MINECRAFT: LIGHTS AND SENSORS

CoderDojo

BUILD YOUR CIRCUIT

Card 1 of 4

I'm Learning: Python

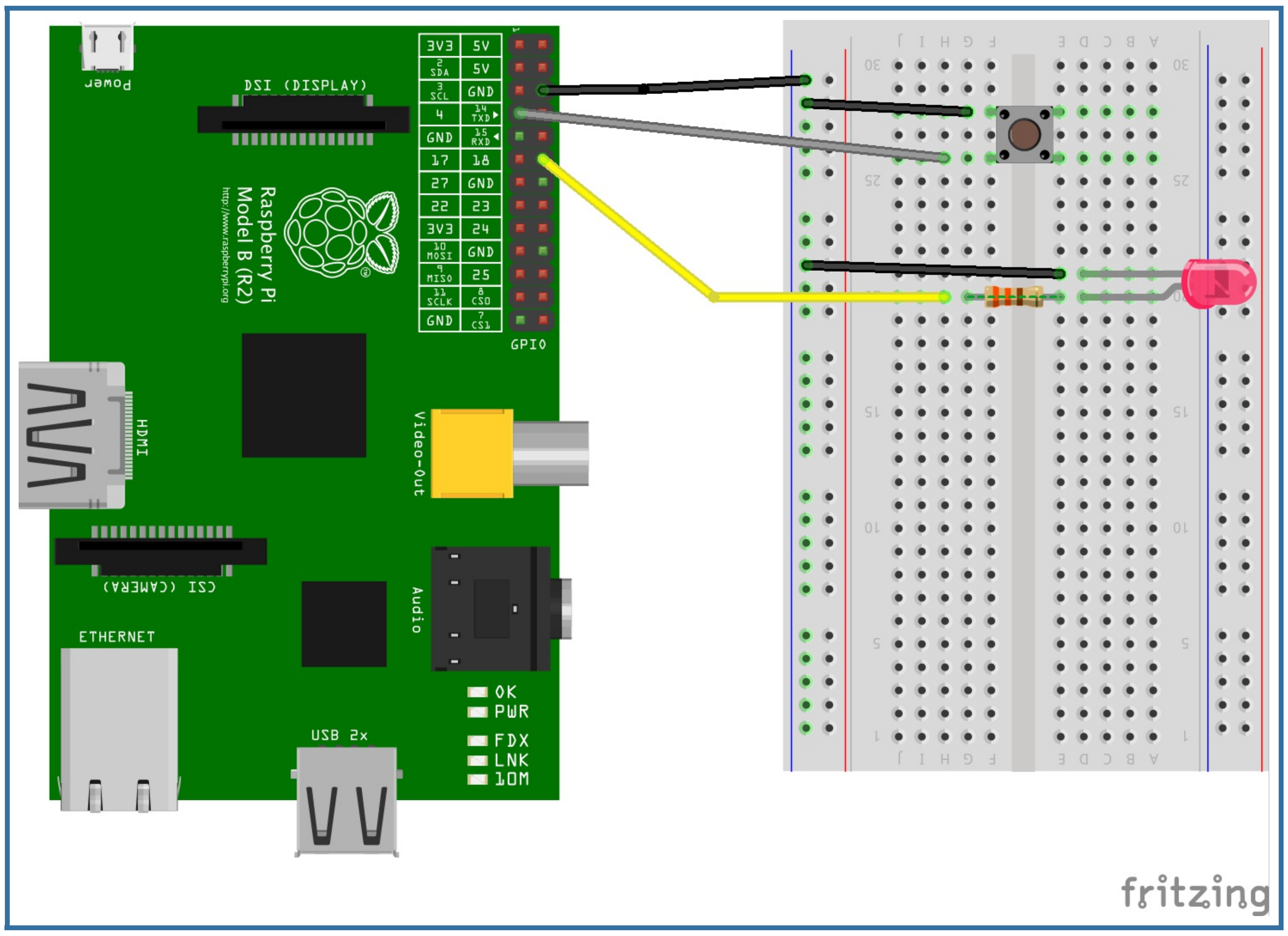
Produced by CoderDojo Banbridge // @CoderDojoBanb

1 What is it?

A Rapsberry Pi has a set of metal pins sticking out of the top of the board. The pins can be used to send signals to and from the computer and what's amazing is that they can be controlled by your programmes.

You will need:

- Raspberry Pi (any model) running Raspbian Jessie
- Breadboard
- 330 Ohm resistor X 1
- A LED
- Push Button Switch
- Male-Female & Male-Male jump leads



CoderDojo

BUILD YOUR CIRCUIT
Card 1 of 4
I'm Learning: Python

Produced by CoderDojo Banbridge // @CoderDojoBanb

2 Your turn!

Let's build the circuit!

- Shutdown the Pi and remove power supply

> **Tip**
> If your Pi does not contain labels for the pins, then the fantastic Pi Leaf is an easy and **free** aid to help you for this purpose.

- Build the circuit shown on other side of this card by connecting **(this contains a number of steps so take your time to reduce chance of mistakes :D)**:
  - One of the **GND** pins to the **-/negative** rail on the breadboard
  - **GPIO pin 4** to the **push button switch** (it doesn't matter which side) and the other side of the switch to the **-/negative** rail on the breadboard.
  - **GPIO pin 18** to the 330 ohm resistor which is connected to the positive leg of the **LED**. **The positive leg is the longer one** (the negative leg is also the one on the same side as the flat end of the LED). Then connect the negative leg back to the **-/negative** rail on the breadboard.
- Reattach the power supply and turn on the Pi

> **Test your circuit!**
> Once the Pi has powered up, test out the circuit and note down the answers to these questions:
> - What happens when you press the button?

CoderDojo

LET'S GET STARTED WITH PYTHON

Card 2 of 4

I'm Learning: Python

Produced by CoderDojo Banbridge // @CoderDojoBanb

1 Let's get started with Python

Python is a perfect language to dive into programming. Python has a gentle learning curve yet is also used by professional software developers. You don't need to know all the complexities of algorithms and syntax, you just want to write basic programs to automate mundane computer tasks.

- Start a Python 3 editor. You can do this by either clicking on `Menu > Programming > Python 3 (IDLE)` or by opening a terminal and typing `idle3 &`
- When the Python shell window opens up, click `File > New Window` to open a new window. This is where you'll enter your code.
- Save the file as **blinking_led.py**
- Enter the following code:

```python
from gpiozero import LED
from time import sleep

led = LED(18)

while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

- Save with `Ctrl + S` and run by pressing the `F5` button.

CoderDojo

Produced by CoderDojo Banbridge // @CoderDojoBanb

2 How does it work?

- To interact with the GPIO pins we'll use the `gpiozero` library and the `LED` class which is used to interact with the pin as an LED.
- We set pin 18 to operate as a LED with `led = LED(18)`.
- In the loop we alternate between setting the LED on using `led.on()` and then waiting for one second before turning the LED off with `led.off()`

Challenge:

- Make the light stays on longer than it goes off
- Replace the code to turn the light on and off with the `led.toggle()` remembering you still need to sleep(1) to ensure the lights stays on or off for one second.

Hint

You should just have two lines of code inside the while loop

- Replace the entire `while` loop with `led.blink()`

Advanced

Try to modify the behavior of `led.blink()` using various arguments as documented on pythonhosted.org/gpiozero/outputs/#led.

MINECRAFT: LIGHTS AND SENSORS

CONTROLLING A LIGHT FROM MINECRAFT

Card 3 of 4

I'm Learning: Python

Produced by CoderDojo Banbridge // @CoderDojoBanb

1 Controlling an LED from Minecraft

Using `gpiozero` along with `mcpi`, a python module that provides classes and functions to interact with the Minecraft Python API, we can add some pretty cool interactions between our LED and the world within Minecraft.

The following exercise shows us how to control the LED by standing on a block of Redstone Ore. Get coding!

- Start Minecraft running on your Pi `Menu > Games > Minecraft`
- In the Python shell, click `File > New Window` to open a new window.
- Save the file as `minecraft_led.py`
- Enter the following code:

```
from mcpi.minecraft import Minecraft
from gpiozero import LED

# Initialize minecraft and our LED pin
mc = Minecraft.create()
led = LED(18)

# blocks
redstone_ore = 73

while True:
  x, y, z = mc.player.getTilePos()
  block = mc.getBlock(x, y-1, z)
  if block == redstone_ore:
      led.on()
  else:
      led.off()
```

CONTROLLING A LIGHT FROM MINECRAFT

Card 3 of 4

I'm Learning: Python

CoderDojo

Produced by CoderDojo Banbridge // @CoderDojoBanb

2 Exercise continued

- Save with `Ctrl + S` and run by pressing the `F5` button.
- Remember to add some redstone ore blocks so you can stand on them (unless your world already has some just lying around :D)

How does it work?

- We interact with the Minecraft API using `mcpi.minecraft` and the `Minecraft` class and as before we're interacting with GPIO using the `gpiozero` library and the `LED` class.
- We initialize the minecraft library's connection to the game using `mc = Minecraft.create()`
- We set pin 18 to operate as a LED with `led = LED(18)`.
- We set a variable to hold the type of block which we will need to find to turn on the LED. Each type of block in minecraft is represented by a different number. For example air is 0, grass is 2, snow is 78. in this case we define `redstone_one = 73` as redstone ore is represented by 73.
- In the `while` loop we fetch the players current position using `mc.player.getTilePos()`. This is returned as a set of co-ordinates and is mapped to x, y, and z
- y represents vertical position of the player so we use y-1 to find the block one place beneath the player. The type of block is returned from `mc.getBlock`
- Finally we check if that block is redstone ore. If it is then turn the `led.on()`. Otherwise, turn the `led.off()`

Challenge:

- Read about the Minecraft API and change the block type from Redstone Ore to something else. Remember to save and run your code again after any updates!

CoderDojo

LET'S ADD A TELEPORT SWITCH

Card 4 of 4

I'm Learning: Python

Produced by CoderDojo Banbridge // @CoderDojoBanb

1 Let's add a switch

So far we've looked at using software to control the physical world. Let's look at how we can use a switch to modify behavior within the Minecraft world.

- In the Python shell, click `File > New Window` to open a new window.
- Save the file as `minecraft_teleport.py`
- Enter the code shown below.
- Save with `Ctrl + S` and run with `F5`.

```
from mcpi.minecraft import Minecraft
from time import sleep
from gpiozero import LED, Button

mc = Minecraft.create()
led = LED(18)
button = Button(4)
start = mc.player.getTilePos()

while True:
  sleep(0.1)
  if button.is_pressed:
      led.on()
      mc.postToChat("Teleport activated!")
      sleep(1)
      mc.player.setPos(start.x, start.y, start.z)
      led.off()
```

CoderDojo

LET'S ADD A TELEPORT SWITCH

Card 4 of 4
I'm Learning: Python

Produced by CoderDojo Banbridge // @CoderDojoBanb

2 How does it work?

Lets look at some elements we've not encountered before:

- As before we're interacting with GPIO using the `gpiozero` library and this time we're also interacting with the Button class.
- We set pin 4 to operate as a button with `button = Button(4)`.
- We set a variable start to hold the position that the player started in using `mc.player.getTilePos()`.
- In the `while` loop we check if the button has been pressed using `button.is_pressed`. If it is then we turn on the LED but also post a message to the game to indicate teleport has started using `mc.postToChat("Teleport activated")`. We then teleport the player back to their starting position using `mc.player.setPos` and turn off the LED.

Challenge:

- What happens when you run the code and press the button?

Advanced #1

Can you add any other cool effects when the button is pressed?

Advanced #2

How about changing the type of blocks that surround the player so it appears like a transporter beam?

Advanced #3

Can you make the player jump instead of teleport?