

# Higher Engineering Science

## Flowcharts & PBasic



Name: \_\_\_\_\_

Teacher: \_\_\_\_\_

Class: \_\_\_\_\_



Mrs Gault

## Learning Intentions

- Programmable Control
- Pulse width modulation
- Control routines up to four inputs and outputs
- Processing Analogue inputs
- Infinite and finite loops

## Success Criteria

- I can use Pbasic code to programme a microcontroller.
- I can understand and describe the effect of Pulse Width Modulation on a motor.
- I can create a control flowchart for a microcontroller with up to 4 inputs and outputs.
- I can use a microcontroller which is able to read analogue inputs and create a flowchart to control it.
- I can create a flowchart with infinite and finite loops.

In this unit we will build on the skills learned in National 5 Engineering Science.

## 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.

## Advantages of a microcontroller

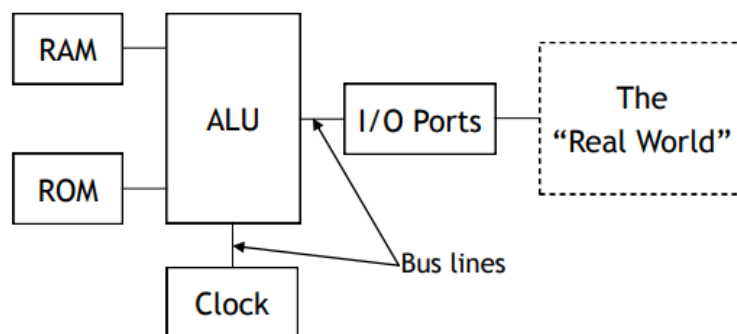
The microcontroller can be programmed to perform different tasks. This means it can be used in the same product in many different ways or it can be used in many different products, all of this simply by changing the programmed routine. This therefore is its number one advantage, it is very flexible and 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

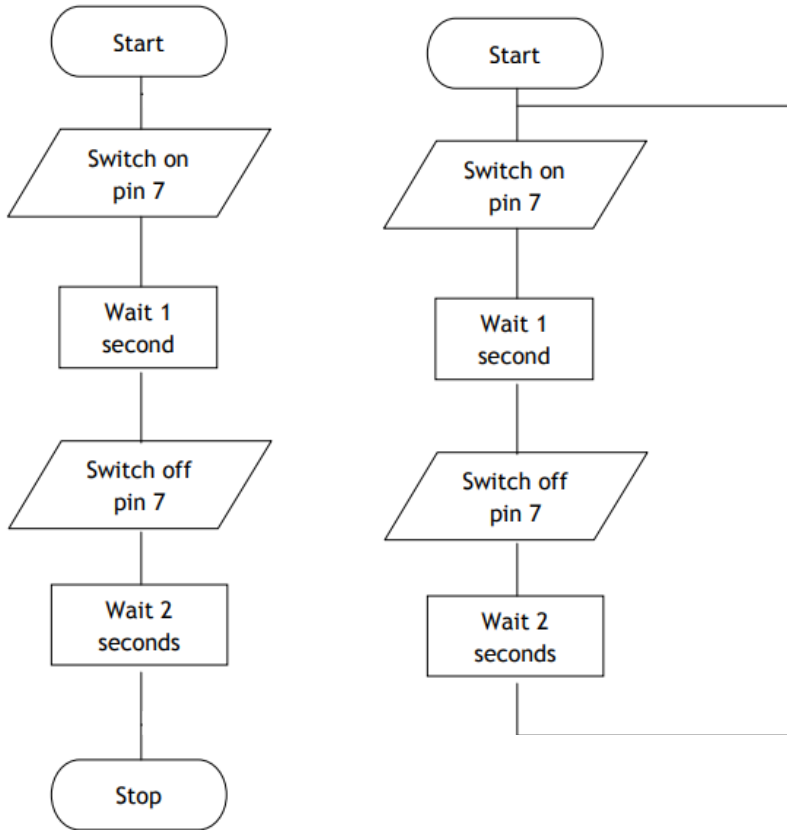
## Inside a Microcontroller

- 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

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





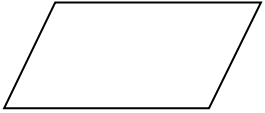
The flowchart on the left is a basic flowchart that completes a simple process.

The flowchart on the right is an example of a flowchart that uses three different Symbols and a Feedback Loop.

Some programs will never stop, and will repeatedly loop again and again, such as traffic lights, moving security cameras etc. To do this we can use an arrow to loop back to any point in the program, as shown on the right. In flowcharts like this, we do not use a 'Stop' symbol.

## Basic Symbols

Have a look at the flowchart on the right hand side of the page. You will see the three most common symbols.

	<p><b>'Terminator' A.K.A Start/Stop Symbol:</b></p> <p>The 'Start' or 'Stop' symbol shape is a rectangle with rounded ends. Each flowchart must contain only one 'Start' symbol and, usually, only one 'Stop' symbol.</p>
	<p><b>'Process' Symbol:</b></p> <p>The 'process' symbol is a rectangle. This is used for anything which happens inside the microcontroller. The most common uses are for time delays and saving numbers to the memory.</p>
	<p><b>'Outputs' Symbol:</b></p> <p>The 'Outputs' symbol is a parallelogram. The text inside the symbol explains which output pins are switched on or off at any time.</p>

## Pin Out Diagrams

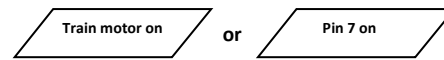
Pin out diagrams are given to show you how pins are connected to inputs and outputs. When you are writing your flowchart you can choose whether you write down the name of the component, or the name of the pin that it is attached to.

### Pin Out Diagram:

Input connection	Pin	Output connection
	7	train motor
	6	points
	5	green signal
	4	red signal
	3	
	2	
	1	
	0	

### Example

Using the pin out diagram on the left, we can switch the train motor on by writing either 'train motor on' or 'Pin 7 on'.



Either command can be used, however it is sensible to avoid 'programming language' where possible so that flowcharts are as simple as possible to read and understand.

N5 recap

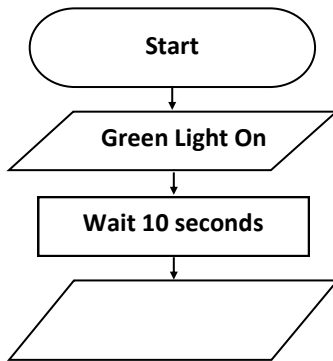
### Task 1 - Traffic Lights

A set of temporary traffic lights are required for road works. The project manager has decided that the best timing for the lights is as follows:

Green	10 s
Amber	2 s
Red	10 s
Red and Amber	2 s



Complete the flowchart below then simulate on Yenka.



N5 recap

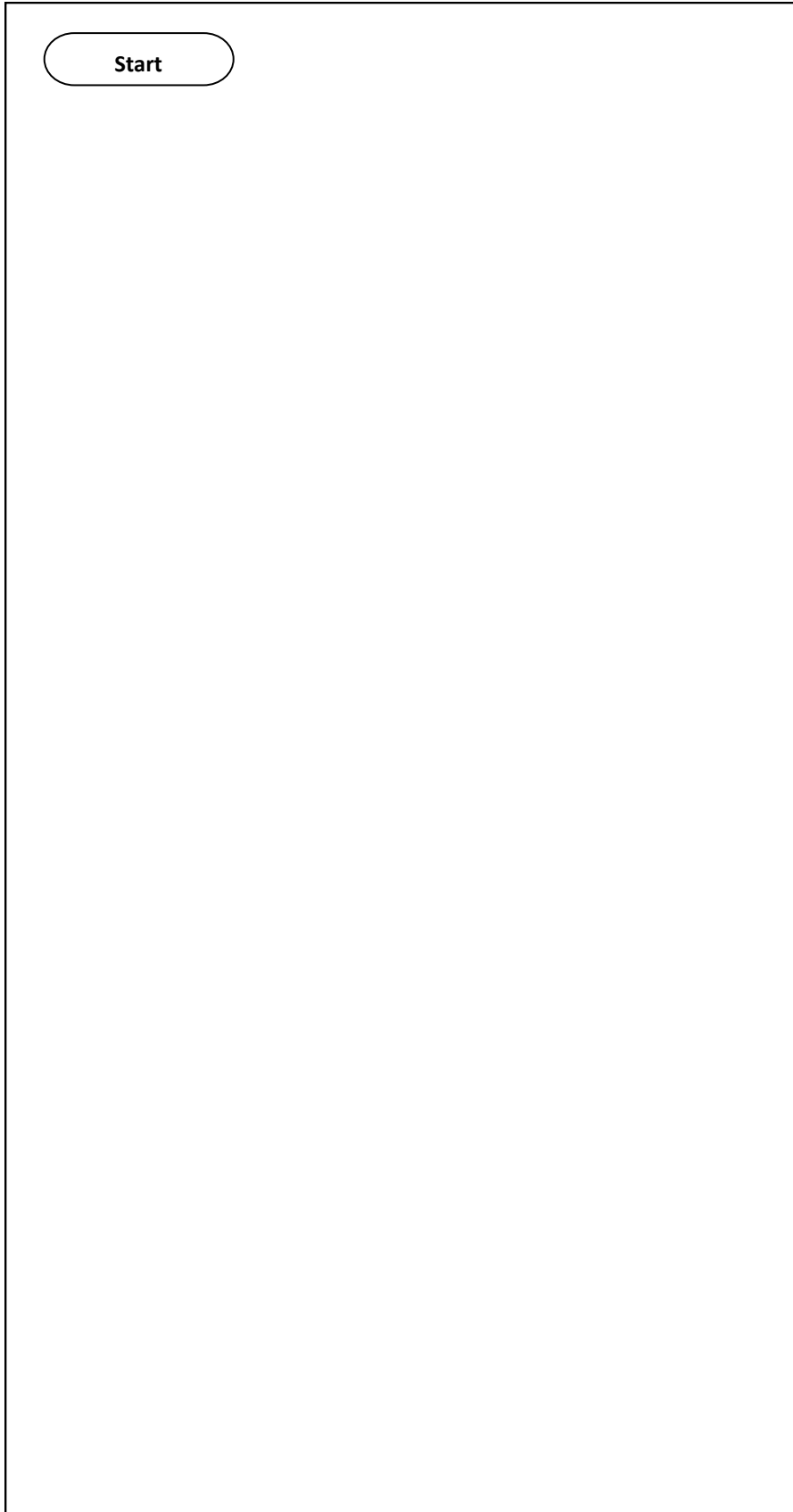
### Task 2 - Microwave

You are involved with the design of a microwave oven. The cooking sequence is shown to the right. Draw a flowchart for this sequence.



<u>Step</u>	<u>Process</u>
1	<i>Light on</i>
2	<i>Turntable on</i>
3	<i>Magnetron on</i>
4	<i>Wait 30 seconds</i>
5	<i>Magnetron off</i>
6	<i>Wait 10 seconds</i>
7	<i>Turntable off</i>
8	<i>Buzzer on</i>
9	<i>Wait 0.5 seconds</i>
10	<i>Buzzer off</i>
11	<i>Light off</i>

Complete the flowchart below then simulate on Yenka.





N5 recap

### Task 3 - Toy Train

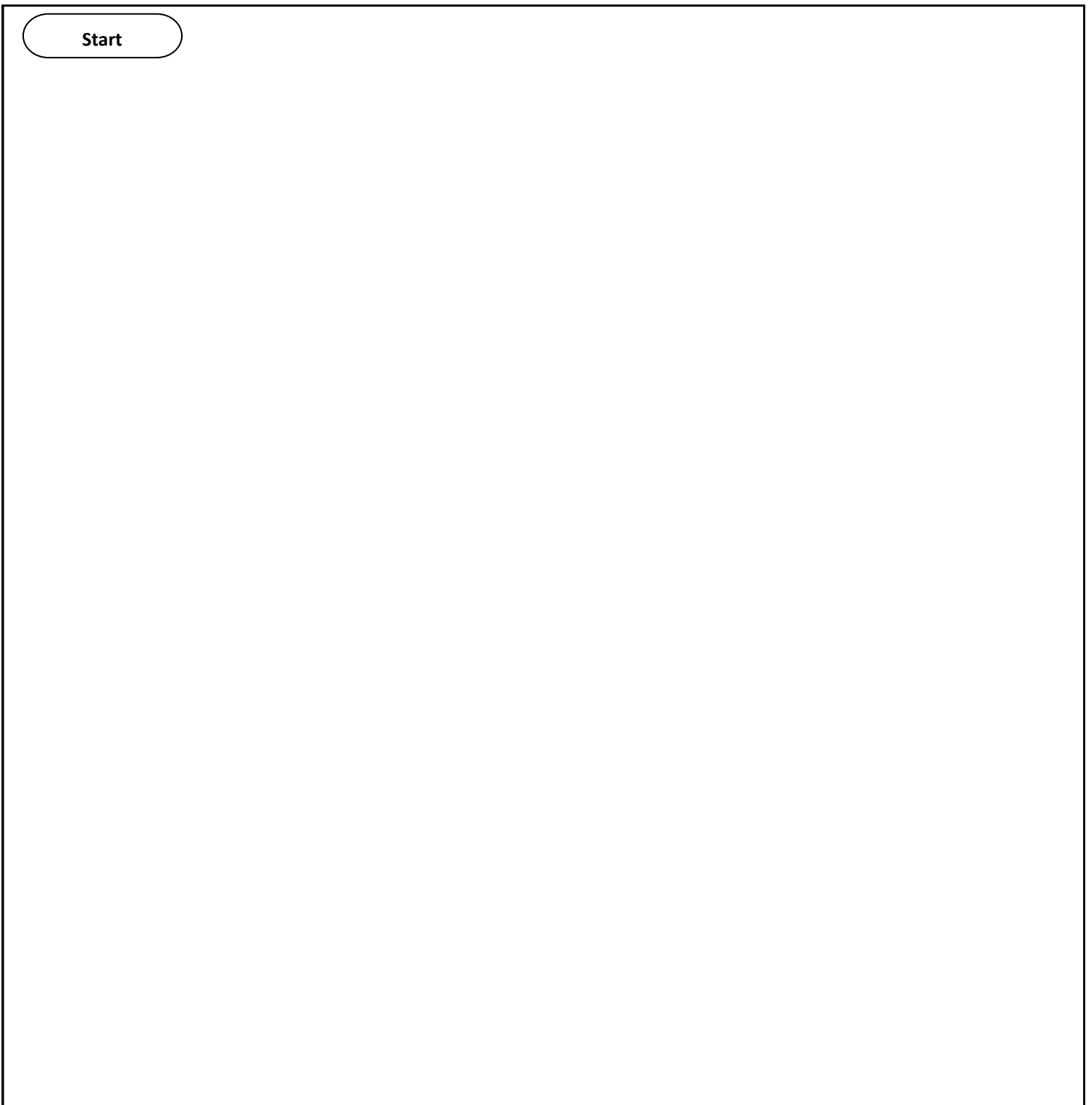
A toy shop owner has asked for your assistance to develop a program which will carry out the sequence shown on the right.

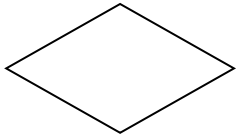
#### Pin Out Diagram:

Input connection	Pin	Output connection
	7	train motor
	6	points
	5	green signal
	4	red signal
	3	
	2	
	1	
	0	

Step	Process
1	Switch the green signal on
2	Switch the train on for 30seconds
3	Stop the train
4	Switch the green signal off and the red signal on
5	Switch the points on
6	Switch the train on for 30 seconds
7	Stop the train
8	Switch the points off
9	Switch the red signal off and the green signal on
10	Jump back to step 2

Complete the flowchart below then simulate on Yenka.



	<p><b>'Decision' Symbol:</b></p> <p>The decision symbol is mainly used for creating 'For...next' loops and for testing whether inputs are high or low.</p>
---	--

## Input Testing

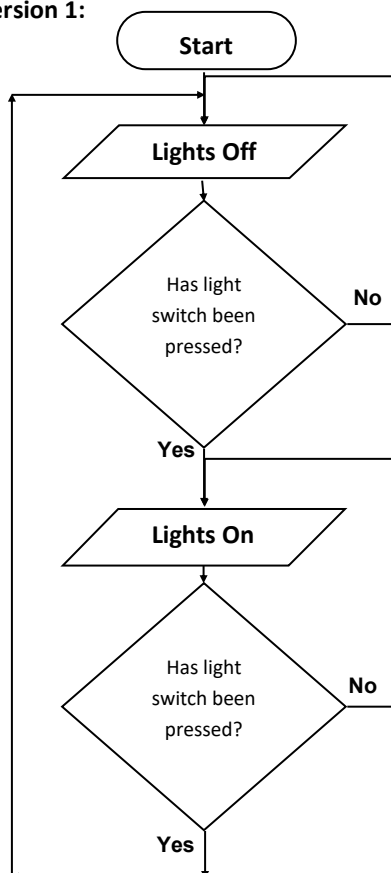
Decision symbols are used to determine the value of an input. They are most commonly used in this manner to check if an input is high or low, however they can also be used to check analogue readings such as temperature levels, light levels etc.

The flowchart below shows a program for a light switch.

**Note:** There is now an input connection listed in the pin out diagram. When Pin0=0, the light switch is set to 'off'. When Pin0=1, the light switch is set to 'on'.

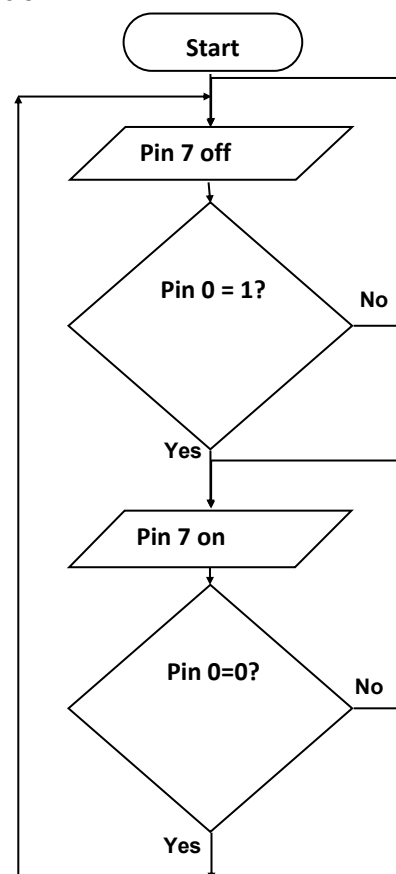
Input connection	Pin	Output connection
	7	Lights
	6	
	5	
	4	
	3	
	2	
	1	
Light Switch	0	

Flowchart Version 1:



Flowchart Version 2:

OR



Although there are differences, both of the above flowcharts are correct. It is up to you whether you choose to write out 'label' style steps which explain each command (as seen above on the left), or you can write out steps referring directly to the output (1 or 0) of the pins (as seen above on the right).

## Task 4 - Olympic Canoe Conveyor System

N5 recap



An inclined conveyor system on the Olympic canoe downhill circuit is operated by a microcontroller.

The control program will activate in the following sequence:

- When the system is switched on the conveyor belt starts and a barrier opening motor runs for 3 seconds before stopping;
- When a canoeist is sensed on the conveyor the barrier closing motor runs for 3 seconds before stopping;
- After an additional delay of 15 seconds the conveyor belt stops;
- Sequence repeats until switched off.

Pin Out Diagram

Input connection	Pin	Output connection
	7	
	6	Barrier Closing motor
	5	Barrier Open motor
	4	Conveyor belt signal
	3	
	2	
Canoe sensor (=1 when	1	
Main ON/OFF switch	0	

Complete the flowchart below then simulate on Yenka.

## Task 5 - Car Wash

N5 recap



A garage jet wash is operated by a microcontroller.

The control program will activate in the following sequence:

- The customer must press the start button outside the car wash.
- The car wash will not start until the sensor detects the car inside the car wash.
- The soap jets will run for 30 seconds, then the rinse jets will run for 30 seconds, then the dryer will run for 30 seconds.
- The system will restart, waiting for the next user.

**Pin Out Diagram**

Input connection	Pin	Output connection
	7	Dryer
	6	Rinse Jets
	5	Soap Jets
	4	
	3	
	2	
Car detection sensor (=1 when car is detected)	1	
Start Button (=1 when pressed)	0	

Complete the flowchart below then simulate on Yenka.

## Task 6 - Garage Door

N5 recap



An automatic door for a garage is operated by a microcontroller. The door is opened/closed by an electric motor with 2 limit switches to detect if the door is fully open or fully closed.

The user has a remote with an 'open' button and a 'close' button.

- When the 'open' button is pushed, the motor must rotate clockwise until the 'fully open' limit switch is pressed.
- When the 'close' button is pushed, the motor must rotate anti-clockwise until the 'fully closed' limit switch is pressed.
- Once the door has opened or closed, the program should restart.

**Pin Out Diagram**

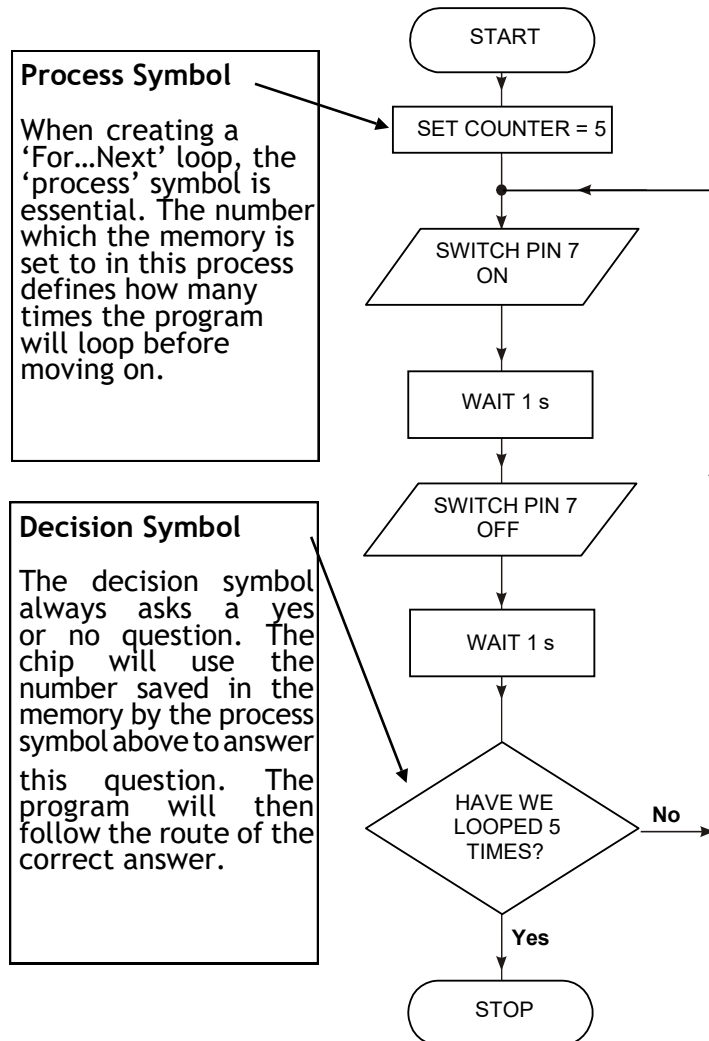
Input connection	Pin	Output connection
	7	Motor Clockwise
	6	Motor Anti-clockwise
	5	
	4	
'Fully Closed' Limit Switch (=1 when door is fully closed)	3	
'Fully Open' Limit Switch (=1 when door fully open)	2	
Close Button (=1 when pressed)	1	
Open Button (=1 when pressed)	0	

Complete the flowchart below then simulate on Yenka.



## 'For ... next' loops

It is often useful to repeat the same part of a program a number of times, for instance when flashing an LED. In these cases a 'decision' symbol is used, and a 'For... next' loop is created.



## Task 7 - Baseball Pitcher

N5 recap

A sports training facility has an automatic pitching machine for athletes to practice their batting technique. A solenoid actuator is used to control the 'firing' of the baseball. When the actuator is set to high, the ball is 'fired', the machine automatically reloads with the next ball when the actuator is set to low. The program must follow the sequence below.

1. User presses the start button to begin process.
2. Ball is fired.
3. Wait 5 seconds
4. Next ball is fired.
5. Process repeated 10 times then program ends.



### Pin Out Diagram

Input connection	Pin	Output connection
	7	Solenoid Actuator
	6	
	5	
	4	
	3	
	2	
	1	
Start Button (=1 when	0	

Complete the flowchart below then simulate on Yenka.



## Task 8 - Nightclub Lighting

N5 recap



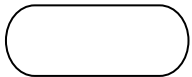
A large nightclub uses an AV technician to control the lighting around the dancefloor. He is looking to use a microcontroller to run the following lighting sequence:

1. 'Start' button is pressed.
2. Strobe light flashes on and off at 0.2s intervals 100 times.
3. Red lasers on for 0.5s then off, Green lasers on for 0.5 seconds then off.  
Step 3 is repeated 20 times.
4. Program ends.

### Pin Out Diagram

Input connection	Pin	Output connection
	7	Strobe Light
	6	Red Laser
	5	Green Laser
	4	
	3	
	2	
	1	
Start Button (=1 when	0	

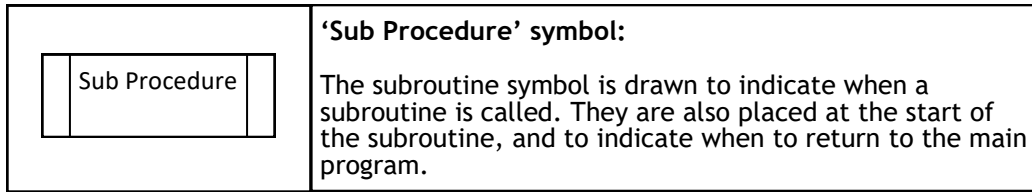
Complete the flowchart below then simulate on Yenka.





## Sub Procedures

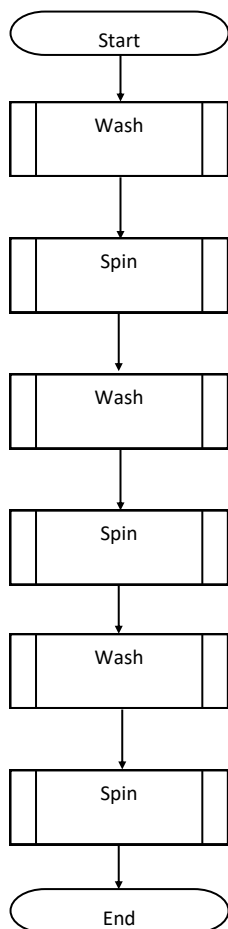
A sub-procedure is a separate 'mini-program' that can be called from the main program. Once the sub-procedure has been carried out, the main program continues.



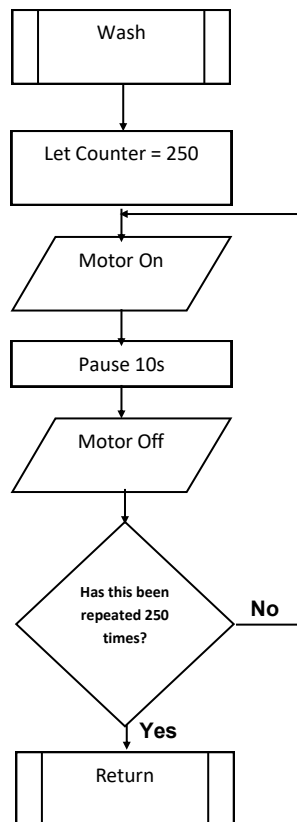
## Washing Machine (Sub-procedure example)

The flowchart below shows the control of a washing machine by calling sub-routines. Observe that when each sub-routine is finished the main program carries on.

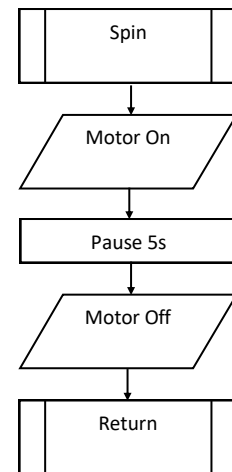
**Main Program:**



**Sub-routine 1:**



**Sub-routine 2:**



### Task:

Describe what happens in the program above.

What are the advantages of using sub-routines when creating programs?

## Task 9 - Stadium Roof

An automatic sliding roof for a sports stadium is operated by a microcontroller. Part of the control program includes a sub-procedure 'close' which will run when frost is detected. The sequence is as follows:

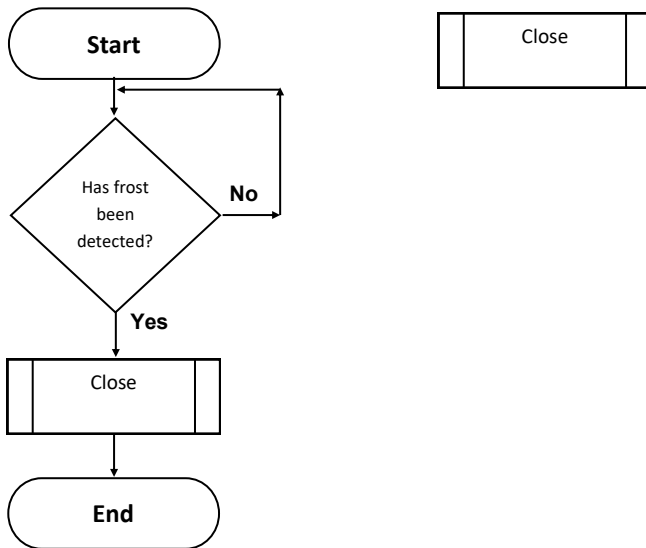
- the floodlights switch on;
- after a 5 second delay the roof motor switches on;
- when the stadium roof is fully closed a limit switch is activated which stops the motor;
- the sequence ends and returns to the main program.

### Pin Out Diagram

Input Connection	Pin	Output Connection
	7	
	6	Roof motor
	5	
	4	Floodlights
	3	
	2	
Limit Switch	1	
	0	



Complete the flowchart below then simulate on Yenka.



## Task 10 - School Minibus

A warning system on a school minibus is operated by a microcontroller. Part of the control program includes a sub-procedure 'warning' which will activate when the minibus' reverse gear is engaged. The sequence of operations is as follows:

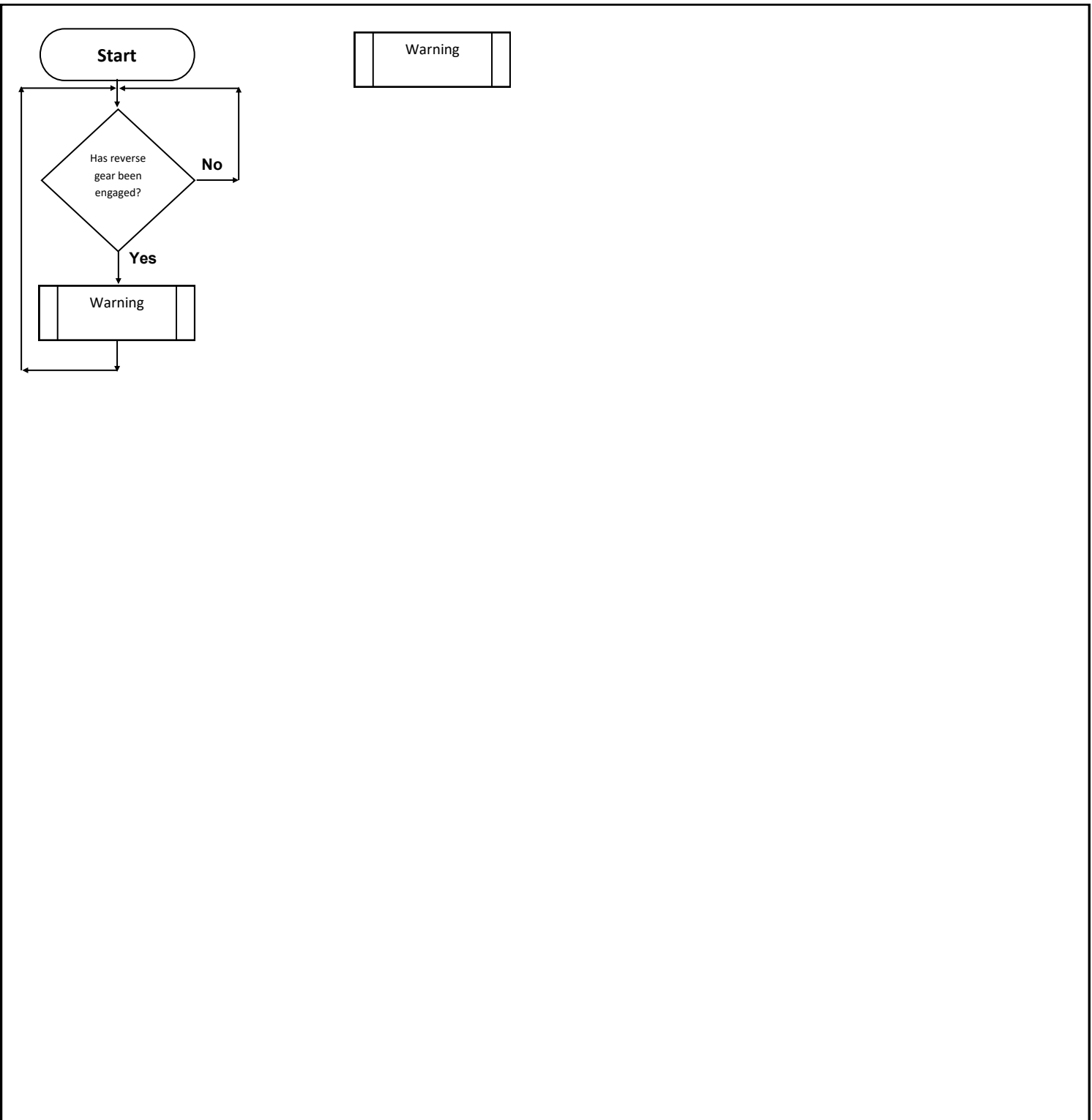
- A buzzer and lights go on;
- A delay of ½ second;
- The buzzer and lights go off;
- A delay of ½ second;
- Only when the reverse gear is disengaged the sequence ends and returns to the main program.



### Pin Out Diagram

Input Connection	Pin	Output Connection
	7	
	6	
	5	Buzzer
	4	
	3	Lights
	2	
Reverse gear sensor	1	
	0	

Complete the flowchart below then simulate on Yenka.



## Task 11 - Shop Christmas Lights

As part of a Christmas decoration in a shop, a lighting sequence is to be controlled by a microcontroller. The output connections are shown below.



When a visitor treads on a pressure mat under the carpet, the lights should flash on and off in sequence three times.

<u>Light</u>	<u>Time</u>
1. Red On	0.5 seconds
2. Yellow On	0.5 seconds
3. Green On	0.5 seconds

### Pin Out Diagram

Input connection	Pin	Output connection
	7	red light
	6	yellow light
	5	green light
	4	
	3	
	2	
	1	
pressure mat	0	

Complete the flowchart below then simulate on Yenka.



## Task 12 - Burglar Alarm

A burglar alarm must sound a buzzer and light a warning signal for 20 seconds when any of the four windows in a house are opened. Each window contains a reed switch that is connected to the alarm.

Draw a flowchart and write a PBASIC program that will operate the burglar alarm correctly. Use the following input and output connections.

**Hint:** You will need to make use of a sub-system.

Complete the flowchart and Pbasic code then simulate on Yenka.



**Pin Out Diagram**

Input connection	Pin	Output connection
	7	red light
	6	buzzer
	5	
	4	
switch 3	3	
switch 2	2	
switch 1	1	
switch 0	0	

**Pbasic code:**

## Task 13 - Washing Machine

A washing machine has connections as shown in the pin out diagram below. The washing machine operates as follows:

- Wait until the start switch is on.
- Wait until the door switch is on.
- Switch on the LED.
- Switch on the solenoid bolt.
- Wash cycle: repeated 20 times - motor forwards for five seconds, motor backwards for five seconds.
- Spin cycle: repeated 10 times - motor forwards for two seconds, motor backwards for two seconds.
- Switch off the solenoid bolt.
- Switch off the LED.



Input connection	Pin	Output connection
	7	motor reverse
	6	motor forward
	5	solenoid bolt
	4	LED
	3	
	2	
door microswitch	1	
start switch	0	

Complete the flowchart and then simulate on Yenka.



## Task 14 - Bank Safe

The bank safe operates as follows,

- Switch on the solenoid bolt/red LED.
- Wait until code switch 1 is pushed.
- Wait until code switch 2 is pushed.
- Wait until code switch 3 is pushed.
- Switch off the solenoid bolt.
- Switch off the red LED.
- Wait until the door is opened (microswitch off).
- Switch on the green LED.
- Wait 10 seconds.
- Switch off the green LED.
- Switch on the buzzer.
- Wait until the door is closed (microswitch on).
- Switch off the buzzer.
- Loop back to step 1.



**Pin Out Diagram**

Input connection	Pin	Output connection
	7	buzzer
	6	
	5	solenoid/red LED
	4	green LED
code switch 3	3	
code switch 2	2	
code switch 1	1	
door microswitch	0	

Complete the flowchart and then simulate on Yenka.

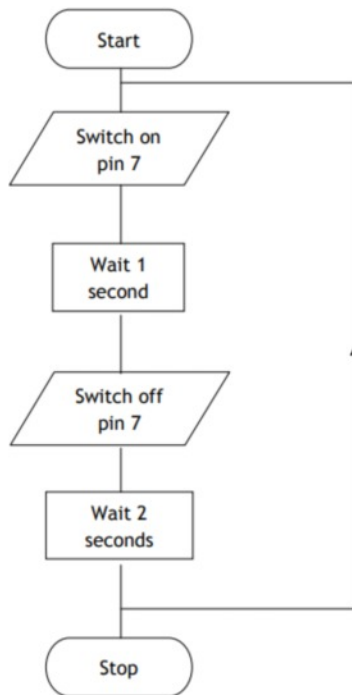
## PBasic for Higher

### 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

Create a Pbasic programme for a set of traffic lights to operate as follows:

- Green on for 10 seconds
- Amber on for 3 seconds
- Red on for 10 seconds
- Red and Amber on for 3 seconds
- Sequence restarts

Pin	Output
7	Red
6	Amber
5	Green

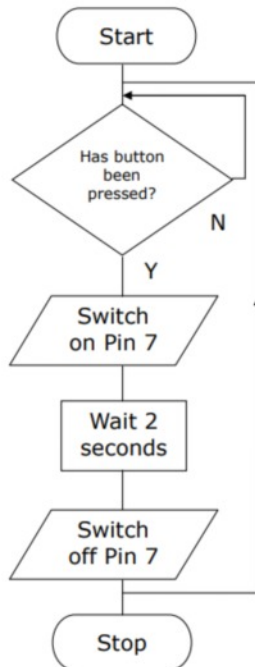


## Testing inputs

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 2

Adapt your PBASIC program from task 1 for a pedestrian crossing, adding in the Red and Green man. Use pin 0 for your push button to stop the traffic.

Input	Pin	Output
	7	Red
	6	Amber
	5	Green
	4	Red Man
	3	Green Man
Switch	0	

## “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.

```
symbol counter = b0          'rename variable b0 'counter'
symbol green = 5            'rename output 5 'green'

flash:
  for counter = 1 to 10     'start a for... next loop
    high 5                  'switch green on
    pause 1000              'wait 1 second
    low 5                   'switch green off
    pause 1000              'wait 1 second
  next counter              'add 1 to counter
end                          'ends program
```

### Task 3

Adapt your PBASIC program from task 2 for a pedestrian crossing where 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.

Input	Pin	Output
	7	Red
	6	Amber
	5	Green
	4	Red Man
	3	Green Man
Switch	0	

## Using Motors

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 Pbasic program to turn a motor 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

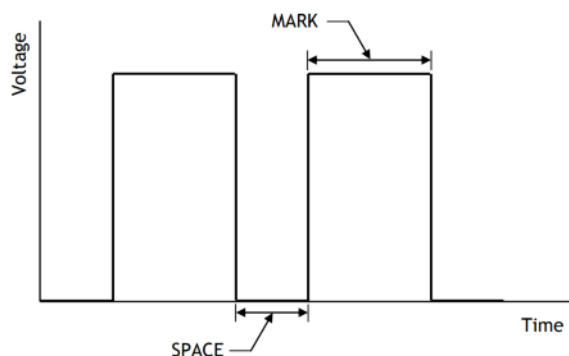
There are three main ways that you can slow down or change the output speed of a motor:

- 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.
- An electrical method of slowing down the speed of an electric motor is to change the supply voltage across it.
- The speed of a motor can also be controlled by switching the power supply on and 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.



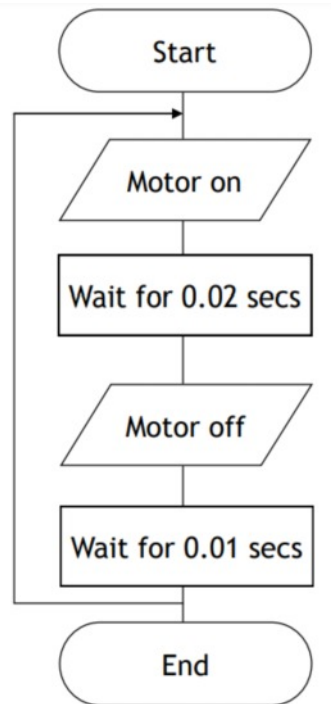
In the graph shown above, the “mark” is twice the “space” time. So the mark to space ratio is 2:1. this will produce an average output voltage of two thirds of the supply.

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

### 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.

## 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:
    readadc 0, b0      'read channel 0 into variable b0
    debug b0           'transmit value to computer screen
    pause 100         'short delay
    goto test         'jump back to the start
```

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.

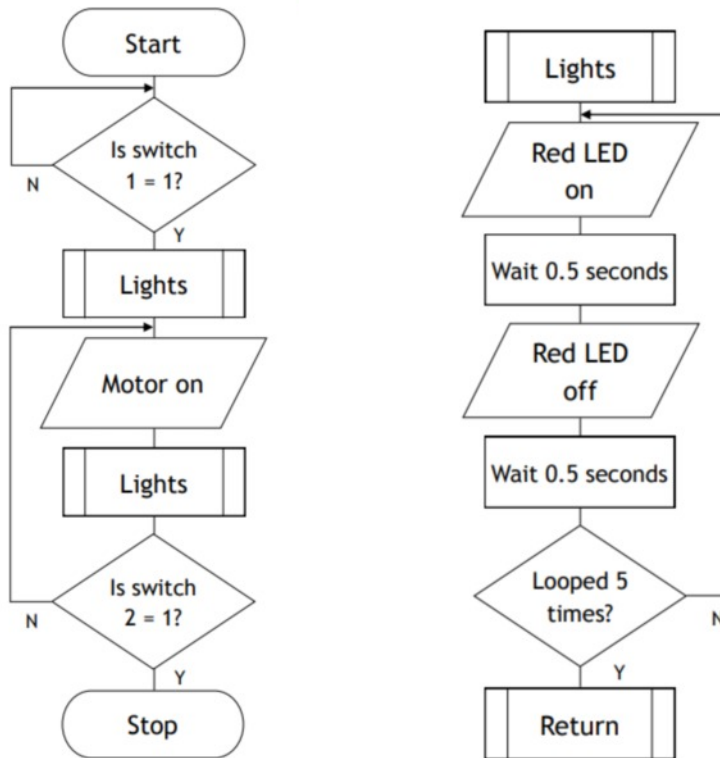


To call in this subprocedure in PBasic, a new word is needed and it is gosub. When writing you main program, to activate a subprocedure called "lights" you could simple write:

```
gosub lights      'call up subprocedure lights
```

## 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 Programme

This programme was for a model clown which had an input thermistor, a motor to make it's bow tie spin, LED eyes and a buzzer.

This programme is an example of reading an analogue input. The part of the microcontroller that is storing the information from the thermistor is b0.

The first line of code clears any information from b0. The 3rd line of code reads the thermistor value.

The if then statement says that if the thermistor has been heated above 180 to go to sub procedure "buzz". If the thermistor has not been heated to reach 180 the programme then goes to the line below.

### 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.



```
symbol counter = b0
```

```
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:
```

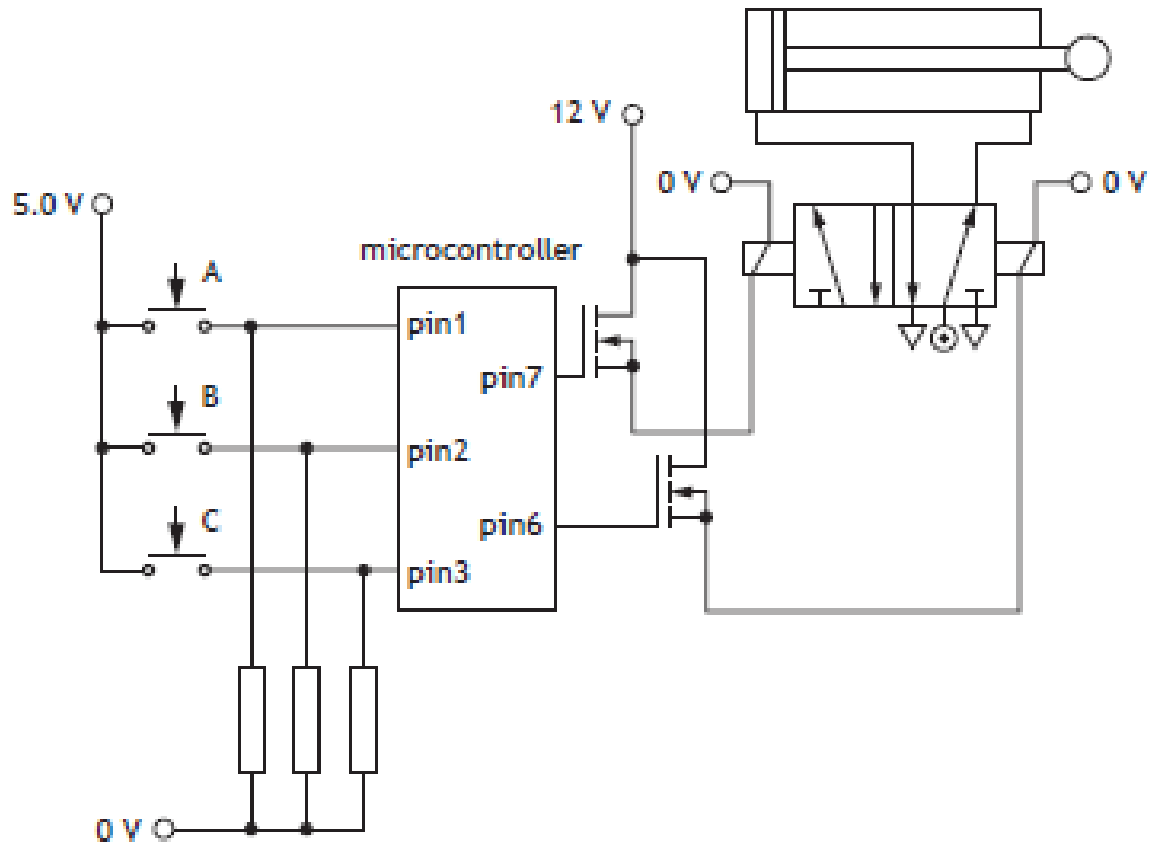
```
  high buzzer
```

```
  pause 100
```

```
  goto main
```

10. (continued)

To reduce costs and allow greater flexibility, an electronic engineer is asked to produce an alternative solution using a microcontroller.



The program to control the cylinder must meet the following criteria:

- $\text{outstroke} = \overline{A} \cdot \overline{B} + C$
- the cylinder must remain outstroked for 1500 milliseconds before instroking
- 200 milliseconds after the command to instroke, the system must start monitoring the switches again.



10. (continued)

The program shown below in PBASIC and ARDUINO code is known to have faults.

<pre>main: if pin3 = 1 then jump       if pin2 = 1 then main       if pin1 = 0 then main  jump: high 7       pause 1500       high 6       pause 200       low 6        goto jump</pre>	<pre>void loop(){   if (digitalRead(3)==HIGH){jump0;}   else if (digitalRead(2)==HIGH){loop0;}   else if (digitalRead(1)==LOW){loop0;}   else{loop0;} }  void jump0{   digitalWrite(7,HIGH);   delay(1500);    digitalWrite(6,HIGH);   delay(200);   digitalWrite(6,LOW); }</pre>
---	---

(b) Describe three different faults in the program shown above.

3

Fault 1 \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Fault 2 \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Fault 3 \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## 2023 Past Paper

### 12. (continued)

A second prototype using a microcontroller instead of a logic circuit is to be tested. The system must meet the following criteria:

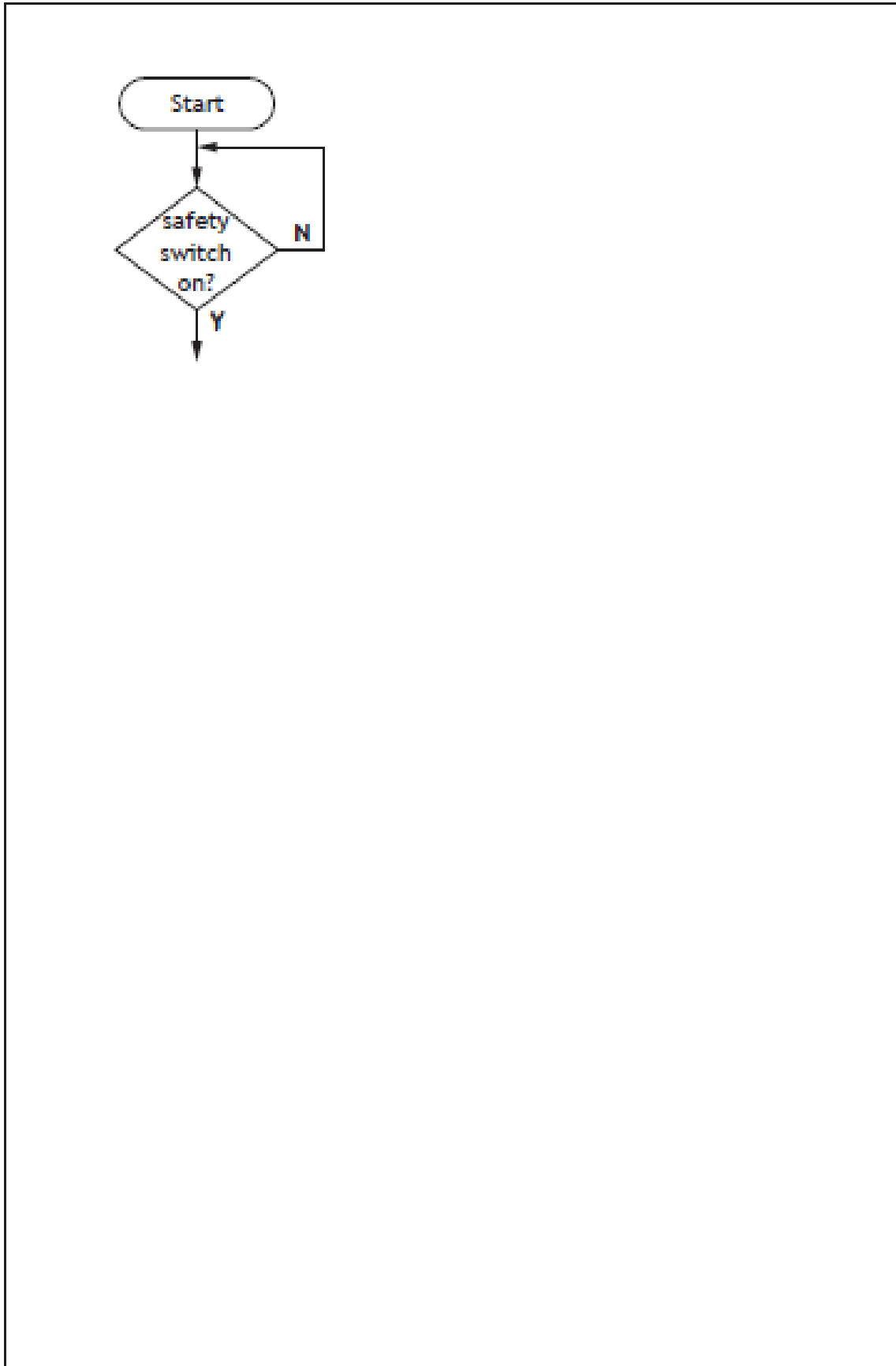
- the system must wait until a safety switch has been activated
- when either of the manual start switches A or B are pressed a signal must be sent to open the hopper and the amber LED must come on
- if the weight reading is between a value of 100 and 120 the green LED alone must come on and the hopper must close
- if the reading is greater than 120 the hopper must close and the red LED must flash 5 times per second until the reading drops to less than 120 (material will be removed manually by an operator to reduce the weight).

Input	Pin	Output
	7	hopper open (1)/close (0)
	6	red LED
	5	green LED
	4	amber LED
safety switch	3	
manual start switch A	2	
manual start switch B	1	
weight (analogue value)	0	

12. (continued)

- (d) Complete, with reference to the pinout table and the criteria given opposite, a flowchart for the system.

10



2022 Past Paper

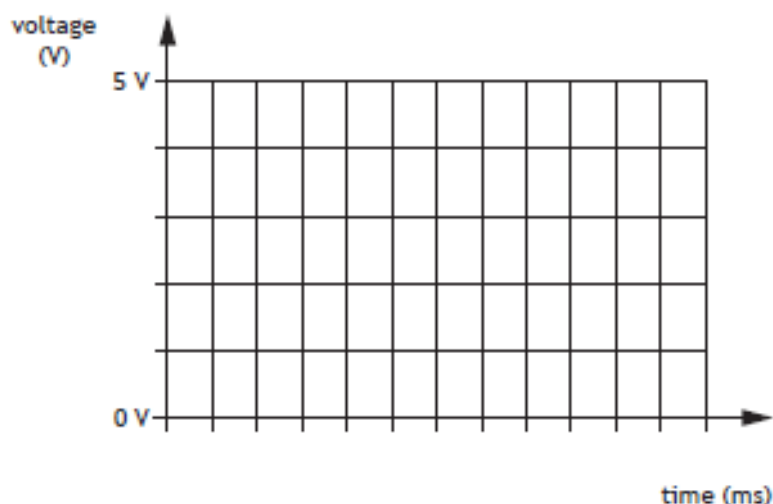
4. The speed at which an automatic garage door opens is controlled by a motor using pulse-width modulation (PWM).



- (a) Complete the graph below to show how PWM could be used to make the motor rotate at half speed.

You should include at least three pulses.

2



- (b) Describe how the speed of this motor could be decreased using PWM.

1

---

---

---

---

4. (continued)

An alternative method of speed control involves varying the size of the DC voltage supplied to the motor.

- (c) Describe one advantage of using PWM in comparison to varying the size of the voltage supply.

1

---

---

---

## 2022 Past Paper

8. An anti-lock braking system is used to control the speed of an elevator as it descends. This uses a form of pulse-width modulation operated by a microcontroller.

If the speed of the elevator is too fast, the brakes will increase the proportion of operating time.

The table below identifies the connections to the microcontroller.

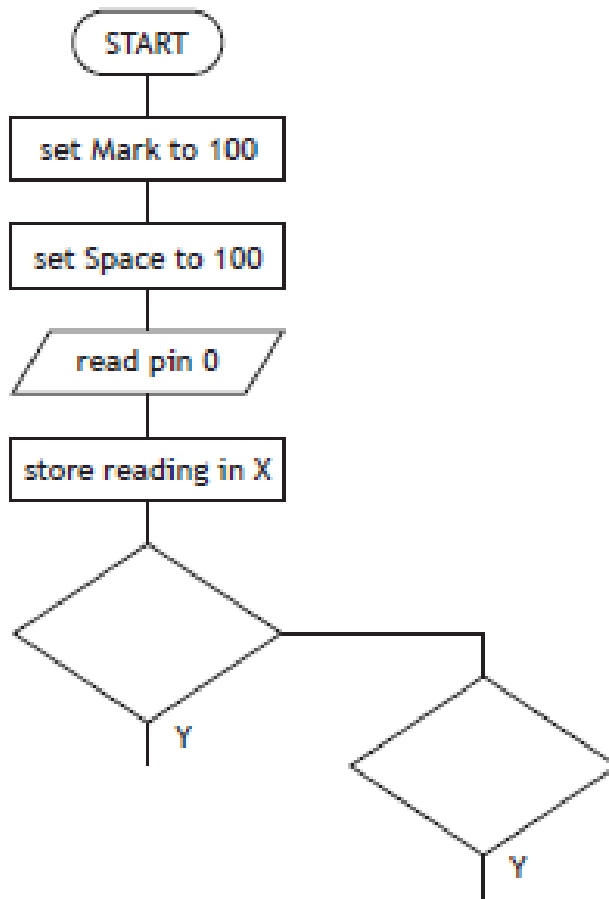
Input	Pin	Output
	7	brake
ground level sensor	1	
speed sensor (analogue)	0	

The system must perform the following steps.

- Values for mark and space both need to be initially set to a value of 100
- A reading must be taken from a speed sensor and its value stored in variable X
- If the value of X is greater than 128 then mark increases by 1 and space decreases by 1
- If the value of X is smaller than 128 then mark decreases by 1 and space increases by 1
- If the value of X is 128 then mark and space do not change
- The brake must be switched on and off for the times specified mark and space (this will be in milliseconds)
- The process must continue until the ground level sensor is activated

8. (continued)

- (a) Complete, with reference to the specification and the input/output table, this flow chart for the control of the system.



8. (continued)

An alarm (Z) is part of the elevator's operating system and needs to be activated under the conditions given by the following Boolean equation.

$$Z = \bar{A} \cdot (B + C)$$

(b) Complete the following flowchart to perform the function described above. 3

