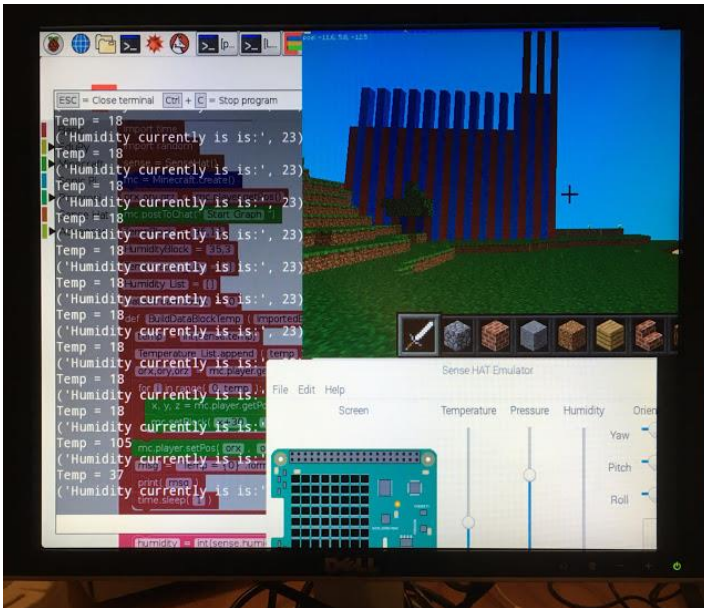
here is the link for the Block ids sheet I've borrowed them from and Raspberry Pi spy this is sourced in the document:

[https://github.com/ncscomputing/Hackpack/blob/master/Minecraft%20Pixel%20Art%20sheet.p  
df](https://github.com/ncscomputing/Hackpack/blob/master/Minecraft%20Pixel%20Art%20sheet.pdf)

**Hack 2: Edu Blocks graphing Sense hat emulator data in Minecraft**



## Picture of code working



This code graphs data from the sense hat emulator in Minecraft using EduBlocks.  
The potential of Edublocks to introduce Python to secondary students

### Python 3 code:

```
from sense_emu import SenseHat
import mcpi.minecraft as minecraft
import mcpi.block as block
import time
import random
```

```
"""
```

NCS team Pixel Astro Pi competition entry

We have created a bar graph in minecraft and store the values for temp and humidity in two seperate lists.

Team members are:

Archie

Tom

Adrian

```
"""
```

```
sense = SenseHat()
```

```
mc = minecraft.Minecraft.create()
```

```
orx,ory,orz = mc.player.getPos()
```



```
mc.postToChat("Start Graph")
```

```
TempBlock = 35,14
```

```
HumidityBlock = 35,3
```

```
Temperature_List = []#stores temp data
```

```
Humidity_List = []#stores humidity data
```

```
DataStreamCount= 0
```

```
def BuildDataBlockTemp(ImportedBlock):# take data for temp
```

```
    temp = int(sense.temp)
```

```
    Temperature_List.append(temp)
```

```
    orx,ory,orz = mc.player.getPos()
```

```
    for i in range (0,temp):
```

```
        x,y,z = mc.player.getPos()
```

```
        mc.setBlock(x+30,i,z,ImportedBlock)
```

```
    mc.player.setPos(orx,ory,orz+1)
```

```
    msg = "Temp = {0}".format(temp)
```

```
    #sense.show_message(msg, scroll_speed=0.10 )
```

```
    print(msg)
```

```
    time.sleep(1)
```

```
def BuildDataBlockHumidity(ImportedBlock):# take data for humidity
```

```
    humidity = int(sense.humidity)
```

```
    Humidity_List.append(humidity)
```

```
    orx,ory,orz = mc.player.getPos()
```

```
    for i in range (0,humidity):
```

```
        x,y,z = mc.player.getPos()
```

```
        mc.setBlock(x+30,i,z,ImportedBlock)
```

```
    mc.player.setPos(orx,ory,orz+1)
```

```
    msg = "Humidity currently is: ",humidity
```

```
    print(msg)
```

```
    msg = "humidity = {0}".format(humidity)
```

```
    time.sleep(4)
```

```
while True:
```

```
    BuildDataBlockTemp(TempBlock) # 46
```

```
    BuildDataBlockHumidity(HumidityBlock)
```

## EduBlocks code:

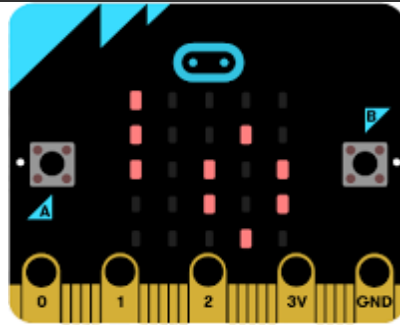
Top of code:

```
from mcpi.minecraft import Minecraft
from sense_hat import SenseHat
import mcpi.block as block
import time
import random
sense = SenseHat()
mc = Minecraft.create()
orx,ory,orz = mc.player.getPos()
mc.postToChat(" Start Graph ")
TempBlock = 35,14
HumidityBlock = 35,3
Temperature_List = []
Humidity_List = []
DataStreamCount = 0
def BuildDataBlockTemp ( ImportedBlock ):
    temp = int(sense.temp)
    Temperature_List.append ( temp )
    orx,ory,orz = mc.player.getPos()
    for i in range( 0,temp ):
        x, y, z = mc.player.getPos()
        mc.setBlock( x+30 , i , z , ImportedBlock )
    mc.player.setPos( orx , ory , orz+1 )
    msg = "Temp = {0}".format(temp)
```

Second half of code:

```
print( msg )
time.sleep( 1 )
def BuildDataBlockHumidity ( ImportedBlock ):
    humidity = int(sense.humidity)
    Humidity_List.append ( humidity )
    orx,ory,orz = mc.player.getPos()
    for i in range( 0, humidity ):
        x, y, z = mc.player.getPos()
        mc.setBlock( x+30 , i , z , ImportedBlock )
    mc.player.setPos( orx , ory , orz+1 )
    msg = "Humidity currently is is:", humidity
    print( msg )
    msg = "Humidity = {0}".format(humidity)
    time.sleep( 4 )
while True:
    BuildDataBlockTemp ( TempBlock )
    BuildDataBlockHumidity ( HumidityBlock )
```

## BitIO tutorial 1: Getting Started with BitIO



David Whale has developed the BitIO library. It essentially lets you code your Microbit in normal Python 3.

Last summer I created these resources for Micro:bit:

<http://warksjammy.blogspot.co.uk/2017/04/cpc-ucreate-microbit-resources-all-in.html>

These were a mostly micro python using MU IDE. Some involved using the serial connection from Micro:bit and a Raspberry Pi. This again built on top of the technical feats of David Whale.

Each program involved a Micro python program which sat on the Micro:bit and then a separate Python program that then interpreted the data sent by the Micro:bit. This was technically possible but elongated.

David has essentially stripped out much of the complexity. Now there is a generic hex file which sits on the Micro:bit. The only coding that you do is in Python 3.

Here are the steps to get a simple demo working:

1. Go to Davids github:

<https://github.com/whaleygeek/bitio>

2. Look on the far right of the webpage for a green box which says 'clone or download'. Then click 'Download Zip'. Then save the zip file to your Pi.

3. Using the file manager find the downloaded Zip file and extract it.

4. Now plug in your Micro:bit to the Pi. Go into the Bitio master folder and you will find a hex file called 'bitio.hex'. Copy this and paste it onto the Micro:bit. This should now be flashed.

5. Go to the Bitio master folder and find the src folder. You will see several examples. The one that we will test uses the accelerometer to control the position of the player in Minecraft.

6. The file is called 'tilt\_mc.py' open this in Python 3.

7. Open Minecraft and create a new world. Press f5 to run it the Python script.

8. The following dialogue will display in the Python shell window:

```
No micro:bit has previously been detected  
Scanning for serial ports  
remove device, then press ENTER
```

9. So unplug the Micro:bit and press enter.

10. Next the following text will display:

```
scanning...  
found 67 device(s)  
plug in device, then press ENTER
```

11. Now press enter and the following text will display:

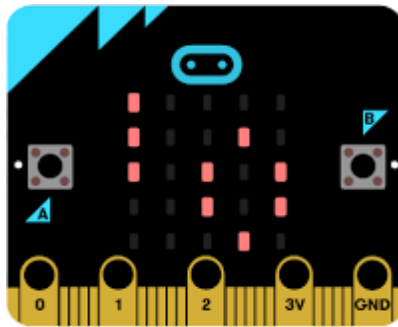
```
scanning...  
found 68 device(s)  
found 1 new device  
selected:/dev/ttyACM0  
Do you want this device to be remembered? (Y/N)
```

12. Press enter to confirm and the following text will display:

```
Your micro:bit has been detected  
Now running your program
```

13. If you pick up your Micro:bit then and open Minecraft you should see the player move according to the direction you are tilting the Micro:bit.

## BitIO tutorial 2: Extending the Accelerometer demo: Walking Rainbow Blocks



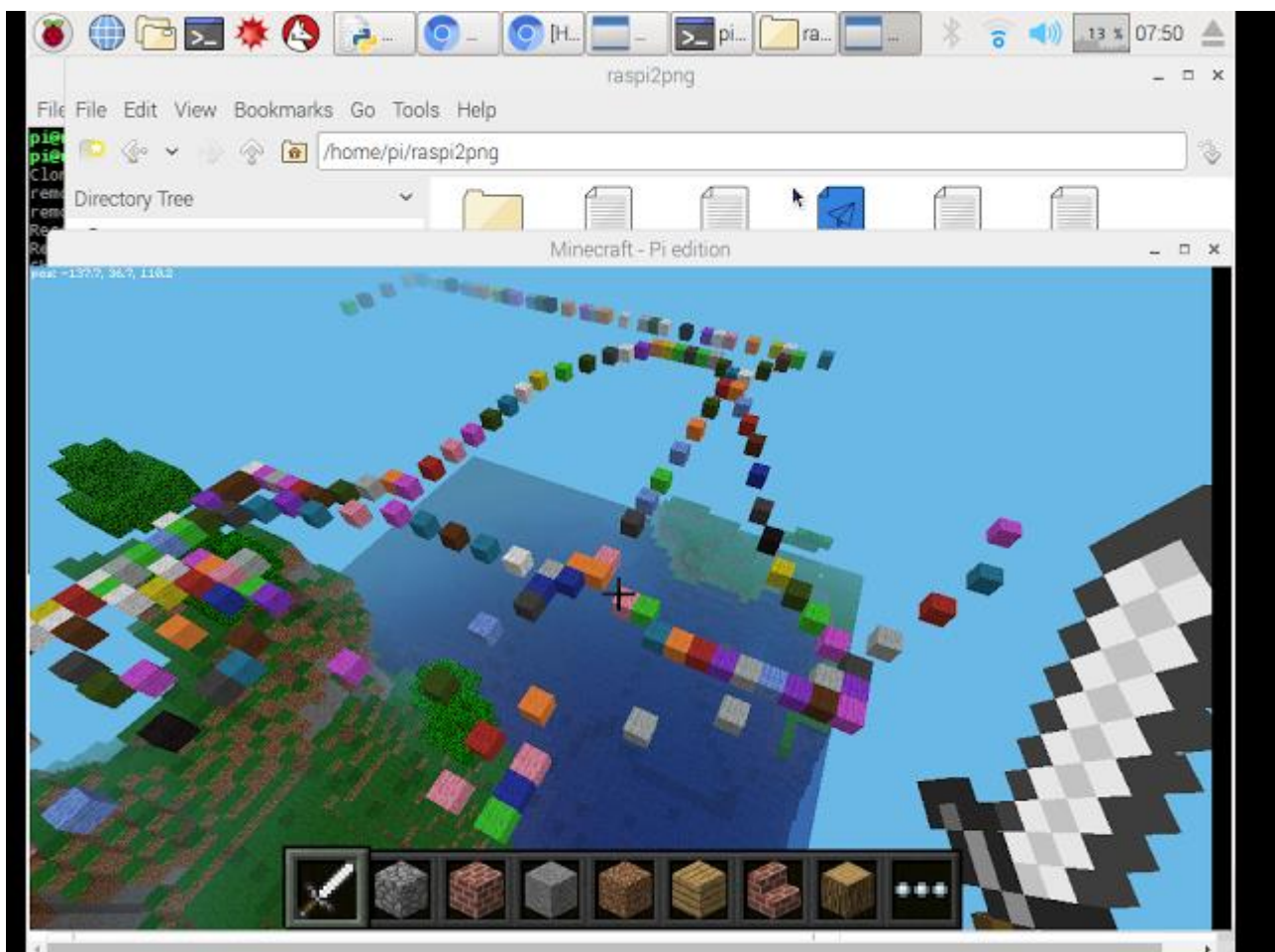
This takes the basic 'tilt\_mc.py' demo from David Whales BitIO master available here:

<https://github.com/whaleygeek/bitio>

In the first blog the demo moves the character around the screen according to the direction you tilt the Micro:bit. The code below extends this to:

1. Use a list to randomly select a wool block.
2. Drop this block underneath the player the end result is a colourful mess. :) (See below)

The code will need to be saved in the 'src' folder to work.



Below is the code, you can download it here:

<https://github.com/ncscomputing/HpAnthologyV2/raw/master/Walking%20tilt.py>

### **Code:**

```
#Original written by David Whale located here as part of his BITIO library
#https://github.com/whaleygeek/bitio/blob/master/src/tilt_mc.py
#adapted by @ncscomputing 19/07/17
```

```
import mcpi.minecraft as minecraft
import mcpi.block as block
import microbit
import time
import random
```

```
mc = minecraft.Minecraft.create()
```

```
blocksList = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]
```

```
while True:
```

```
    pos = mc.player.getTilePos()
    x = microbit.accelerometer.get_x()/300 # -ve=left/+ve=right
    y = microbit.accelerometer.get_y()/300 # -ve=forward/+ve=backward
```

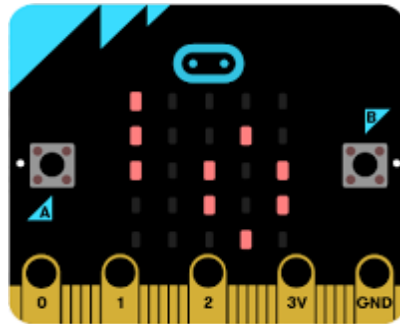
```
    pos.x += x # east/west
    pos.z += y # north/south
```

```
    mc.player.setTilePos(pos.x, pos.y, pos.z)
    mc.setBlock(pos.x, pos.y-1, pos.z,35,random.choice(blocksList))
```

```
# time.sleep(0.5)
```



## BitIO tutorial 3: Extending the accelerometer 'Tilting Rainbow Road'



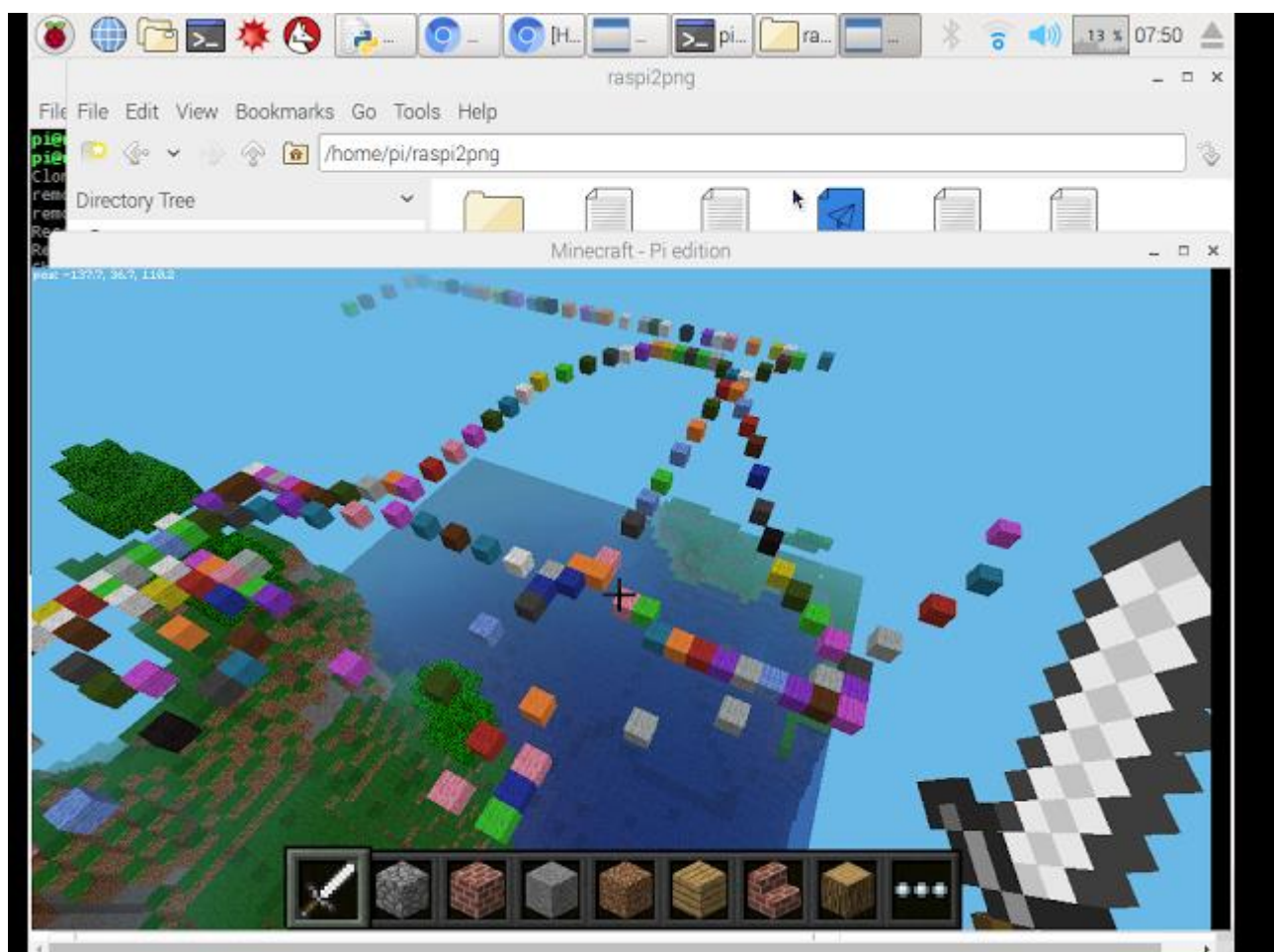
This takes the basic 'tilt\_mc.py' demo from David Whales BitIO master available here:

<https://github.com/whaleygeek/bitio>

In the second blog the 'tilt\_mc.py' demo was extended to:

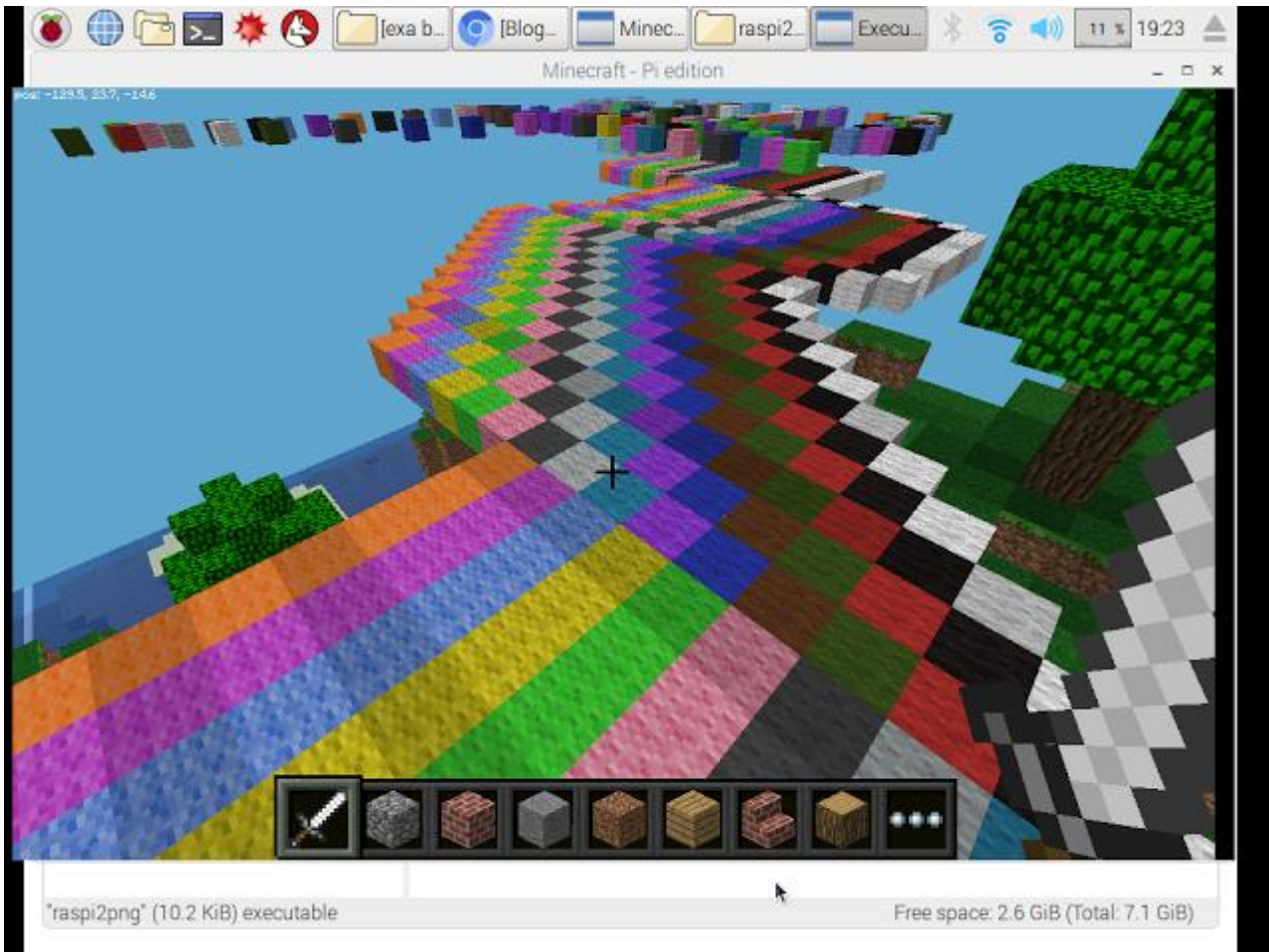
1. Use a list to randomly select a wool block.
2. Drop this block underneath the player the end result is a colourful mess. :) (See below)

The code was saved in the 'src' folder to work and did this >>>



This third blog will extend this further to drop a roads worth of blocks each time the character is moved from a tilt of the Micro:bit. The end result will look like this:





Again you will use the accelerometer to move player and it will drop the road to the left of the character. The code will need to be saved in the src folder. (This is all explained in blog 1) Here is the code (a marginally more complex version of the previous blog):

### Code:

```
#Original written by David Whale located here as part of his BITIO library
#https://github.com/whaleygeek/bitio/blob/master/src/tilt_mc.py
#adapted by @ncscomputing 20/07/17
```

```
import mcpi.minecraft as minecraft
import mcpi.block as block
import microbit
import time
import random

mc = minecraft.Minecraft.create()

blocksList = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]
time.sleep(7)
mc.postToChat("Rainbow Road 3.0 Micro:bit controlled")
while True:
    pos = mc.player.getTilePos()
    x = microbit.accelerometer.get_x()/300 # -ve=left/+ve=right
```

```
y = microbit.accelerometer.get_y()/300 # -ve=forward/+ve=backward

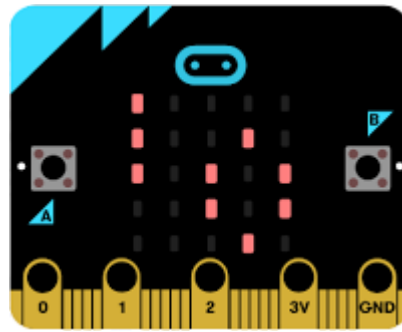
pos.x += x # east/west
pos.z += y # north/south

Count = 1
while Count <=16 :
    mc.player.setTilePos(pos.x, pos.y, pos.z)
    mc.setBlock(pos.x-Count, pos.y-1, pos.z,35,Count)
    Count = Count+1
    time.sleep(0.25)
```

**Download the code from here:**

<https://raw.githubusercontent.com/ncscomputing/HpAnthologyV2/master/Rainbow%20road%20tilt.py>

## BitIO tutorial 4:Tilt me 'Around the World'



This takes the basic 'tilt\_mc.py' demo from David Whales BitIO master available here:

<https://github.com/whaleygeek/bitio>

In the second blog and third blog I extended the basic demo to drop random wool blocks where ever you tilted the Micro:bit and in tutorial three I extended this to lay a road of sorts.

### Blog 2

<http://warksjammy.blogspot.co.uk/2017/07/blog-2-extending-accelerometer-demo.html>



### Blog 3

<http://warksjammy.blogspot.co.uk/2017/07/bitio-blog-3-extending-accelerometer.html>



In this blog post we will hike up the challenge to combine both push buttons in addition to the accelerometer.

### Section 1 of the code:

If you press the a button on the Micro:bit then a data file is read and world map is then rendered into the Minecraft world. You can then use the accelerometer to navigate the character around the map by tilting the Micro:bit.

**Here is a video to show this in action:**

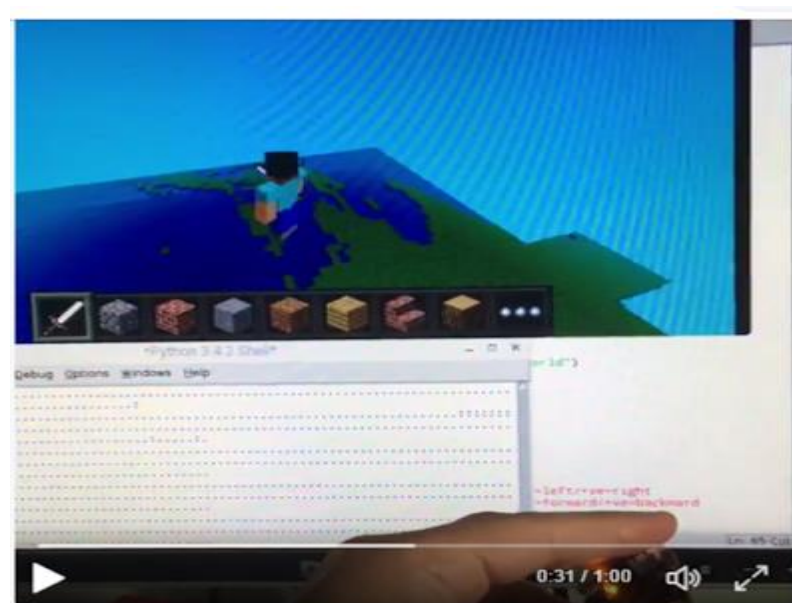
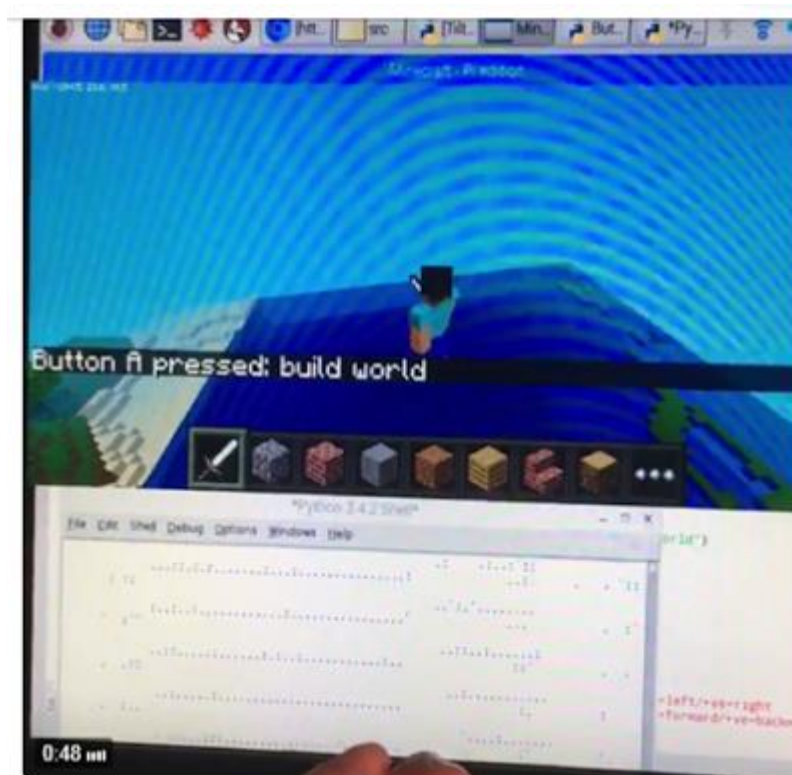
<https://twitter.com/ncscomputing/status/885213278007328772>

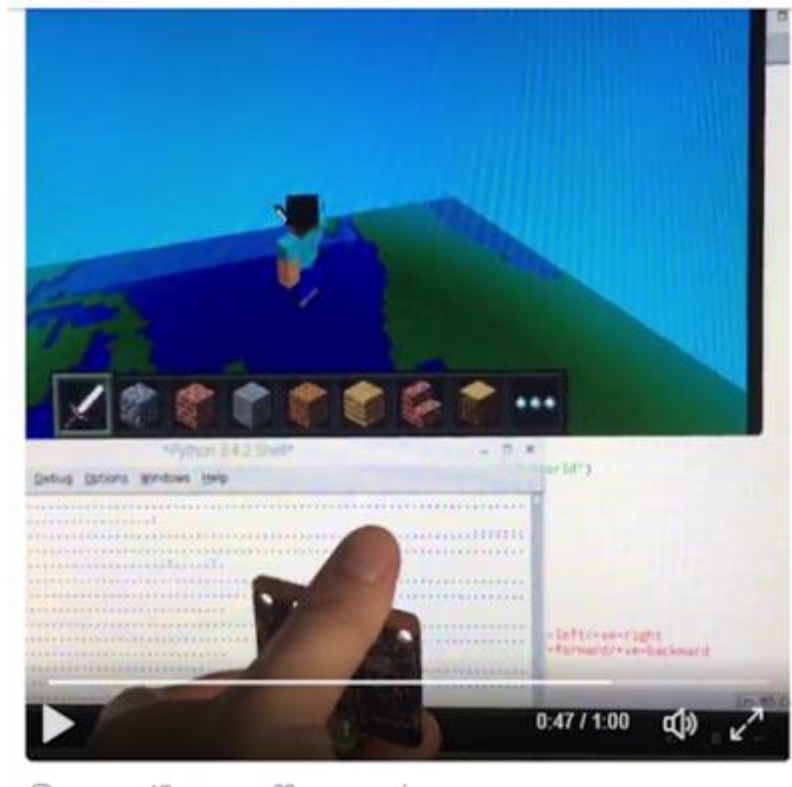
## Section 2 of the code:

If you press the B button then the character teleports to predetermined locations on the map. Here is a video of this in action:

<https://twitter.com/Warksraspijam/status/894492363330113540>

Here are some static images:





The files you will need are here should you not want to code the BitIO python file:

**Files:**

World map data file:

<https://github.com/ncscomputing/HpAnthologyV2/blob/master/world3.txt>

Build world library:

<https://github.com/ncscomputing/HpAnthologyV2/blob/master/BuildWorldDM.py>

BitIO file:

<https://github.com/ncscomputing/HpAnthologyV2/blob/master/blog4AWACC.py>

If you do want to have a go at making the BitIO file then here are your instructions:

This is what the code will do:

**Instructions**

1. Ensure that you have read blog 1:

<http://warksjammy.blogspot.co.uk/2017/07/blog-1-getting-started-with-bitio.html>

and downloaded the BitIO master files from David Whales github

2. Go into the 'Bitio master' folder, find the 'src' folder and save 'World map data file' and 'Build world library' from the links above into that folder.

3. Create a new Python 3 script and type out the following:



## Code

```
#"Written by : @ncscomputing /@warksraspijam"

from mcpi import minecraft as minecraft
from mcpi import block as block
from datetime import datetime
import time
import random
import BuildWorldDM as bw

###
"""
Written by @ncscomputing on top of the bitio produced by David Whale
https://github.com/whaleygeek/bitio

world building code imported library from Damien Mooney's blog:
https://damianmooney.wordpress.com/2016/02/16/raspberry-pi-minecraft-iss-tracker/

"""
import serial
from mcpi.minecraft import Minecraft
import time
from mcpi import block as block
import random

mc = Minecraft.create()

# button.py - demonstrates using a button

import microbit

print("micro:bit connected - press button A to test")
WoolList = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]

EngX = 1.0
EngZ = 50.0

UsaX = 53.8
UsaZ = 39.7

CanadaX = 59.0
CanadaZ = 61.3

IceLandX = 10.0
IceLandZ = 61.1

def Teleport(x,z,Country):
    mc.player.setPos(x,20,z)
    mc.camera.setFollow()
    mc.setBlock(x,1, z,35,random.choice(WoolList))
    mc.postToChat(Country)

while True:
    time.sleep(0.25)
    #=====buildworld call when a button pressed
    if microbit.button_a.was_pressed():
        mc.postToChat("Button A pressed: build world")
        print("Button A pressed: build world")
```

```

bw.Build()
time.sleep(2)

microbit.display.show("build world")

# manual teleport using accelerometer readings
pos = mc.player.getTilePos()
x = microbit.accelerometer.get_x()/300 # -ve=left/+ve=right
y = microbit.accelerometer.get_y()/300 # -ve=forward/+ve=backward

pos.x += x # east/west
pos.z += y # north/south

mc.player.setTilePos(pos.x, pos.y, pos.z) # set player position

#time.sleep(0.5)
#=====
if microbit.button_b.was_pressed():
    print ("Button B pressed: Manual teleport")
    mc.postToChat("Button B pressed: Teleport")
    time.sleep(0.5)
    Teleport(EngX, EngZ, "England")
    time.sleep(8)
    Teleport(UsaX, UsaZ, "USA")
    time.sleep(8)
    Teleport(CanadaX, CanadaZ, "Canada")
    time.sleep(8)
    Teleport(IcelandX, IcelandZ, "Iceland")
    time.sleep(8)

```

4. Save it as something.py in the same 'src' folder as the previous files.
5. Open Minecraft, create a new world.
6. Make sure your Micro:bit is plugged in to your PC/Pi.
7. Run the python file that you created above.
8. Try pressing the 'a' button let it build the world map.
9. Try using tilting the Micro:bit to see if you can navigate around the world map.
10. Now try and press the 'b' button and see if the teleportation works.

Extension:

Try extending the teleportation code to go to extra countries on the word tour.

Enjoy :)

You can now download a pdf version of these tutorials as part of the updated #hackpack resource booklet here:  
<https://github.com/ncscomputing/HPAnthologyV2/raw/master/Hackpack%20Anthology%20V2%200.3.pdf>