DETECTING DIAMONDS IN MINECRAFT

LESSON PLAN 2

Table of Contents

1. Detecting diamonds in Minecraft

1.1 General information

1.1.1 Short description

In this lesson plan, we will create a diamond detector using an LED light. The goal is for our LED to work as a detector, lighting up every time a diamond is found underneath (until 15 blocks).

1.1.2 Learning objectives

The main learning objectives of this lesson plan are:

- familiarizing with circuits and programs to interact with GPIO pins of Raspberry Pi
- utilizing a breadboard to interact with Minecraft Pi.
- understanding basic structures of Python programming language

1.1.3 Links to curriculum

The domains, subdomains, subjects/topics that this lesson plan can be linked to are:

- Computer Science/Informatics: processing unit and peripherals, interfaces, programming language and main structures, coding
- Technology: electronics, open-source hardware and software, sensors, digital signal, circuits, single board computers
- Physics: resistance and the unit of electrical resistance
- Chemistry / Science: different kind of minerals (diamonds)

1.1.4 Materials required

For this lesson plan (and for each student group) besides the STEMKIT console and its Raspberry Pi we'll need:

- 1 x Breadboard
- 1 x 40P T-Cobbler Plus GPIO Breakout Board
- 1 x GPIO 40P Rainbow Ribbon Cable
- 1 x LED
- 1 x 220 Ohm resistors
- Male-to-Male jumper wires
- Female-to-Male jumper wires

1.1.5 Duration

The duration of this lesson plan is estimated to be about 50 - 60 mins, i.e., one classroom hour.

1.2 Lesson plan

The lesson plan is divided in four phases, which are introduction, preparation, execution and conclusion.

1.2.1 Introduction

In this lesson we will wire up an LED to the raspberry Pi that will light up when the Minecraft character is over a diamond (within 15 blocks).

1.2.2 Preparation

First thing we need to do is to make a circuit and connect our sensor to GPIO pins of our Raspberry Pi. Before we proceed, we turn off our Raspberry Pi and unplug it. For our circuit, we will need a breadboard, resistors, jumper wires and an LED light. In addition, we use a 40P T-Cobbler GPIO Breakout Board and a GPIO 40P Rainbow Ribbon Cable to help us set up the circuit on the breadboard instead of connecting jumper wires directly on the Raspberry.

Here is how we can set up our circuit:

1 - If you have a sticky label in your kit for the Pi-T-Cobbler, stick it on, this will help you to find the correct GPIO pins more easily. Attach the Pi-T-Cobbler to the breadboard so that the black connector is on the right hand just hanging off the edge of the breadboard. Push it into the right-hand side of the breadboard so that it lines up with the holes on the far-right hand edge of the breadboard. Half of the pins should push into the top half of the breadboard, and half of the pins should push into the bottom half of the breadboard. Push quite hard to make sure it goes all the way in.

2 - Connect the ribbon cable between the Pi-T-Cobbler and the Raspberry Pi. There is a notch in the socket and a slot in the plug of the Pi-T-Cobbler that means it will only go in one way. When you connect the ribbon cable to the Raspberry Pi, make sure that the plastic lug on the connector faces away from the edge of the Raspberry Pi circuit board; otherwise, all the pins will be connected the wrong way and it won't work!

3 - Run a wire between the positive power rail at the top of the breadboard and the pin on the Pi-T-Cobbler labelled as 3V3

4 - Run a wire between the negative power rail at the bottom of the breadboard and the pin on the Pi-T-Cobbler labelled as 0V

Here is some more information about the circuit: Provided your Raspberry Pi is powered up, your LED should now light up, as it is being powered by the power supply of your computer
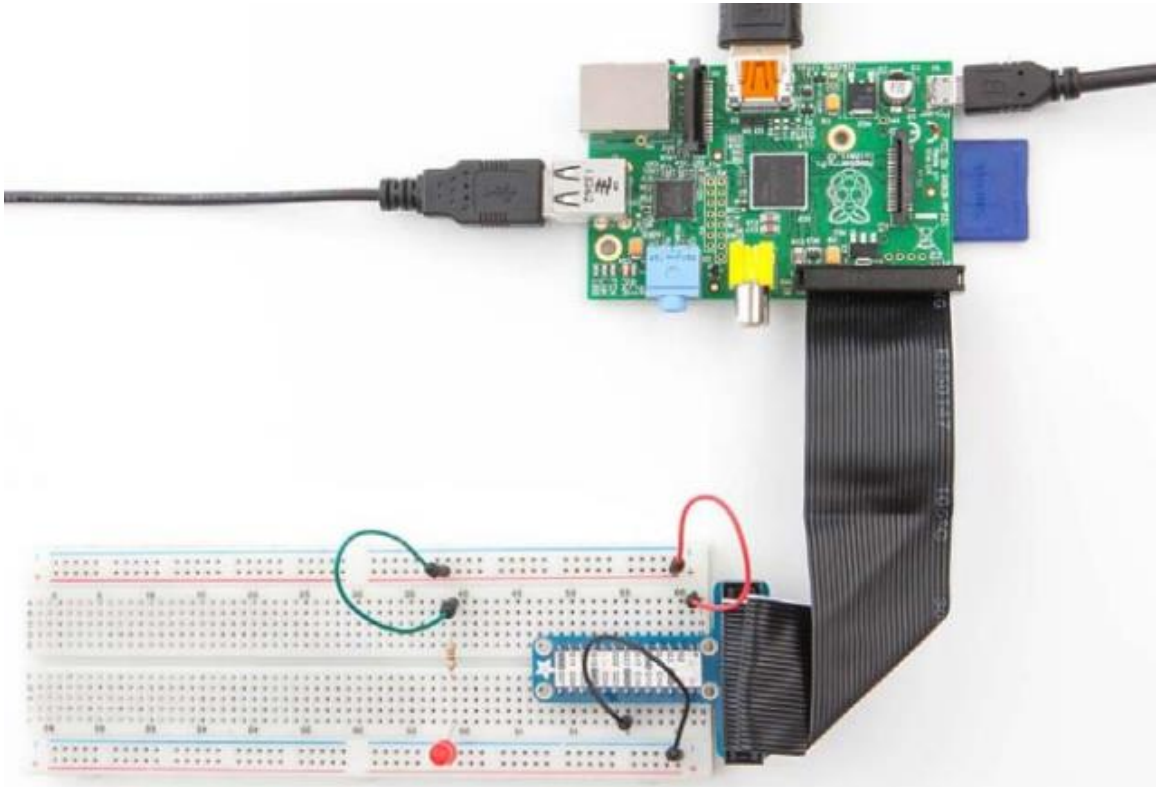

FIGURE 1 CIRCUIT DIAGRAM OF AN LED CONNECTED TO RASPBERRY PI

1.2.3 Execution

Now that we have wired our LED, we are ready to write the detector program in Python.

We start a new program by choosing File → New File and save it as **detector.py**.

Then we start writing our program by importing the necessary modules:

```
import mcpi.minecraft as minecraft
import time
```

Then we configure the GPIO setting for our computer:

```
import RPi.GPIO as GPIO
```

We then connect to the Minecraft game:

```
mc = Minecraft.create()      #create Minecraft Object
led_pin = 14                 # store the GPIO pin number
```

We setup the GPIO for the LED so it is as an output, and configure the display GPIOs:

```
GPIO.setmode(GPIO.BCM)       # tell the Pi what headers to use
GPIO.setup (14, GPIO.OUT)    # tell the Pi this pin is an output
```

Now we write a function that allows our LED to light up when it finds a diamond:

```
while True:                      # repeat indefinitely
    x,y,x = mc.player.getPos()
    for I in range(15):   #look at every block until block 15
    if mc.getBlock(x,y - i, z) == 56:
        GPIO.output(led_pin, True) # turn LED on
        time.sleep(0.25)      # wait
        GPIO.output(led_oin, False)     # turn LED off
        time.sleep(0.25)      # wait
```

We save our program and we run it.

1.2.4 Conclusion

We have been able to successfully create a diamond block detector.

1.3 References

O'Hanlon M. & Whale D., 2015, Adventures in Minecraft, Wiley Publications.

https://www.makeuseof.com/tag/learn-python-electronics-minecraft-pi-edition/