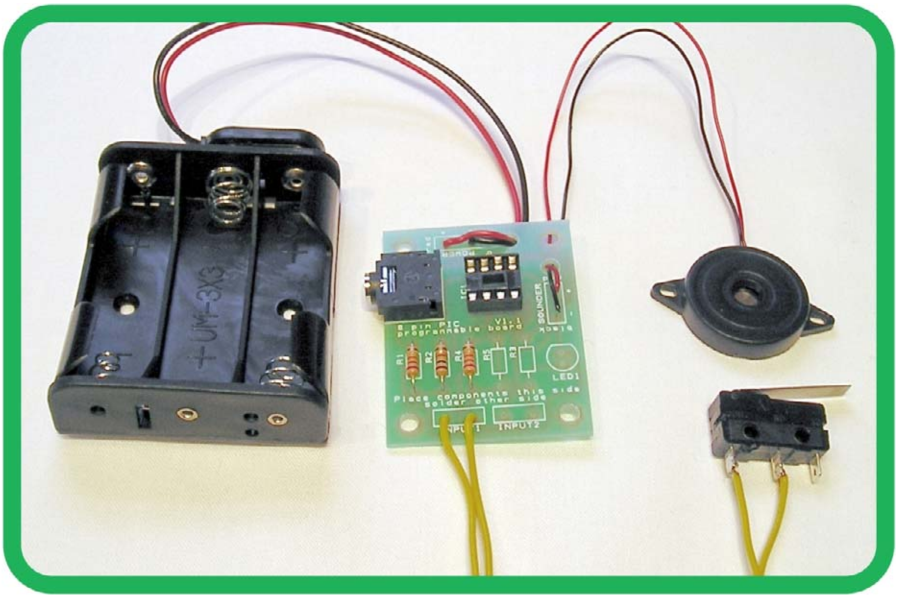
Kitronik

Programmable music box

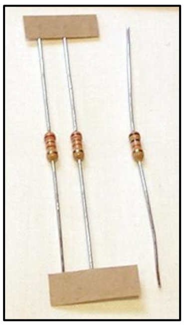


Build Instructions

Issue 1.2

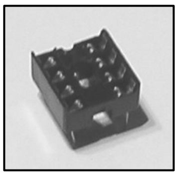
Build Instructions

Before you put any components in the board or pick up the soldering iron, just take a look at the Printed Circuit Board (PCB). The components go in the side with the writing on and the solder goes on the side with the tracks and silver pads. You will find it easiest to start with the small components and work up to the taller larger ones. If you've not soldered before get your soldering checked after you have done the first few joints.

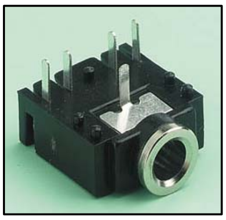


Step 1
Start with the three small resistors (shown right):
R1& R4 are 22KΩ (Red, Red, Orange coloured bands).
R2 is a 10KΩ (Brown, Black, Orange coloured bands).

The text on the board shows where R1, R2, etc go. Make sure that you put the resistors in the right place.



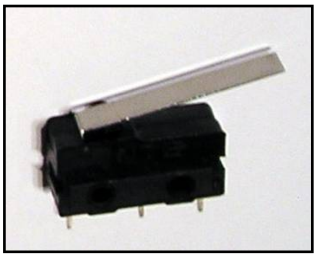
Step 2
Solder the Integrated Circuit (IC) holder (shown left) in to IC1. When putting this into the board, be sure to get it the right way around. The notch on the IC holder should line up with the notch on the lines marked on the PCB.



Step 3
Solder the programming connector (shown right) into the board where it is labeled 'PROG'.

Step 4
The battery clip (shown left) should be soldered into the 'POWER' terminal. First start by feeding the wire through the strain relief hole (feed from the solder side). The red wire must go to the '+' terminal (also marked 'red') and the black wire must go to the '–' terminal (also marked 'black').

Step 5
The buzzer (shown right) should be soldered into the 'SOUNDER' terminal. First start by feeding the wire through the strain relief hole (feed from the solder side). The red wire must go to the '+' terminal (also marked 'red') and the black wire must go to the '–' terminal (also marked 'black').



Step 6
Cut and strip two pieces of wire to the required length for connecting to the switch (shown left). Solder the end of one piece of wire to the terminal of the switch labeled 'COM' (this is printed on the plastic body of the switch). Then solder one end of the other piece of wire to the terminal of the switch labeled 'NO'. Now solder the other ends of the wires to the terminals labeled 'INPUT1' on the PCB. It does not matter which wire goes to which terminal on the PCB.

Step 7
The IC can be put into the holder ensuring the notch on the chip lines up with the notch on the holder.

Checking Your PCB

Check the following before you insert the batteries:

Check the bottom of the board to ensure that:

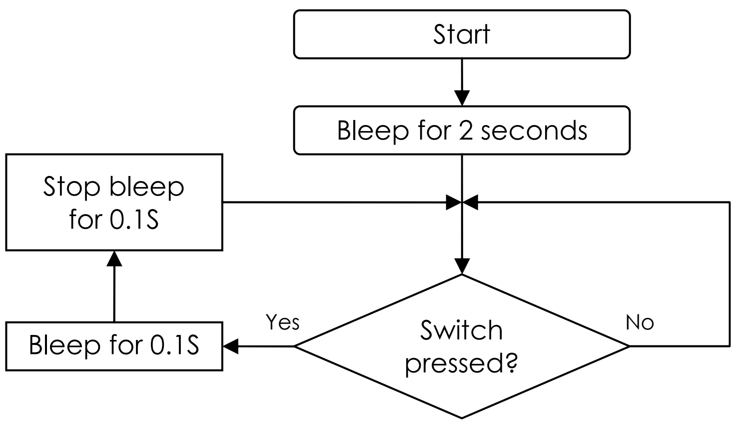
- All the pads with wires in them are properly soldered.
- Pins next to each other are not soldered together.

Check the top of the board to ensure that:

- Nothing is fitted in INPUT2, LED1, R3 and R5, but all the other outlines have parts in them.
- The notch on the IC and the IC holder are in the same orientation as the markings on the printed circuit board.
- R2 has Brown, Black, Orange coloured bands.
- R1 and R4 have Red, Red, Orange coloured bands.
- The red wire on the battery connector goes to the + terminal on the power terminals and the black wire goes to the – terminal.

Testing the PCB

The circuit has been designed to allow easy testing of the PCB. To test the PCB you will first have to insert a chip programmed with some software to allow it to be tested. The purpose of the test program is to test that the buzzer can sound and that the switch is functioning (used to turn the buzzer pattern on and off). The test software works as shown below.



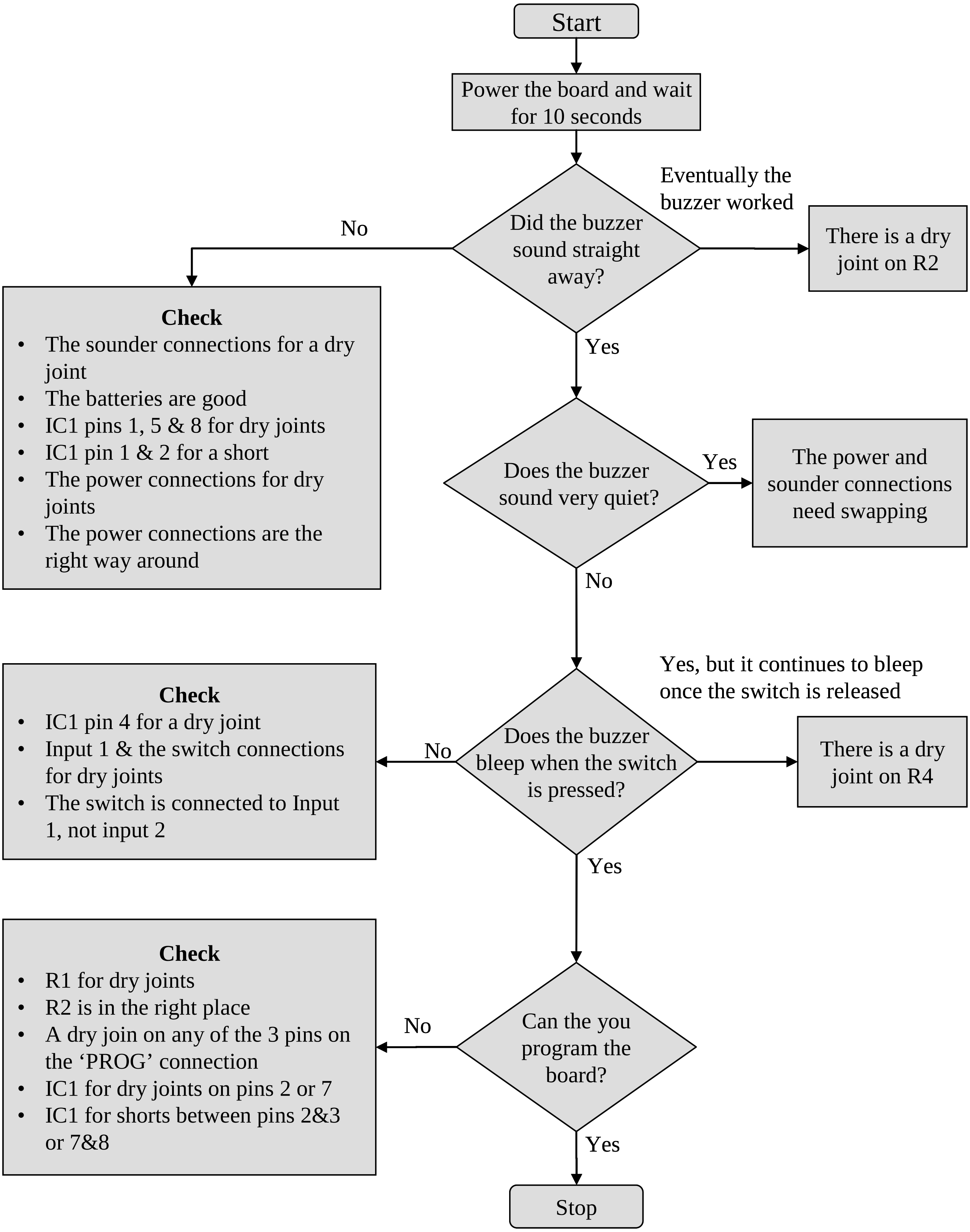
If you have problems with any of the above use the fault finding flow chart to find the cause of the fault. Please note that these fault finding diagrams have been based around using the test software outlined above. If you are using an alternative test program they will not be suitable.

Why use a PIC micro-controller?

There are a number of advantages of using a micro-controller; some of these are outlined below:

- Complex functionality can be produced a very low cost.
- Circuit size can be very small for and still provide complex functionality.
- It is very easy to make minor alterations to the function of the product.
    - For instance changing the tune that is played.
- Flexibility. The circuit could easily be used for a completely different function by simply re-writing the software.
- Easy to develop and debug. Most software packages allow you to simulate the software while it is being developed making it much more likely to work when used. It is also possible to break the functionality down into small steps which is easier to get right then jumping straight to the final design.

Fault finding flowchart



Student programming task

Test program

The test program referenced in the 'Testing the PCB' section can be downloaded from the resource section of the Kitronik website at the web address listed at the bottom of this page.

Task overview

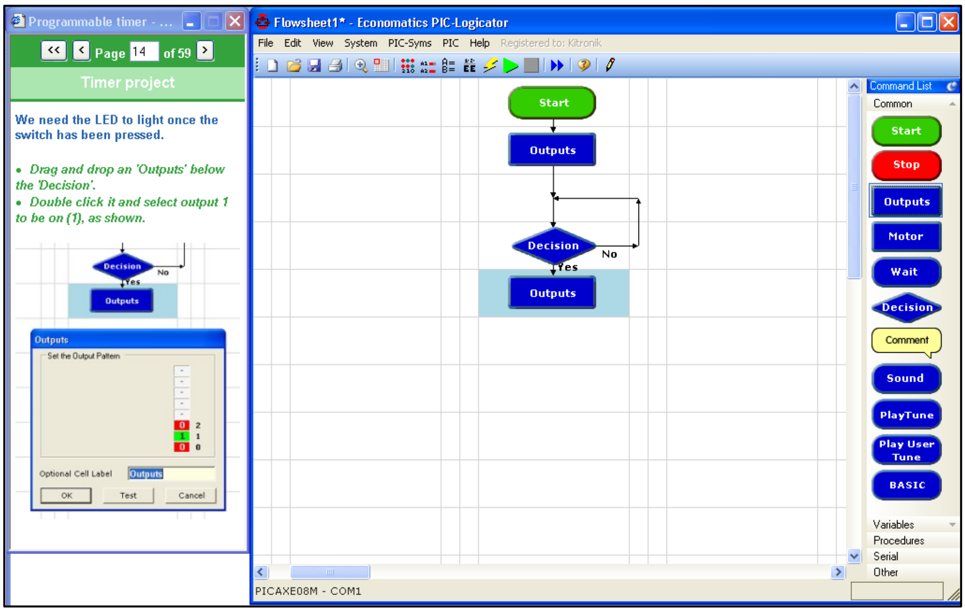
Programming task - Play music when the switch is activated

When the button is opened (or closed if required) the buzzer will play a musical tune.

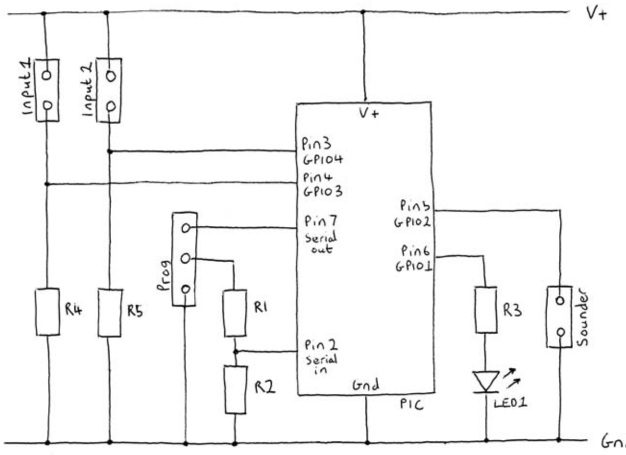
Student guides available on-line

To guide you through the programming task detailed above an electronic guide can be run or downloaded from our website. This guide runs in any web browser, so you don't need to install any software, however the files can be copy onto the PC if you don't have internet access. It is intended that the guide takes up ¼ of the screen, leaving space for the flowchart software. The screen shot below is of the PIC logicator guide, but guides for other software packages are available. To find out if there is a guide for your software and to give it a try go to:
www.kitronik.co.uk/music_tasks.htm

Programming guide Programming software



How the music box hardware works



The music box is based around an eight pin PIC microcontroller device. A PIC is in effect a small computer that behaves in a way determined by the software it's programmed with. This software is generated by the user / student. It is this code that will determine the eventual function of the project. To aid the design of this software the following describes the function of the hardware (circuit) that this software controls. From the music box circuit above you can see that the PIC has two usable inputs and two usable outputs.

Input/Output	**Pin**	**Connected to**
Input 1	Pin 4, GPIO3	Switch
Input 2	Pin 3, GPIO4	Spare (connected to 'INPUT2')
Output	Pin 6, GPIO1	Spare (connected to 'R3' and 'LED1')
Output	Pin 5, GPIO2	Sounder

The other connections to the PIC are to provide it with power (V+ and Gnd) and also allow it to be programmed with the user defined software (Pins 2 and 7).

Input 1 is connected to the micro-switch, using the common and normally open connections. When this switch is not pressed resistor R4 pulls the voltage on the input to the PIC to a low voltage. When the switch is press (closed) the voltage on the PIC pin is pulled up to a high voltage. You will be able to read this change of state in your software.

Input 2 has been left as a spare input should you wish to add another input device. The pull down resistor R5, which is the same as that used on Input 1 would need to be added.

The sounder (buzzer) that has been used in the circuit does not have any drive circuitry. This means that it needs driving with square wave (alternating high low voltage signal) to make it produce a tone. The frequency of the tone will be the same as that of the square wave that is used to drive it. It is by varying this tone that the sounder can be used to play a musical tune.

If you wish you can add an LED to the board. If you do this you will also need to add R3. The purpose of resistor R3 is to limit (restrict) the flow of current into the LED. This controls the brightness of the LED and prevents it from becoming damaged, which would happen if no resistor was used. We would suggest using a 330Ω resistor if you use a standard LED.

One other point worth noting is the processor clock. For any micro-controller to work it requires a clock source. The micro-controller uses this clock so that it knows when to execute the next line of software. Often these clocks are generated externally but in the chip used in this circuit the clock is built into the chip itself. This is why it does not appear on the circuit diagram.