

Contents

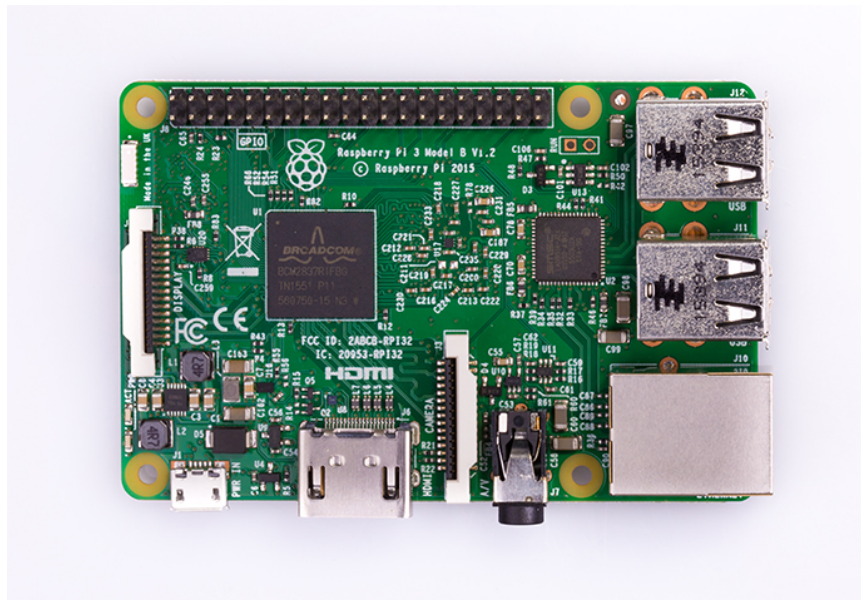
1	Carlsbad Library Beginning Raspberry PI Circuit Class	2
1.1	What is Raspberry PI	2
1.2	Expansion Header	2
1.3	Electrical Considerations	2
1.4	First Circuit with Raspberry PI	2
1.5	Using gpio command to test your Circuit	2
1.6	Using processing to control your circuit	2
2	What is a Raspberry Pi?	2
3	Expansion Header	3
4	Electrical Considerations	3
5	First Circuit with Raspberry PI	3
6	Using gpio command to test your Circuit	4
6.1	Example Commands (don't type the // comment portion) . .	5
6.2	gpio -g mode 18 out // set gpio number 18 (header pin 12) to be an output	5
6.3	gpio -g mode 25 in // set gpio number 25 to be an input . . .	5
6.4	gpio -g write 18 1 // turn on led	5
6.5	gpio -g read 25 // read the button (try with button held down and not held down)	5
6.6	excercise - can you turn on and off the second LED in the circuit?	5
6.7	hint - use the up and down arrow to edit previous commands without retyping	5
7	Using processing to control your circuit	5
8	Simple Input	5
9	Simple Output	5
10	Processing Excercises	6
11	Resources, other things to try	6

1 Carlsbad Library Beginning Raspberry PI Circuit Class

- 1.1 What is Raspberry PI
- 1.2 Expansion Header
- 1.3 Electrical Considerations
- 1.4 First Circuit with Raspberry PI
- 1.5 Using gpio command to test your Circuit
- 1.6 Using processing to control your circuit

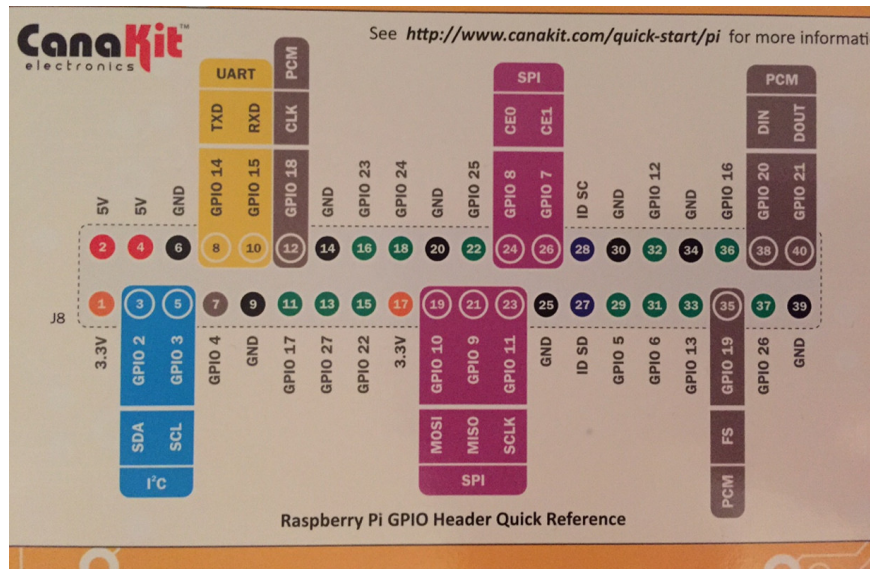
2 What is a Raspberry Pi?

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. There is a tradition of naming computers after fruits (Tangerine, Apricot, Acorn) which is where the "Raspberry" comes from. The "Pi" part refers to the Python programming language



3 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.

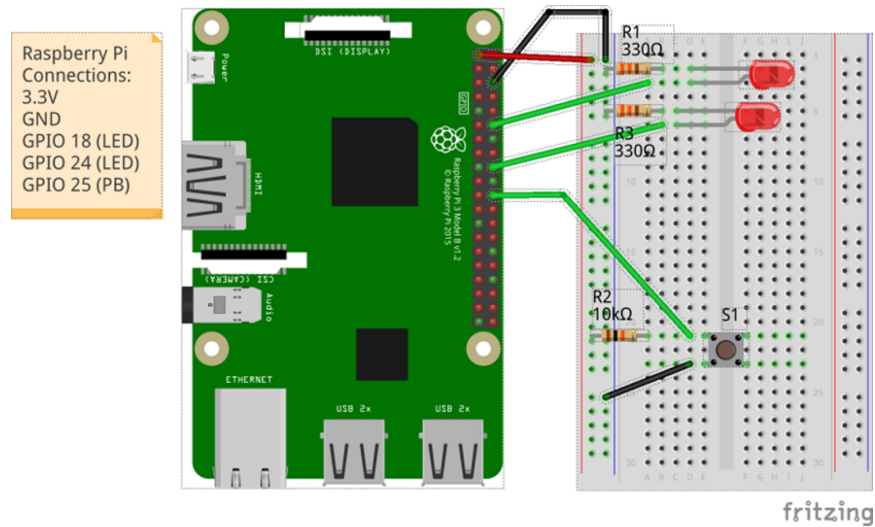


4 Electrical Considerations

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. $3.3v/330\text{ ohms}$ 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.

5 First Circuit with Raspberry PI

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



6 Using gpio command to test your Circuit

Once you have your circuit connected we will need to test it. The simplest way to test your circuit is to use the gpio command in a terminal. You can find out about the GPIO command at the [wiring pi](http://www.wiringpi.com) website. Similar to the arduino there is one command to set the pin to an input or output and another command to write or read the pin. Also note there are several numbering conventions available for the pins. We will use the Broadcom GPIO numbers which is the -g option

- 6.1 Example Commands (don't type the // comment portion)
- 6.2 `gpio -g mode 18 out // set gpio number 18 (header pin 12) to be an output`
- 6.3 `gpio -g mode 25 in // set gpio number 25 to be an input`
- 6.4 `gpio -g write 18 1 // turn on led`
- 6.5 `gpio -g read 25 // read the button (try with button held down and not held down)`
- 6.6 exercise - can you turn on and off the second LED in the circuit?
- 6.7 hint - use the up and down arrow to edit previous commands without retyping

7 Using processing to control your circuit

Once you have your circuit working, there are many ways to control your circuit on the raspberry pi using a program. We are going to use the processing language which is based on java. It's IDE is very similar to Arduino (which is based on processing) but instead of setup and loop, you have setup and draw. To Open processing, Programming->Processing to start processing. Once you have started Processing goto File->Examples. In the Examples Window open Libraries->Hardware I/O->Simple Input

8 Simple Input

The Simple Input example used GPIO 4 as an input. Our circuit uses GPIO 25, So change 4 -> 25 in the GPIO.pinMode and GPIO.DigitalRead commands. Notice there is a setup and draw routine, similar to Arduino setup and loop routines.

9 Simple Output

Next open the Simple output example. Again it uses GPIO 4. Change the GPIO.pinMode commands and GPIO.digitalWrite commands to use 18 instead of 4. Can you change the sketch to use both GPIO 18 and GPIO 24 and have GPIO 18 LED on while GPIO 24 LED is off?

10 Processing Exercises

No that you have these pieces, can you change the Simple Output to use the input to change it so when the button is pressed is Blinks GPIO 18 and 24 both on and then off when pressed, otherwise GPIO LED is on while GPIO 24 LED is off?

11 Resources, other things to try

You can find a similar class online from the library using lynda.com. Access it for free with your library card through the library portal (e-research) under Buisness an Investing <http://www.carlsbadca.gov/services/depts/library/services/research.asp>

<http://cbcl.idm.oclc.org/login?url=http://iplogin.lynda.com>