

[CHAPTER **NINE**] USING THE GPIO TO FIND A BLOCK

Program Minecraft, and connect an LED and a buzzer to a Raspberry Pi.



Above A gold block will be hidden at a random position; the buzzer and LED will help you find it

You're going to learn how to connect an LED and buzzer to your Raspberry Pi and create a program which will hide a gold block in Minecraft with their help. The LED will light up when you are close to the block, and the buzzer will let you know when you are walking towards the block.

Get Started

The first task is to start your program and get a message to appear on the Minecraft screen:

01. Start Minecraft by clicking Menu > Games > Minecraft and create a new world.
02. Press ESC to go back to the Minecraft menu but leave the game playing.
03. Open IDLE by clicking Menu > Programming > Python 3.
04. Use File > New Window to create a new program and save it as **findablock.py**.
05. At the top of your program, type the following code to import the Minecraft modules you'll need:

```
import mcpi.minecraft as minecraft  
import mcpi.block as block
```

You'll Need

- ▶ Breadboard
- ▶ LED and resistor
- ▶ Buzzer
- ▶ GPIO Zero
magpi.cc/294zLHK

06. Create a connection to Minecraft using this code:
`mc = minecraft.Minecraft.create()`
07. Post a message to the chat window:
`mc.postToChat("Go find the block")`
08. Run your program by clicking Run > Run Module.

You should see your message appear in the Minecraft chat window.

Hide a Block

Using the **random** module, you can generate a position near the player and use the Minecraft API to create a gold block there which the player will have to find:

01. At the top of your program import the **randint** function from the **random** module.
`from random import randint`
02. Add the following code to the bottom of your program to find out the player's position:
`p = mc.player.getTilePos()`
03. Generate 3 random numbers for the coordinates of the gold block:
`x = p.x + randint(-20, 20)`
`y = p.y + randint(-5, 5)`
`z = p.z + randint(-20, 20)`
04. Create the gold block at this position:
`mc.setBlock(x, y, z, block.GOLD_BLOCK.id)`
05. Run your program by clicking Run > Run Module or by pressing F5.

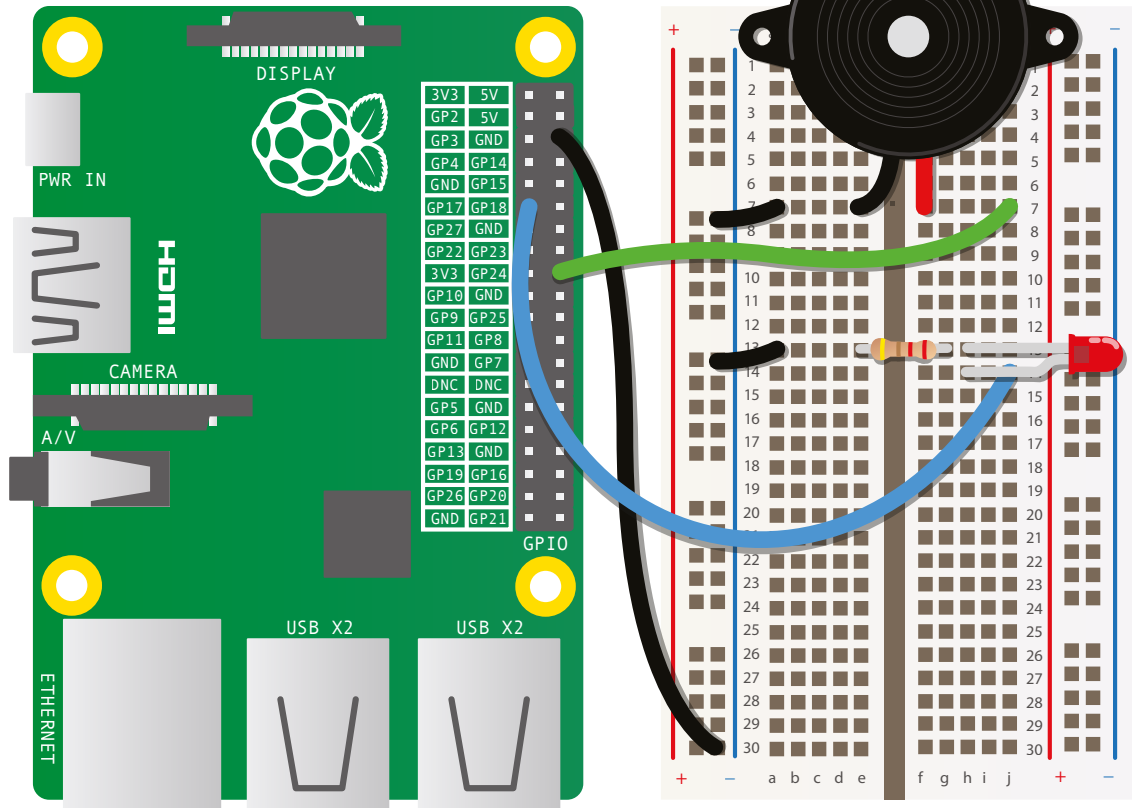
A gold block will be created within 20 blocks of the player – see if you can find it!

LEDs & Buzzers

To help the player find the block, you're going to use an LED which will indicate when the player is close by, and a buzzer to let the player know they are walking towards the block.

Use your breadboard, jumper cables, LED, resistor, and buzzer to build the circuit as shown in the diagram [fig1].

Fig1 You will build this circuit to help find the gold block



When the circuit is complete, update your program to flash the LED and buzz the buzzer to let the player know the game has started:

01. Import **LED**, **Buzzer** and **sleep** from the **gpiozero** and **time** Python modules:

```
from gpiozero import LED, Buzzer
from time import sleep
```
02. Create the LED, which is connected to GPIO 24:

```
led = LED(24)
```
03. Create the buzzer, which is connected to GPIO 17:

```
buzz = Buzzer(17)
```
04. Turn the LED and buzzer on, sleep for 1 second and then turn them back off:

```
led.on()
buzz.on()
sleep(1)
led.off()
buzz.off()
```
05. Run your program; the LED and buzzer will turn on for 1 second.

Making the Buzzer 'Buzzzzzzz'

The buzzer should buzz when the player is getting further away from the gold block, so if the buzzer remains silent you're getting closer.

Update your program so that it works out the distance between the player and the block, and turns the buzz on or off:

01. To calculate the distance you will use the maths square root (`sqrt`) function, so import it now:
`from math import sqrt`
02. Create a variable called `dist` (for distance) and set it to 0:
`dist = 0`
03. Create a variable called `gameover` and set it to False – it will be set to True at the end of the game when the player has found the block:
`gameover = False`
04. Create a loop which will continue until the game is over:
`while gameover == False:`
05. Indented under the while loop, add the code to get the player's position:
`p = mc.player.getTilePos()`
06. Work out the distance between the player and the gold block:
`xd = p.x - x`
`yd = p.y - y`
`zd = p.z - z`
`dist_now = sqrt((xd*xd) + (yd*yd) + (zd*zd))`
07. If the distance is going up, turn the buzzer on, else turn it off:
`if dist_now > dist:`
`buzz.on()`
`else:`
`buzz.off()`
08. Set the variable `dist` to `dist_now` so it can be compared next time around the loop:
`dist = dist_now`
09. Run the program. The buzzer should buzz when the player is getting further away from the gold.

Nearly there LED

To give the player a chance to find the block when he is close by, the LED should light up.

01. Turn the LED on when the distance to the gold block is less than 5:

```
if dist_now < 5:
    led.on()
else:
    led.off()
```
02. Run your program and when you get close, the LED should light up.

Game over

Once the player finds the block, the game is over.

01. When the distance between the player is less than 1.5, set the **gameover** variable to True and post a message to let the player know:

```
if dist_now < 1.5:
    gameover = True
    mc.postToChat("You got GOLD")
```
02. Finally, clean up by turning off your LED and buzzer:

```
led.off()
buzz.off()
```
03. Run your program and find the gold block!

Next steps

Can you take this program forward and make it your own? Some ideas:

- › Create a timer so that you can have races trying to find the gold block.
- › Add more LEDs (i.e. red, yellow and green) which show when you're getting nearer to the gold.
- › Hide several blocks which the player has to find in order.