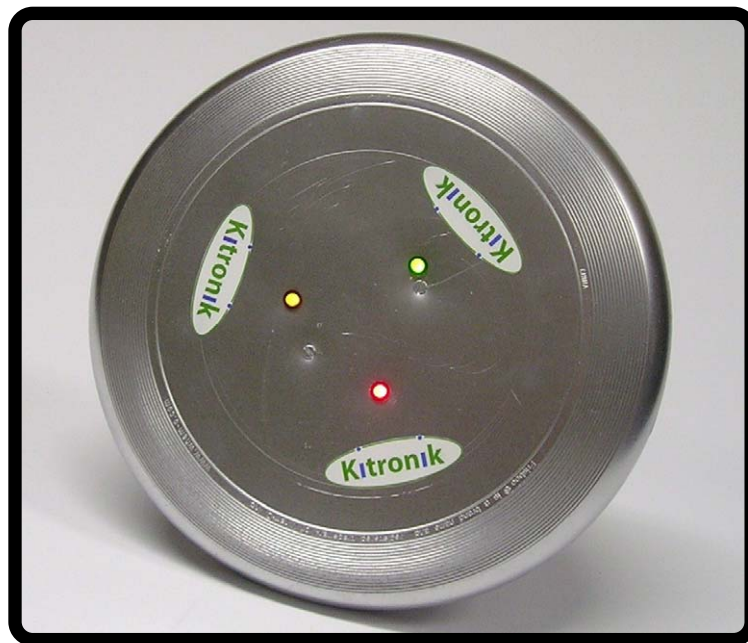




PIC Frisbee



Build Instructions

Issue 1.1



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

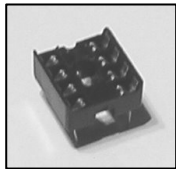
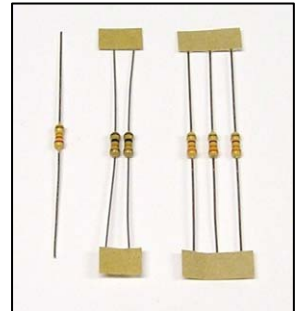
Start with the six small resistors (shown right):

R1 & R2 are 10K Ω (Brown, Black, Orange coloured bands).

R3 is a 22K Ω (Red, Red, Orange coloured bands).

R4, R5 & R6 are 330 Ω (Orange, Orange, Brown coloured bands).

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

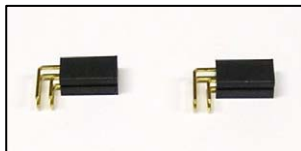


Step 2

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 two tilt switches (shown left) should be soldered into the PCB where it is labeled SW1 and SW2. Make sure these are inserted the correct way around, which is indicated by the outline on the PCB.

Step 5

The two battery holders (shown right) should be soldered into the PCB where it is labeled BAT1 and BAT2. Make sure these are inserted the correct way around, which is indicated by the outline on the PCB.



Step 6

Solder the electrolytic capacitor (shown left) into the board where it is labeled C1. Make sure the device is the correct way around. The capacitor has a '-' sign marked on it which should match the same markings on the PCB.

Step 7

There are three LEDs that are used in the circuit (shown right). These are to be soldered in to the PCB where it is marked LED1, LED2 and LED3. It doesn't matter which coloured LED goes where but it is important that they are inserted the correct way around. Again this is indicated by the outline on the PCB. The flat edge on the outline should match the flat on the body of the LED. You may want to stand these LEDs off the PCB so they are higher than the other components.



Step 8

The PIC can be put into the holder ensuring the notch on the chip lines up with the notch on the holder.



Step 9 (optional)

The buzzer (shown left) should be soldered into the 'SOUNDER' terminal. The red wire should go to the terminal marked 'red' and the black wire should go to the terminal marked 'black'.



Checking Your PCB

Check the following before you insert the batteries:

Check the bottom of the board to ensure that:

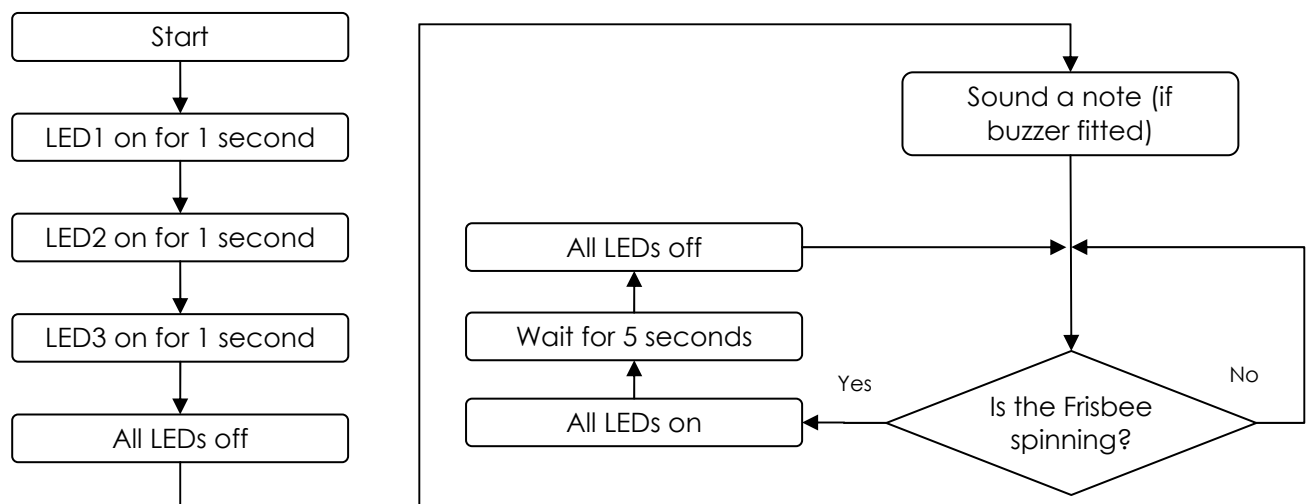
- All the pads with wires in them are properly soldered.
- Pins next to each other are not soldered together.

Check the top of the board to ensure that:

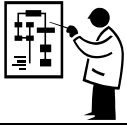
- The notch on the IC and the IC holder are next to the text 'IC1'.
- R1 & R2 have brown, black and orange coloured bands.
- R3 has red, red and orange coloured bands.
- The negative '-' markings on the capacitor match the same markings on the PCB.
- The battery holders are in the same orientation as the outlines on the PCB.
- The tilt switches are in the same orientation as the outlines on the PCB.
- The flat edges of the LEDs match the outlines on the PCB.

Testing the PCB

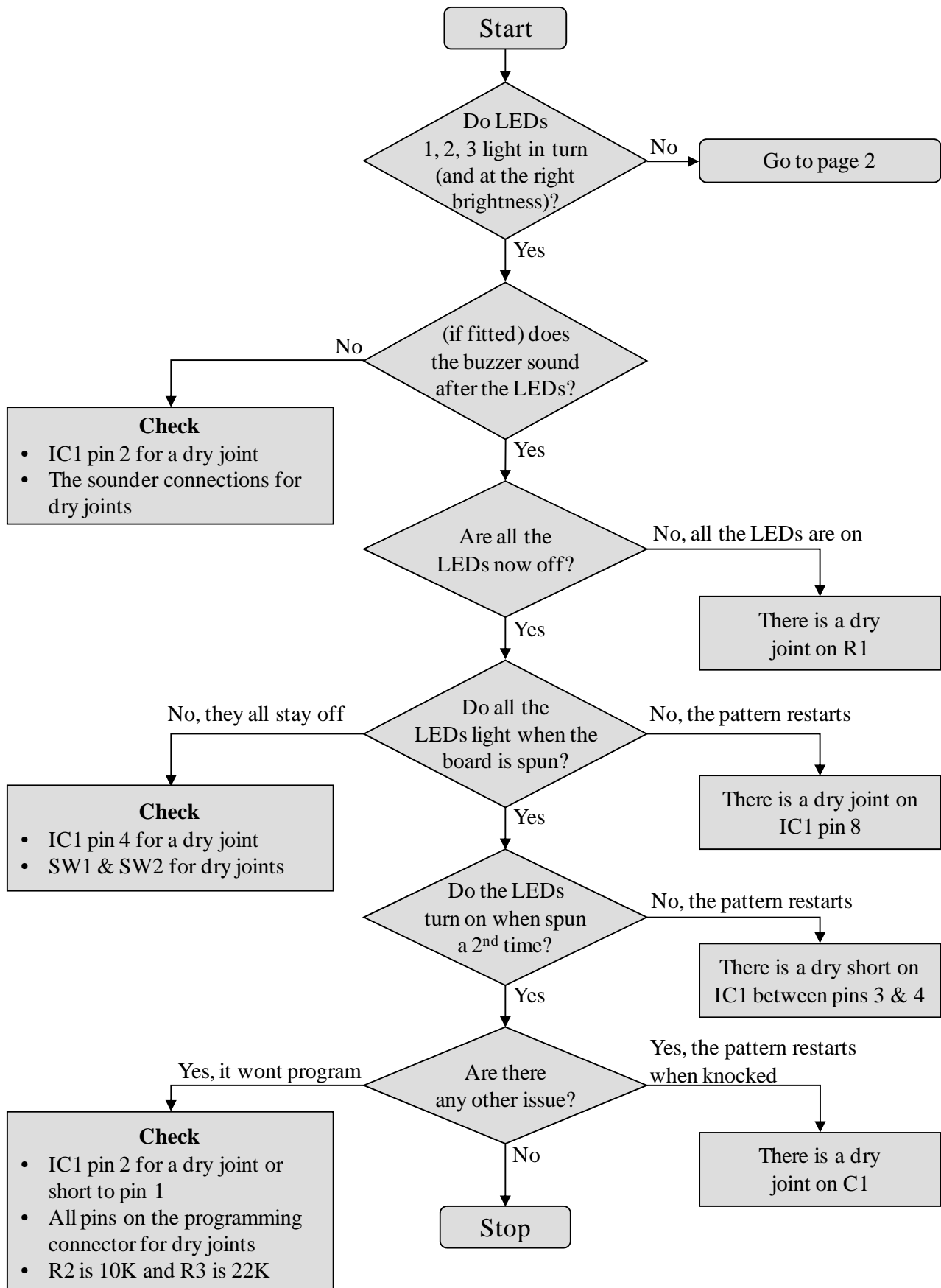
The circuit has been designed to allow easy testing of the PCB. To test the PCB you will first have to insert a chip programmed with some software to allow it to be tested. The purpose of the test program is to test that the LEDs work, the buzzer can sound (if fitted) and that the switches are functioning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

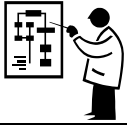


If you have problems with any of the above use the fault finding flow chart to find the cause of the fault. Please note that these fault finding diagrams have been based around using the test software outlined above. If you are using an alternative test program they will not be suitable.

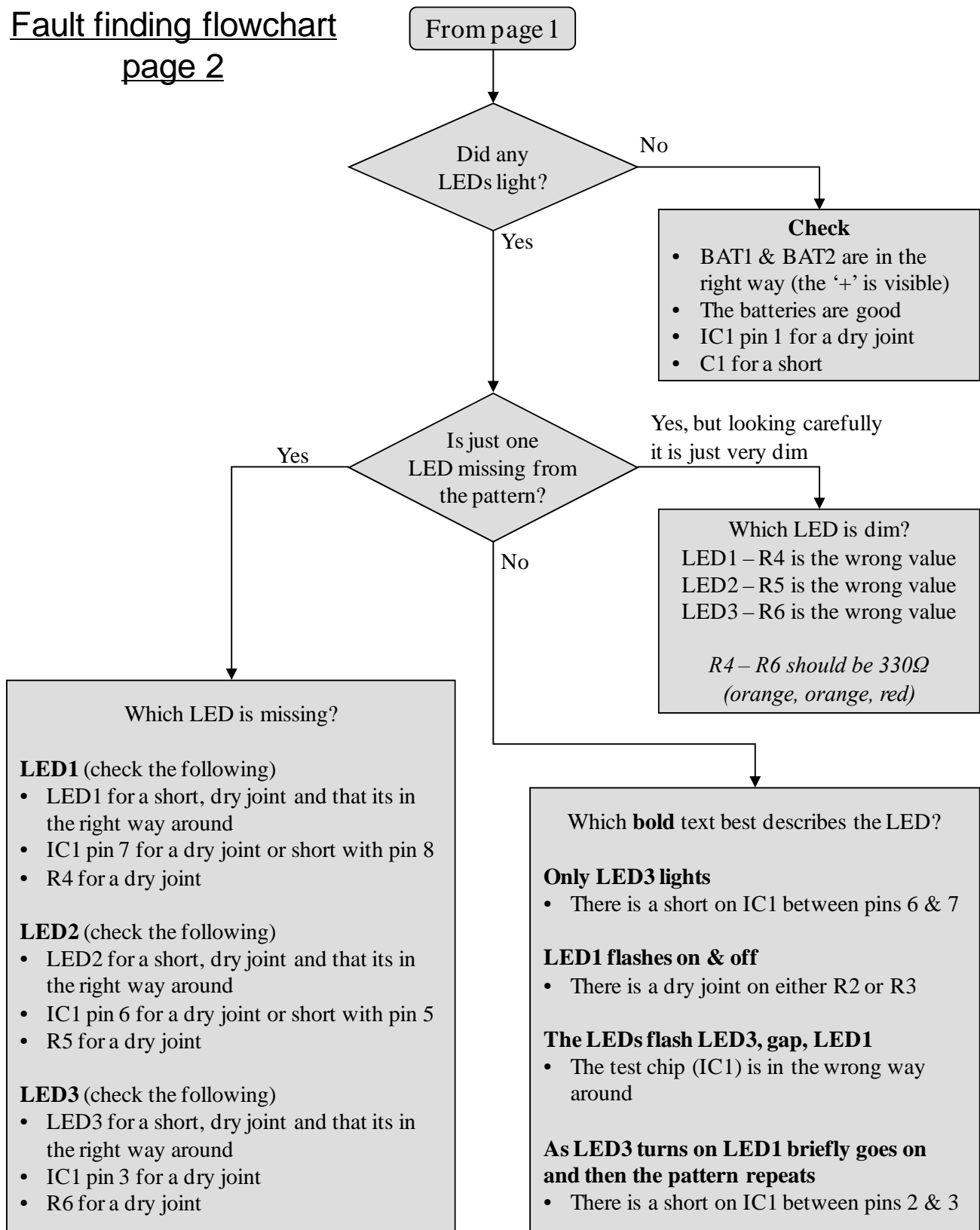


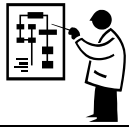
Fault Finding Flowchart





Fault finding flowchart page 2





Programming Tasks

Here are the PIC connections for reference when writing your software.

| Input / Output | Pin | Connected to |
|----------------|--------------|---------------|
| Input | Pin 4, GPIO3 | Tilt switches |
| Output | Pin 3, GPIO4 | LED 3 |
| Output | Pin 6, GPIO1 | LED 2 |
| Output | Pin 7, GPIO0 | LED 1 |
| Output | Pin 5, GPIO2 | Sounder |

Simple task overview

Programming task – Turn LEDs on in a pattern when the Frisbee is spun

When the Frisbee spins and the switches are closed, the input pin goes to a high voltage. When this happens turn the LEDs on to produce a visual pattern. If you have connected a buzzer then a tune can be played at the same time. When the Frisbee stops spinning the LEDs and buzzer should be turned off.

Advanced task overview

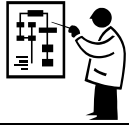
Programming task – In addition to the functionality in the simple task above use the sleep function to increase the products battery life

The circuit normally uses 0.9mA of current while it is waiting to see if the switches have closed. This will give a battery life of around 2 weeks.

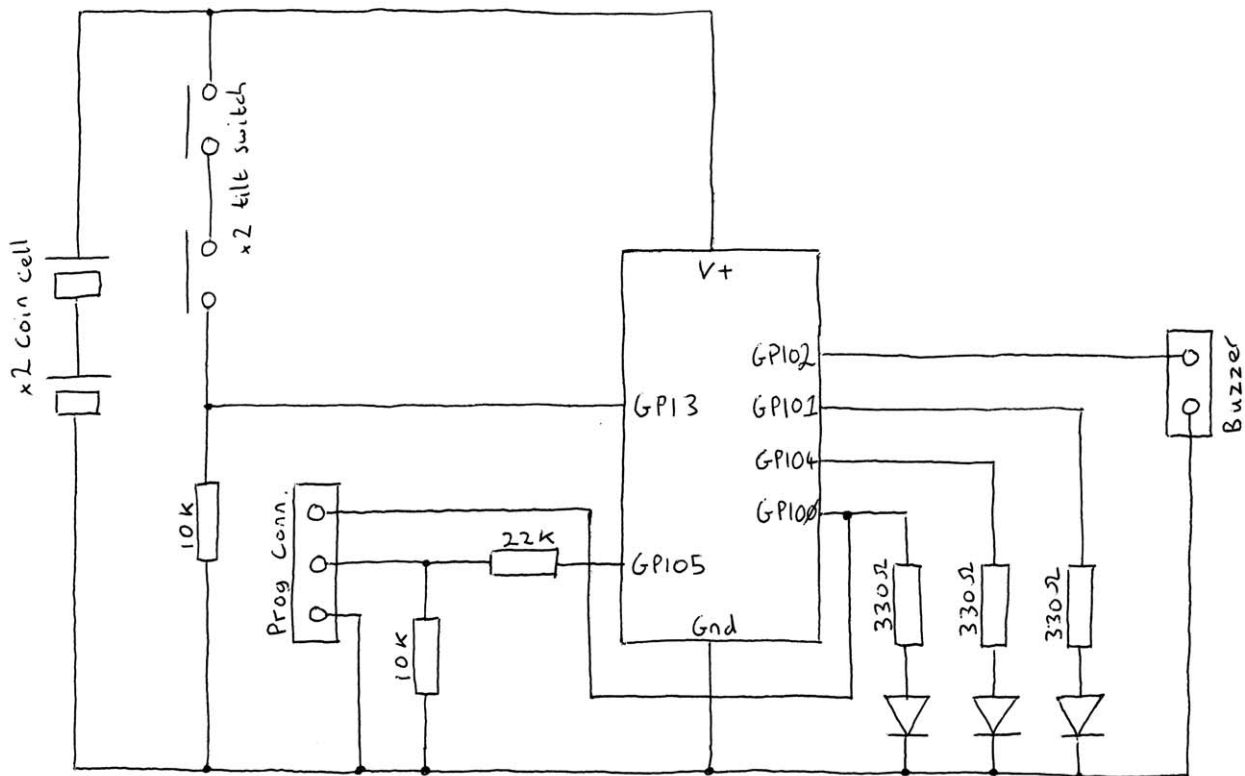
If the sleep function (this turns off a lot of the functions within the PIC for around 2 seconds) is used it is possible to get the current the circuit uses down to around 0.140mA. This will mean the battery life will increase to about 3 months.

To do this you will need to add the following operations to your software program:

- If the Frisbee hasn't been spun for 30 seconds then go to sleep. This can be done by adding a short 0.2 second delay where the switch is checked and keeping a count of how many times this delay has run. If 30 seconds passes and the count has reached 150 then go to sleep. When a spin is detected the count should be set to zero.
- When the PIC periodically finishes the sleep function check the switch input.
 - If the Frisbee is not being spun call the sleep function again.
 - If the Frisbee is being spun the start the LED pattern (and tune), then stay awake until the Frisbee hasn't again been spun for 30 seconds, at which point call the sleep function again.



How the Frisbee Hardware Works

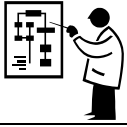


The Frisbee is based around an eight pin PIC microcontroller device. A PIC is in effect a small computer that behaves in a way determined by the software it's programmed with. This software is generated by the user / student. It is this code that will determine the eventual function of the project. To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

| Input / Output | Pin | Connected to |
|----------------|--------------|---------------|
| Input | Pin 4, GPIO3 | Tilt switches |
| Output | Pin 3, GPIO4 | LED3 |
| Output | Pin 6, GPIO1 | LED2 |
| Output | Pin 7, GPIO0 | LED1 |
| Output | Pin 5, GPIO2 | Sounder |

The other connections to the PIC are to provide it with power ($V+$ = pin 1 and Gnd = pin 8) and also allow it to be programmed with the user defined software (Pins 2 and 7). When pin 7 is not being used to program the device it is used as the output which controls LED1.

The input is connected to the two tilt switches and a resistor. When either or both of these switches are open the 10K resistor pulls the voltage on the input to the PIC to a low voltage. When these switches are both closed the voltage on the PIC pin is pulled up to a high voltage. The software is able to read this change of state. These switches are both closed when the board is spun. This is because the centrifugal force pushes both of the elements in the tilt switches outwards. If it is not spinning only one of these switches at a time will be closed because of the way they are orientated (opposite to each other).



Three of the outputs have been connected to LEDs. Between each of the PIC outputs and each of the LEDs is a resistor. The purpose of these resistors is to limit (restrict) the flow of current into the LEDs. This controls the brightness of the LED and prevents it from becoming damaged, which would happen if no resistor was used. A 330Ω resistor has been used for this purpose.

The circuit has been designed so that a buzzer can be added. The type of buzzer used should 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.

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.

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.