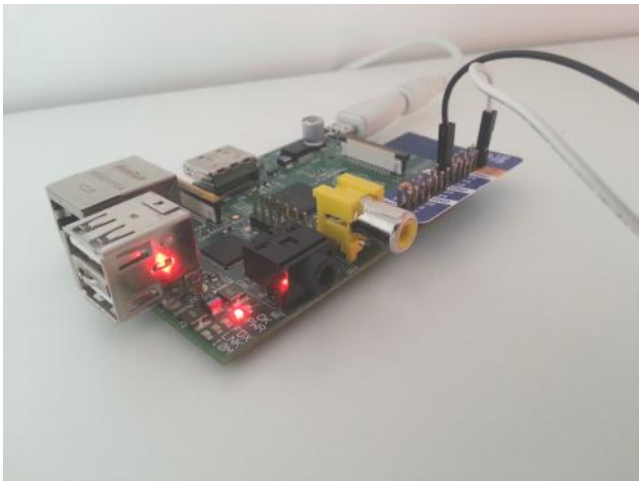
LIGHTS 3
USE BUTTON TO CONTROL LED

LEVEL: **BEGINNER**



COMPUTING PoS KS1:
- Understand what algorithms are
- Create and debug simple programs
- Use logical reasoning to predict the behaviour of simple programs

COMPUTING PoS KS2:
- Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems
- Use sequence, selection, and repetition in programs
- Use logical reasoning to explain how some simple algorithms work

PROGRESSION PATHWAYS:
- I know what an algorithm is and I can express simple algorithms using symbols
- I can create a simple program
- I can run, check and change programs

COMPUTER SCIENCE CONCEPTS COVERED:
- Algorithm
- Sequence
- Selection

PERFORM

How does a light switch work?
How do you turn light on and then off?

Equipment: Double sided A4 card: Light bulb on/ off; 2 actors: Lightbulb & Switch

Act Out:

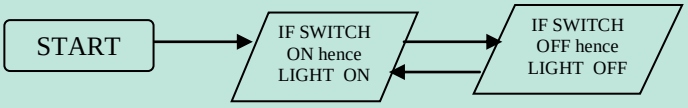
1) Switch and Light bulb stand next to each other – Light bulb is off
2) Switch on and light switches on
3) Switch off and light bulb switches off

PLAN

Pseudocode:

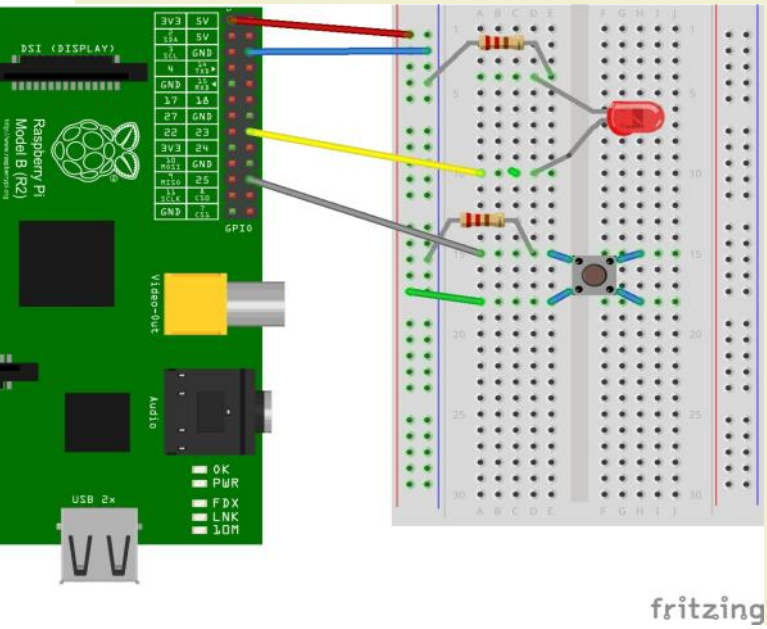
```
Start
     If switch pressed
          Turn Light On
     Otherwise
          Turn Light Off
```

Flow Diagram:



HARDWARE SETUP

Diagram:



Equipment: Raspberry Pi; 1 x LED; 1 x button; 4 x Male to Female Jumper Leads; 1 x Male to Male Jumper Lead; 1 x 330 ohm resistor; 1 x 1k ohm resistor; Breadboard

Method:
1) Set up breadboard as in the diagram opposite. *Note:* The 1k ohm resistor is used for the button. The 330 ohm resistor is used to protect the LED. The short leg of the LED should be next to the resistor.
2) Attach GPIO pin 1 to power rail (red) of breadboard (red jumper in diagram).
3) Attach GPIO pin 6 to ground rail (blue) of the breadboard (blue jumper in diagram).
4) Attach GPIO pin 16 to the same row as long leg of LED (yellow jumper in diagram).
5) Attach GPIO pin 22 to the same row as 10k ohm resistor (grey jumper in diagram).
6) Attach power rail (red) to the same row as the leg of button without the resistor (green jumper in diagram).

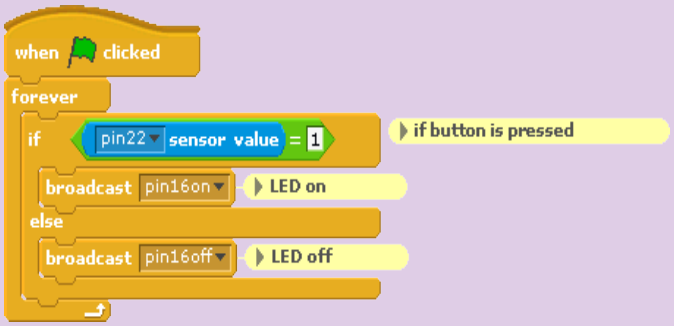
LIGHTS 3
USE BUTTON TO CONTROL LED

PROGRAM: SCRATCH

Using pseudocode/flow diagram what blocks in Scratch will we need to use?
We need to sense an input on pin 22
The LED is attached to pin 16.

You need to use Scratch GPIO
See http://cymplecy.wordpress.com/scratchgpio/ for details.

Code:



PROGRAM: PYTHON3

You need to download and install the RPi GPIO library – see http://pythonhosted.org/RPIO/ for instructions

You also need to run the python script as *sudo* in order to access the GPIO pins on the Raspberry Pi.

Code:

```
import RPi.GPIO as GPIO   #GPIO Library
import time

GPIO.setmode(GPIO.BOARD)  #use Board numbering

GPIO.setup(16, GPIO.OUT) #set pin 11 to output (LED)
GPIO.setup(22, GPIO.IN) #set pin 22 to input (button)

while True:
    if GPIO.input(22):     #if button is pressed
        GPIO.output(16, True) #switch on LED
    else:
        GPIO.output(16, False) #switch off LED
    time.sleep(0.1)

GPIO.cleanup()
```

EXTENSION:

Could you turn this into a reaction game?
For example, time how long it takes you to press the button from when the LED is lit.

Could you create a high score to beat?

PERFORM the process first and then create the associated pseudocode and/or flow diagram as your PLAN.
Then PROGRAM for Scratch and/or Python3.

Pseudocode:

```
Start
     Turn Light Off
     Wait random length of time
          Reset timer to 0
          Turn Light On
          If switch pressed
               Record time taken
               Stop
```