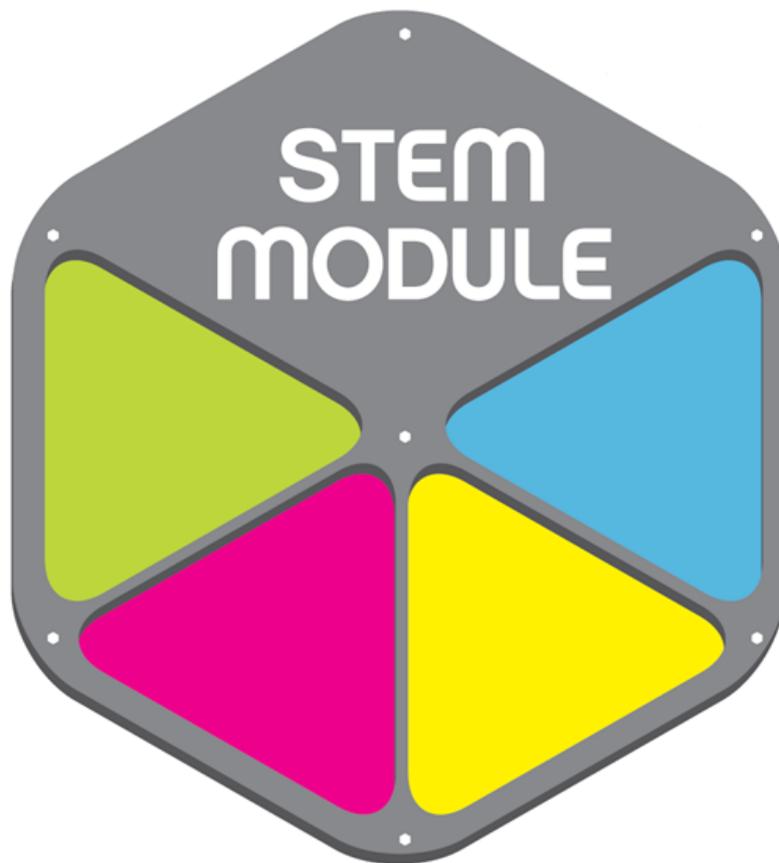


Introduction to PIC programming with Logicator



Teaching Resource Booklet 2010/2011

Project funded by:



Project managed by:



Introduction to PICAXE microcontrollers

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 micro controller 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).

What is a PIC microcontroller?

PIC is a family of Harvard architecture microcontrollers made by microchip technology, derived from the PIC1640 originally developed by General Instruments Microelectronics Division. The name PIC was used to refer to “programmable interface controllers”.

These PIC microcontrollers quickly became popular with industrial developers, educational establishments and hobbyists alike due to their low costs, wide availability, large user base and the availability of low cost or free development tools.

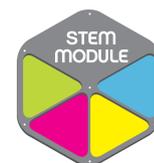
The PICAXE system then came about as a development of these original microcontroller chips by UK based company Revolution Education. There are 13 PICAXE variants of differing PIN counts from 8 to 40 PIN. This system was initially marketed for use in education and by electronic hobbyists. However is also used in commercial technical fields, including rapid prototype development.

The unique feature of the PICAXE system is that it uses a new generation of low cost “flash” memory based microcontrollers. These microcontrollers can be programmed and reprogrammed upwards to 1000, 000 times without the need for an expensive programmer.

This is because the PICAXE chips are pre-programmed with a bootstrap code which allows the microcontroller to be programmed directly via a simple serial connection.

These programs can be created in one of two ways graphically using flowcharts or programmatically using a simple BASIC style language. This can be achieved either using free software provided by Revolution Education called “Programming Editor” or with other specialized software such as Logicator.

Additional benefits are that as the PICAXE is a “chip” level device, students can design their own PCBs to build into their own PIC controlled projects.



Logicator Flowchart Exercises

In the following exercises you will learn the basics of PIC programming using Logicator for PIC and PICAXE software.

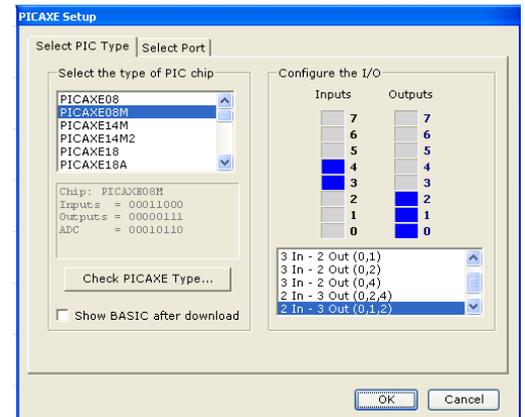
Exercise one

Step 1:

Open Logicator for PIC and PICAXE

Step 2:

In the first two simple exercises you will be programming a simulated circuit within the software package. You must first select the PIC type and configuration.

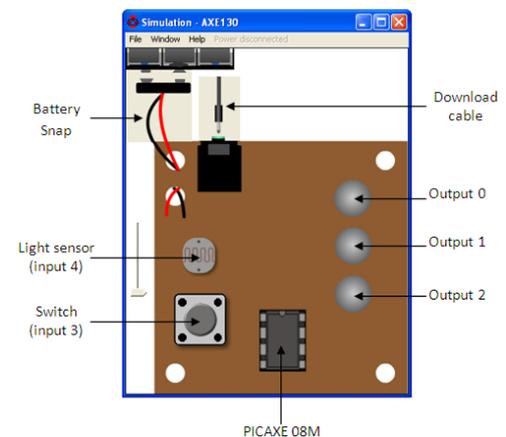


To do this first choose options on the top menu, then PIC type. Then set the PIC type as PICAXE 08M and the configuration I/O as 2in3 out (0, 1, 2).

Step 3:

Now to open the PIC simulation first choose simulate on the top menu, then select open simulation > PICAXE kits > AXE 130 simple PIC

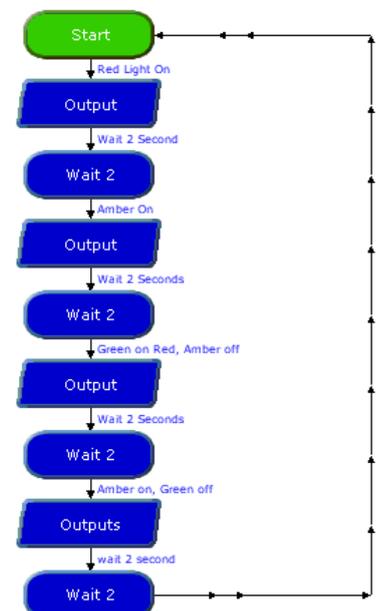
The simulation is configured as shown with the LED outputs displaying in red, amber and green like on a traffic lights system.



Step 4:

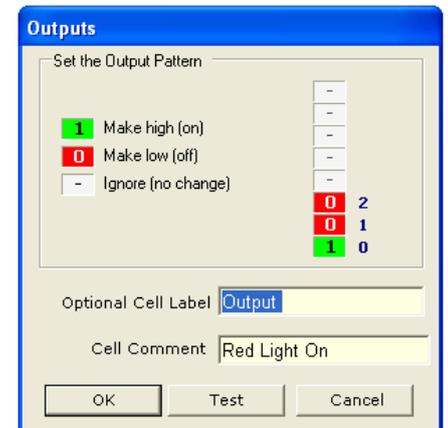
Now you can begin to construct a flowchart program using the commands in the command list on the right hand side of the screen. Simply move the commands onto the screen by clicking on them and dragging them across and placing them where you want them.

Start by dragging in 4 output and 4 wait commands and setting them out as shown. Then use the line tool (pencil) in the top command menu to connect the cells up as shown click first on the starting command and then the destination point to draw your line in the correct direction.



With each command you have the option to set the outcomes, label the cell and add a comment. To do this you must understand the purpose of the circuit and program, in this case the program is controlling a simple traffic light circuit to turn the lights on in sequence Red > Red +Amber > Green > Amber.

So the first output cell must turn the Red light on, to set this cell first double click on it. This will display a setting menu. The numbers on the far right of the menu in this case 0, 1, 2 indicate the available output pins on your chosen PIC chip. Each pin can be set in one of three ways high (on), Low (off) or ignore (leave in current state). As the Red light is pin 0 set it to 1 for on set the others as 0 for off. Label the cell as output and the comment as red light on.



Now set the other three output cells to carry out their intended function and label them also.

As well as the outputs you must also set wait commands to allow a pause between each change. Again double click on the cell and set the time to 2 and set the cell comment as wait 2 seconds.

Set the other three wait cells the same.

Step 5:

Once the program flowchart is completed you can then download the program to the simulated chip.

The first thing to do is ensure the download cable and battery snap are connected in the simulation by clicking on them. Then select the program PIC command (jack plug) on the tool bar at the top of the screen.



Now watch the simulation of the programme you created.

Results:

A traffic light sequence that is on a continuous loop

Problems:

No user input.



Exercise two

In the last exercise you created a programme to control three LEDs configured like traffic lights. However the problem with this initial program was it had no user input, so in this exercise you will adapt your program to include a switch.

Step 1: If you look in the commands list on the right hand side of the screen you will see that there is no command for either input or switch this is because a switch is considered to be a decision in terms of programming. So click and drag a decision cell on to the screen.

Now place the decision cell in-between the first output and wait command cells and connect it first to the wait cell below and then back to the start cell. You will see that it adds two labels to the lines Yes and No.

Step 2:

Now that you have placed the decision cell in position you must configure it to the correct input. First double click on the cell, then as before a setup menu will appear. Set the input pattern so that pin 3 must be on by placing a 1 in the box.

This means that the program flowchart will only continue past the decision cell if the switch is pressed, if not it will loop back to the start and check again if the switch is pressed. Once the switch is pressed then the program can continue as normal until the start of the next loop.

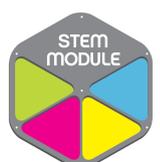
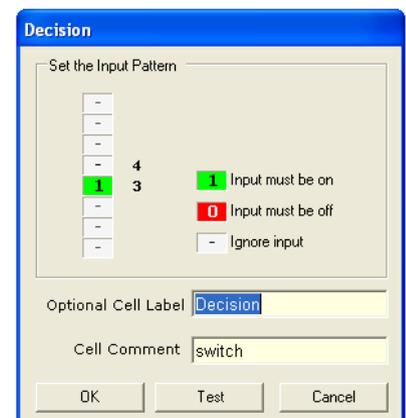
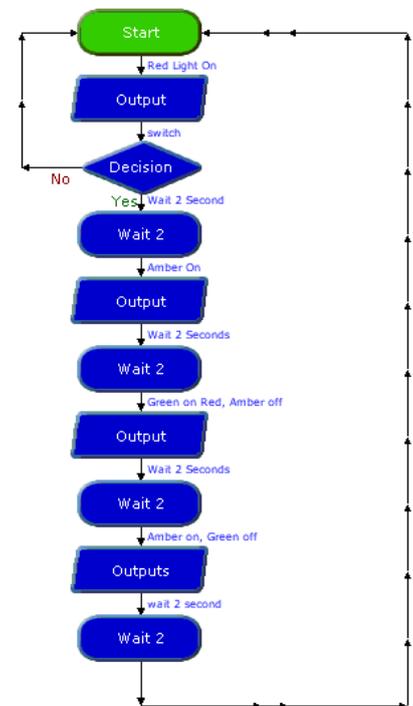
Step 3:

Once you have completed the changes to the program flowchart you can download the program to the simulated chip again.

First ensure the download cable and battery snap are connected in the simulation. Then select the program PIC command (jack plug) on the tool bar at the top of the screen.



Now watch the simulation of the programme you created.

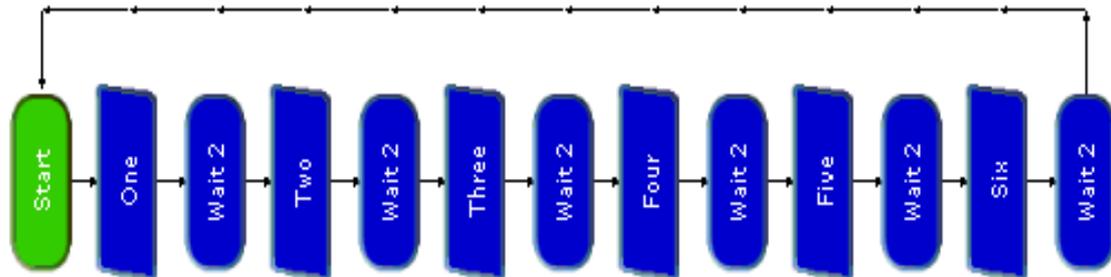


Programming the Electronic Dice

In the following exercises you will learn two different methods of programming the electronic dice project using Logicator for PIC and PICAXE software.

Method One:

Method one uses the following program:



Step 1:

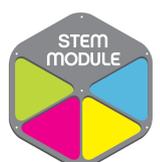
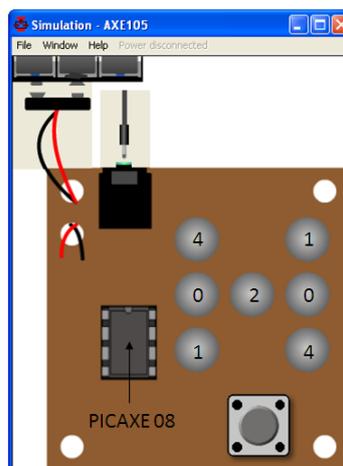
Start as before by selecting the required PIC type and configuration.

To do this first choose options on the top menu, then PIC type, then set the PIC type as PICAXE 08 and the configuration I/O as 1in4 out.

Step 2:

Now to open the PIC simulation first choose simulate on the top menu, then select open, simulation > PICAXE kits > AXE 105 dice.

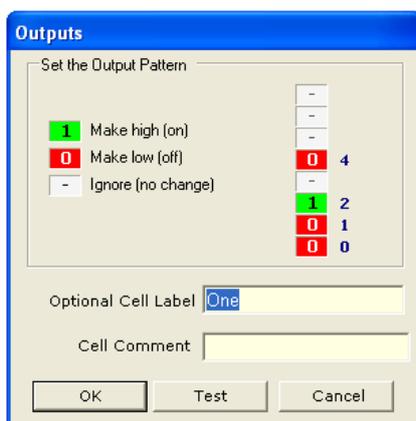
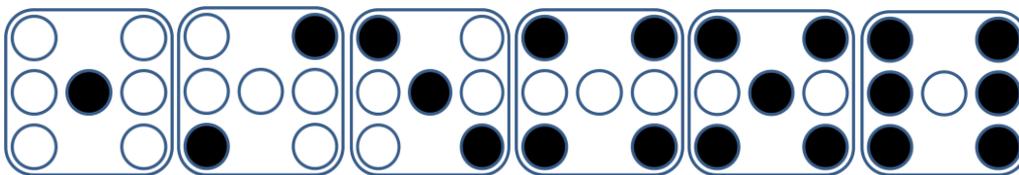
The simulation is configured as shown with the 7 LED outputs configured in 4 groups. Each group is attached to one pin 1, 2, 3, or 4.



Step 3:

Now you can begin to construct the flowchart program using the commands in the command list on the right hand side of the screen. Simply move the commands onto the screen by clicking on them, dragging them across and placing them where you want them.

Start by dragging in 6 output and 6 wait commands and setting them out as shown. Then use the line tool (pencil) in the top command menu to connect the cells up as shown click first on the starting command and then the destination point to draw your line in the correct direction.



In order to set the output cells simple look at the above diagram and decide which outputs need to be turned on in order to display the number as it would be seen on a standard gaming dice.

Here you can see the output set to show the number one with only the centre LED output turned on. Work out the rest yourself.

Set the wait commands as before.

Step 4:

Once the program flowchart is completed you can then down load the program to the chip and check the results.

The first thing to do is ensure the download cable and battery snap are connected. Then select the program PIC command (jack plug) on the tool bar at the top of the screen.



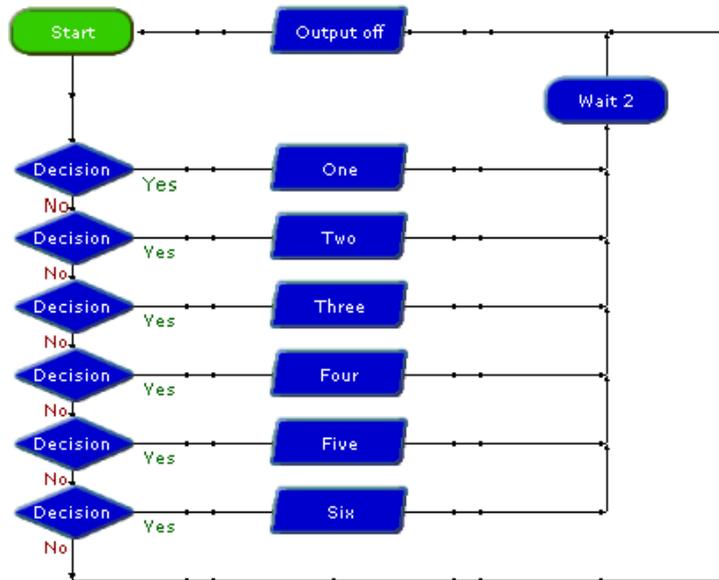
The Results:

If your program is working correctly you will see the LEDS change to show the number one to six in sequence and then repeat.



Method two:

Method two uses the following program:



Start by dragging in 7 output, 1 wait and 6 decision commands and setting them out as shown. Then use the line tool (pencil) in the top command menu to connect the cells up as shown click first on the starting command and then the destination point to draw your line in the correct direction..

Once the command cells are in place configure the settings and label the cells as shown. Ensure that the output cells are set to turn on all the LEDs required to give each number as in the previous exercise.

Results:

When you click the switch (input device) a 'random' number will be displayed.

This is just the beginning there are many other possible ways to program your electronic dice experiment yourself with different programmes and see how well you can make it work.

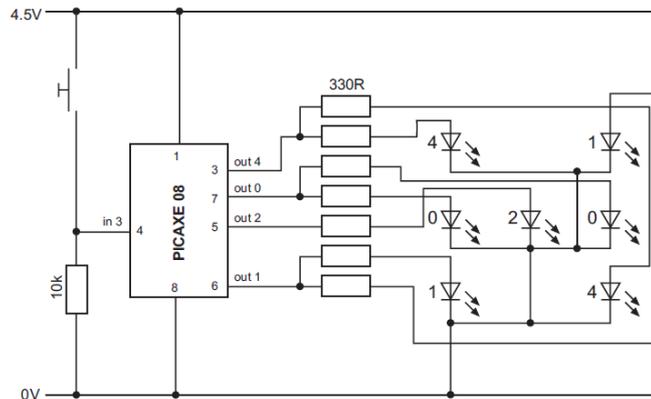


Build your own Electronic Dice

Brief:

Build your own electronic dice which can then be operated by a PICAXE microcontroller chip that you will learn to programme on the STEM Module.

Circuit Diagram:

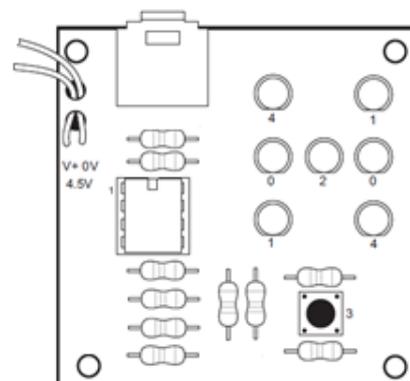


Component list:

R1 to 7	330R resistor (orange orange brown gold)
R8	22k resistor (red red orange gold)
R9, 10	10k resistor (brown black orange gold)
LED1 to 7	5mm LEDs
SW1	push switch
IC1	8 pin IC socket
IC1	PICAXE-08 microcontroller
CT1	PICAXE download 3.5mm socket
BT1	4.5V (3xAA) battery box
PCB	printed circuit board
(Single core wire to connect LEDs and battery box, if required)	

Tools:

Soldering iron and solder
Side cutters

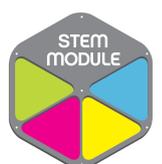


Steps to soldering:

- 1) Ensure the soldering iron is hot by checking it will quickly melt solder.
- 2) Clean the soldering iron tip on the damp sponge.
- 3) Press the soldering iron tip against the pad on the PCB AND the leg of the component. Count to 3 to give the joint time to warm up.
- 4) Keep the soldering iron in position and touch the solder against the joint. Allow enough solder to melt to cover the joint.
- 5) Take the solder away first, then the soldering iron
- 6) Always start with the smallest components, move onto larger components and finish with the tall components like capacitors and transistors.

Step by Step guide:

- 1) Place the 22k (red red orange gold) resistor and the two 10k (brown black orange gold) resistors in position. Bend the legs to hold the resistors in position and then solder.
- 2) Place the seven 330R (orange orange brown gold) resistors in position and solder. This may be easier if the resistors are positioned in two groups. Bend the legs to hold the resistors in position and then solder.
- 3) Push the PICAXE stereo download socket onto the PCB and **make sure it clicks into position**. Then solder the five metal square contacts.
- 4) Push the IC socket into position. Make sure the notch at one end points towards the top. Fold the legs over to hold the socket in position and then solder.
- 5) Solder the push switch in position. The switch only fits one way around.
- 6) Solder the LEDs into position. Make sure that the flat on one side of the LED aligns with the flat marked on the PCB
- 6) Thread the battery clip wires into place through the holes connect the black wire to the hole marked 0v and the red wire into the hole marked V+
- 7) Insert the microcontroller into the socket, ensuring pin1 faces the resistors.



Testing your circuit

Step 1 – Check the solder joints:

Check that the solder does not accidentally bridge between two pads. This is most likely to happen on the LEDs. On the stereo socket the two square pads close together on each side can be joined as they are already joined by a track on the board. However they must not be joined to the central round hole.

Step 2 - Check the components:

- 1) Check that the black battery clip wire is in the hole marked '0V' and the red battery clip wire is in the hole marked 'V+'
- 2) Check that the PICAXE-08 chip is in the socket correctly, with the dent (showing pin1) closest to the stereo socket.
- 3) Check that the flat edge of the LEDs is connected to the correct hole on the PCB.
- 4) Check that the stereo socket is correctly soldered, including the middle square pad which is often forgotten by mistake.

Step 3 - Connect the battery

Check the 3 AA batteries are in the battery box correctly. Connect to the battery snap and then put your finger on the PICAXE chip. If it starts to get hot remove the battery box immediately as there is a problem – most likely that the chip or the battery wires are around the wrong way.

