

1 Carlsbad Library Beginning Raspberry Pi Circuit Class with Python

1.1 Pi Expansion Header

1.2 Electrical Considerations

1.3 Python gpiozero library for Raspberry Pi

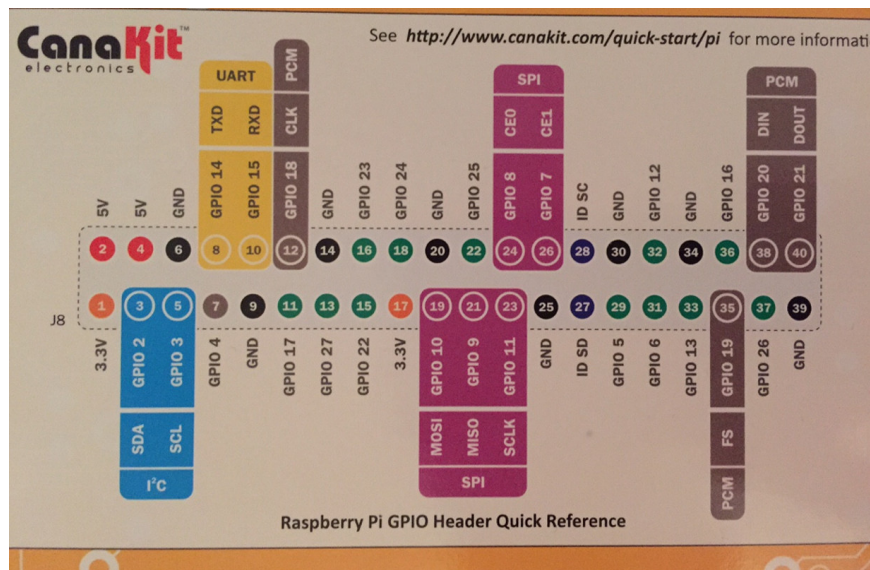
1.4 First Circuit with Raspberry Pi

1.5 Using python shell to test your Circuit

1.6 Modify turtle race to use button and leds

2 Expansion Header

The expansion header is a 40 pin connector on the raspberry pi that allows connect to additional devices or circuits. A add-on board is usually called a "HAT" (Hardware Attached on Top). However it is possible to wire up to the pins directly to a solderless breadboard.



3 Electrical Considerations

The Raspberrry Pi pins are only made to accept 3.3V maximum input. Many common device use 5V as a logic high level, and cannot be directly connected

to a raspberry pi. If you need to connect such a device you need a "level shifter" circuit. Also the maximum current a raspberry pi pin can supply (or sink) is 16mA. The maximum total gpio current is 50mA. Our circuit has 2 LED's with 330 ohm current limit resistors. $3.3\text{V}/330\text{ ohms}$ is 10mA per LED max (actually much lower) so we are OK. For comparison the maximum current on Arduino Uno is 40mA per pin and 200mA total.

4 Python gpiozero library for Raspberry Pi

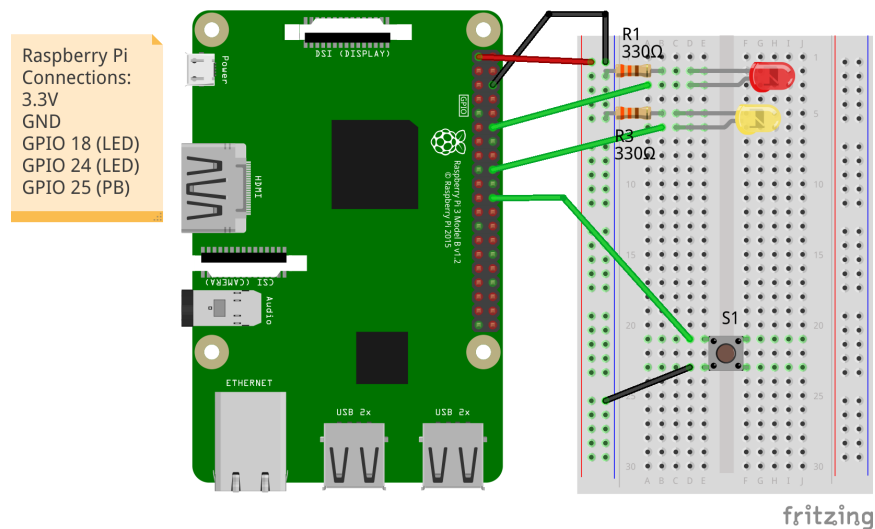
In order to read the button or write the leds we wil need to use a new python library. We will use the gpiozero library, which is usually already installed. You can google "gpiozero python" to find the library, or look for it here.

<https://gpiozero.readthedocs.io/en/stable/>

Let's take a look at the web page documentation.

5 First Circuit with Raspberry PI

You will find this drawing with the handout material on the raspberry pi



6 Where is the pull up resistor for the button?

The pizero button class will enable a pull up resistor internal to the Raspberry pi, so it's not necessary.

7 Using python shell to test your Circuit

Once you have your circuit connected we will need to test it. We will use the python shell

```
### First tell the shell you want to use LED
>>> from gpiozero import LED
### Make a red LED
>>> red = LED(18)
### Turn it on then off
>>> red.on()
>>> red.off()
### Repeat it with the yellow LED
>>> yellow = LED(24)
>>> yellow.on()
>>> yellow.off()
### Test the button.
>>> from gpiozero import Button
>>> button = Button(25)
>>> button.is_pressed
False

### Hold down the button while you press enter
>>> button.is_pressed
True
>>>
```

8 Python Exercises

We have an version of last weeks turtle race program that uses the button to start the race, and indicates the winner with a red or yellow led. Pressing the button again will exit the program. We have given you most of the program but left some parts for you to fill out. Open the file `studentturtleracesingle.py` and look for "TODO" in the comments.

You can also run the race from the command prompt by going to the directory where the file is and typing

- `python studentturtleracesingle.py`

9 Additional things to try?

How could you change the program to make it so it will run a new race every time? How would you then tell the program to end? You can try it on your own, or we have an example `turtleracebutton.py` that you can look at.