

# LIGHTS 2 TRAFFIC LIGHTS

LEVEL: BEGINNER

## COMPUTING PoS KS1:

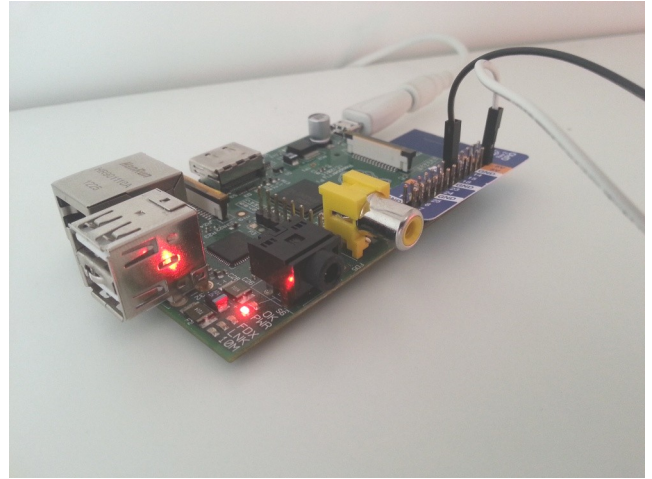
- Understand what algorithms are
- Create and debug simple programs
- Use logical reasoning to predict the behaviour of simple programs

## 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
- Repetition

## PERFORM

What is the sequence for traffic lights?  
(Red; Red/Amber; Green; Amber; repeat)  
Find actual lights or video to record sequence.

Equipment: Double sided A4 cards: Red/Amber/  
Green bulb on/off; 3 actors: Red/Amber/Green  
Lightbulbs

### Act Out:

- 1) Lightbulb actors stand next to each other – Red on – Amber & Green off
- 2) Amber actor turns light on.
- 3) Red and Amber off
- 4) Green Off – Amber on
- 5) Repeat 1) through 4)

## PLAN

Can you write the pseudocode for the traffic light sequence?

### Pseudocode:

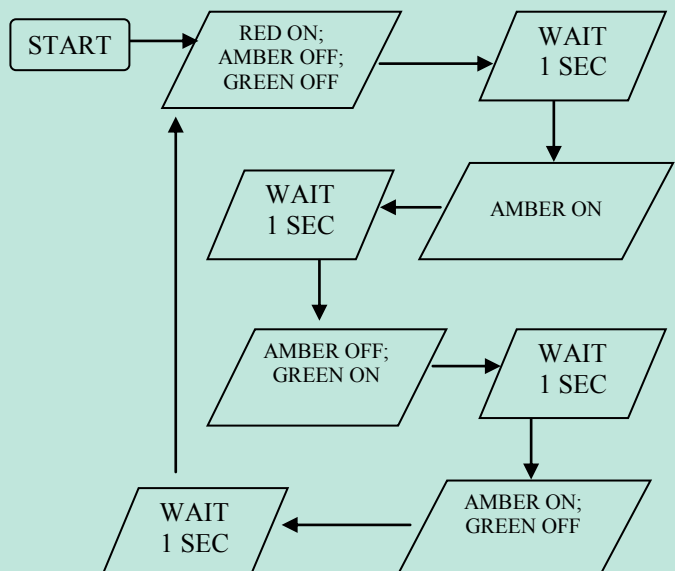
Start

Repeat

```

Turn Red On; Amber Off; Green Off
Wait 1 Second
Turn Amber On
Wait 1 Second
Turn Green On; Red off; Amber Off
Wait 1 Second
Turn Amber On; Green Off
Wait 1 second
    
```

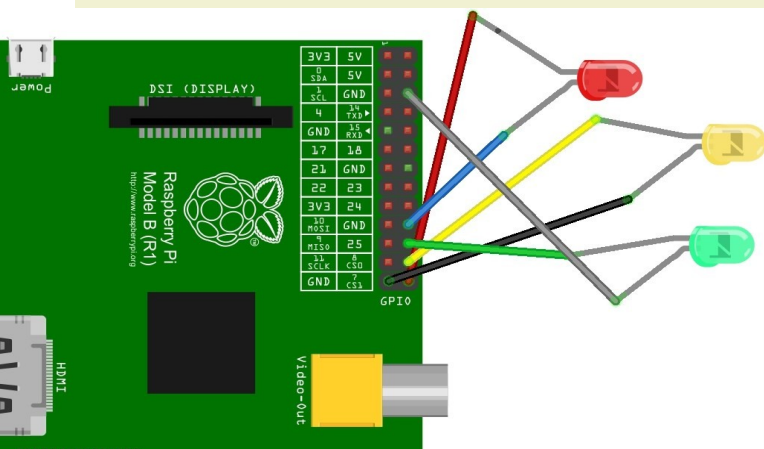
### Flow Diagram:



# LIGHTS 2 TRAFFIC LIGHTS

## HARDWARE SETUP

Diagram:



Equipment: Raspberry Pi; 3 x LEDs (Red, Amber, Green); 2 x Female to Female Jumper Leads; or use 1 x Pi-Stop

Method:

- 1) Attach +ve (long) leg of Red LED to Pin 26 of the Raspberry Pi using one of jumper leads (red in diagram); Amber LED to Pin 24 (yellow in diagram); Green LED to Pin 22 (green in diagram)
- 2) Attach -ve (short) leg of Red LED to Pin 20 (blue in diagram); Amber LED to Pin 25 (black in diagram); Green LED to Pin 6 (grey in diagram)

(If using Pi-Stop then use Pins 20, 22, 24 and 26 ensure Ground is Pin 20.)

## PROGRAM: SCRATCH

You need to use Scratch GPIO

See <http://cymplecy.wordpress.com/scratchgpio/> for details.

To buy Pi-Stops use 4tronix

[http://4tronix.co.uk/store/index.php?rt=product/product&product\\_id=390](http://4tronix.co.uk/store/index.php?rt=product/product&product_id=390)

To use Pi-Stops see detailed information here

<https://github.com/PiHw/Pi-Stop>

Code:



## PROGRAM: PYTHON3

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

Code:

```
import RPi.GPIO as GPIO #GPIO Library
import time             #for "sleep"

#set up Pins
GPIO.setmode(GPIO.BOARD) #use Board numbering
GPIO.setup(22, GPIO.OUT) #pin 22 to output
GPIO.setup(24, GPIO.OUT) #pin 24 to output
GPIO.setup(26, GPIO.OUT) #pin 26 to output
```

```
For i in range(0,10): #ten times
    GPIO.output(26, true) #red ON
    GPIO.output(24, false) #amber OFF
    GPIO.output(22, false) #green OFF
    time.sleep(1) #1 sec
    GPIO.output(24, true) #amber ON
    time.sleep(1) #1 sec
    GPIO.output(26, false) #red OFF
    GPIO.output(24, false) #amber OFF
    GPIO.output(22, true) #green ON
    time.sleep(1) #1 sec
    GPIO.output(24, true) #amber ON
    GPIO.output(22, false) #green OFF
    time.sleep(1) #1 sec
```

GPIO.cleanup()

## EXTENSION:

How would a second set lights interact with this one at a junction? Or opposite side of a bridge? What should be on opposite light if green is displayed on one?

PERFORM the two sets of lights. Create associated pseudocode and flow diagram as your PLAN. Then PROGRAM for Scratch and/or Python3.