



CyberCamp at UNK

GPIO project one: LEDs

What you will learn with this tutorial . . .

In this tutorial, we will develop together a little project using the GPIO pins of our Raspberry Pi. We will build a circuit and code the Python script responsible for turning on a LED light directly from our Pi. After understanding how it works you will be asked to code your own LED project.

1 GPIO: Raspberry Pi

GPIO stands for ***General Purpose input/output***. The Raspberry Pi has a row of GPIO pins along the edge of the board. These pins are a physical interface between the Pi and the outside world.

The Raspberry Pi GPIO is composed by 26 pins, 17 are GPIO pins and the others are power or ground pins.

Many things can interact and being controlled by the GPIO pins of the Raspberry Pi. In this project we will learn how to use the pins to connect a LED to our Pi, and after that code a Python script that will be responsible to turn this LED on. In the image below you can see where the pins are located on our board, and the description of each one of them. It may seem a little complicated but don't worry, it is actually really simple, and we will go together step by step in order to build our project.

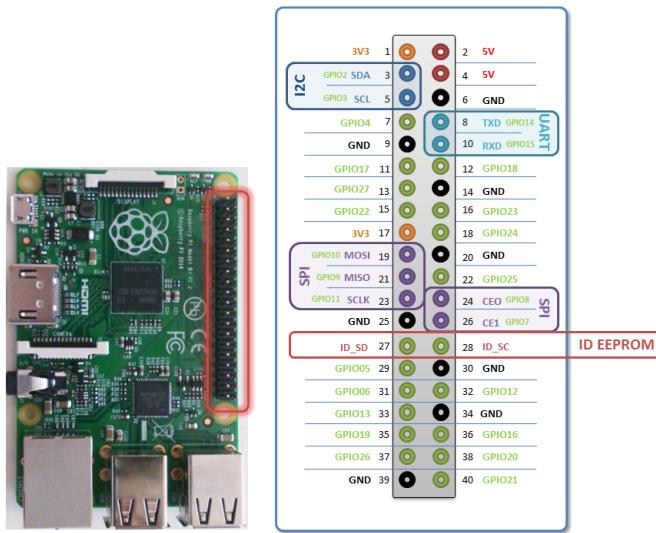


Figure 1: Raspberry Pi GPIO location and pins description

In the next section we will take a look at the components that we will need to develop our project.

2 Required components

The image below shows the components required for our LED project.

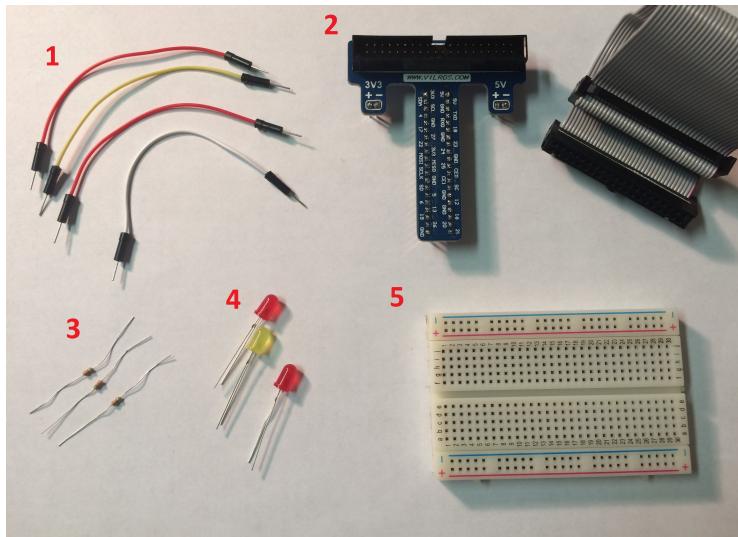


Figure 2: Required components

Let's go through a brief description of our components:

- 1. Wires:** We will use simple wires to connect the components of our circuit.

2. **Cobbler:** This component is used to extend the GPIO pins of our Pi, so we can have more space to work with our pins.
3. **Resistors:** Electrical component that implements electrical resistance as a circuit element. We need it to create the necessary resistance to turn our LED on. In this project we are using 330 ohm resistors.
4. **LEDs:** Simple LEDs that we want to control with our Python script.
5. **Breadboard:** Construction base for prototyping of electronics. We will build our circuit over it.

Now that we know everything we need, let's start building our circuit.

3 Building the circuit

Our first circuit will be a simple LED connected to a resistor that we will turn on using a Python script later. First thing to do is to attach the cobbler to the breadboard and connect it to our Pi. The result should be something like this:

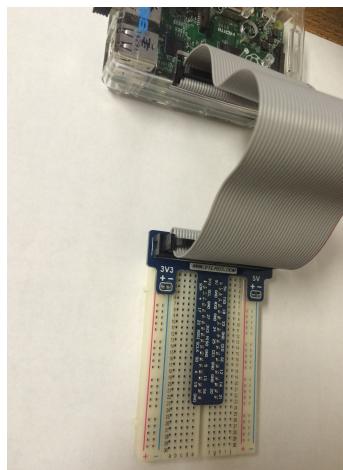


Figure 3: Breadboard and cobbler connected to the Pi GPIO

Now that we have our cobbler properly connected to the breadboard and to the Pi, we will build our circuit.

The image below shows how your circuit should look like. Labeled numbers are described right after the picture, explaining how the components should be properly connected.

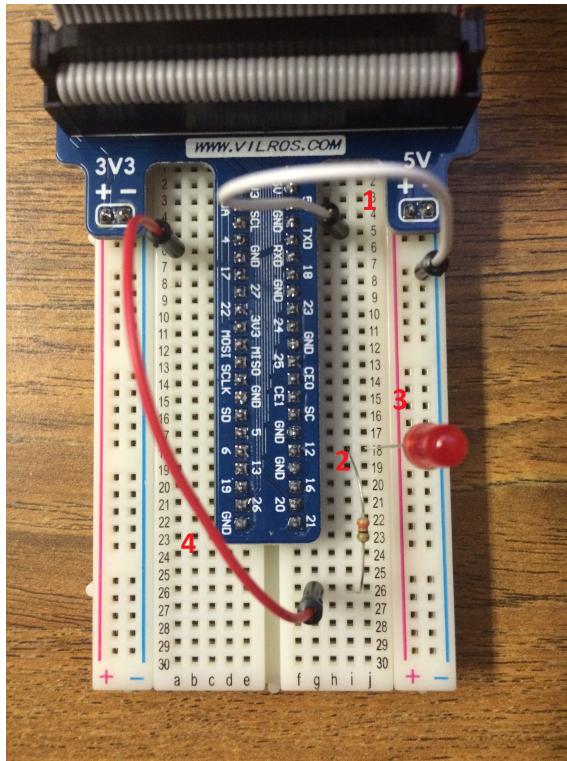


Figure 4: Complete one LED circuit

1. Connect one side of a wire on a ground pin (**GND**) and the other side in a negative (-) spot of the breadboard.
2. Get one 330 ohm resistor and put it on a place of your choice. Here we put one side into row 18 and the other one into row 26, this way we get a distance between the cobbler and have more space to connect the next components, making our work easier.
3. Place the LED in the same row of the first side of your resistor. LEDs have a shorter and a longer leg. The shorter leg is negative, hence it must be connected into a negative (-) pin. The longer leg is the positive one and must be put in any spot in the same row of your resistor.
4. Now we just have to connect our LED to a GPIO pin. Place one side of another wire into the same row of the second side of your resistor, and the other side of the wire must be connected to a GPIO port of our Pi. In our circuit, we connected it to the port 4. You could use other ports like 17 and 22, but to keep it easier since we will code these ports in a moment, please select also port 4.

Our circuit is done! What we have to do now is to write our Python script that will be responsible for turning our LED on. Let's do it.

4 Coding the script

This is how our script will work: Our LED will turn on, keep alight for 3 seconds, and then turn off. Simple, right?

But first, in order to do so, we need to install the **GPIO module** in our Raspberry Pi. A *module* is basically a file containing Python definitions and statements that we can use in our codes. In this case, the GPIO module contains functions and definitions that we will use in our LED script.

To install the module open your **terminal** and type:

```
sudo apt-get install python-dev python-rpi.gpio
```

Hit **enter**. It will probably ask for your confirmation, type **Y** and hit enter again. This is all we need to install. Now we are ready to write our code.

Open your code editor and type the following code:

```
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setwarnings(False)
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setup(4, GPIO.OUT)
6 GPIO.output(4, True)
7 time.sleep(3)
8 GPIO.output(4, False)
```

This is all that we need of Python code. Let's understand what is going on. First and second lines tell Python that we will use functions of these two libraries, **RPi.GPIO** (responsible for controlling the GPIO pins) and **time** (responsible for the function **sleep** used later).

The third line simple turn off the warnings that Python may send to us about the GPIO (they are really annoying and we don't need to worry about them now). Fourth line specifies the mode of the labeled pins we will use, this line is necessary, but you don't need to worry exactly how it works now.

Fifth line tells Python that we will use port 4 of the GPIO pins. This is the same port we connected our LED, remember? The number port must be the same where the LED is connected, otherwise our LED will not turn on.

Sixth line turns our port 4 (our LED!) on. After that, **time.sleep(3)** makes our script wait for 3 seconds. During this time out LED will remain on. The last line turns the port 4 off, hence our LED turns off too.

Save your file and run it! If your circuit and your script are correct your LED will turn on, wait for 3 seconds, and then turn off. Our project is done! If it worked well for you, go ahead to the exercise.

Now try for yourself!

*It's your turn! Create a new file and name it **threeLedTime.py**.*

Our project had one LED that kept on for 3 seconds. How about have three LEDs and make them be turned on for 1 second each? When the first turns off the second turn on, and when the second turns off the third turns on.

You will have to modify a little bit the circuit. Add two more resistors and two more LEDs, and remember to connect the new LEDs to new GPIO ports. Choose whatever ports you want.

*When writing your script remember that now you have two more ports to setup!
Good luck!*