**GPIO Memory game!**

Today we’re going to learn about:

* GPIO outputs
* GPIO inputs
* Random numbers & lists in Python
* Github!

**The Game**

This game is similar to the Simon game. We’ll use 3 LEDs and the program will display a random sequence that the user has to repeat using buttons.

First, lets get the code

Go to <https://github.com/coderdojo-newtampa/pi-gpio-memory>

The README page on github has the instructions on how to get the code.

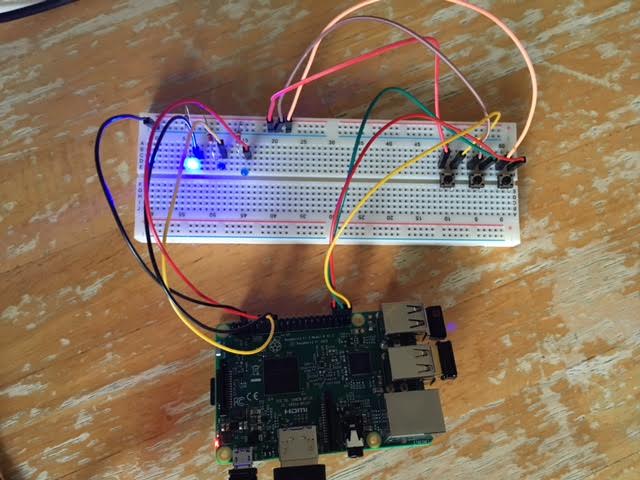
**Wiring**

Here’s what we need:

* 1 breadboard
* Raspberry Pi with wifi adaptor
* 3 LEDs
* 3 buttons
* 3 resistors (could be 1k Ohm, 470Ohm, 220 Ohm)
* 7 male-female jumper wires
* 3 male-male jumper wires

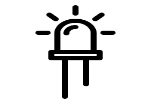
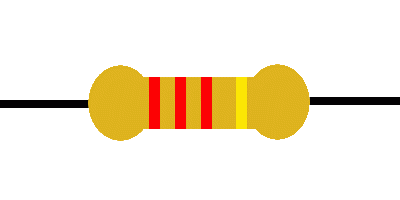
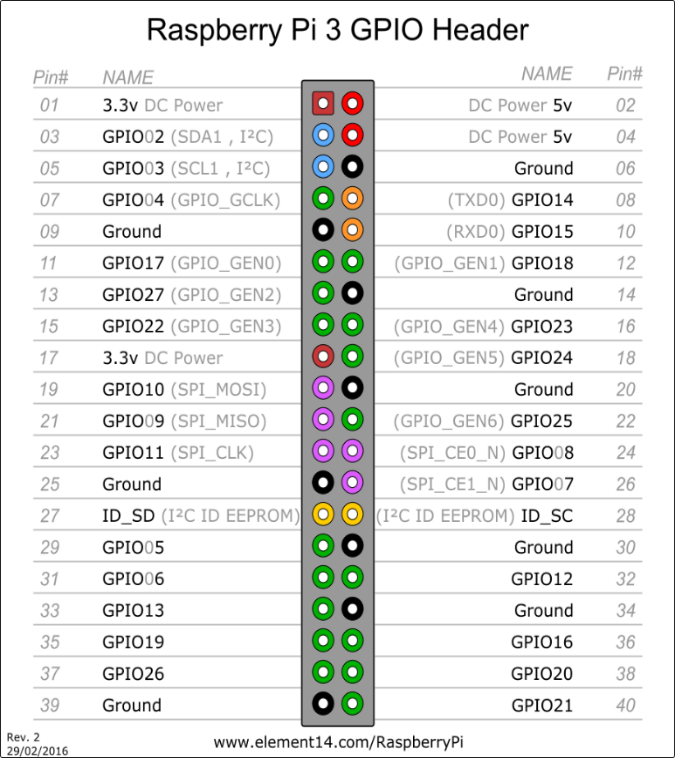
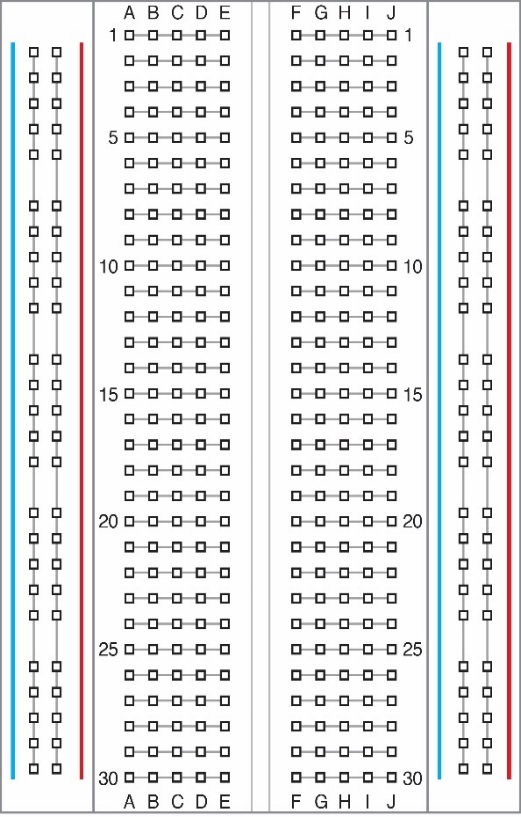


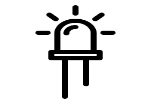
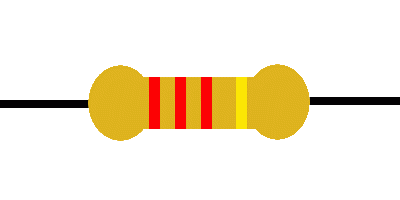
-

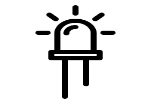
Here’s what we’re going to build!

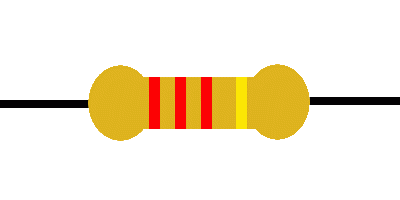
LEDs connected to GPIO 17,27,22

Buttons to GPIO 16,20,21















Remember, LEDs connect to ground on the short/bent end

**Let’s test**

To test your wiring, run these programs

To just test the LEDs

cd pi-gpio-memory/src/test

python ledtest.py

You should see the lights turn on in different patterns

To test the lights and the buttons

cd pi-gpio-memory/src/test (same dir)

python buttontest.py

If you push a button, the corresponding LED should light up. If these programs don’t work, we have an issue with the wiring!

The Game code

cd pi-gpio-memory/src/

idle3 memory-game.py

Add this code at the end of the file

|  |
| --- |
| setup() |
|  | animate(7, 0.1) |
|  |  |
|  | while not gameOver: |
|  | time.sleep(0.5) |
|  | ledSequence.append(random.choice( [led1, led2, led3] )) |
|  | display() |
|  |  |
|  | for led in ledSequence: |
|  | button = readInput() |
|  |  |
|  | if (led == button): |
|  | blink([button], 0.5, 0.2) |
|  | else: |
|  | blink([button], ledWait, 0.1) |
|  | gameOver = True |
|  | animate(3, 0.3) |
|  | break |
|  |  |
|  | print("Your score is [%d], thank you for playing" % (len(ledSequence)-1)) |
|  | gpio.cleanup() |

To run the game, type:

python memorygame.py