Example Materials

All of this material is available, for every unit taught, on the DVD ROM of teaching materials.

Unit 1

Ultra Violet Spy Kit

Lesson 5

Ultra violet Spy Pack: Lesson 5

Lesson Objectives
To write a simple program (algorithm) to light the ultra violet L.E.D. for ten seconds. Then to connect and use the Secret Code Machine.

Vocabulary
ultra violet, positive, negative, L.E.D. (light emitting diode), circuit, program

Introduction
In this lesson, pupils will write/enter a simple program (algorithm) into Scratch, to turn on the Ultra violet L.E.D.; and after connecting the Secret Code machine (made in previous lesson) to the Pibrella, they will use it to write and decode secret messages. The PowerPoint presentation outlines the requirements of the simple program needed to light the ultra violet L.E.D.

Main Activity
There are five possible activities that can be undertaken here. This is so that work can be differentiated; extension activities used; or simply so that some pupils can use the Raspberry Pi and Pibrella whilst other pupils do other activities, then swap over (if resources are limited). It is not expected that every pupil should complete every activity - but all pupils should have managed to use their Secret Code Machine by the end of the lesson.

1. **Enter Program into Scratch (**)** Pupils should write a simple set of instructions (possibly using Worksheet 1G) to turn the ultra violet L.E.D. on for ten seconds, then turn it off. This can be entered into Scratch.

2. **Jigsaw Puzzle Programming Activity (*)** Less able pupils can open a puzzle from the Puzzles folder, called 'Secret Code Machine Puzzle'. This is a simple jigsaw type activity where pupils just need to connect four pieces of code to make the program

3. **Connect Secret Code Machine to Raspberry Pi to use it (**)** Once the program (algorithm) is written, pupils are to connect it to the Pibrella, and run their program (click on the green flag) to start using the Secret Code machine. If the cables are not long enough to reach the Pibrella, the kits contain spare cables. Simply connect another red cable to the red lead, and another black cable to the black lead.

4. **Write an Explanation(** to ***)** Pupils could be asked to write an explanation for how their program works, or to explain it verbally to the class (in plenary session). This explanation could be used, alongside the Secret Code Machine, as part of a classroom display.

5. **Write and Read Secret Messages (*)** Pupils can write a secret message using the ultra violet Security pen that is included in the kit. If the message is complex or long, pupils could write it on paper inside the Secret Code Machine, so that they can see what they are writing. Once their message is written, it will take about a minute for the ink to dry, and the message to become invisible to the naked eye. At this point, placing the paper into a Secret Code Machine will enable pupils to decode the message. Pupils could use Worksheet 1J to write and decode messages in Morse Code using the Secret Code Machine.

6. **Creative Writing (***)** This activity gives rise to lots of scope for creative writing - to fit in with Literacy work. For example, the Secret Code could be the answer to a mystery story, or poetry could be written about the activity and messages decoded.

For extension activities, making more complex algorithms, see next lesson.

Plenary
Pupils could be asked to share with the class their experiences, explaining how they got on, and how they resolved any difficulties that they encountered.

Resources
Worksheet 1G, 1J
Helpsheets
Raspberry Pi complete with Pibrella, ultra violet Spy Pack

Health and Safety
Ultra violet light is most commonly emitted by the sun, and although it is invisible to the human eye, most people are aware of its effects on the skin, called suntan and sunburn. Ultra violet is also responsible for the formation of vitamin D in organisms that make this vitamin (including humans). The UV spectrum thus has many effects, both beneficial and damaging, to human health. For safety considerations, although pupils are subjected to ultra violet light every day, pupils should not look directly at a source of ultra violet light, which is why the L.E.D. has been housed and sheltered inside a box for the purposes of this activity.

As the Raspberry Pi uses electricity, pupils should as usual be reminded about the dangers of electricity; not having liquids near the equipment, not touching mains devices etc. In practise, both the Raspberry Pi and the Pibrella run on just 5 volts of power once the electricity leaves the mains power source, so will not cause any danger to pupils working with them.

Recommended practise is to have the Raspberry Pi receiving the electricity, and for the Pibrella to receive its power from the Raspberry Pi, via a GPIO ribbon. For further protection, it is recommended for the Raspberry Pi to be secured in a case to protect it from damage, possibly even mounted to the back of a monitor via a vesa mount. Pupils are then only working with the Pibrella at 5 volts charge, with no mains power directly in view.

Unit 5

Intruder Alarm Kit

Lesson 6

Lesson 6

Lesson Objectives
To use variables to control and determine whether a system (such as a Quiz Game or a Burglar Alarm) is 'On', 'Off' or in some other state

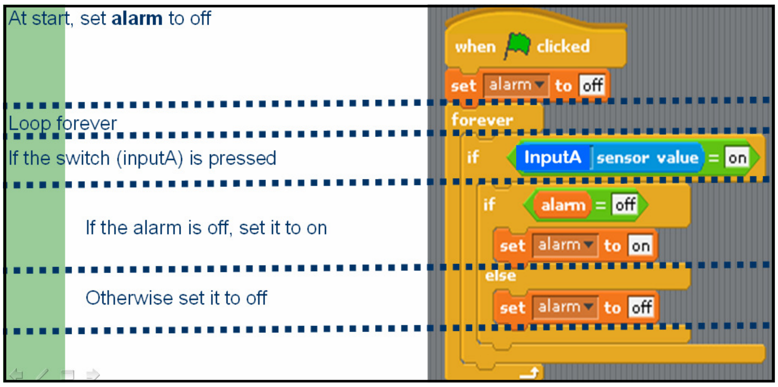
Vocabulary
Variable, state, IF, IF/ELSE, comment, variable-controlled system, input

Introduction
Start by talking about systems that are just ON or OFF, and how a variable can be used to remember this, so that when a user presses a switch, if the system was OFF it goes to ON, and vice versa. Explain how pupils can add comments to annotate their work in Scratch, simply by right-clicking in the programming area and selecting 'Add Comment'. This can be useful so they can explain what different parts of their program do, and explain settings such as what different values for a variable actually mean.

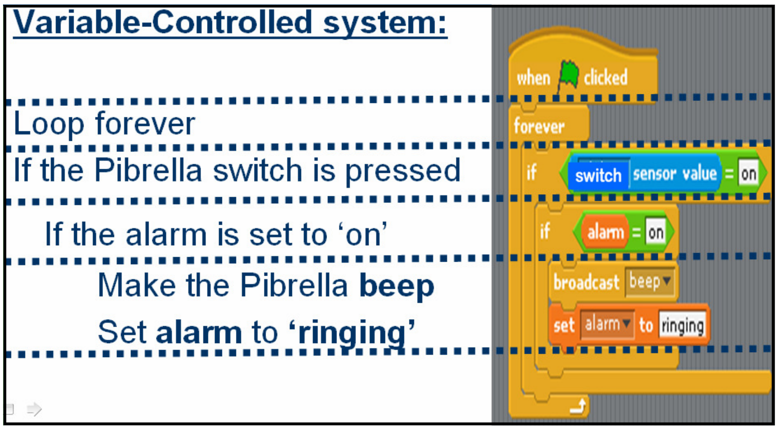
The PowerPoint presentation covers all of these elements, and gets pupils to look at some simple code that changes between ON and OFF to work out if the code is correct. Finally, a correct solution for switching a system from ON to OFF, using an IF/ELSE command, shows pupils a way to switch between ON and OFF when a switch is pressed, correctly.

A more complex system has more than two possible states. So in the case of a burglar alarm, as well as the ON and OFF states, if the alarm is ON and a burglar then sets off the alarm, maybe a third state, RINGING, or INTRUDER DETECTED, needs to be set. This is most easily done by having a system to control between ON and OFF (top, opposite), and another program that checks, if the sensor detects a burglar, whether the alarm is also set, and if so, it changes to a third state, RINGING, and makes a buzzer sound (like a burglar alarm rings).

Simple ON/OFF system



More Complex System with third state, 'RINGING'



Main Activity
There are six possible activities that can be undertaken here. This is so that work can be differentiated; extension activities used; or simply so that some pupils can use the Raspberry Pi and Pibrella whilst other pupils do other activities, then swap over (if resources are limited). It is not expected that every pupil should complete every activity - but all pupils should have managed to at least create a simple ON/OFF program in Scratch by the end of the lesson.

1. **Devising a simple ON/OFF system using a variable (**).** Pupils are to create a system, controlled by a switch, that changes a variable from ON to OFF, or OFF to ON, each time it is pressed.

2. **Devising a simple ON/OFF system using a variable with support (*).** Pupils can use **Worksheet 5L** to help them to develop a simple ON/OFF system.

3. **Recording a simple ON/OFF System (**)** Pupils could be asked to use **Worksheet 5M** to record and explain the logic behind the program they've developed to make an ON/OFF system.

4. **Devising a Variable-Controlled Complex System (***)** Pupils are to create a system, controlled by a switch, that changes a variable from ON to OFF, or OFF to ON, each time it is pressed *(as in (1) above. This activity could be used as an extension for pupils who achieve activity 1, and are ready for a more difficult task!).* Then, they are going to write a second piece of code (algorithm), that checks if the burglar alarm sensor (in this case just the switch on the Pibrella) is triggered, and if so, if the alarm is also set to ON, then it changes the state of the alarm to **RINGING**, and makes the alarm ring (or makes the buzzer on the Pibrella ring by using the command **Broadcast 'Beep').**

5. **Recording a Variable-Controlled Complex System (**)** Pupils could be asked to use **Worksheet 5N** to record and explain the logic behind the program they've developed to make an ON/OFF system.

6. **Investigating Window Alarm Sensors (*)** Pupils should be given the window alarm sensor (2 pieces - one with wire attached), and connect it to the Pibrella as explained in the PowerPoint presentation. Worksheet 5O explains to pupils how to do this, and also the code to write in Scratch to use while investigating the Window Alarm Sensor. The Window Alarm Sensor is basically a Reed Switch inside a plastic case - which is controlled by a magnet. When a magnet is near to the sensor, it pulls two small wires in the sensor together, and makes the switch 'ON'. But when the magnet is removed, the two small wires come apart; and the switch is now 'OFF'. This idea is used as part of burglar alarms, to detect when a window or door is opened, and pupils have the task in this activity of working out when and how the window alarm could be used.

Plenary
Any pupils who have investigated the use of the Window Alarm Sensor could explain how the sensor works, and how this could be used in a burglar alarm, to the class.

Pupils who've only managed the 'ON/OFF' system could explain how the basic ON/OFF system works, showing their code and comments to the class.

Pupils who've managed to extend this, to also be able to detect an input such as a burglar triggering a sensor, could explain their code to the class. The PowerPoint presentation does include model answers (as shown on previous page) that can be used to explain each system to pupils, if required.

Resources
Worksheets 5L, M, N, O
Helpsheets (esp. 4B)
Raspberry Pi complete with Pibrella
Switch, red jumper cable, black jumper cable

Health and Safety
As the Raspberry Pi uses electricity, pupils should as usual be reminded about the dangers of electricity; not having liquids near the equipment, not touching mains devices etc. In practise, both the Raspberry Pi and the Pibrella run on just 5 volts of power once the electricity leaves the mains power source, so will not cause any danger to pupils working with them.

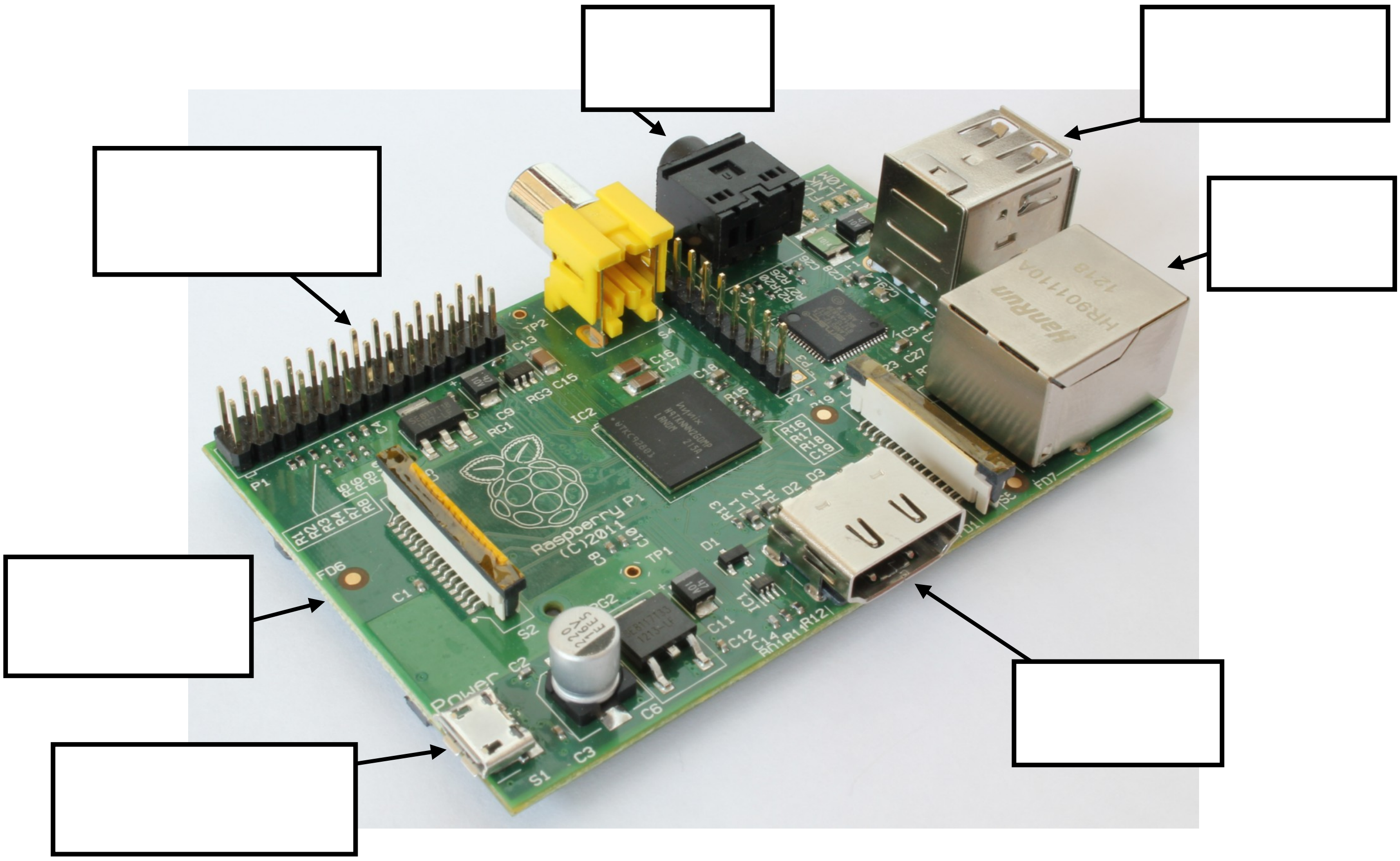
Recommended practise is to have the Raspberry Pi receiving the electricity, and for the Pibrella to receive its power from the Raspberry Pi, via a GPIO ribbon. For further protection, it is recommended for the Raspberry Pi to be secured in a case to protect it from damage, possibly even mounted to the back of a monitor via a vesa mount. Pupils are then only working with the Pibrella at 5 volts charge, with no mains power directly in view.

Name: ______________________________ Date: ________________________ 2A

Parts of a Pi

Label the following parts of the Pibrella.

USB ports	Audio Out Socket	SD Card Slot
Ethernet Port	HDMI out Socket	GPIO Header
Micro USB Power Socket		

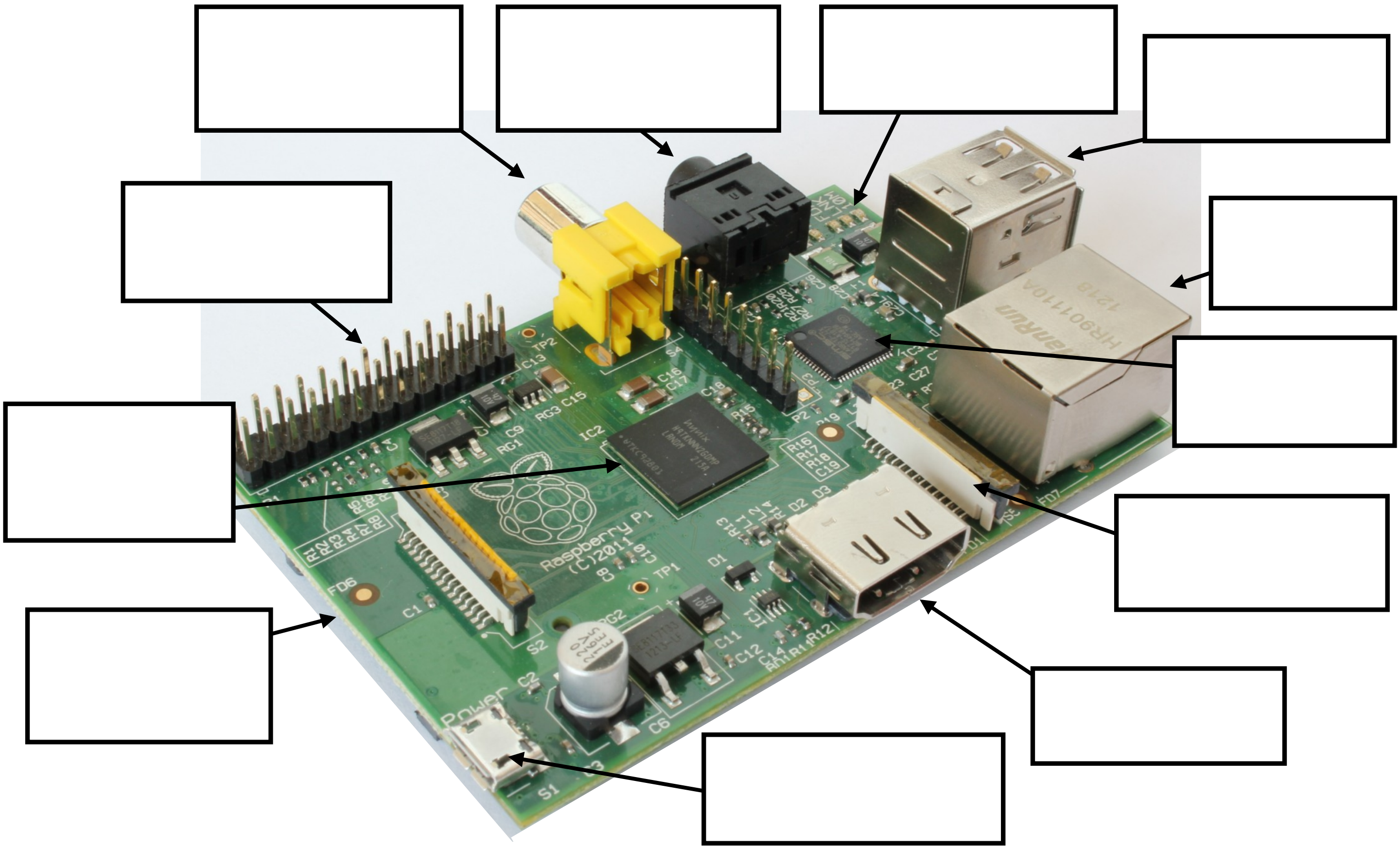


Name: ______________________________ Date: ________________________ 2B

Parts of a Pi

Label the following parts of the Pi

USB ports	Audio Out Socket	SD Card Slot
Graphics & Central Controller	HDMI out Socket	Ethernet Port
Micro USB Power Socket	Video Out	LED Indicators
Camera Connector	Networking Controller	GPIO Header



Name: ______________________________ Date: ________________________ 2C

Parts of a Pibrella

Label the following parts of the Pibrella.

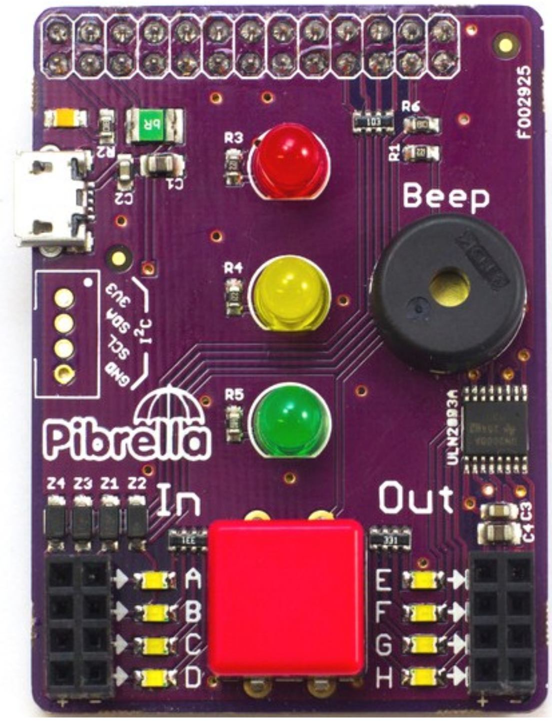
Output Connections	Amber LED
Input Connections	Green LED
Buzzer	Push Button
Red LED	Power Socket

Name: ______________________________ Date: ________________________ 2D

Parts of a Pibrella

Label the following parts of the Pibrella.

Output Connections	Input Connections	Buzzer	Red LED
Amber LED	Green LED	Push Button	Power Socket



Use the space below to explain the function of each part of the Pibrella.

Name: ______________________________ Date: ________________________ **2E**

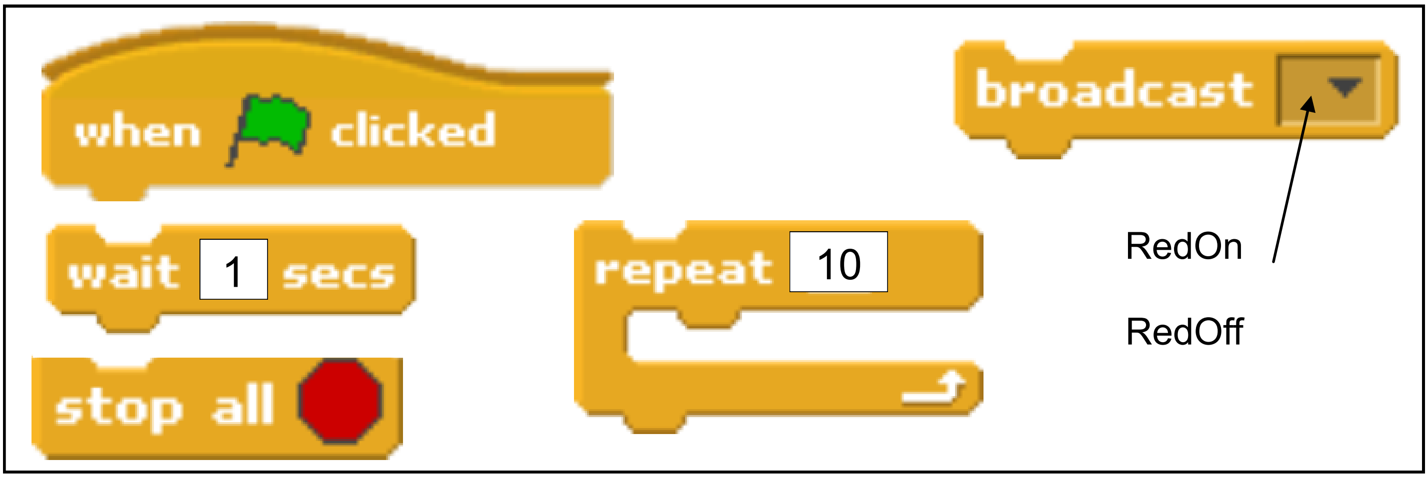
Lighting an LED - a Program or Algorithm

Using the commands below to help you, draw out the sequence of instructions you used to light the Red LED for 10 seconds.

Name: ______________________________ Date: ________________________ 2F

Lighting an LED - a Program or Algorithm

Using the commands below to help you, draw out the sequence of instructions you used to light the Red LED for 1 second, ten times.



Name: ______________________________ Date: ________________________ 2F

Lighting an LED - a Program or Algorithm (Suggested Answer)

Using the commands below to help you, draw out the sequence of instructions you used to light the LED.

The program starts when the green flag is clicked

The following instructions are repeated ten times

Turn on the Red LED

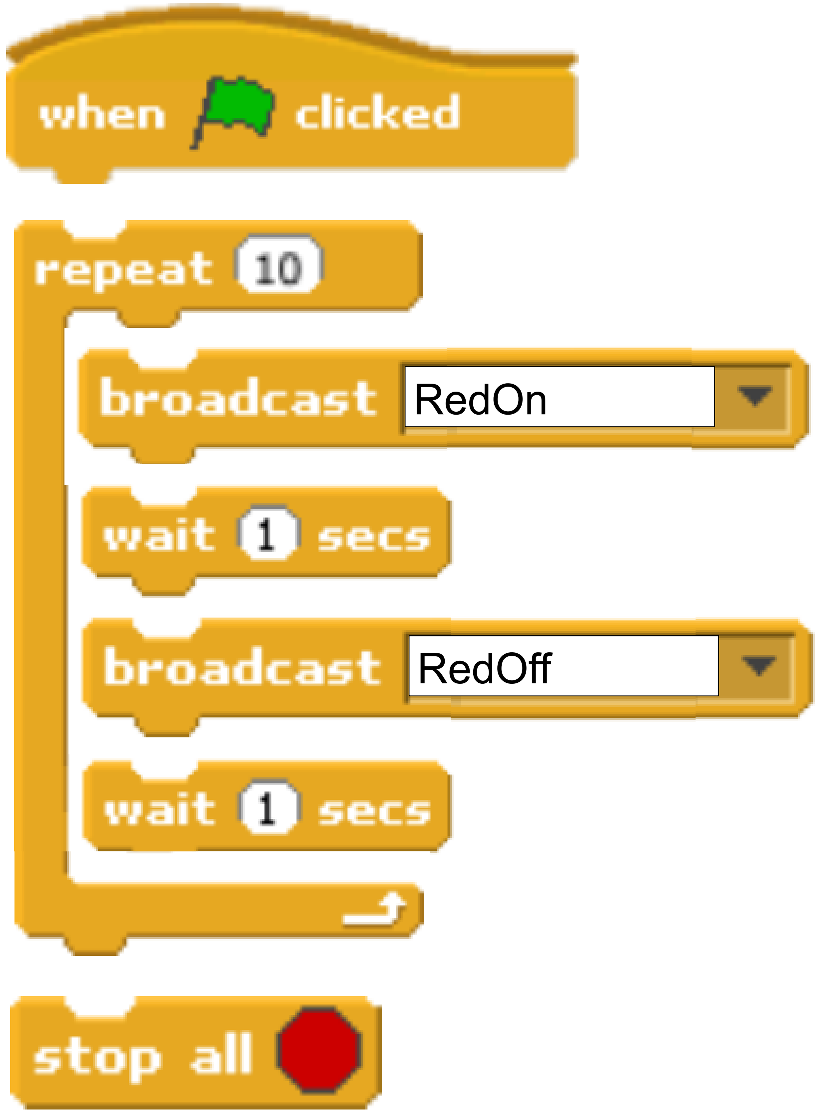
Wait ten seconds

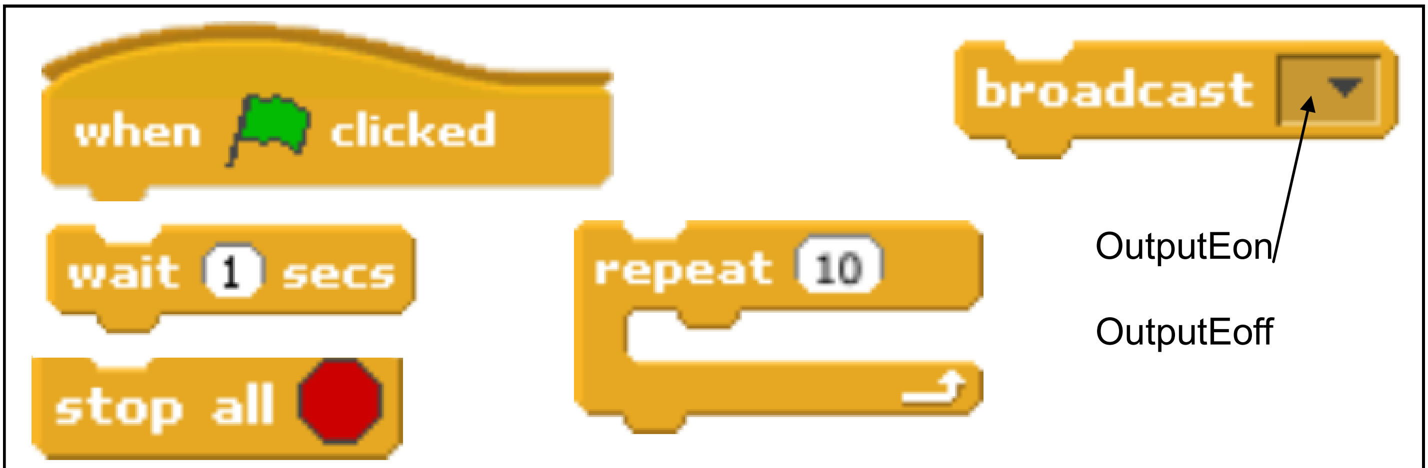
Turn off the Red LED

Wait ten seconds

These instructions above are repeated ten times

This command stops the program





Name: ______________________________ Date: ________________________

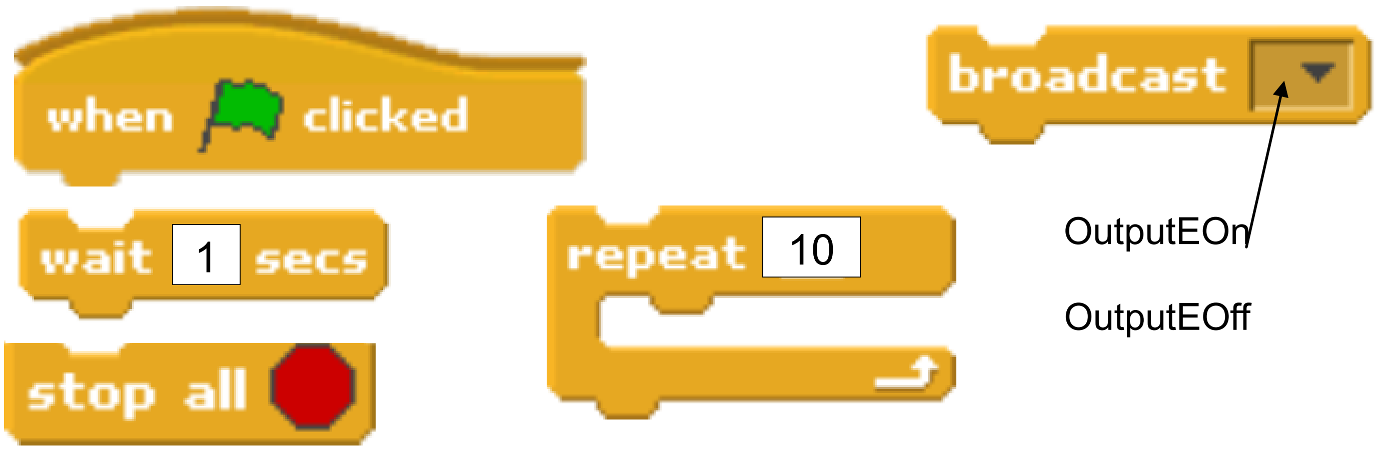
Lighting an LED - a Program or Algorithm

Using the commands below to help you, draw out the sequence of instructions you used to light the LED that you connected to OutputE, for 10 seconds.

Name: ______________________________ Date: ________________________ 2H

Lighting an LED - a Program or Algorithm

Using the commands below to help you, draw out the sequence of instructions you used to light the LED that you connected to OutputE for 1 second, ten times.



Name: ______________________________ Date: _______________________ 2H

Lighting an LED - a Program or Algorithm (Suggested Answer)

Using the commands below to help you, draw out the sequence of instructions you used to light the LED.

The program starts when the green flag is clicked

The following instructions are repeated ten times

Turn on the Ultra Violet LED

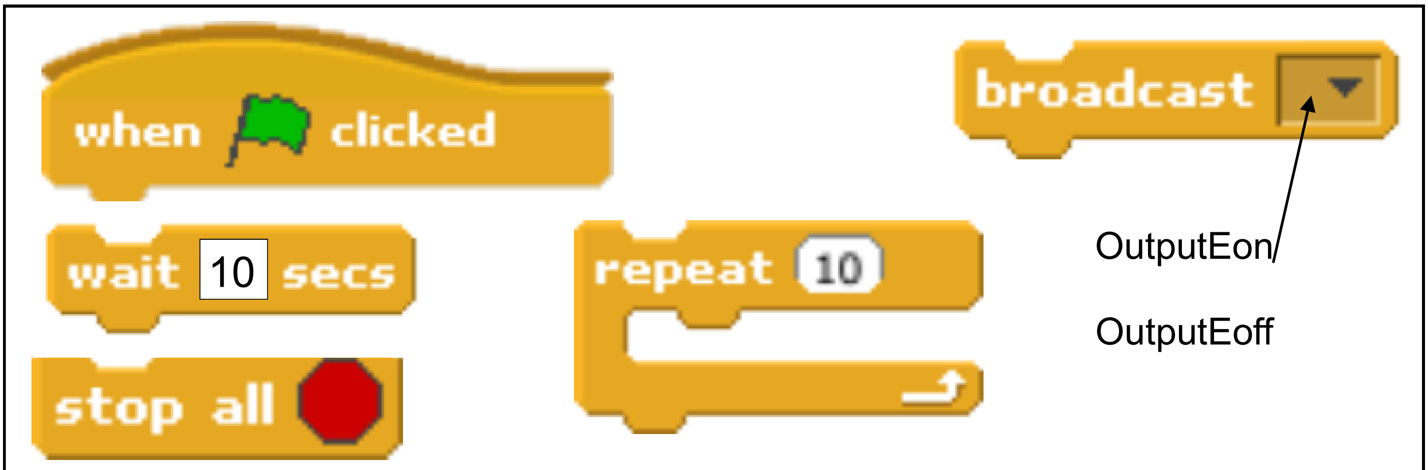
Wait ten seconds

Turn off the ultra violet LED

Wait ten seconds

These instructions above are repeated ten times

This command stops the program



Name: ______________________________ Date: _______________________

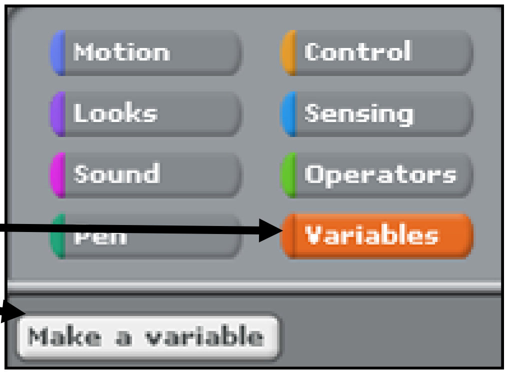
Varying the brightness of an LED

The ‘Power’ command lets you change the power of an output on the Pibrella from 0 (off) to 100 (maximum). The three outputs on the Pibrella can be controlled by variables, called:

PowerRed **PowerAmber** **PowerGreen**

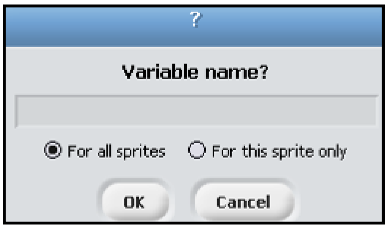
1. Load ScratchGPIO

2. Open ‘Illusion Spinner Template’



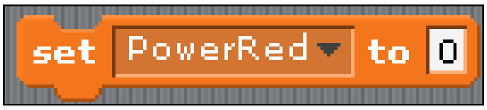
3. To make a new variable (PowerRed), you need to go to the Variables tab and then click on ‘Make a variable’

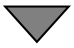
4. Enter a name for the variable (PowerRed) then click on ‘OK’.



5. Start a new Scratch program with 

6. To set the Red LED to be off at the start, you need to set variable PowerRed to be equal to zero.
To do this:



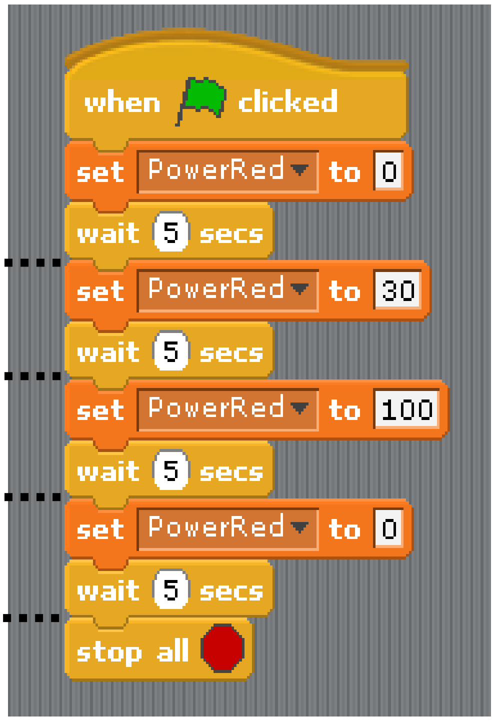
- Choose the ‘Set’ variable command as shown opposite
- Choose the variable you want to control, PowerRed, from the pull down menu, by clicking on 
- Type in the value you want (0)

7. Carry on entering this simple program. It sets the red LED to off (Set PowerRed to 0), then waits 5 seconds.

Then it sets the red LED to be dim (Set PowerRed to 30) then waits 5 seconds

Then it sets the red LED to be bright (Set PowerRed to 100) then waits 5 seconds

Finally, it sets the red LED to be off (Set PowerRed to 0) then waits 5 seconds.



Click on the Green Flag to see your program running!

Name: ______________________________ Date: ________________________ 2J

Electrical Devices with a variable Power Output

Make a list of electrical devices that are used everyday, which have variable power controls (that means that you can turn them up or down!)

Name: ______________________________ Date: ________________________ 2K

Investigating Illusions

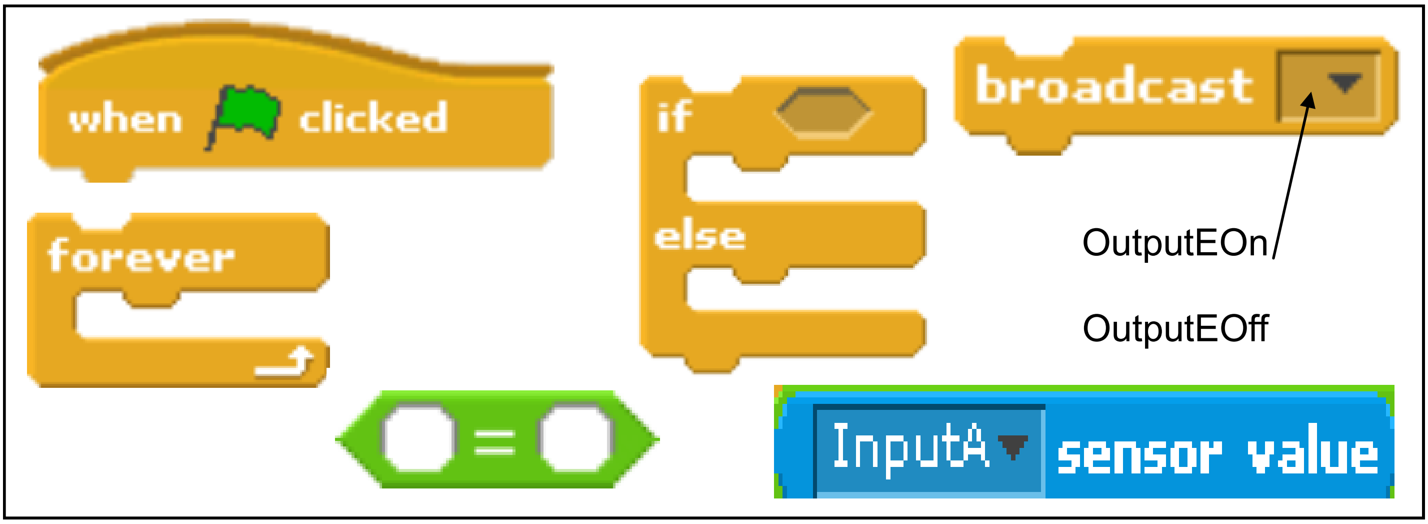
When the Illusion Disks were controlled by the Raspberry Pi and spun around, we recorded what happened with each illusion disk. With the yellow disk, what happened depended upon which way the disk was spinning, clockwise or anti clockwise.

Spinner	What happened when it turned?
(Turning Clockwise)	
(Turning Anti-Clockwise)	

Name: ______________________________ Date: ________________________ 2L

Lighting an LED when a switch is pressed - a Program or Algorithm

Using the commands below to help you, draw out the sequence of instructions you used to light the LED that you connected to OutputE, whenever the switch (InputA) is pressed.



Name: ______________________________ Date: ______________________ **2L**

Lighting an LED when a switch is pressed (Suggested Answer)

Using the commands below to help you, draw out the sequence of instructions you used to light the LED when the switch (InputA) is pressed..

The program starts when the green flag is clicked

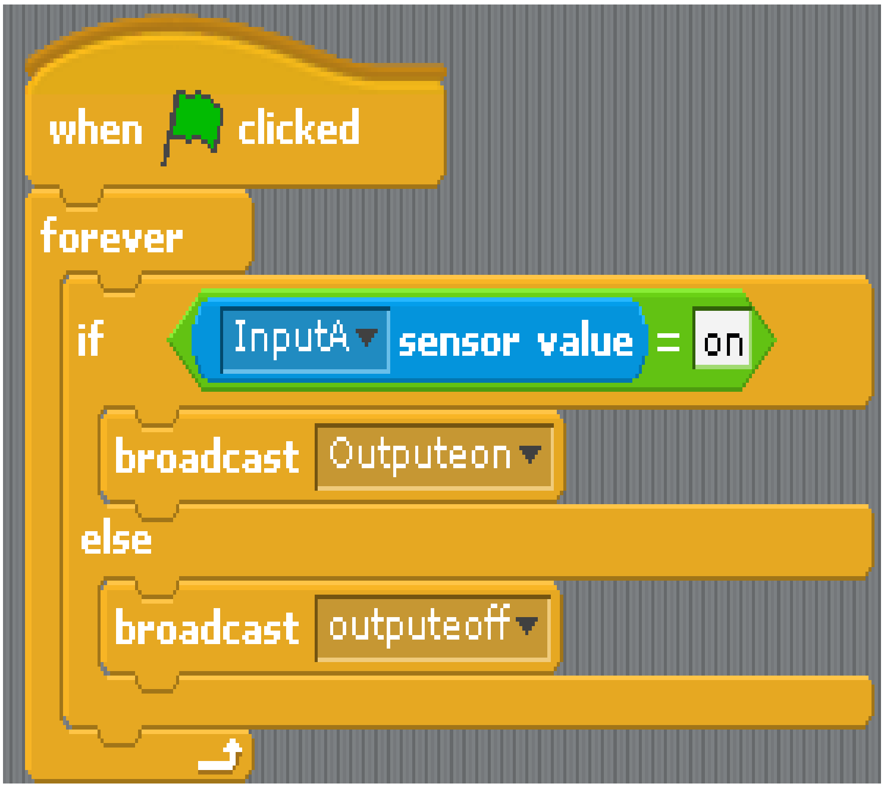
It loops forever

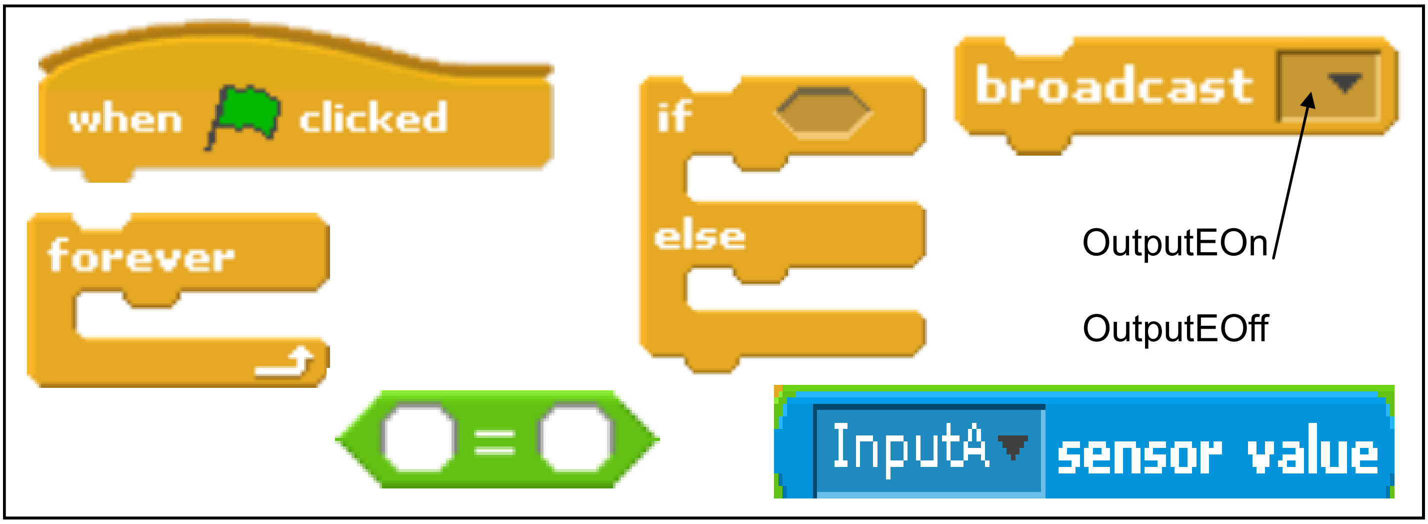
If Switch (InputA) is pressed

Turn on the LED (OutputE)

Otherwise (if the switch is not pressed)

Turn off the LED (outputE)





Name: ______________________________ Date: ________________________ 2M

Everyday Objects where an Input causes an Output to happen

Make a list of devices that are used everyday, where an input (such as pressing a doorbell) causes an output (such as a bell ringing) to happen.

Name: ________________________________ Date: _________________________ 2N

Using the variable 'SCORE' to count how many times a switch is pressed

Record below the program you wrote in Scratch to count the number of times that the switch (InputA) was pressed.

Instruction in Scratch	Explanation of what it does

Investigating the sequence of Traffic Lights

Lesson 4

Wishtrac

Lesson Objective

- To be able to program the correct sequence for a set of traffic lights, including using a loop

Wishtrac

Introduction

Options:

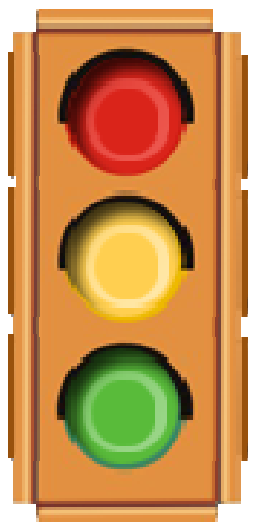
- In what order do traffic lights go?
- Repeating loops
- Controlling L.E.Ds on the Pibrella

Go to Main Activity

Wishtrac

In what order do traffic lights go?

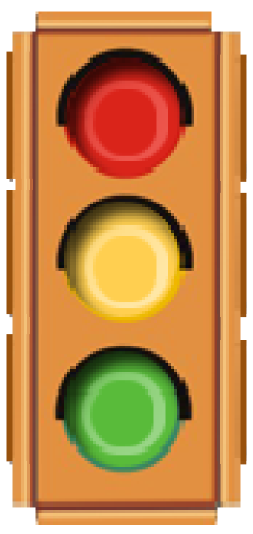
- Spend a few minutes trying to write down the correct order that traffic lights go in.
- Start with Red, and carry on until you get back to Red.



Wishtrac

In what order do traffic lights go?

- Where you right?
- Watch the video on the next page to check



Wishtrac

In what order do traffic lights go?

Wishtrac

In what order do traffic lights go?

The correct sequence is:

RED
RED and AMBER
GREEN
AMBER
RED

Wishtrac

Introduction

Options:

- In what order do traffic lights go?
- Repeating loops
- Controlling L.E.Ds on the Pibrella

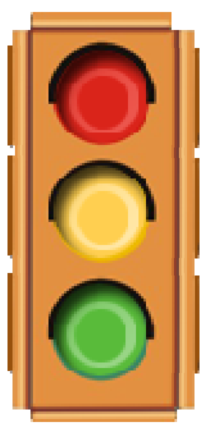
Go to Main Activity

Wishtrac

Lighting the Pibrella L.E.Ds using Scratch

The following commands control each L.E.D:

- Broadcast (RedOn)
- Broadcast (RedOff)
- Broadcast (AmberOn)
- Broadcast (AmberOff)
- Broadcast (GreenOn)
- Broadcast (GreenOff)
- Broadcast (AllOff)



Wishtrac

Introduction

Options:

- In what order do traffic lights go?
- Repeating loops
- Controlling L.E.Ds on the Pibrella

Go to Main Activity

Wishtrac

Repeating Loops

- We've got a sequence for traffic lights of:

RED
RED and AMBER
GREEN
AMBER
RED

But how long does this go on for?

Wishtrac

Repeating Loops

RED
RED and AMBER
GREEN
AMBER
RED
RED and AMBER
GREEN
AMBER
RED
RED and AMBER
GREEN
AMBER
RED

Wishtrac

Repeating Loops

How many times should we repeat?

RED
RED and AMBER
GREEN
AMBER

Wishtrac

Repeating Loops

Scratch lets us use a **REPEAT** command:

- Whatever we put inside here is repeated the number of times we want – simply change the number 10 to the number we want

Wishtrac

Repeating Loops

- But with traffic lights, what would happen if they got to the end of the Repeat instruction and had finished the program?

Wishtrac

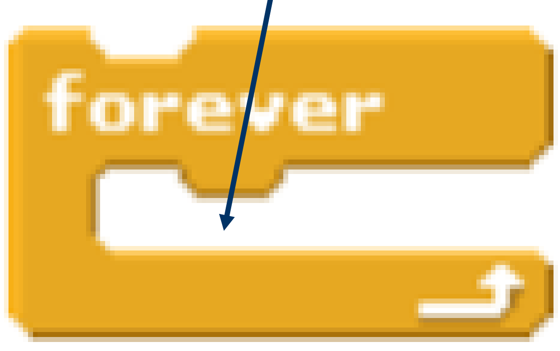
Repeating Loops

- So Scratch has another way of repeating things:

Wishtrac

Repeating Loops

- Whatever we put inside here is repeated 'forever' until the program is stopped.



Wishtrac

Introduction

Options:

- In what order do traffic lights go?
- Repeating loops
- Controlling L.E.Ds on the Pibrella

Go to Main Activity

Wishtrac

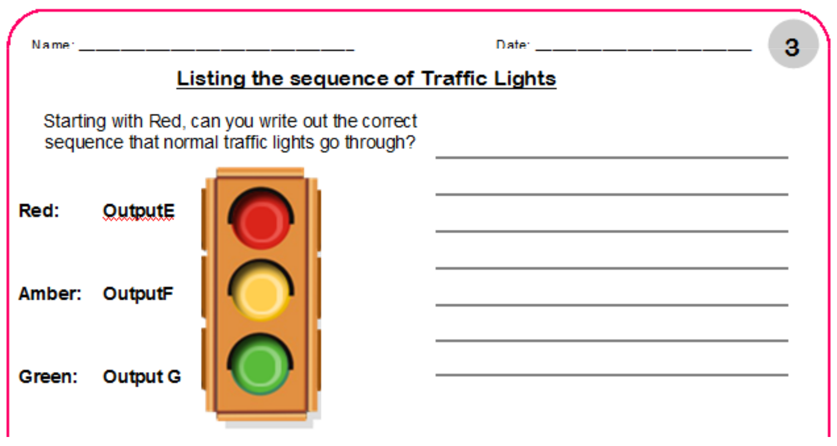
Main Activities

- Working out the correct sequence of instructions
- Testing your sequence using Scratch and the Pibrella
- Help with writing a sequence of instructions (a program)

Go to Plenary

Working out the correct sequence of instructions

- Use the worksheet to record your program



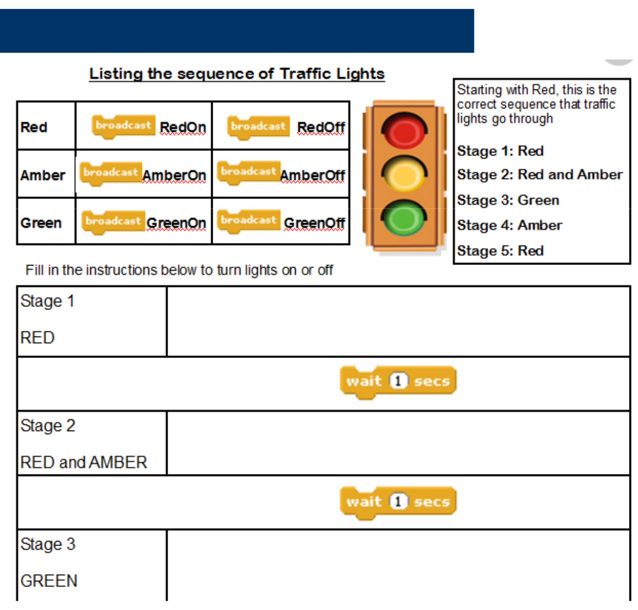
Main Activities

- Working out the correct sequence of instructions
- Testing your sequence using Scratch and the Pibrella
- Help with writing a sequence of instructions (a program)

Go to Plenary

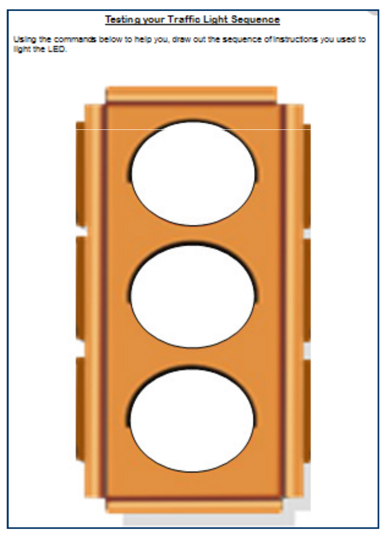
Writing down the correct sequence

- Work out what instructions are needed for each stage.
- Don't forget to turn lights off when you don't need them to be lit!



Writing down the correct sequence

- You can use the three coloured counters (Red, Amber and Green) to check that your program works
- Only put them on the sheet when they are turned on
- Only take them away when there is an instruction to turn them off



Main Activities

- Working out the correct sequence of instructions
- Testing your sequence using Scratch and the Pibrella
- Help with writing a sequence of instructions (a program)

Go to Plenary

Plenary

- Did you manage to get the sequence working correctly?
- What did you find most difficult?
- Next lesson – building a model set of traffic lights

Wishtrac