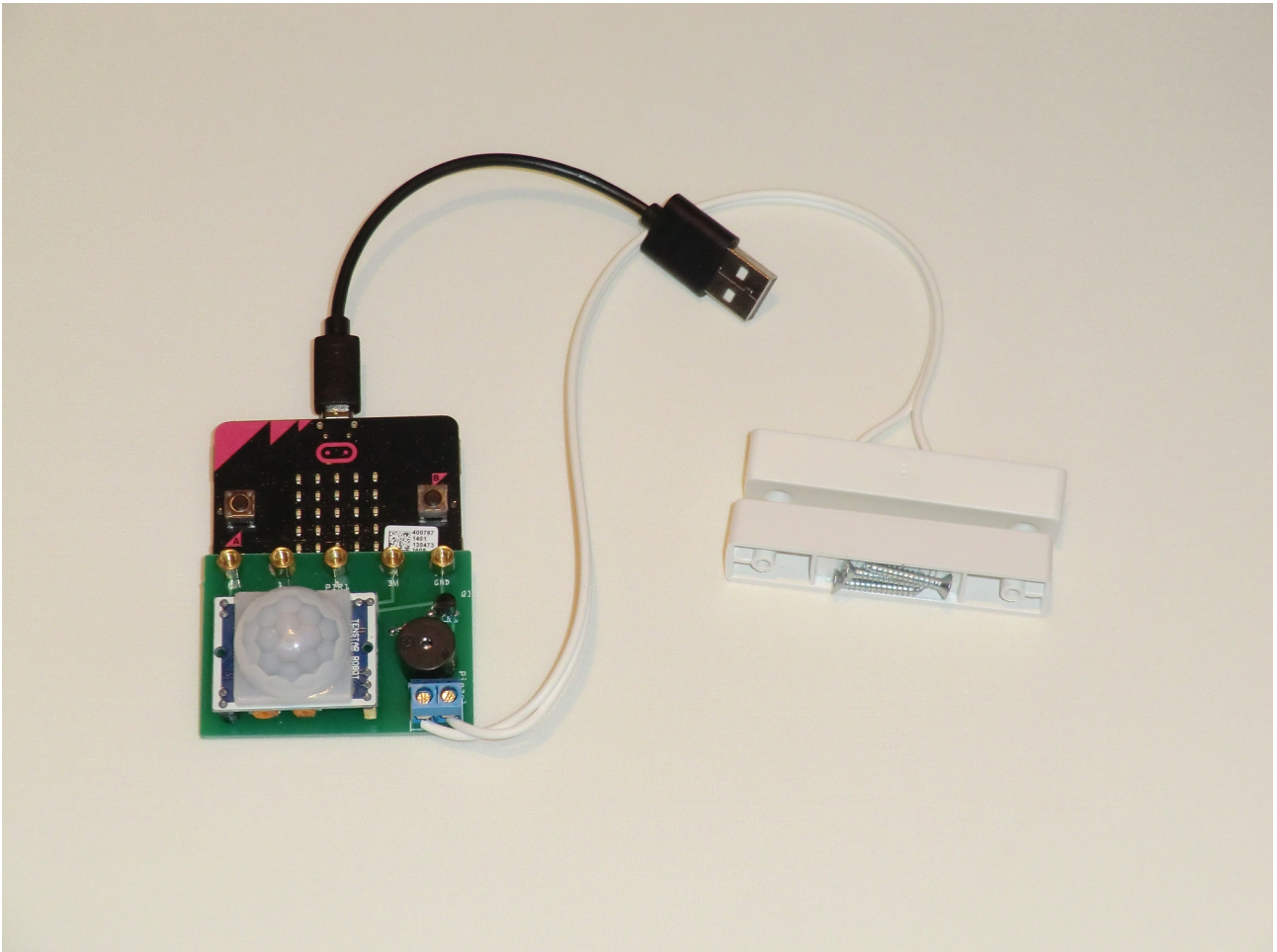


Alarm project board for micro:bit.

The alarm project board adds a PIR movement sensor, buzzer and the facility to attach a magnetic door/window alarm sensor to your Microbit.

The board attaches either using the 5 supplied nuts and bolts, or if you prefer you can use your own crocodile clips.

Here's a photo of the Microbit attached to the alarm board, with a magnetic door sensor attached.



The PIR is attached to pin 0, the magnetic door sensor is attached to pin 1, and the buzzer is attached to pin 2. We can read the state of the sensors by reading the state of pin 0 or 1. We can make the buzzer sound by setting pin 2 to 1, and turn it off again by setting pin 2 to 0.

The Microbit has the ability to send radio signals from one unit to another, so it's possible to have two Microbits with alarm boards attached, and have the buzzer sound on the remote microbit, when the local one's PIR or door sensor is triggered.

Also, because the Microbit can send text strings over a USB serial port, it's possible to connect the Microbit to a PC or Mac's USB port and have the PC email your phone when the PIR or door sensors are triggered.

We have a test program you can download to your Microbit here:

<https://makecode.microbit.org/0LyhE9Axx3xy>

This code will display a smiley face when the PIR is covered up, and if the PIR senses movement it displays an unhappy face. If you press button A the buzzer will sound for 1 second. If you press button B the display will show an open or closed door, depending on the position of the magnetic door sensor.

```
forever loop
  set pir to (digital read pin P0)
  if (pir = 1)
  then
    show leds (smiley face)
  else
    show leds (unhappy face)

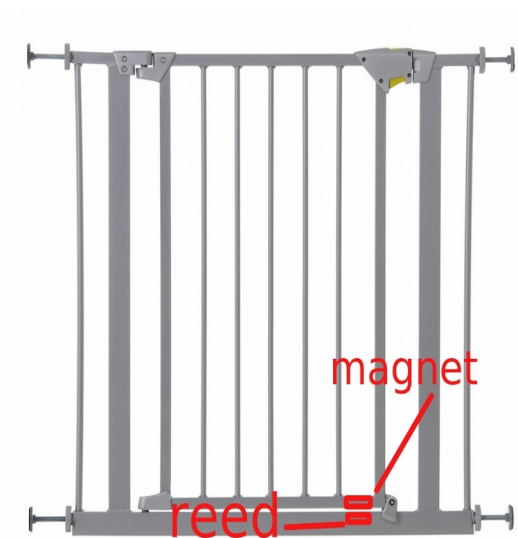
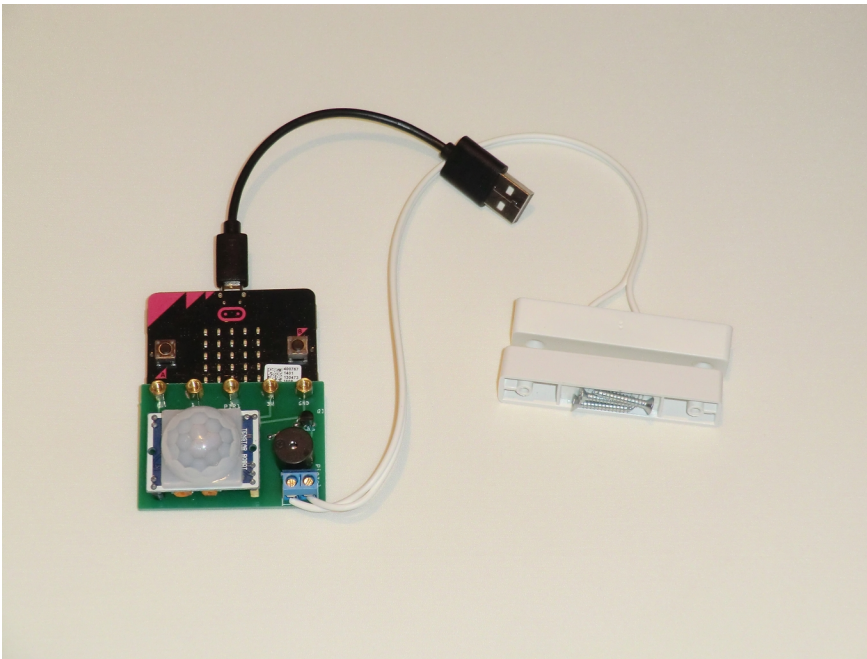
on button B pressed
  set doorStatus to (digital read pin P1)
  if (doorStatus = 1)
  then
    show leds (open door)
  else
    show leds (closed door)
  pause (ms) 2000

on button A pressed
  digital write pin P2 to 1
  pause (ms) 1000
  digital write pin P2 to 0
```

Childgate alarm project with the Alarm Board.

This project shows you how to make an alarm that warns when a stair safety gate has been left open for more than 20 seconds. We also show you how to send a radio signal to a 2nd Microbit with a buzzer attached which could be in another part of the house.

If you attach a 5 volt buzzer to the output pin and GND on a Microbit, and turn the output pin on, it doesn't supply enough current to make a clearly audible noise, but if you attach the buzzer to 3v and GND it does make a audible beep. So, with the alarm board we get around this issue by using a small npn transistor to switch full power to the buzzer when the output pin is set high in our code.



When the magnet is next to the reed switch, I/O port 1 on the Microbit reads 1, and when the magnet moves away I/O port 1 on the Microbit reads 0. So if we attach the magnet to the moving part on the bottom of a stair gate and the reed switch to the part below that doesn't move, we can know whether the gate is open or not.

We can turn on the buzzer on the alarm board by setting I/O port 2 to "1" for one second, and then setting it back to "0".

Here's the code to make it work. Download from <https://makecode.microbit.org/iVL8MV6EWaFb>

```
on start
  radio set group 99

forever
  set gatestate to (digital read pin P1)
  if (gatestate = 0)
  then
    pause (ms) 20000
    set gatestate to (digital read pin P1)
    if (gatestate = 0)
    then
      show number gatestate
      digital write pin P2 to 1
      radio send number 0
      pause (ms) 1000
      digital write pin P2 to 0
      show leds
```

If you aren't using a 2nd Microbit with an extra alarm board you can delete the 2 pink radio blocks. The code waits until you open the gate, at which point the gatestate variable becomes 0, it then waits another 20 seconds and tests again to see if the stairgate has been left open. If the gate's still open the LEDs display a 0, the buzzer attached via I/O pin 2 gets turned on for 1 second, and a radio signal is transmitted on group 99 to any other Microbits running slave buzzers.

Here's the code to run on a 2nd slave Microbit with an alarm board attached, that listens out for the radio code from the main board with the magnetic door/gate sensor attached:

```
forever
  radio set group 99
  on radio received receivedNumber
    if (receivedNumber = 0)
      then
        digital write pin P2 to 1
        show number receivedNumber
        pause (ms) 2000
        digital write pin P2 to 0
        show leds
```

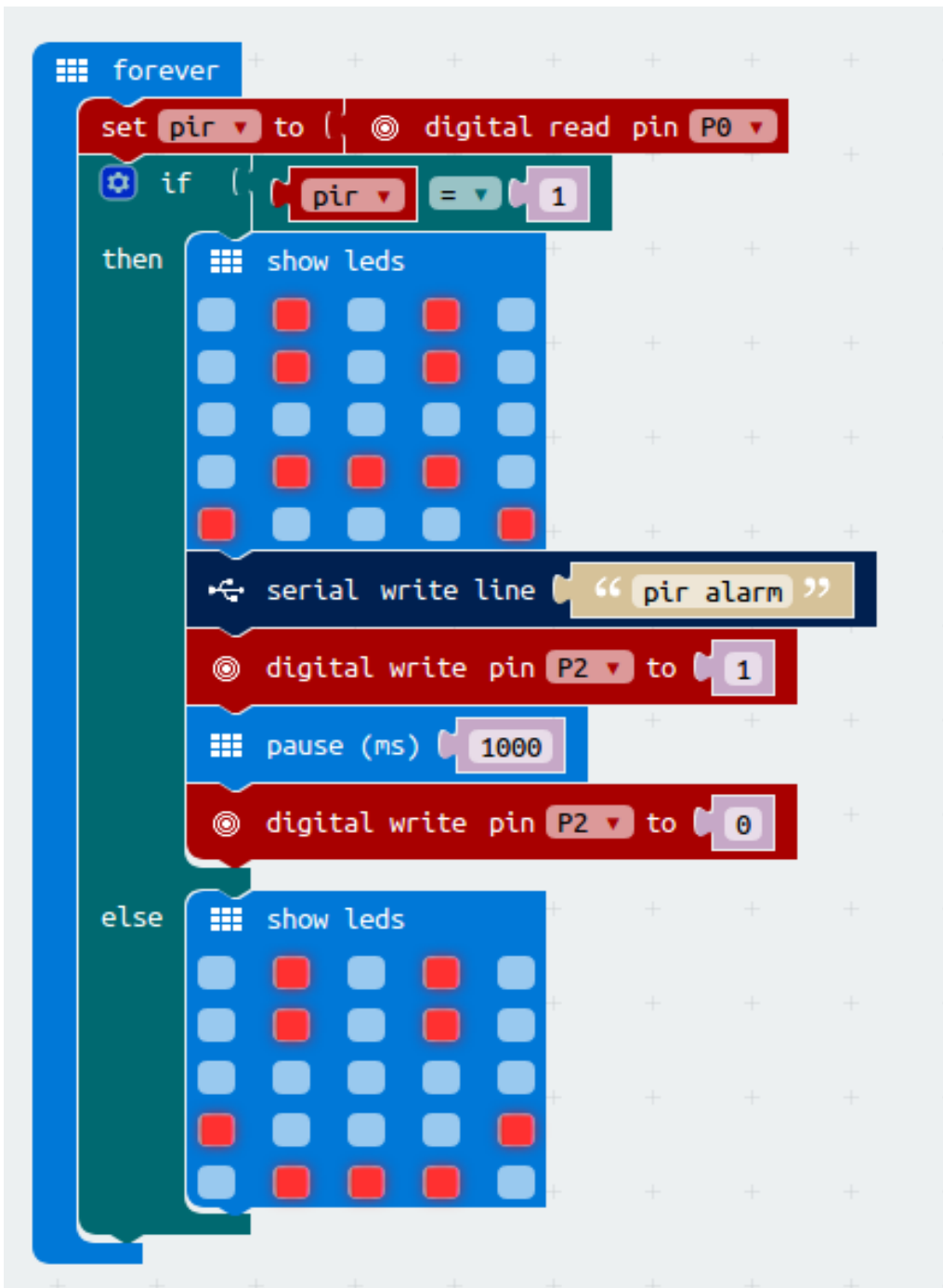
You can download it from : https://makecode.microbit.org/_7Jyesm6q5Uw6

When the radio code is received the buzzer turns on for two seconds (2000ms) and displays a 0 on the LED panel. If the gate remains open the buzzer will sound every twenty seconds until someone closes it.

PIR Alarm

This example uses the PIR movement sensor plugged into the alarm board and the on-board buzzer.

Whenever movement is detected the buzzer sounds for 1 second, an unhappy face appears on the LEDs, and a “pir alarm” message is sent on the USB serial interface to a PC or Mac that might be running a program listening for the message (we'll show you how to read that message on a PC in the next section)



```
forever loop
  set pir to (digital read pin P0)
  if (pir = 1) then
    show leds (5x5 grid with 10 red LEDs)
    serial write line "pir alarm"
    digital write pin P2 to 1
    pause (ms) 1000
    digital write pin P2 to 0
  else
    show leds (5x5 grid with 10 red LEDs)
```

The image shows a Scratch code editor with a 'forever' loop. Inside the loop, the first block is 'set pir to (digital read pin P0)'. This is followed by an 'if' block with the condition '(pir = 1)'. The 'then' branch of the 'if' block contains: a 'show leds' block with a 5x5 grid of LEDs where 10 LEDs are red; a 'serial write line' block with the text 'pir alarm'; a 'digital write pin P2 to 1' block; a 'pause (ms) 1000' block; and a 'digital write pin P2 to 0' block. The 'else' branch of the 'if' block contains a 'show leds' block with a 5x5 grid of LEDs where 10 LEDs are red.

You can download the code here: https://makecode.microbit.org/_H7aW45E4dLeX

Reading USB serial messages from the Microbit on your PC

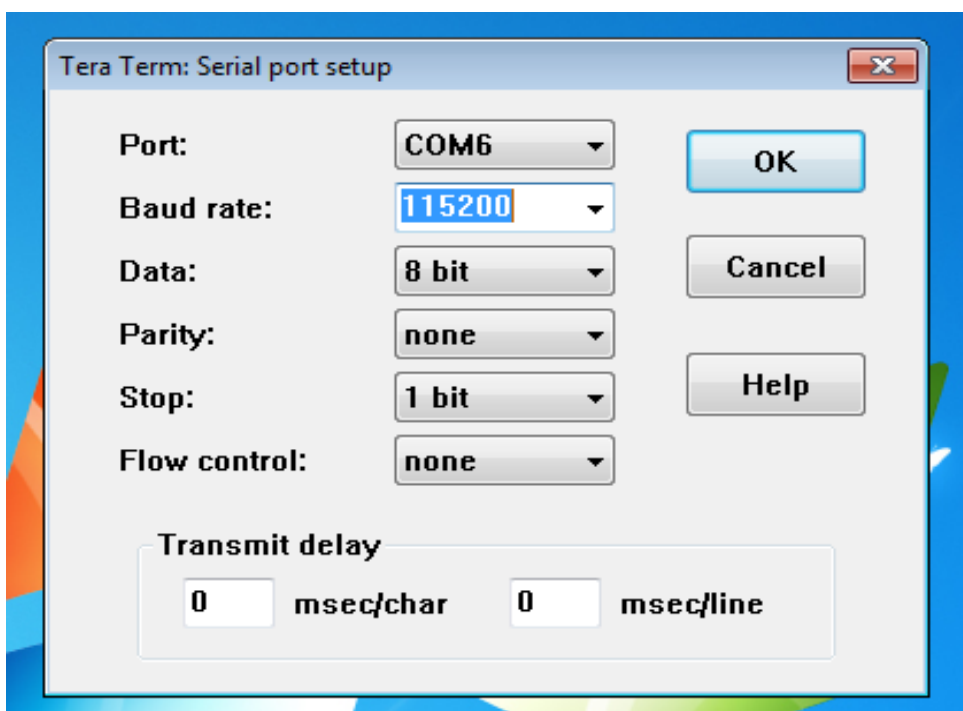
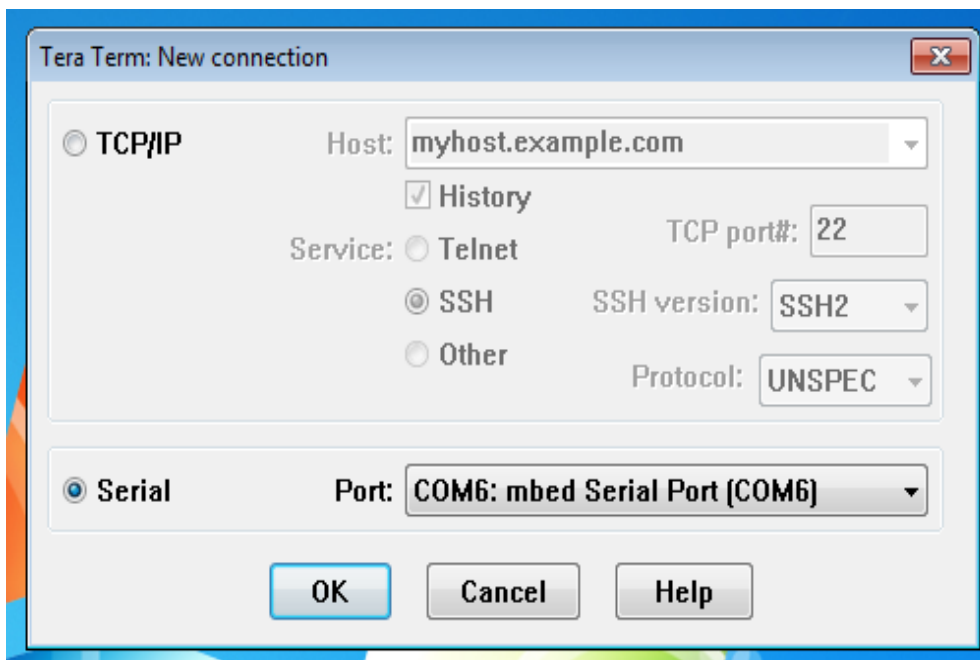
In the last example we sent a “serial write line” message over the USB port on the Microbit. If the Microbit is plugged into a PC you can read that message on the PC using the Python3 programming language and have the PC do something useful, such as write the time and date of each alarm to a log file, or email an alert to your phone.

Install Windows serial driver for Microbit from:

https://developer.mbed.org/media/downloads/drivers/mbedWinSerial_16466.exe

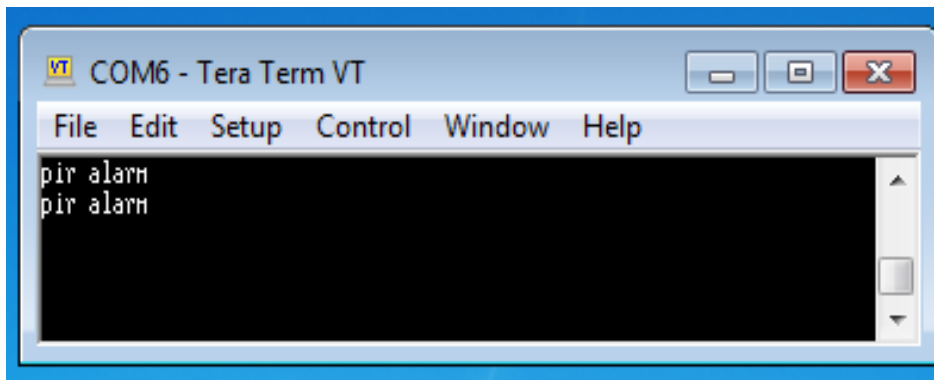
Setup up terminal emulator Tera Term to check you're receiving the serial messages:

<https://osdn.net/projects/tssh2/downloads/68252/teraterm-4.96.exe/>

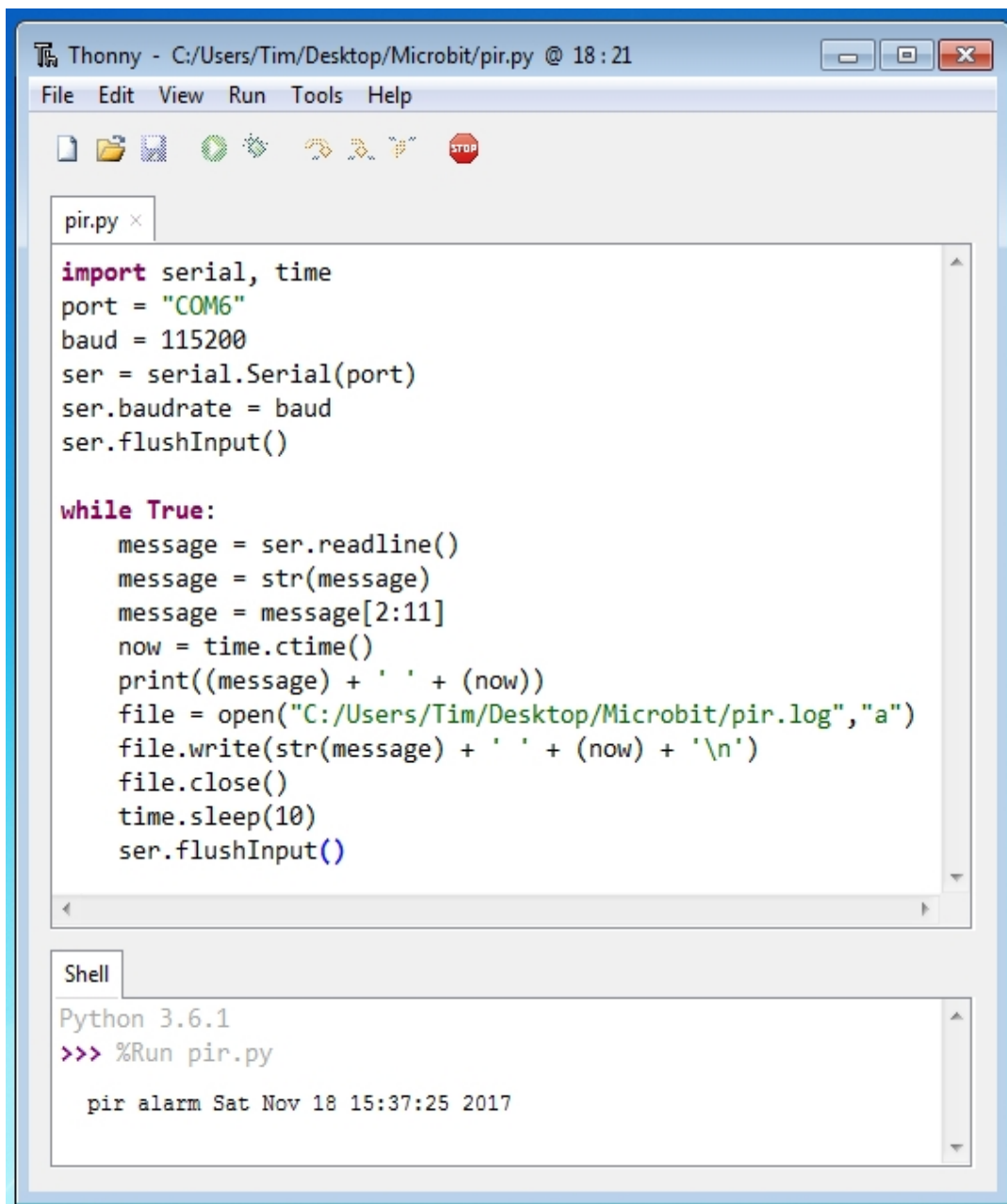


In Tera Term menus: setup → serial port → set baud rate to 115200

Serial messages from Microbit should appear in Terminal Window each time the PIR is triggered.



Install Thonny for PC, Mac or Linux. Thonny is Python 3.6 and a simple editor. <http://thonny.org>



Once you have Thonny installed go to Tools → Manage Packages and install package pyserial

You may need to change the serial port number on your Windows PC to something else. You can discover the correct serial port, from the list available in the Tera Term software.

If you're using a Linux PC the port will be described as something like /dev/ttyUSB0 instead of COM6. You should also change the folder where the PIR logfile will be stored to something that exists on your system.

Here's the script pir.py

```
import serial, time
port = "COM6"
baud = 115200
ser = serial.Serial(port)
ser.baudrate = baud
ser.flushInput()

while True:
    message = ser.readline()
    message = str(message)
    message = message[2:11]
    now = time.ctime()
    print((message) + ' ' + (now))
    file = open("C:/Users/Tim/Desktop/Microbit/pir.log", "a")
    file.write(str(message) + ' ' + (now) + '\n')
    file.close()
    time.sleep(10)
    ser.flushInput()
```

The first six lines setup the serial port, change COM6 to the correct port for you.

The while loop looks for a new text message from the Microbit over the USB serial port. When a message arrives the message is first turned into a string, and then the 3rd to 11th characters of the message are extracted. The message and date/time are then printed to the display and also to a text file called pir.log. The script then sleeps for 10 seconds, flushes the serial port and waits for a new message to arrive.

You can open the pir.log file using notepad in Windows or select Tools → Open System shell in Thonny, and a DOS command window will appear.

In the DOS command window on a Windows system you can use these commands:

<i>dir</i>	list files in folder	<i>type pir.log</i>	show the contents of pir.log
<i>del pir.log</i>	delete the file pir.log	<i>cls</i>	clear the displays
<i>cd c:/Users</i>	change directory		

In the next section i'll show you how to send an email from the PC to your phone letting you know the Microbit PIR has been triggered. This will require setting up a new free gmail account for the Python script to use.

Setting up a spare Gmail account.

The simplest way of sending emails & photos as attachments from your PC, is to setup a new Gmail account for the PC to use, even if you already have an existing Gmail account. For one thing, it gets you 15GB of new cloud storage for your alert photos & secondly it removes the complication of generating application specific passwords for other apps on your existing Gmail account.

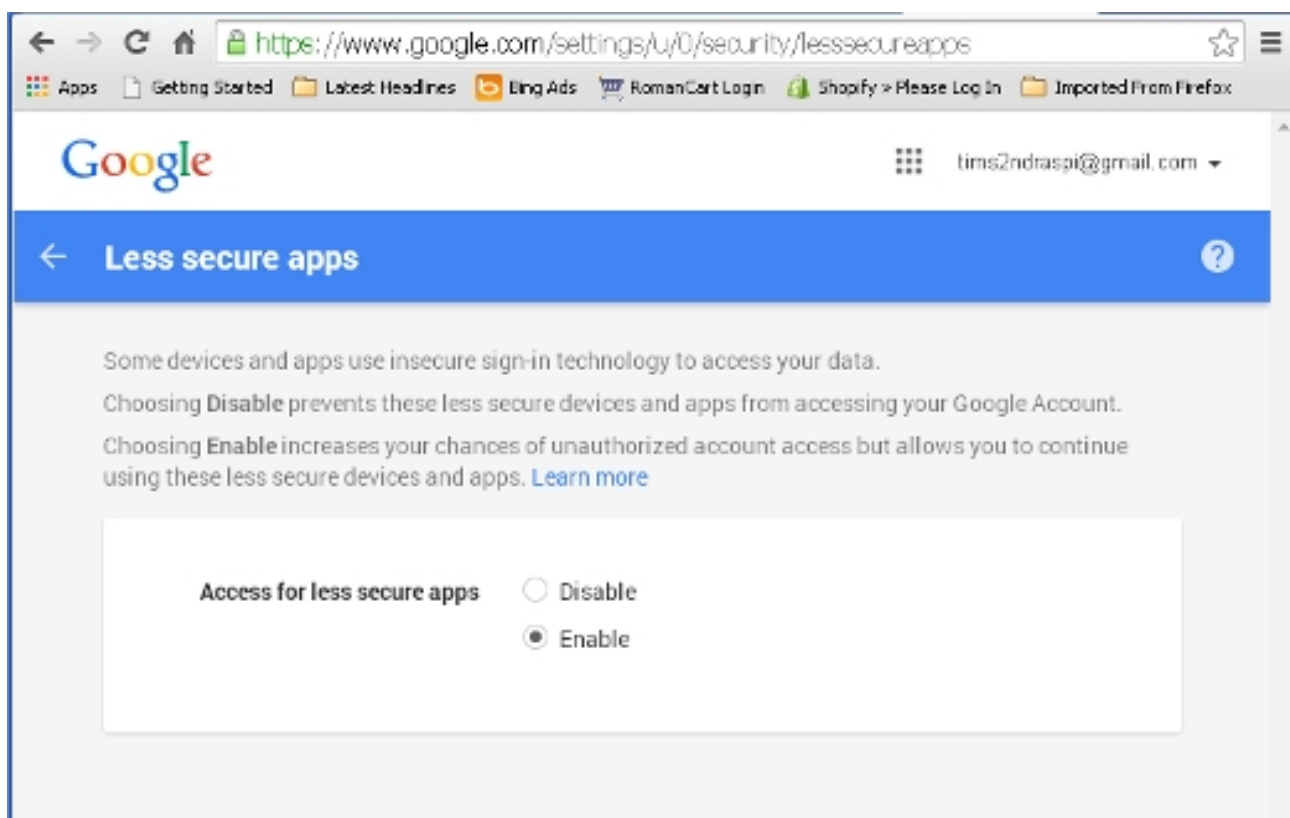
You need to create the new Gmail account in the web browser on your PC or Mac @

<https://accounts.google.com/SignUp?service=mail>

and note down the login & password for later.

Next, you need to set the new Gmail account to Enable “less secure apps”. While logged in go to:

<https://www.google.com/settings/u/0/security/lessecureapps>



Now we have a working email account, just for the PC to use when sending emails from Python. Emails from the PC running Python can be sent to any other email address on your phone or a different PC.

Pir2.py a script that emails you when the PIR attached to the Microbit detects motion

```
import serial, time, smtplib
port = "COM6"
baud = 115200
ser = serial.Serial(port)
ser.baudrate = baud
TO = 'you@yourphonesemailaddress.co.uk'
GMAIL_USER = 'newaccountyoujustmade@gmail.com'
GMAIL_PASS = 'newhardtoguesspasswordyoujustmade'
SUBJECT = 'Intruder'

def send_email(TEXT):
    print("Sending Microbit Email")
    smtpserver = smtplib.SMTP("smtp.gmail.com",587)
    smtpserver.ehlo()
    smtpserver.starttls()
    smtpserver.ehlo
    smtpserver.login(GMAIL_USER, GMAIL_PASS)
    header = 'To:' + TO + '\n' + 'From: ' + GMAIL_USER
    header = header + '\n' + 'Subject:' + SUBJECT + ' ' + now + '\n'
    print (header)
    msg = header + '\n' + TEXT + ' ' + now + ' \n\n'
    smtpserver.sendmail(GMAIL_USER, TO, msg)
    smtpserver.close()

while True:
    message = ser.readline()
    message = str(message)
    message = message[2:11]
    now = time.ctime()
    print(str(message) + ( ' ') + (now))
    send_email(message)
    time.sleep(10)
    ser.flushInput()
```

Remember to change the port from COM6 to your own, also change the TO field to the email address you use on your phone, and the GMAIL_USER and PASS to the login and password for the new Google mail account you just created.

Edublocks and MicroPython.

The Edublocks website enables you to use a Block style interface to create Python code for the Microbit <https://microbit.edublocks.org/> - We made a demo that uses all functions of the Alarmbit board:

```
from microbit import *
while True:
    if pin_0.read_digital() :
        display.on()
        image = (" 0 9 0 9 0 :"  
              " 0 9 0 9 0 :"  
              " 0 0 0 0 0 :"  
              " 0 9 9 9 0 :"  
              " 9 0 0 0 9 ")
        display.show( image )
        pin_2.write_digital( 1 )
        sleep( 1000 )
        pin_2.write_digital( 0 )
        sleep( 2000 )
        display.clear()
    if button_a.is_pressed() :
        pin_2.write_digital( 1 )
        sleep( 1000 )
        pin_2.write_digital( 0 )
    if button_b.is_pressed() :
        display.on()
        if pin_1.read_digital() :
            image2 = (" 0 0 0 0 0 :"  
                   " 0 0 0 0 0 :"  
                   " 9 9 9 9 9 :"  
                   " 0 0 0 0 0 :"  
                   " 0 0 0 0 0 ")
            display.show( image2 )
        else:
            image3 = (" 0 0 0 0 9 :"  
                   " 0 0 0 9 0 :"  
                   " 0 0 9 0 0 :"  
                   " 0 9 0 0 0 :"  
                   " 9 0 0 0 0 ")
            display.show( image3 )
        sleep( 3000 )
        display.clear()
```

and another demo that just uses the PIR to trigger the buzzer for one second and draw an unhappy face each time movement is detected:

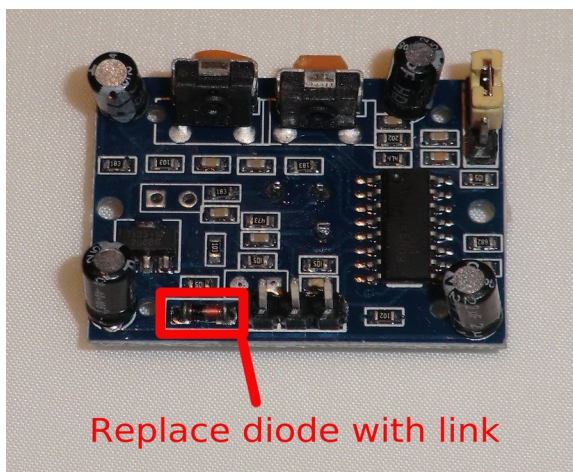
```
from microbit import *
while True:
    if pin 0 .read_digital() :
        display.on()
        image = (" 0 9 0 9 0 :"  
              " 0 9 0 9 0 :"  
              " 0 0 0 0 0 :"  
              " 0 9 9 9 0 :"  
              " 9 0 0 0 9 ")
        display.show( image )
        pin 2 .write_digital( 1 )
        sleep( 1000 )
        pin 2 .write_digital( 0 )
        sleep( 2000 )
        display.clear()
```

You can download the EduBlocks demos from www.securipi.co.uk/alarmbit-edublocks.zip

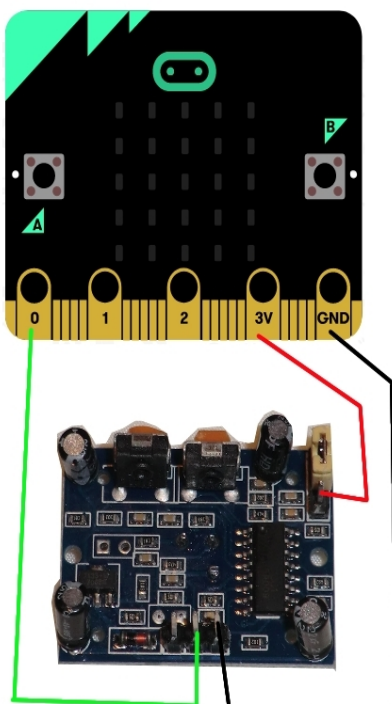
Microbits and PIR sensors

The HC-SR501 PIR sensor that comes with our Alarm Board has been modified to run at 3 volts. If you try to use an unmodified PIR with a Microbit it will appear to work correctly for the 1st detection and will usually then flip between 1 and 0 seemingly randomly. This is just because it's a 5 volt device that's being under-powered. You'll see advice on other websites telling you to adjust the two dials on the back of the PIR when using a Microbit, but that is only partially masking the problem, not fixing it. You don't need to alter the two dials on the PIR board.

If you're using the PIR clipped into a 3 way socket on a circuit board, such as our Alarm Board for Microbit, you'll need to remove a diode and replace it with a soldered link. If you do this you can't then use the PIR again with a 5V supply such as a Raspberry Pi.



If you're using a PIR connected to a Microbit using breadboard cables or crocodile clips, then you can attach the 3V feed from the Microbit to the spare pin next to the jumper, rather than the VCC pin on the PIR, attach centre signal pin to I/O contact 0 on the Microbit, and GND to GND.



If you don't already have one, our Alarm Board for Microbit is available ready-assembled from eBay and Amazon:

Without door sensor and cables

<https://www.amazon.co.uk/dp/B0779G7X49>

<https://www.ebay.co.uk/itm/132392772120>

With door sensor and cables

<https://www.amazon.co.uk/dp/B0778WBMJB>

<https://www.ebay.co.uk/itm/132392027807>

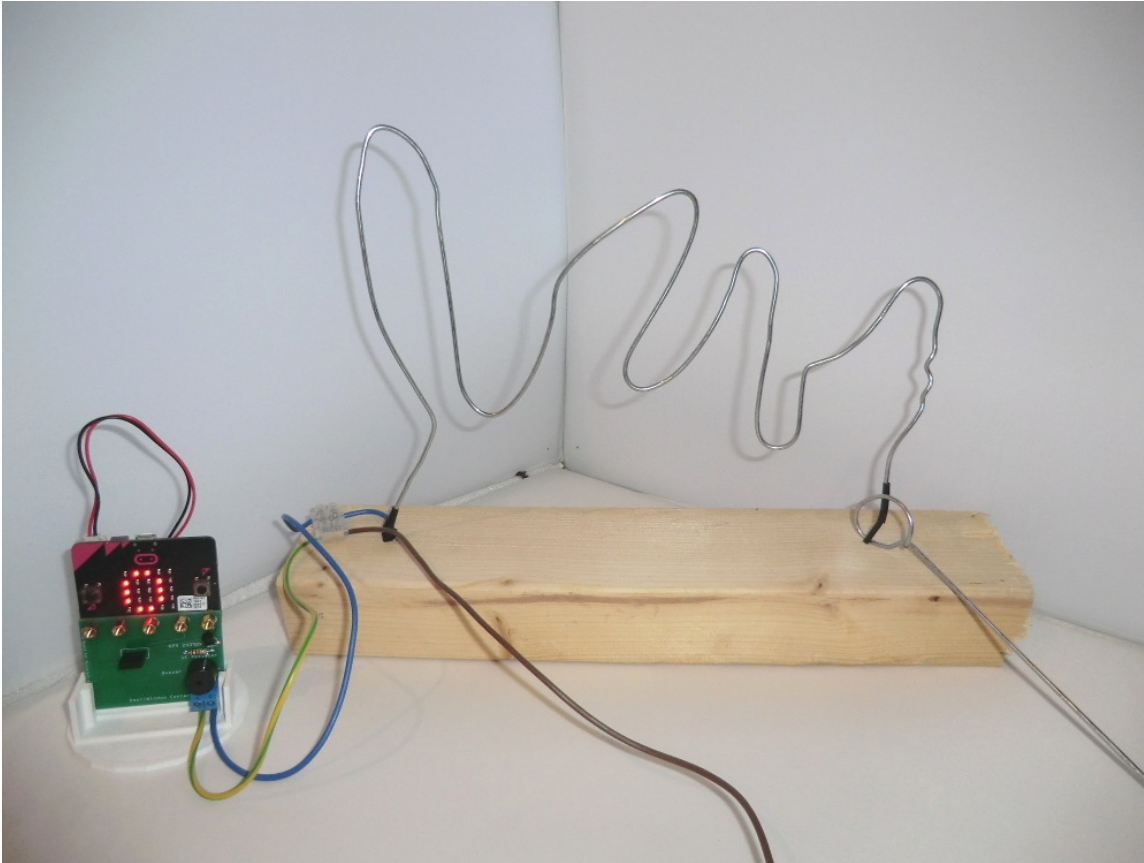
If you want to buy more than a couple of units email tim@trcomputers.co.uk for bulk pricing.

We also have them available as kits you can solder together yourself in our eBay shop.

Next page, DIY Wire Loop Game using the AlarmBit board.

Wire loop game for Microbit & Alarmbit board.

The wire loop game uses the buzzer and door contacts terminal on our Alarmbit board to recreate the classic wire loop game, where you feed a metal loop over a bendy wire course without the buzzer sounding, in the fastest possible time.



To build our version you'll need:

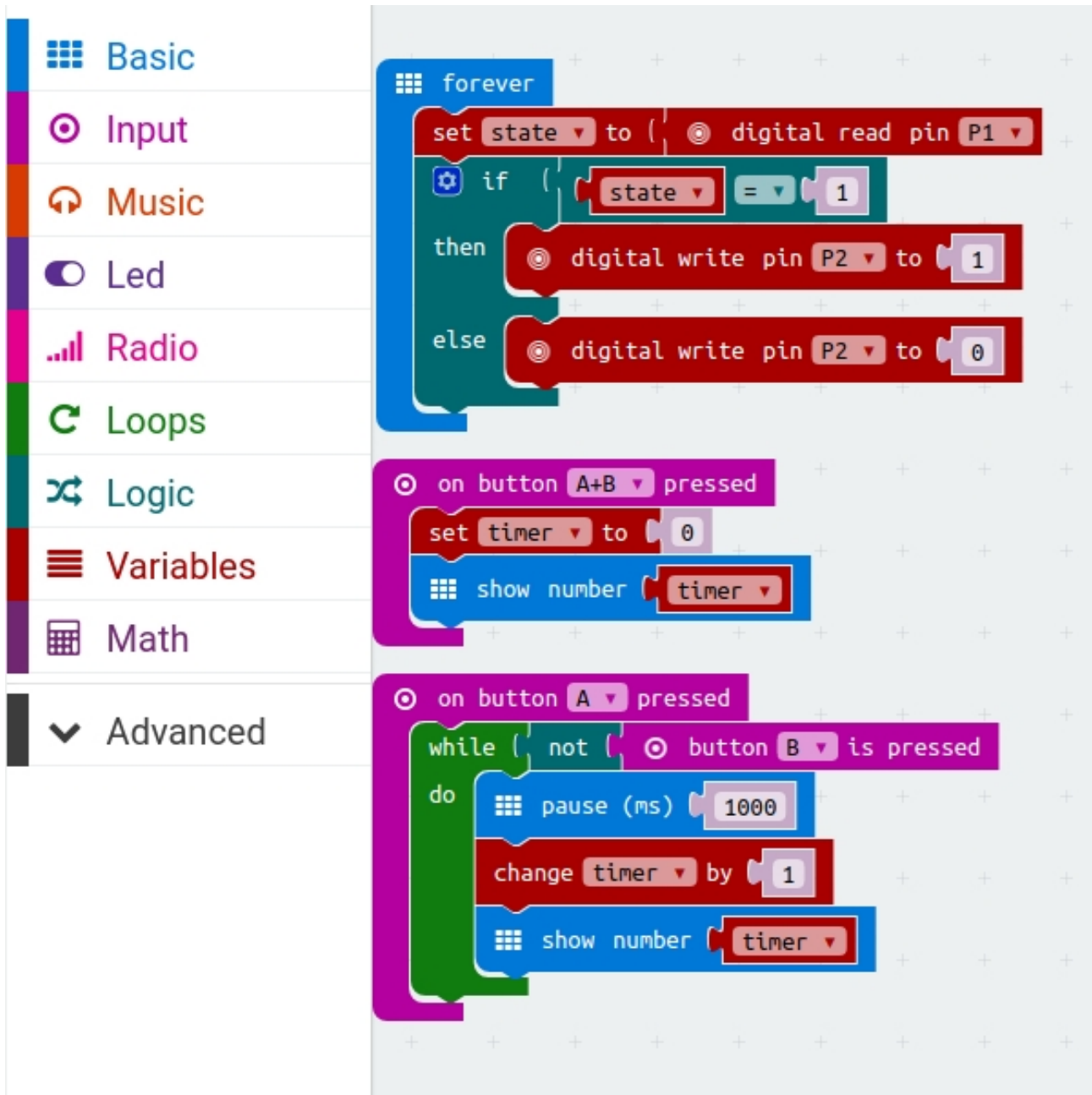
- Offcut of wood from a length of 3x2 or a decking board.
- Wire coathanger silver uncoated variety, available at dry cleaners.
- Wire – we used some wire stripped from an old mains cable.
- Pliers to help straighten out and bend the coathanger
- Insulation tape or sellotape
- Drill or screw to make two holes in the wood

Attach two wires to the door contact terminals on the Alarmbit board. One wire goes to the handle/loop, the other wire connects anywhere on the bendy wire course. When the handle/loop touches the course it completes a circuit and Pin 1 on the Microbit will read 1. If that happens we sound the buzzer, which is on Pin 2.

We also have a timer to show how long it takes to complete the course. Pressing buttons A + B on the Microbit together sets the timer to 0. Pressing button A starts the timer. Holding button B stops the timer.

The code for your Microbit is available to download here:

https://makecode.microbit.org/_YjtY5FUriX7h



The Alarmbit Board for Microbit is available from eBay and Amazon:

£10.99 Without door sensor and cables <https://www.amazon.co.uk/dp/B0779G7X49>
<https://www.ebay.co.uk/itm/132392772120>

£13.99 With PIR movement sensor, door sensor and cables
<https://www.amazon.co.uk/dp/B0778WBMJB>
<https://www.ebay.co.uk/itm/132392027807>