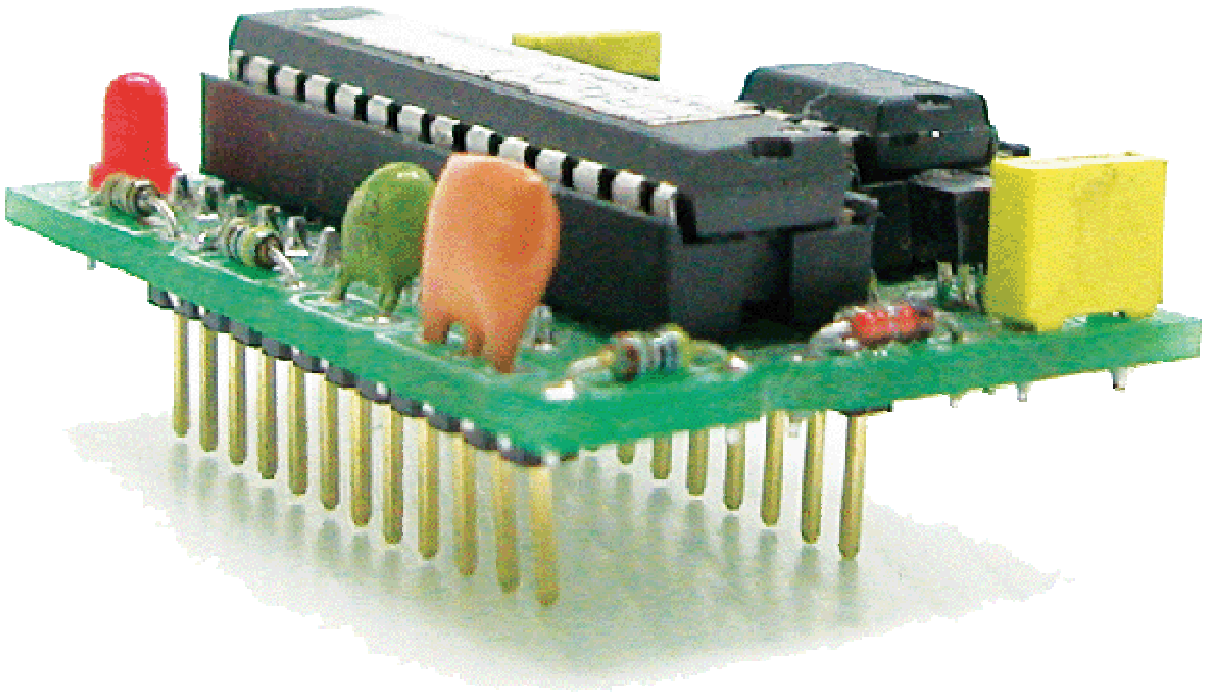
ENGINEERING SCIENCE

Programmable Control



Name	
Class	
Teacher	

Ellon Academy
Technical Faculty

Learning Intentions

- Gain the ability to design and evaluate solutions to engineering problems in a range of contexts
- I will gain knowledge and understanding of key concepts related to electronic and microcontroller-based systems, and their application
- To know what a microcontroller is and how its used
- I will know advantages and disadvantages of microcontroller based control systems compared to a hard-wired electronic equivalent

Success Criteria

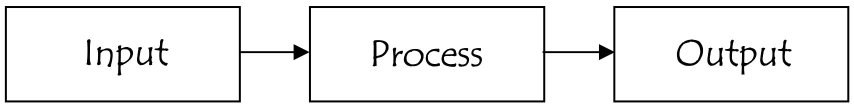
- I can explain what a microcontroller is, how it works and can give examples of where they are used
- I can state the advantages & disadvantages of using control systems in industry.
- I can explain the following terms; RAM, ROM, ALU, EEPROM, I/O port, bus and clock.
- I can use and understand binary.
- I can convert binary to decimal and vice versa.
- I can identify and use various input and output transducer devices.
- I can use correct symbols (start, stop, input, output, branch, loop) to construct flowcharts showing solutions to simple control programs, involving time delays, continuous and fixed loops
- I can use suitable commands, including high, low, for…next, if…then, pause, end (or their equivalents) to construct programs to solve simple control problems, involving time delays, continuous and fixed loops
- I can use the computer and stamp controller to test my programs

To access video clips that will help on this course
go to www.youtube.com/MacBeathsTech

Electronic Control Systems

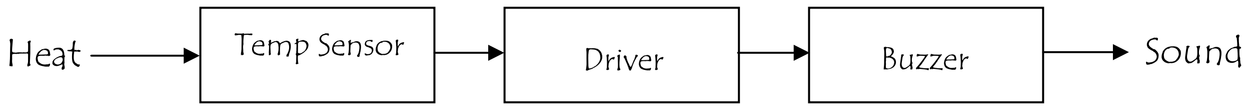
Many electronic devices have been developed to make life easier, safer, to help with work and for entertainment purposes.
Some devices are purely electronic devices (for example a digital watch) however many of these devices also control mechanisms (for example the eject mechanism in a video recorder) and so can be described as mechatronic devices.
Both electronic and mechatronic devices all have one thing in common they are all *electronic control systems.*

System diagram

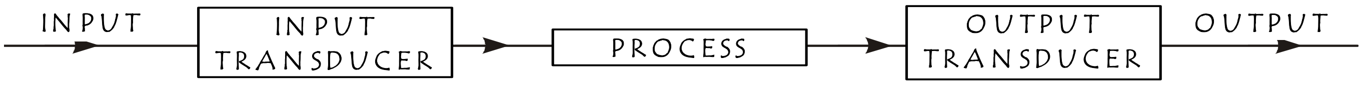


A system diagram is a more detailed block diagram that also shows the *real world input signal* (for example light and heat) and the *real world output signals (for* examples movement or sound).

Example - The system diagram for a warning device for a freezer in a restaurant would be drawn as shown below:



Or to put it another way:



Input transducers are electronic devices that detect changes in the 'real world' and send signals into the process block of the electronic system.

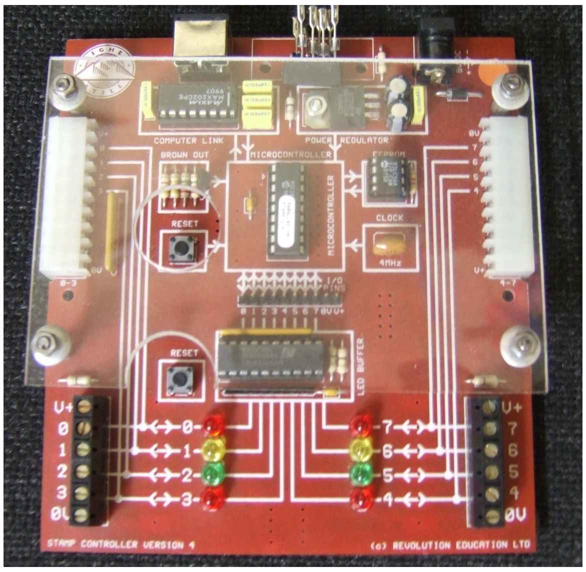
Output transducers are electronic devices that can be switched on and off by the process block of the electronic system.

The Stamp Controller

The 'Stamp' software runs on a computer and allows you to use the computer keyboard to type in programs.

The stamp controller 'runs' programs that have been downloaded to it. It has indicator L.E.D.s to show which inputs and outputs are off or on, and has connectors for the input and output modules.

The 'brain' of the stamp controller board is the 18-pin micro controller chip in the centre of the board.



Micro controller

A micro controller is often described as a *'computer on a chip'.*
Microcontrollers have a controller and memory all built into a single chip. As they are *small, inexpensive* they can easily be built into other devices to make these products more intelligent and easier to use.
Microcontrollers are usually programmed for a specific electronic product – for instance, a microwave oven may use a single microcontroller to process information for all its electronic devices.
By altering the microcontroller program the same 'brand' of chip can do many different tasks.

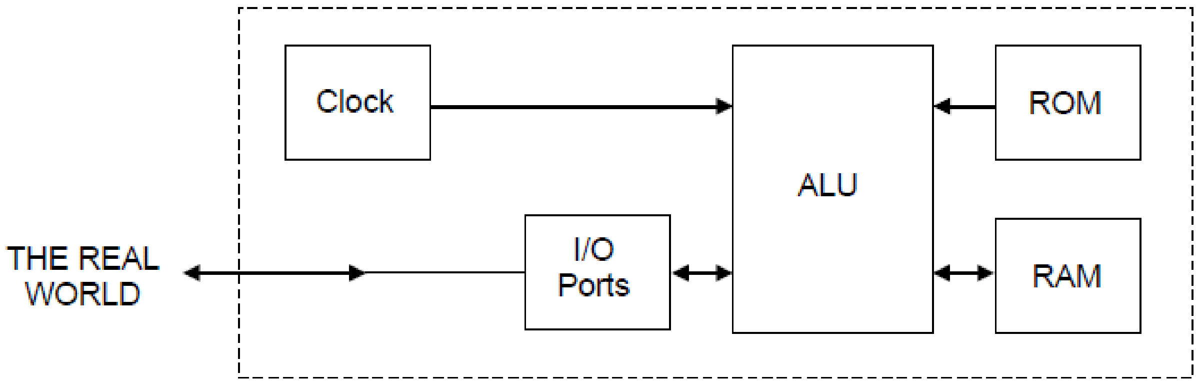
Advantages of using microcontrollers in a product design are:

- Increased reliability and reduced quantity of stock (as one microcontroller replaces several parts)
- Simplified product assembly and smaller end products
- Greater product flexibility and adaptability since features are programmed into the microcontroller and not built into the electronic hardware
- Rapid product changes or development by changing the program and not the electronic hardware

Inside a Microcontroller

The 'brain' of the stamp controller system is the 18 pin micro controller in the centre of the board. Although micro controllers are relatively cheap, they are very complex devices containing many thousands of transistors, resistors and other electronic components.

The main features of the micro controller are shown in the block diagram;



http://www.bbc.co.uk/schools/gcsebitesize/design/systemscontrol/electronicsrev3.shtml

ROM
The ROM (read only memory) contains the operating instructions for the micro controller. The ROM is 'programmed' before the micro controller is installed in the target system, and the memory retains the information even when the power is removed.

RAM
The Ram (random access memory) is 'temporary' memory used for storing information whilst the program is running. This is normally used to store mathematical answers that the micro controller comes out with as it is working. This memory is 'volatile', which means that as soon as the power is disconnected the contents of the memory are lost.

ALU
The processing unit (full name arithmetic and logic unit – ALU) is the 'control centre' of the micro controller. It operates by reading instructions from the ROM and then carrying out the mathematical operations for each instruction.

Clock
The clock circuit controls the speed at which these operations occur.
The *clock circuit* within the micro controller 'synchronises' all the internal blocks (ALU, ROM, RAM, etc) so that the whole system works correctly.

Buses
Information is carried between the various blocks of the micro controller along 'groups' of wires called *buses*. The 'data bus' carries data between ALU and RAM and the 'program bus' carries the program instructions from the ROM to the ALU.

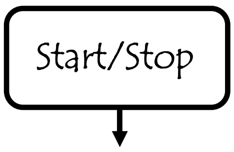
EEPROM
EEPROM (Electrically Erasable Programmable Read Only Memory) is a type of memory that can be reprogrammed when desired, but it also keeps the program when the power supply is removed. This means the stamp controller will start to run the program currently in the memory whenever the power supply is connected.

Flowcharts

Flowcharts are commonly used to explain how a program works. As flowcharts are drawn graphically they often make programs easier to understand. A flowchart should be drawn for each program you develop.

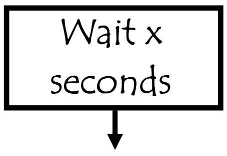
Start/stop symbol

The start/stop symbol shape is a rectangle with rounded ends. Each flow chart must contain one start and usually one stop symbol unless it is a continuous loop.



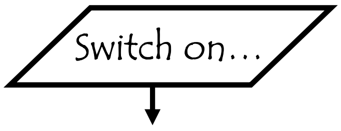
Wait symbol

The 'wait' symbol is a rectangle. The text inside the symbol explains how long the time delay is.



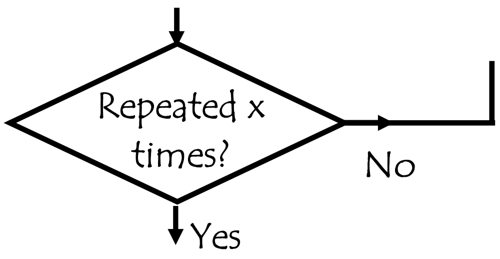
Outputs symbol

The 'outputs' symbol is a parallelogram. The text inside the symbol explains which output pins are switched on or off at any time.



Decision Box

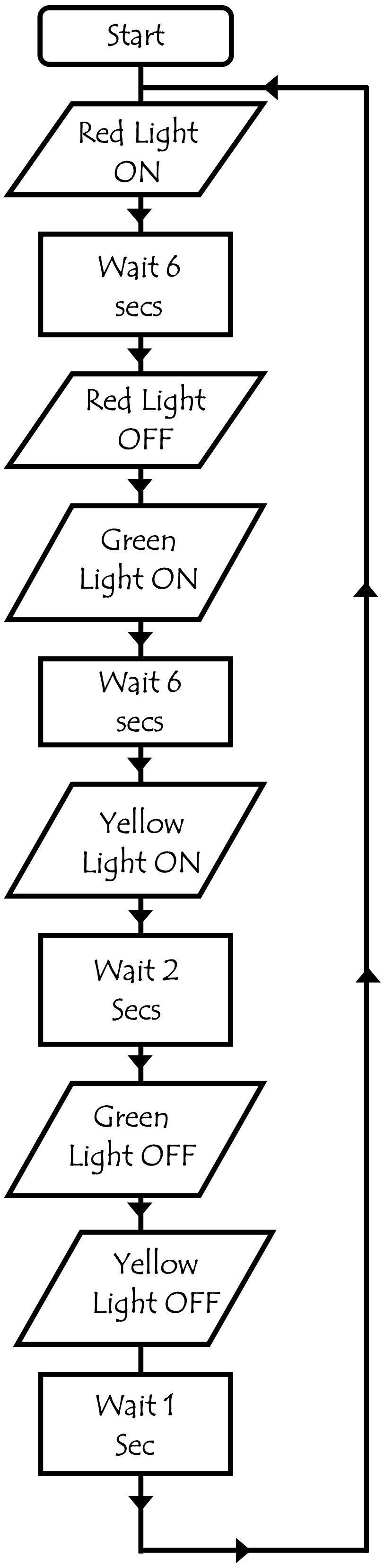
The Decision box is a diamond. The program uses this to check if something has been completed.



Continuous loop

Sometimes it is necessary to create programs that loop' forever', as is the case with traffic lights. There is no 'stop' symbol because the program never ends!

Example



Task 1

Write out the flowchart for making a cup of tea

Task 2

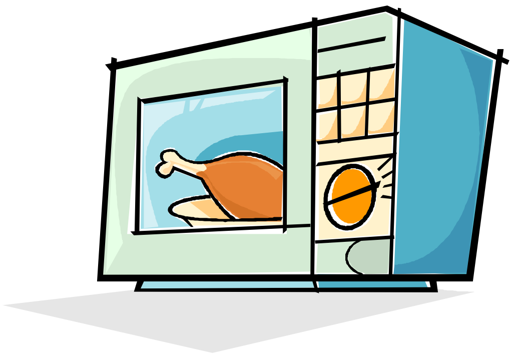
Set of temporary traffic lights is required for a system of road works.

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

Draw a flow chart for the lights sequence shown above by one set of traffic lights. Use the times given in the table.

Task 3

A microwave operates with the following sequence.



1.	Light on
2.	Turntable on
3.	Magnetron on
4.	Wait 30s
5.	Magnetron off
6.	Wait 10s
7.	Turntable off
8.	Buzzer on
9.	Wait 0.5s
10.	Buzzer off
11.	Light off

Draw a flowchart for this sequence.

Writing a programme

Once a flowchart has been drawn it is necessary to convert it into the stamp controller programming language, which is called PBASIC
A PBASIC program for the flowchart shown on the previous page is:

```
main:
        high 7                          ' switch pin 7 (red) on
        pause 6000                      ' wait for 6seconds
        low 7                           ' switch pin 7(red) off
        high 5                          'switch pin 59(green) on
        pause 6000                      ' wait for 6 seconds
        high 6                          ' switch pin 6(yellow) on
        pause 2000                      ' wait for 2 seconds
        low 5                           ' switch pin 5(green) off
        low 6                           ' switch pin 6 (yellow) off
        goto main                       'return to start

        end                             ' end the program
```

An explanation after the apostrophe (') symbol makes each line of a PBASIC program much easier to understand. These comments are ignored by the computer when it downloads a program to the stamp controller.
A label (for example 'main:' in the program above) can be any word (apart from keywords such as 'high'), but it must begin with a letter. When the label is first defined it must end with a colon (:). The colon 'tells' the computer that the word is a new label.

Task 4
Key in the program above into PICAXE and test.
Would this program be suitable for a traffic light system?
If not how could you improve it?

Test your new design in PICAXE

Symbols and Labels

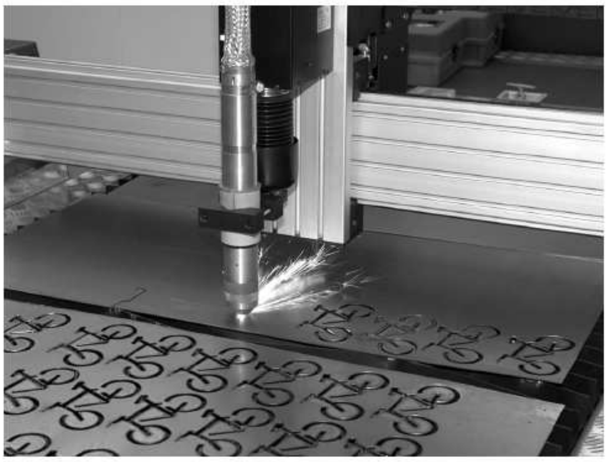
When writing the computer programs for our flowcharts it can sometimes be hard to remember which pins are connected to which devices. The 'symbol' command can then be used at the start of a program to *rename the inputs and outputs.*
In a normal program there will be a 'main program', which includes many 'mini programs'.
Mini programs' or sub procedures are often used to separate the program into small sections to make it easier to understand. Sub-procedures that complete common tasks can also be copied from program to program to save time.

Example

```
symbol red = ouput 7                    'rename output 7 red
symbol buzzer = output6                 'rename output 6 buzzer
symbol counter = b0                     'define a counter using variable b0

main:
        gosub flash                     'make label called main
        gosub noise                     'call the sub-procedure flash
        goto main                       'loop back

        end                             'end of the main program

flash:                                  'make a sub-procedure called flash
        for counter = 1 to 25           'start a for ....next loop
        switch on red                   'red LED on
        pause 50                        'wait 0.05 seconds
        switch off red                  'red LED off
        pause 50                        'wait 0.05 seconds
        next counter                    'next loop
        return                          'return from the sub-procedure

noise:                                  'make a sub-procedure called noise
        switch on buzzer                'buzzer on
        wait 2                          'wait 2 seconds
        switch off buzzer               'buzzer off
        return                          'return from the sub procedure
```

Task 5

A laser cutting machine is used to cut sheet steel, as shown in the photograph below.



The laser is positioned by motors A and B which are operated by a microcontroller.

Input and output connections to the microcontroller are shown in the table below.

Input connection	Pin	Output connection
	7	Motor A forward (move right)
	6	Motor A reverse (move left)
	5	Motor B forward (move forward)
	4	Motor B reverse (move back)
	3	Laser on
	2	
	1	
	0	

The cutter is required to perform the following sequence of operations:

1: Switch on the laser

2: Move right for 0·5 seconds

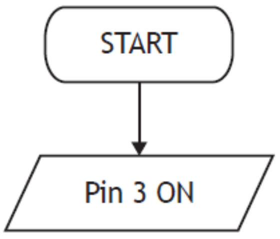
3: Move forward for 0·5 seconds

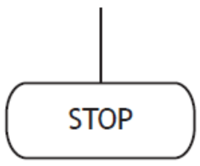
4: Repeat steps 2 and 3 four times

5: Switch off laser and motors

Task 5 (Continued)

(a) Complete the system flowchart below, to produce the required sequence of operations. The flowchart must include appropriate pin numbers. You may use information from the National 4/5 Data Booklet provided.

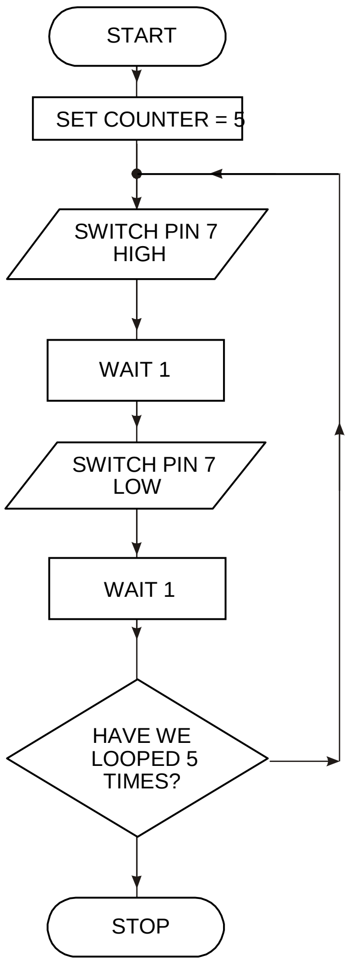




For … next' loop

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 'for … next' loop can be used.

In this flowchart the LED connected to output pin 7 is flashed on and off five times. The number of times the code has been repeated is stored – that is, stored in the RAM memory of the stamp controller. There are 10 available variables, labelled b0 to b9, which can be used in this way. These variables can also be renamed using the *symbol* command to make them easier to remember.



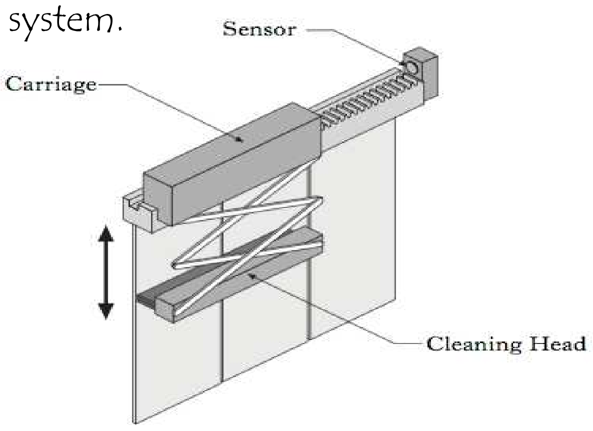
Task 6

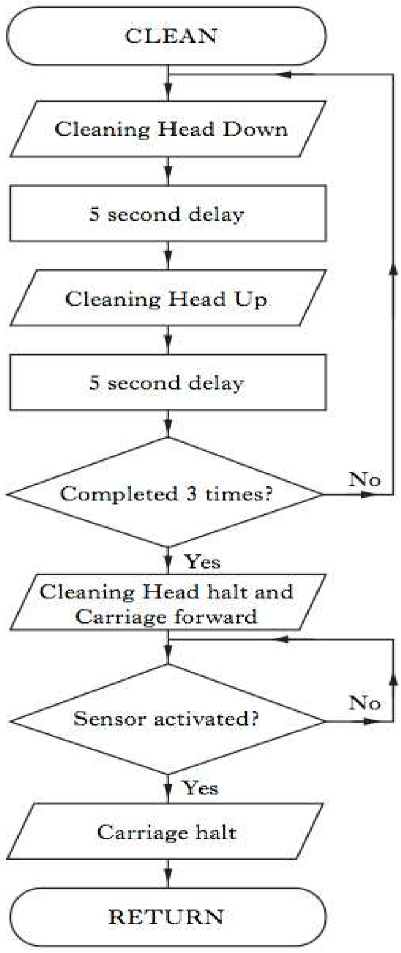
Key in, download and run the following program.

```
symbol counter = b0      ' define the variable 'counter'
symbol red = 7           ' define pin 7 with the name 'red'

main: for counter = 1 to 5      ' start a for … next loop
          high red              ' switch pin 7 high
          pause 1000            ' wait for 1 second
          low red               ' switch pin 7 low
          pause 1000            ' wait for 1 second
      next counter              ' end of for … next loop

      end                       ' end program
```

Task 7

A microcontroller is used to operate an automatic window cleaning system.





Input and output connections to the microcontroller are shown in the table below.

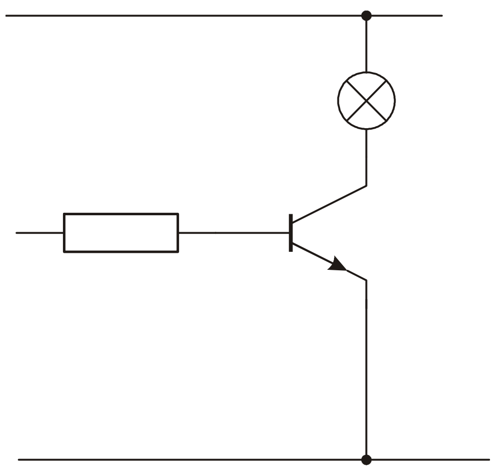
Input Connection	Pin	Output Connection
	7	Cleaning Head Up
	6	Cleaning Head Down
	5	Carriage forward
	4	
	3	
	2	
	1	
Sensor	0	

Task 7 (continued)

Write a program to control the window cleaner.

Connecting output transducers

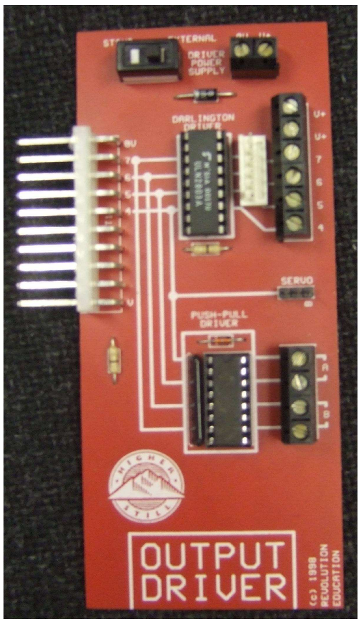
The stamp controller can only drive low-power devices, such as LEDs, directly. It cannot drive devices such as lamps, buzzers, solenoids or motors directly because these devices require a higher current to operate.



A common way to drive these devices is with a transistor, as shown in the diagram above. In this case the lamp is controlled by the transistor switching on and off.

The Output Driver

The output driver module provides four transistor outputs, as in the circuit shown above. Instead of using four separate transistors, the output driver uses an integrated circuit called the ULN2803A, which contains all the transistors in one 18-pin 'chip'.

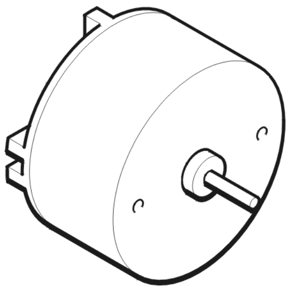


To use the transistor outputs, the output device should be connected between the screw-terminal numbered output (4–7) and a V+ connection. The positive (red) wire on polarised devices (for example a buzzer) should be connected to the V+ connection.

The white 6-pin header beside the screw terminals allows a 'stepper motor' or control model to be connected easily to all four of the outputs.

The Motor Driver

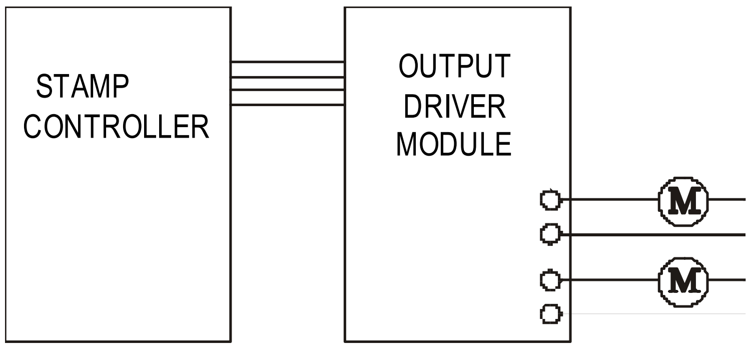
The output driver module also contains a second integrated circuit called the L293D push–pull driver. This chip allows forward and reverse control of two d.c. motors. Each motor output uses two of the stamp controller output pins to control the direction of rotation of the motor.



Pin 4	Pin 5	Motor A
off	off	halt
off	on	forwards
on	off	backwards
on	on	halt

Pin 6	Pin 7	Motor B
off	off	halt
off	on	forwards
on	off	backwards
on	on	halt

To use the push–pull motor output, the motor should be connected between the screw terminals labelled A or B. If the motor turns in the opposite direction to that expected, the two motor wires should be swapped over.

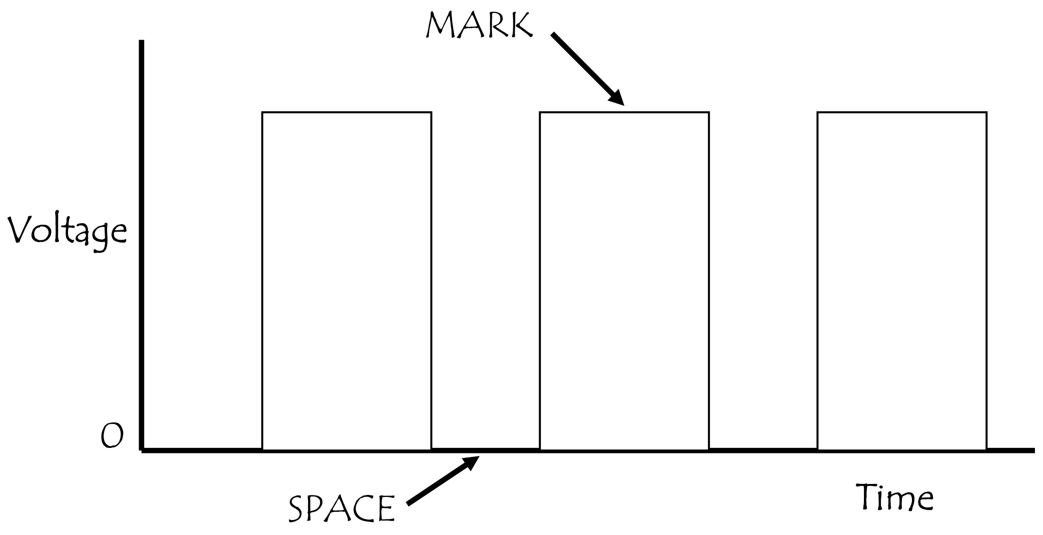


Speed Control of D.C. Motors

There are two ways to control the speed of a D.C. motor. The simplest is to vary the voltage applied to the motor. If, for instance, 3 V is applied to a small D.C. motor it will rotate at a lower speed than if 5 V were applied. Unfortunately the 'turning power' (torque) of the motor will also drop, which means the whole motor system will be less powerful.

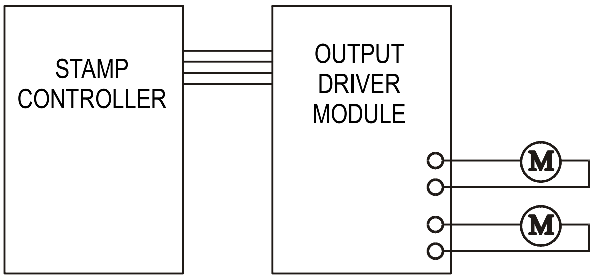
The second way to control the motor is to always apply the full voltage (for example
5 V) across the motor, but then to switch the power supply on and off rapidly. As the power supply is off some of the time, the motor does not receive as much power and so the motor turns more slowly. The advantage of this system is that the torque remains quite high.

This system is called pulse-width modulation (PWM). 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 on (mark)-to-off (space) ratio, the speed of the motor can be varied.



Task 8

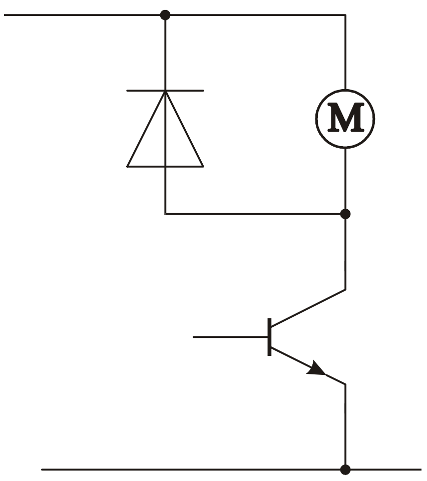
Connect 2 motors to the output driver module as shown in the diagram.



The 'high' and 'low' commands can be used to switch the output pins and control the motors *forward*, *backward* and *halt*.

```
main:                       ' make a label called 'main'
        high 5              ' motor A forward
        high 7              ' motor B forward
        pause 1000          ' wait 1 second
        low 5               ' motor A halt
        low 7               ' motor B halt
        pause 1000          ' wait 1 second
        high 4              ' motor A backward
        high 6              ' motor B backward
        pause 1000          ' wait 1 second
        low 4               ' motor A halt
        low 6               ' motor B halt
        goto main           ' jump back to the start
```

Key-in the program listed above, and then download it to the stamp controller. The motors should start moving.



Task 9
Connect a d.c. solar motor across the 'V+' and '7' terminals on the output driver module. This will provide an interfacing circuit as shown.

a) Key in, download and run the program listed below. This program drives the motor at approximately half speed, as the space (off time) is twice the length of the mark (on time). Note that you must use pause delays, as the power supply must switch on and off very quickly (wait delays would be too long).

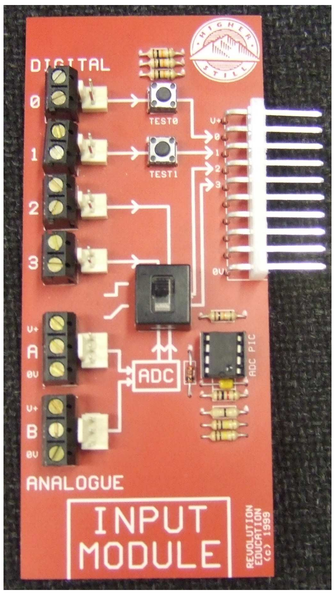
```
main:                               ' make a label called 'main'
        high 7                      ' output high
        pause 5                     ' wait 5 ms
        low  7                      ' output low
        pause 10                    ' wait 10 ms
        goto main                   ' jump back to the start
```

b) Try out different speeds (by experimenting) by altering the length of the pause delays.

c) Describe the advantages and disadvantages of using PWM speed control.

The Input Module

The input module provides the interfacing circuits required to connect switches and sensors to the stamp controller.

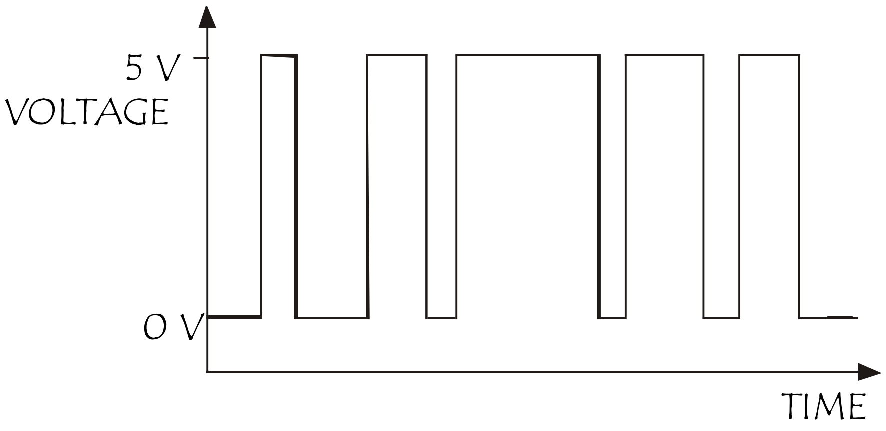


When the slide switch on the input module is 'up' the input module provides four digital (on/off) switch connections. These can be used to connect input switches (for example a microswitch) to the stamp controller. The switches can be connected through the screw-terminal blocks or through the white push-on headers.

Input pins 0 and 1 have 'on-board' test switches. These allow programs to be tested without the need to connect external switches.

Digital sensors

A digital sensor is a simple 'switch' type sensor that can only be 'on' or 'off'.



Common examples of a digital sensor are:

- microswitches
- push-and-rocker switches
- reed switches.

<u>Task 10</u>

Key in, download and run the program listed below. This program makes output pin 7 flash every time the push-switch on input pin 0 is pushed.

```
main:                               ' make a label called 'main'
        if pin0 =1 then flash       ' jump if the input is on
        goto main                   ' else loop back around

flash:                              ' make a label called 'flash'
        high 7                      ' switch output 7 on
        pause 2000                  ' wait 2 seconds
        low 7                       ' switch output 7 off
        goto main                   ' jump back to start
```

In this program the first three lines make up a continuous loop. If the input is off the program just loops around time and time again.

If the switch is then pushed, the program jumps to the label called 'flash'. The program then switches pin 7 on for two seconds before returning to the main loop.

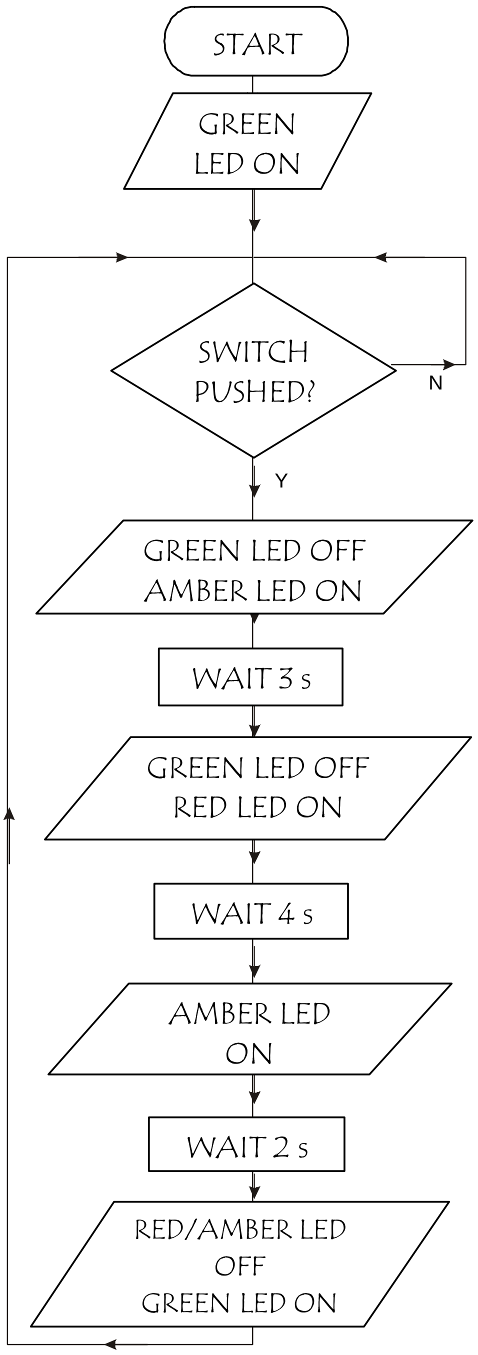
Note carefully the spelling in the 'if … then' line – 'pin0' is all one word (without a space). Note also that only the label is placed after the command 'then' – no other words apart from a label are allowed.

http://www.youtube.com/watch?v=0AItUj7Ctng

Task 11

Develop a PBASIC program that will carry out the instructions shown in the flowchart above. Use the following pin configuration.

Input connection	Pin	Output connection
	7	red light
	6	amber light
	5	green light
	4	
	3	
	2	
	1	
start switch	0	



http://www.youtube.com/watch?v=yF7yKUoGLQk

Task 12

A music venue has a system to cut off the power supply if a band plays too loudly.

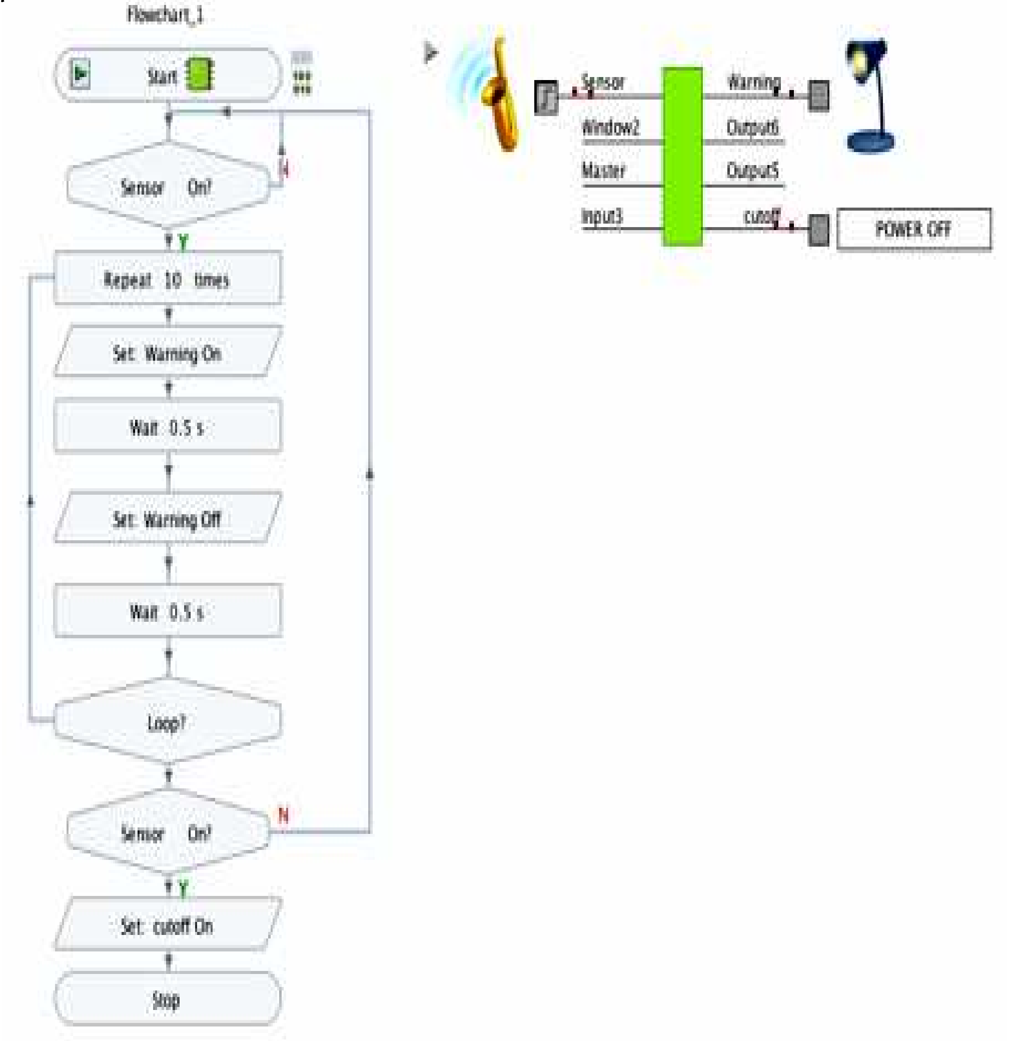
The system is operated by a microcontroller.

Specification

The system must:

- switch on a warning light if the sound level is too high
- flash the light 10 times (on for half a second then off for half a second)
- reset the system if the sound level is then ok
- cut off the power if the sound level is still too high.

Load the following simulation into Yenka to see if it does what is expected.



Task 12 (Continued)

a) Describe the tests you carried out and how the circuit actually performed.

b) Compare the circuit's performance to the specification. Draw meaningful conclusions and where appropriate suggest possible improvements or further work.

Task 12 (Continued)

c) Write the program for this circuit below.

d) Take a photograph of your completed circuit and glue below