

Programmable Control

Introduction

By the end of this unit you should be able to:

- Give examples of where microcontrollers are used
- Recognise the symbols for different processes in a flowchart
- Construct simple flowcharts
- Write control programs using PBasic to:
 - Control Light Emitting Diodes and buzzers
 - Control an input switch
 - Use a For...Next loop
- Know the advantages of a microcontroller
- Understand the basic sub-systems in a microcontroller
- Control a motor
- Understand how Pulse Width Modulation works

What is a microcontroller?

A microcontroller is often described as a 'computer-on-a-chip'. It can be used as an 'electronic brain' to control a product, toy or machine.

The microcontroller is an integrated circuit ("chip") that contains memory (to store the program), a processor (to process and carry out the program) and input/ output pins (to connect switches, sensors and output devices like motors).

Example use of a microcontroller.

Almost all modern buildings are fitted with some type of alarm. For instance a fire detection system may have a number of smoke sensors to detect the smoke from a fire.

However many alarm systems are also safety systems - for instance an alarm system on an oil rig may monitor the temperature and pressure of the crude oil as it is being extracted and automatically shut the system down if a fault is detected. This ensures the safety of both the workers and the environment around the oil rig.

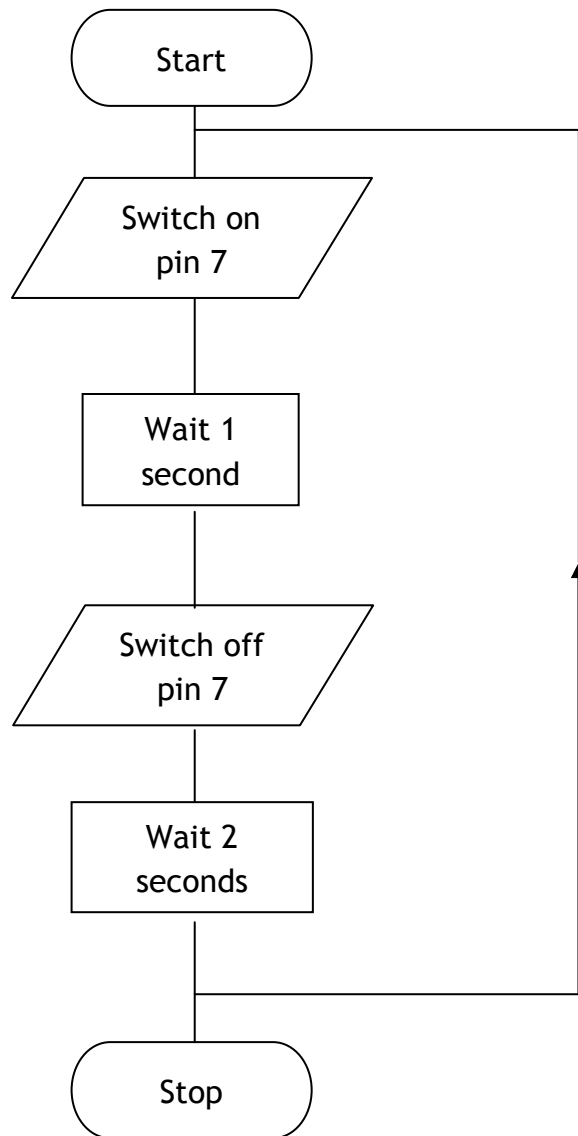
All systems are made up of input and output devices. Often these devices are connected to a microcontroller that interprets the information from the sensors and switches the outputs on and off at the correct time.

In the case of a fire alarm system the inputs may be smoke sensors and the keypad on the front of the control panel. The output devices are the display on the control panel as well as the external siren and strobe light. The microcontroller is the 'brain' of the system.

Microcontrollers are powerful electronic components that have a memory and can be programmed to switch things on and off in a special sequence. The microcontroller in the fire alarm, for instance, has been programmed to switch the siren on and off when the smoke sensor has detected fire.

Flowcharts

Software engineers like to draw a flowchart for each program that they develop. This makes the overall shape of the program easier to understand.



This example of a flowchart uses three different Symbols and a Feedback Loop.

Switching on the outputs

A program can contain instruction 'commands' to switch outputs on or off, or to wait for a set time. Type in the following program and see what it does. Note the use of comments (usually typed in *italics* and preceded by an apostrophe). The comments are only there to tell humans what is going on, they are ignored by the computer.

Symbols

Sometimes it can be hard to remember which pins are connected to which devices. The `symbol` command can be used at the start of a program to rename the inputs and outputs.

```
symbol red = 7           'rename output 7 'red'  
  
main:  
  high red              'set pin 7 on  
  pause 1000            'keep pin 7 on for 1 second  
  low red               'set pin 7 off  
  pause 2000            'keep pin 7 off for 2 seconds  
  goto main            'jump back to main
```

Task 1

Using the appropriate symbols, draw a flowchart to represent the sequence of a set of traffic lights.
Remember to switch off lights before switching other ones on.



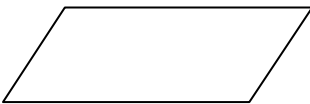
Start / Stop Symbol

This is a rectangle with rounded ends.
Each flowchart must contain only one 'Start' and end with the 'Stop' symbol.



Process Symbol

This is a rectangle. The text inside the rectangle should explain what is happening in the program.



Input / Output Symbol

Anything that is input or output from the microcontroller is shown as text in the parallelogram.

Feedback Loop

Notice that the loop back to repeat the program forever does not have a special symbol. It is shown as a line looping back to the start of the sequence. Often arrows are added to show the direction that it is going in.

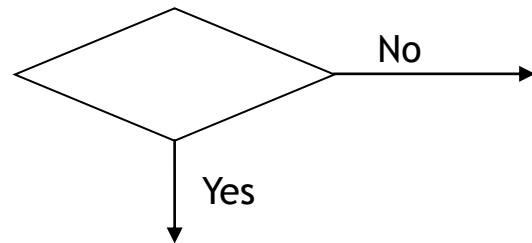
Task 2

With reference to your flow chart of the traffic lights write a PBASIC program that will use pins 7, 6 and 5 to represent the lights as red, yellow and green respectively.

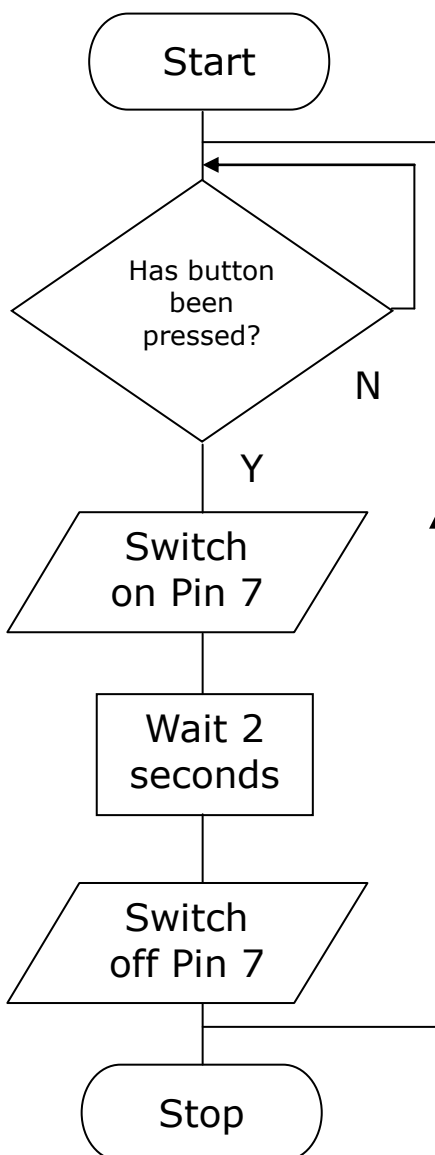
Making Decisions (Asking Questions)

When making a decision or asking a question, there are often just two possible answers. The answer can be YES or NO, in electronic terms this would be HIGH or LOW.

The **Decision Symbol** for a question is a diamond shaped box with the 'yes' and 'no' arrow showing the direction.



Example flowchart & PBasic program with a question



```
main:
  if pin0 = 1 then light
  goto main
light:
  high 7
  pause 2000
  low 7
  goto main
```

Task 3

Using the appropriate symbols, draw a flowchart to represent the sequence of a set of traffic lights at a pedestrian crossing. Remember to switch off lights before switching other ones on.

Task 4

With reference to your flow chart of the pedestrian crossing write a PBASIC program that will use pins 7, 6 and 5 to represent the lights as red, yellow and green respectively. Use pin 0 for your push button.

Read the following passage and answer the questions below.

Light Emitting Diode

A light emitting diode (LED) is an electronic component that gives out light when current passes through it. An LED is a special type of diode. A diode is a component that only allows current to flow in one direction. Therefore when using a diode, it must always be connected the correct way around.

LEDs are mainly used as indicator lights. Red and green LEDs are commonly used on electronic appliances like televisions to show if they are switched on or in 'standby' mode. LEDs are available in many different colours, including red, yellow, green and blue.

LEDs only require a small amount of current to work, which makes them much more efficient than bulbs

(this means, the LEDs will light for a much longer time than a bulb would). If too much current is passed through an LED it will be damaged, and so LEDs are normally used together with a 'series' resistor that protects the LED from too much current.

1. What do the letters LED stand for?
2. What does a diode do?
3. What advantage is there to use LED's rather than ordinary bulbs?
4. What would happen if too much current was passed through a LED and how would you prevent this from happening?
5. Name three devices in your home that use LED's?

Task 5

There is a set amount of time that you are given to cross the road. Towards the end of the time limit the green man starts to flash in order to warn you not to start crossing the road and to get a move on. Write program that will make the green LED flash five times with a gap of half a second on and then half a second off.

That's ok if you just want to switch it on and off a few times but what if you wanted to switch it on and off 100 times?

Just as well there is an easier way to do it using a **for.....next** loop.

For...Next Loop

This is a way to make something repeat a certain number of times without having to type the same lines of programming over and over again. The counter is where the microcontroller stores the number of times the program has looped. The program below shows a green LED flashing on and off 10 times.

Variables

A variable is something that can be varied or changed. With PBasic we can use variables called b0, b1, and b2 etc. Rather than call them b0 etc. it would be easier to understand the program if the variable had a name related to what it was doing. We can do this by using **symbols** as we did for outputs earlier.

Test the following program.

<code>symbol counter = b0</code>	<i>'rename variable b0 'counter'</i>
<code>symbol green = 5</code>	<i>'rename output 5 'green'</i>
flash:	
<code>for counter = 1 to 10</code>	<i>'start a for... next loop</i>
<code>high 5</code>	<i>'switch green on</i>
<code>pause 1000</code>	<i>'wait 1 second</i>
<code>low 5</code>	<i>'switch green off</i>
<code>pause 1000</code>	<i>'wait 1 second</i>
<code>next counter</code>	<i>'add 1 to counter</i>
<code>end</code>	<i>'ends program</i>

Task 6

Write a program that combines the traffic light system with the pedestrian crossing where the lights will go to stop, the green man will come on for a certain amount of time and then start to flash to warn people to get a move on crossing the road.

You have already used the microcontroller with the traffic lights and discovered that this “computer on a chip” can do a variety of tasks.

Advantages of a microcontroller

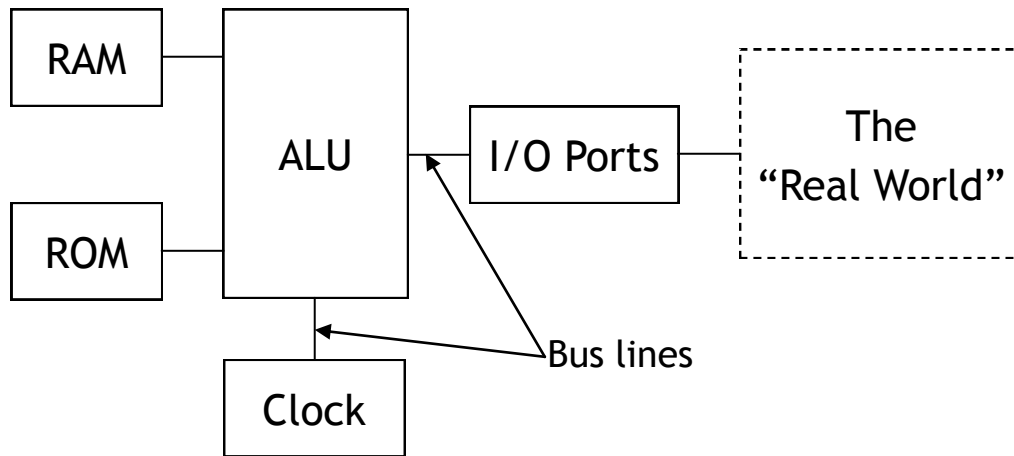
The microcontroller can be programmed to perform different tasks. This means it can be used in the same product in many different ways or it can be used in many different products, all of this simply by changing the programmed routine.

This therefore is its number one advantage, it is very flexible and adaptable - it can easily be reprogrammed to work in a different way in the same machine or even to work in a different machine. Very often, it is only the software that has to be changed and the hardware can remain the same. This means that product development is much quicker and therefore cheaper.

The main advantages of using a microcontroller rather than a hard wired electronic circuit are:

- Easier to reprogram
- Requires fewer components
- Shorter assembly time
- Smaller due to less components
- Cheaper due to less components

Inside a Microcontroller



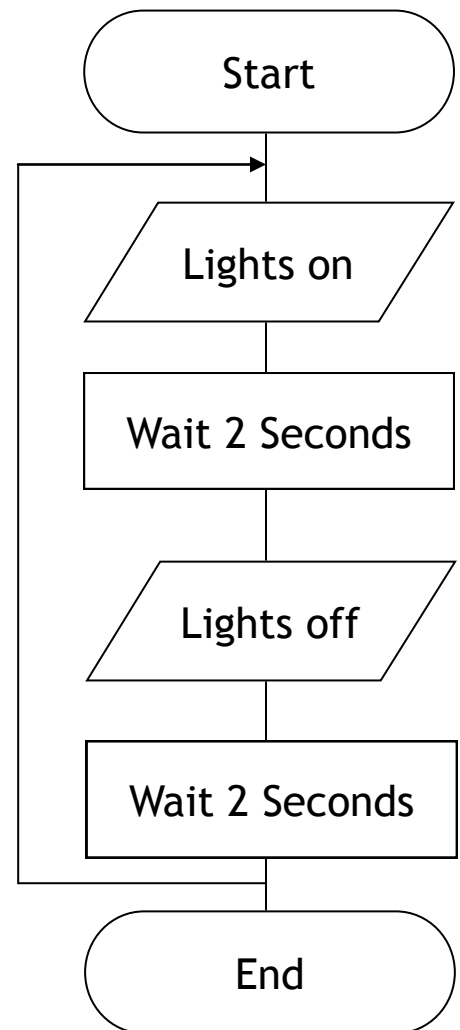
- **ALU** - this is the processing unit, the “brain” or the “control centre” of the microcontroller.
- **Clock** - the clock controls the speed that the ALU operates at and it synchronises or times the movement of data between the ALU, ROM and RAM so that the whole system works correctly.
- **Memory**
 - **ROM** - stands for Read Only Memory and it contains the operating instructions (that is, the program) for the microcontroller. ROM can be thought of as permanent (or non-volatile) memory.
 - **RAM** - stands for Random Access Memory which is temporary (or volatile) memory which means when the power is switched off, it loses all of its data.
- **Buses** - A Bus in a microcontroller is a wire or connection between its different parts.
- **I/O Ports** - the “I/O Ports” of a microcontroller are the Input and Output Ports. A port is a connection so it is through these that the microcontroller can link to the real world.

Flowcharts and Programming

Moving on from the traffic lights, you are going to use a clown which has different input and output components. You will work through some basic tasks which will allow you to complete the challenge at the end of this booklet.

Task 1

Using the given flowchart, write a program that will make the clown's eyes flash. Both LEDs are connected to pin 7.



Task 2

Alter the program to include a for...next loop to make the eyes flash 5 times.

Task 3

Alter your program to include the buzzer to sound for 2 seconds before the lights flash then again for 2 seconds after the lights have completed their sequence. The buzzer is connected to pin 6.

Using Motors

The main difference with the clown compared to the traffic lights is it's use of a motor to spin the bow tie. To control the motor it is as simple as switching on one pin for clockwise and the other pin for anti-clockwise. This is shown below.

Pin 4	Pin 5	Direction
Off	Off	Off
Off	On	Clockwise
On	Off	Anti-clockwise
On	On	Off

Task 4

Develop a flowchart and program to meet the given specification.

The clown's tie must spin clockwise for 2 seconds, then anti-clockwise for 2 seconds. This must repeat 5 times. The buzzer should sound at the end of the sequence.

Motor Speed Control

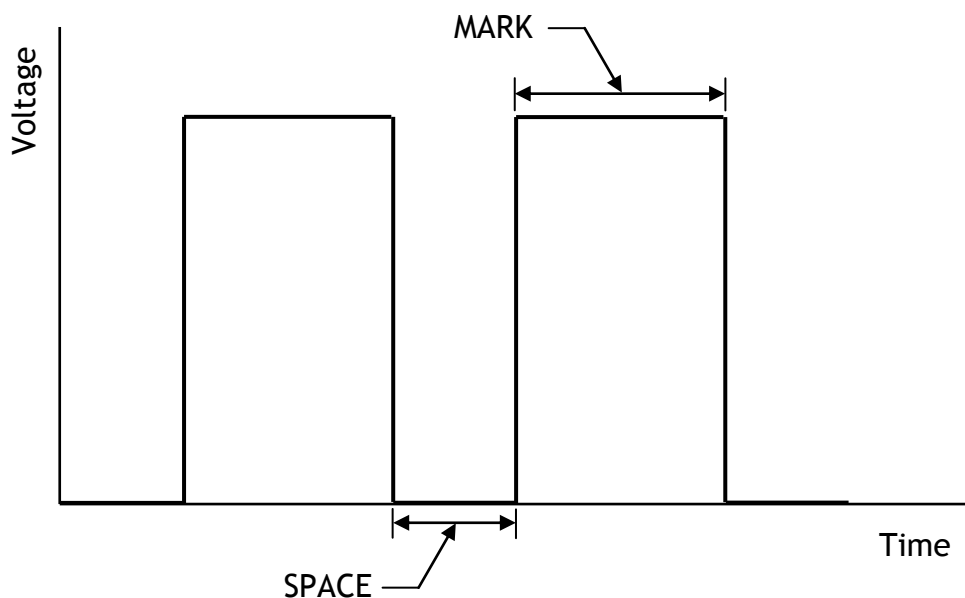
There are three main ways that you can slow down or change the output speed of a motor:

- A) You can do it mechanically by using a belt drive system or gear system. For example, the speed of a motor car can be changed simply by changing gear.
- B) An electrical method of slowing down the speed of an electric motor is to change the supply voltage across it.
- C) The speed of a motor can also be controlled by switching the power supply on an off very quickly. This makes it an ideal method for a microcontroller which can use Pulse Width Modulation.

Pulse Width Modulation

We can now look at changing the speed of the spinning bow tie by changing how long the motor is on and off.

This practice of switching the supply on and off rapidly is called **Pulse Width Modulation (PWM)** which can control the speed of a motor by altering the Mark-Space ratio. The time that the power supply is switched on is called the **mark time**, and the time that the motor is switched off is called the **space time**. By varying the mark-to-space ratio (on-off time), the speed of the motor can be controlled.



In the graph shown above, the “mark” is twice the “space” time. So the mark to space ratio is 2:1. this will produce an average output voltage of two thirds of the supply.

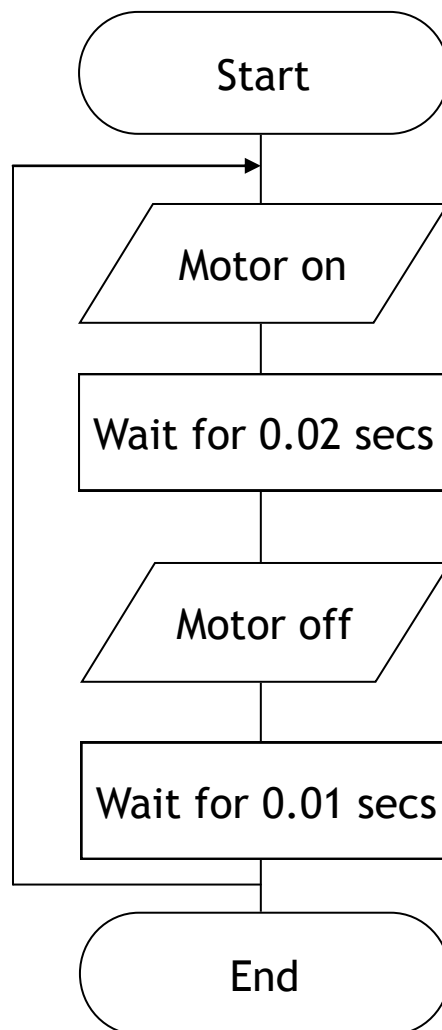
$$\frac{2}{3} \times 5V = 3.33V$$

Task 5

The flowchart for the Pulse Width Modulation to control the speed of a motor is shown below.

Notice that the length of the “mark” and “space” time has been kept very short to prevent the motor turning in a jerky manner. With this mark-to-space ratio the motor should rotate at two-thirds of its full speed.

Develop a PBasic program for the flowchart. Try altering the mark-to-space ratio and note the effect on the motor.



Task 6

Develop a PBasic program to make the motor run at half speed clockwise for 5 seconds.

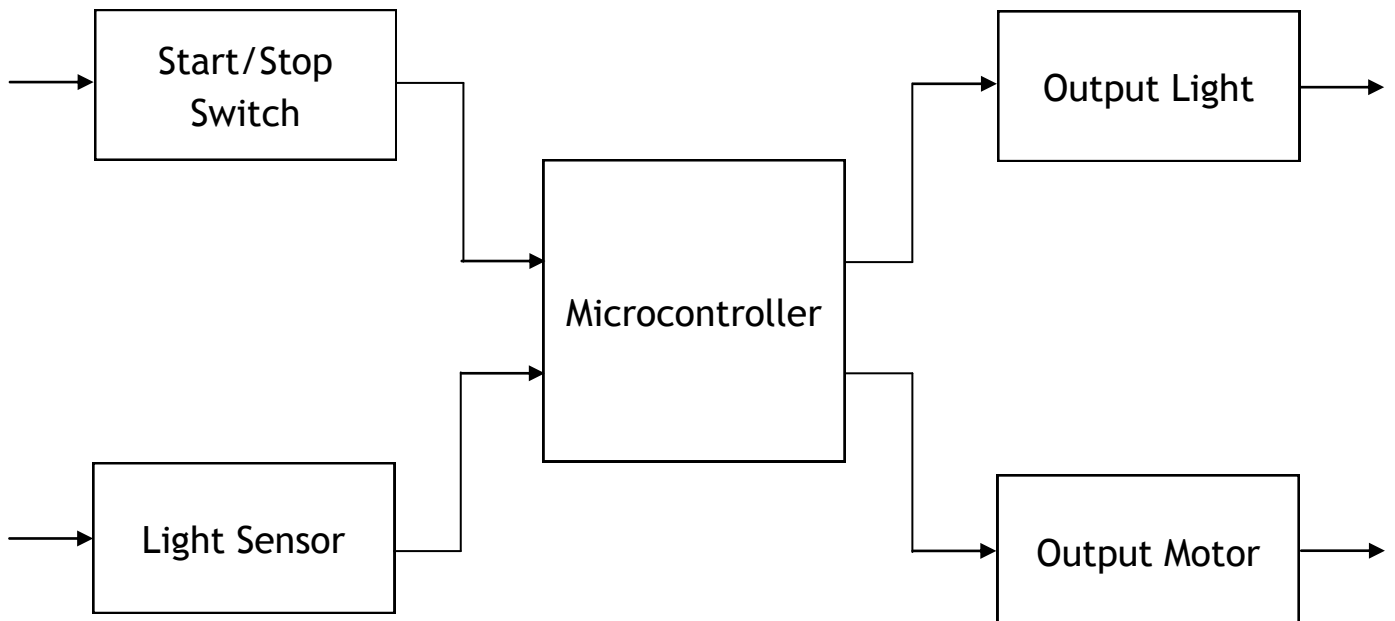
The Systems Approach

A Universal System is shown below.



A Universal System Diagram

After this, we try to break this down by drawing a Subsystems Diagram. In Programmable Control, this process is normally always a microcontroller. You should remember that the microcontroller is really a “computer on a chip”. It is like the brain of the system which processes all of the instructions or information coming from the inputs before sending instructions to the outputs. An example is shown below.

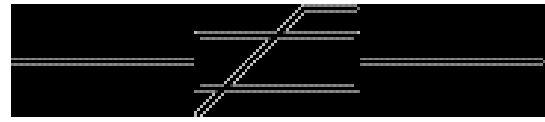
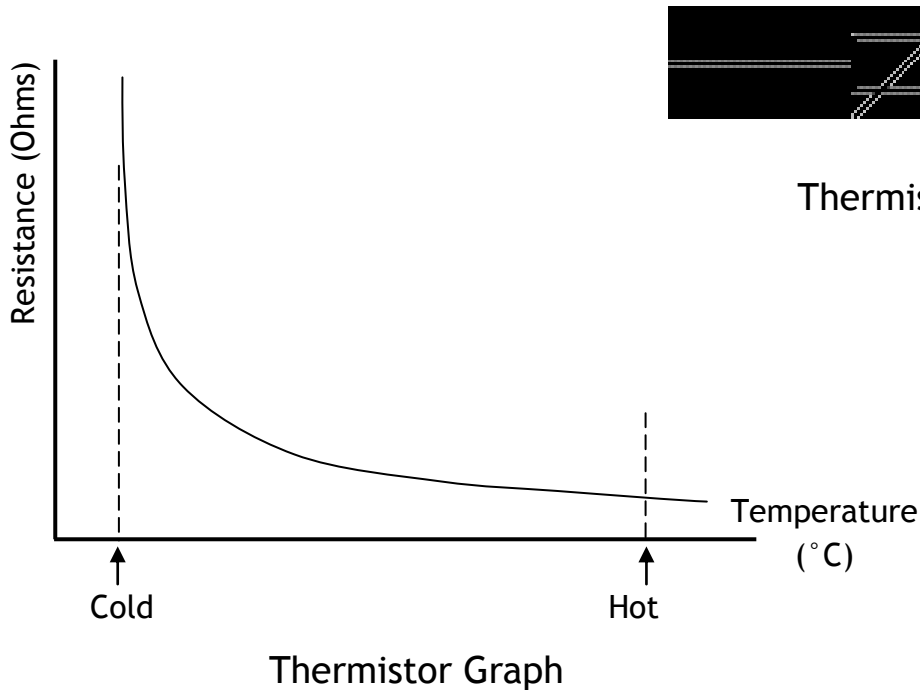


A Subsystems Diagram

The inputs can either be Digital or Analogue. Digital inputs can only be in one of two states - for example, the switch used on the Traffic Light system would either be on or off (1 or 0).

The Thermistor - Analogue Input

A thermistor is a special type of resistor whose resistance changes according to the temperature of its environment. It can therefore be used as an input sensor to detect changes in temperature.



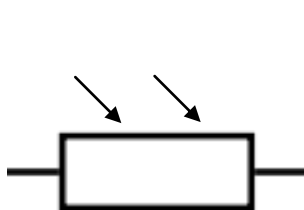
Thermistor symbol



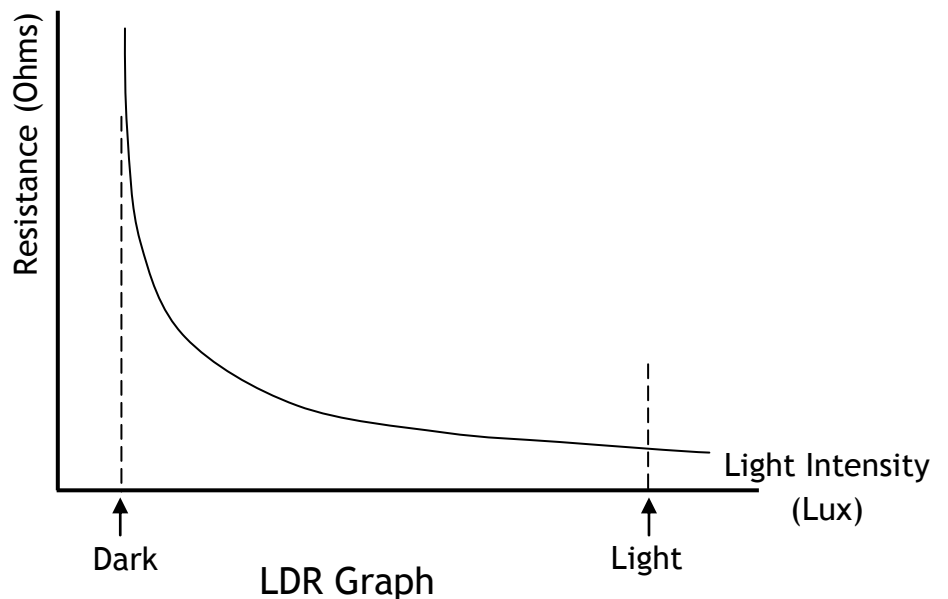
Thermistor

The Light Dependent Resistor

A Light Dependent Resistor (LDR) is a special type of resistor, whose resistance changes according to how much light is falling on it. It can therefore be used as an input sensor to detect changes in light level.



LDR Symbol



Testing an Analogue sensor

To test an analogue sensor, which can show you the actual value as it changes on screen, you are going to use the following test program.

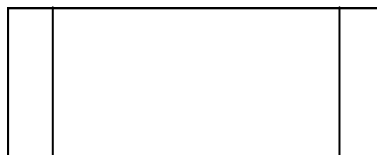
test:

<code>readadc 0, b0</code>	<code>'read channel 0 into variable b0</code>
<code>debug b0</code>	<code>'transmit value to computer screen</code>
<code>pause 100</code>	<code>'short delay</code>
<code>goto test</code>	<code>'jump back to the start</code>

A window showing the value of variable b0 will appear on the computer screen. As the sensor is experimented with the variable value will show the current sensor reading.

Using Subprocedures

Subprocedures are whole blocks of program that, once written, can be used over and over again simply by calling them into the main program. In a flowchart, the subprocedure symbol is a rectangle with two parallel lines down each vertical side as shown below.



To call in this subprocedure in PBasic, a new word is needed and it is **gosub**. When writing you main program, to activate a subprocedure called “lights” you could simple write:

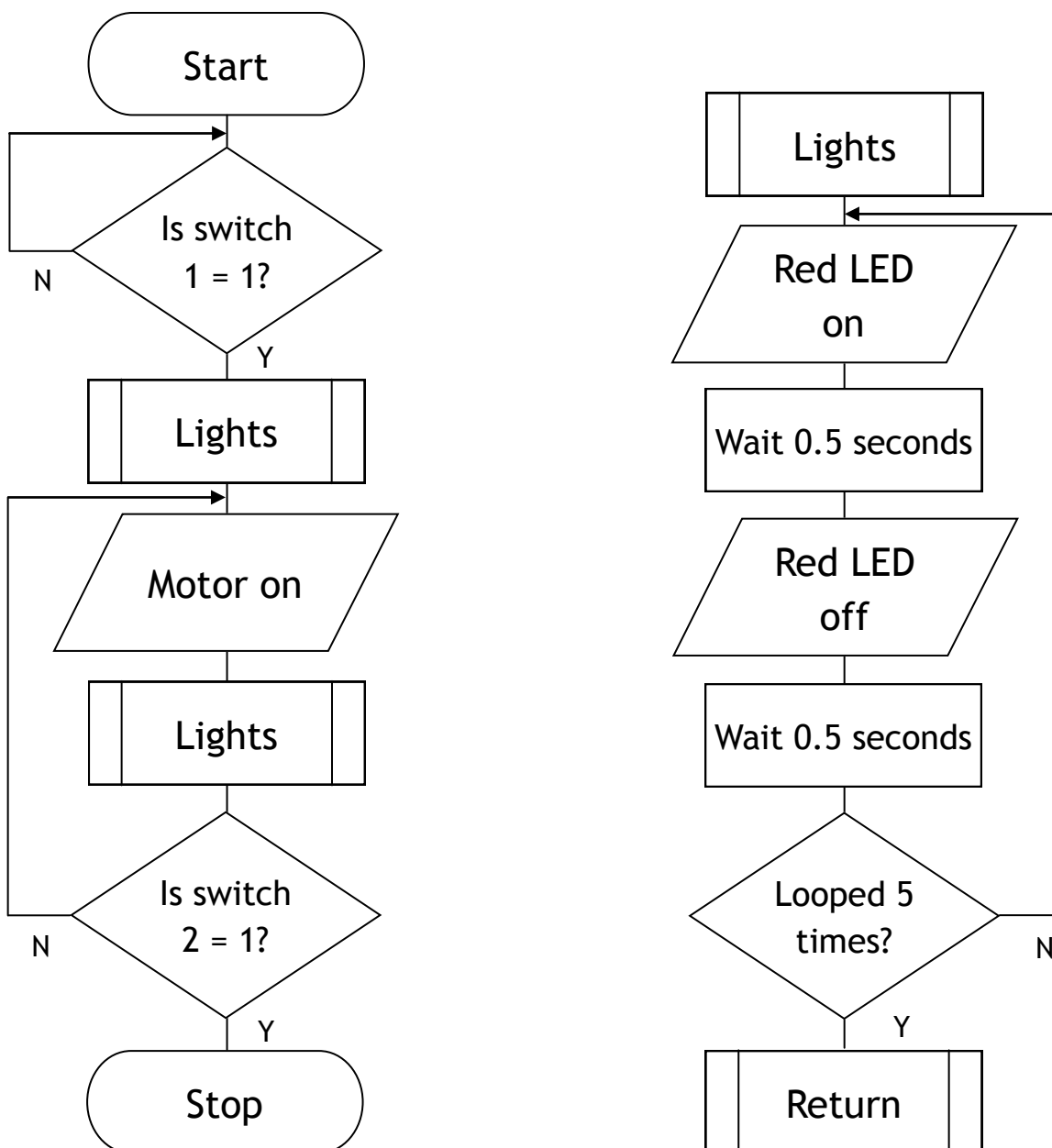
<code>gosub lights</code>	<code>'call up subprocedure lights</code>
---------------------------	---

Defining a Subprocedure

If you simply inserted the line “gosub lights” into your PBasic program, it would return with an “**Error: Label not defined—lights**”.

In other words, it doesn't know what you are talking about because you haven't told the microcontroller what the subprocedure “lights” actually does. At the flowchart stage, the main sequence is drawn out and - if it contains any subprocedures - these are then drawn alongside or underneath the main part.

An example, showing a typical subprocedure is given below to illustrate this, note the use of the word **Return** and how the subprocedure flowchart must start and end with the appropriate symbol.



Writing subprocedures in PBasic

Copy and test the following program.

```
symbol red = 7
```

```
symbol fan = 5
```

```
symbol counter = b0
```

main:

```
if pin1 = 0 then main
```

'check to see if start switch is pressed

```
gosub lights
```

'jump to the subroutine lights

```
gosub motor
```

'jump to subprocedure motor

```
gosub lights
```

'jump to subprocedure lights

```
end
```

'ends program

lights:

```
for counter = 1 to 5
```

'start a for...next loop

```
high red
```

'red LED on

```
pause 500
```

'wait for 0.5 seconds

```
low red
```

'red LED off

```
next counter
```

'repeat LED flashing 5 times

```
return
```

'return to main program

motor:

```
high fan
```

'switch fan on

```
if pin2 = 1 then motor_off
```

'test pin 2

```
goto motor
```

'keep fan on

motor_off:

```
low 5
```

'switch fan off

```
return
```

'return to main program

Clown program example

Copy and test the following program to control the clown.

```
symbol counter = b1
```

```
symbol eyes = 7
```

```
symbol buzzer = 6
```

main:

```
  debug b0
```

```
  pause 100
```

```
  readadc 0, b0
```

```
  if b0 > 180 then buzz
```

```
  low buzzer
```

```
  pause 100
```

```
  if b0 < 175 then flash
```

```
  low eyes
```

```
  pause 100
```

```
  goto main
```

flash:

```
  high eyes
```

```
  pause 100
```

```
  goto main
```

buzz:

```
  high buzzer
```

```
  pause 100
```

```
  goto main
```

Task 7

Write a program that will control the clown motor to run at 3 different speeds. A low speed for a cool temperature, a medium speed for a medium temperature and full speed for a hot temperature.