

**Copyright © 2008, Wimborne Publishing Ltd**  
**(Sequoia House, 398a Ringwood Road, Ferndown, Dorset BH22 9AU, UK)**  
**and TechBites Interactive Inc.,**  
**(PO Box 857, Madison, Alabama 35758, USA)**

**All rights reserved.**

**The materials and works contained within EPE Online — which are made available by Wimborne Publishing Ltd and TechBites Interactive Inc — are copyrighted.**

TechBites Interactive Inc and Wimborne Publishing Ltd have used their best efforts in preparing these materials and works. However, TechBites Interactive Inc and Wimborne Publishing Ltd make no warranties of any kind, expressed or implied, with regard to the documentation or data contained herein, and specifically disclaim, without limitation, any implied warranties of merchantability and fitness for a particular purpose.

Because of possible variances in the quality and condition of materials and workmanship used by readers, EPE Online, its publishers and agents disclaim any responsibility for the safe and proper functioning of reader-constructed projects based on or from information published in these materials and works.

In no event shall TechBites Interactive Inc or Wimborne Publishing Ltd be responsible or liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or any other damages in connection with or arising out of furnishing, performance, or use of these materials and works.

#### READERS' TECHNICAL ENQUIRIES

We are unable to offer any advice on the use, purchase, repair or modification of commercial equipment or the incorporation or modification of designs published in the magazine. We regret that we cannot provide data or answer queries on articles or projects that are more than five years' old. We are not able to answer technical queries on the phone.

#### PROJECTS AND CIRCUITS

All reasonable precautions are taken to ensure that the advice and data given to readers is reliable. We cannot, however, guarantee it and we cannot accept legal responsibility for it. A number of projects and circuits published in EPE employ voltages that can be lethal. You should not build, test, modify or renovate any item of mains-powered equipment unless you fully understand the safety aspects involved and you use an RCD adaptor.

#### COMPONENT SUPPLIES

We do not supply electronic components or kits for building the projects featured; these can be supplied by advertisers in our publication Practical Everyday Electronics. Our web site is located at [www.epemag.com](http://www.epemag.com)

**We advise readers to check that all parts are still available before commencing any project.**



To order your copy for only \$18.95 for 12 issues go to [www.epemag.com](http://www.epemag.com)

# FREEZER ALARM

## HUMPHREY BERRIDGE



Simply protect your food, learn about using the 8-pin PIC12C508, and about applying the Watchdog Timer.

**S**OME modern freezers contain alarms which sound if you leave the door open and allow the internal space to warm up. However, they do not work if the freezer suffers a power failure, which is a bit of a drawback.

Making a temperature-sensitive circuit which can sound an alarm is not too difficult (a typical example was the *Ice Alert* in Feb '01) but what is required here is a low-cost circuit which can run on batteries for a very long time.

This design uses a circuit based on a PIC, using a feature about which little has been written, namely the ability to send it to sleep! The circuit is extremely simple, and the software uses several techniques which could be useful in other projects.

### BABY PIC

The design uses the “baby” of the PIC family, the PIC12C508. This is an extraordinarily versatile device, and in its OTP (one-time-programmable) version is very inexpensive. It is housed in an 8-pin d.i.l. package (see Fig.1), and has the same set of 33 RISC instructions as its big brothers.

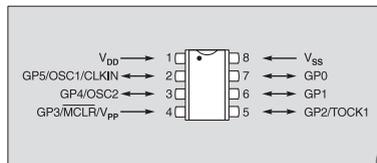


Fig.1 (above). The pin diagram for the PIC12C508 microcontroller.

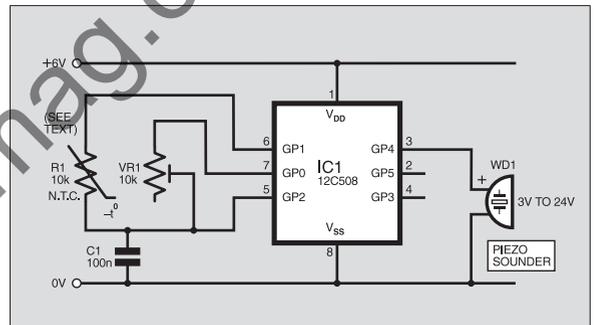


Fig.2 (right). Circuit diagram for the Freezer Alarm.

Two pins are used for power (between 2.5V and 5.5V), the remainder can be configured as five I/O (input-output) pins and one input-only pin.

An in-built oscillator runs at 4MHz – i.e. an instruction every 1μs. It has 25 bytes of

data RAM available, program memory is 512 bytes. There are two internal timers, one of which is a “Watchdog”, and it can drive output devices with currents up to 50mA.

This little PIC is extraordinarily versatile, and for many applications provides adequate microcontroller power. Of course, the more powerful versions such as the PIC16F84 can be programmed to operate in just the same way with few changes to the software, but why use a sledge-hammer to crack a nut?

### CIRCUIT DESCRIPTION

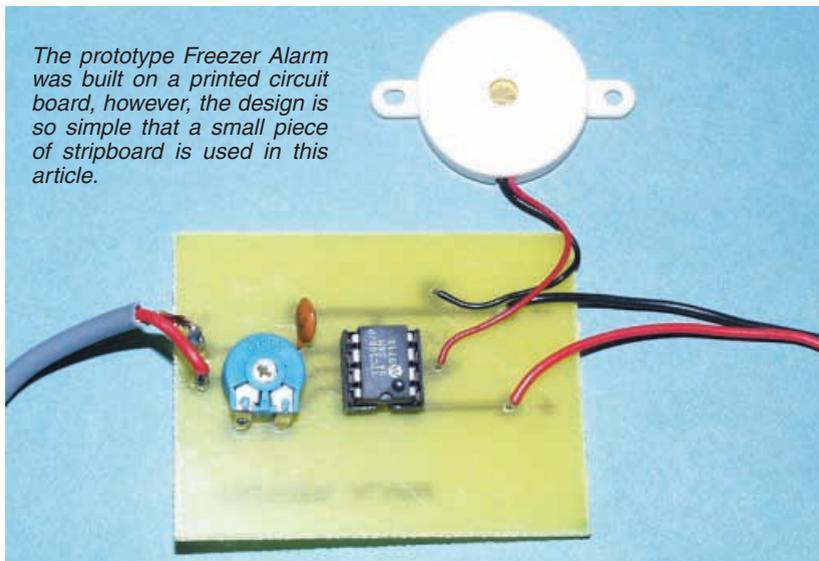
If you are the sort of person who enjoys the challenge of constructing complex circuits, you will be disappointed! The complete circuit contains only five components, as shown in Fig.2. The clever stuff, of course, is provided by the PIC.

The temperature sensor used is a low-cost disc thermistor, R1, which can be attached via a length of 2-core cable. A small preset variable resistor, VR1 is used to set the operating point, the temperature threshold at which the alarm sounds.

Capacitor C1 is used to make the input circuit time-dependant, as described in the next section. For the alarm, a piezo sounder (WD1) is used because it can make a relatively large amount of noise whilst using a very small amount of electrical power. The whole circuit will conveniently run off a 6V battery.

### HOW IT WORKS

Thermistors are basically temperature sensitive resistors – normal n.t.c. (negative temperature coefficient) ones have a high resistance when cold, and a low resistance when hot, and the change of resistance is a very large effect.



The prototype Freezer Alarm was built on a printed circuit board, however, the design is so simple that a small piece of stripboard is used in this article.

An easy way to test a thermistor is to place it in a freezer (which is normally at about  $-18^{\circ}\text{C}$ ) and connect it to a multimeter via leads of suitable length. After a few minutes the thermistor will reach the temperature of the freezer, and with the meter on its resistance range you will be able to measure the approximate value of resistance at this temperature.

The thermistor used here has a resistance of around  $1\text{k}\Omega$  at room temperature (see Table 1). Other types of n.t.c. thermistor could equally well be used with minor alterations to the circuit which are explained later.

**Table 1. Thermistor temperature/resistance**

	Temperature	Resistance
Room	$22^{\circ}\text{C}$	$900\Omega$
Fridge	$2^{\circ}\text{C}$	$2\text{k}\Omega$
Freezer	$-20^{\circ}\text{C}$	$11\text{k}\Omega$

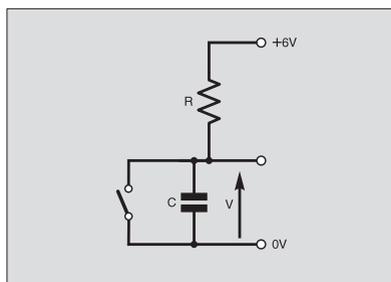
The microcontroller circuit needs to "measure" this resistance in order to evaluate whether the thermistor is too warm – we do not need to know its actual value, just whether it is higher or lower than a preset value. This can conveniently be done by comparing the resistance of the thermistor to that of a preset resistor, VR1.

The PIC12C508 is basically a digital device, so we need a cunning plan to make it capable of measuring resistance. The method is to use the PIC as a timer which can count how long it takes a small capacitor C1 to charge up to a certain voltage.

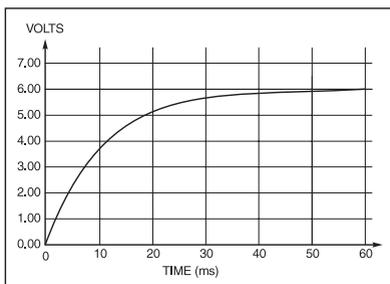
If we consider a very simple circuit (Fig.3) consisting of just a capacitor C and a resistor R, we can see that if the switch is closed, the voltage V across the capacitor is zero and it is uncharged. As soon as the switch is opened, the capacitor charges via the resistor, and the voltage rises along an exponential curve (Fig.4).

In our circuit, we time how long it takes to go from zero to the logic threshold of the PIC (the point at which a logic 0 changes to a logic 1), which is about 3.0V, say.

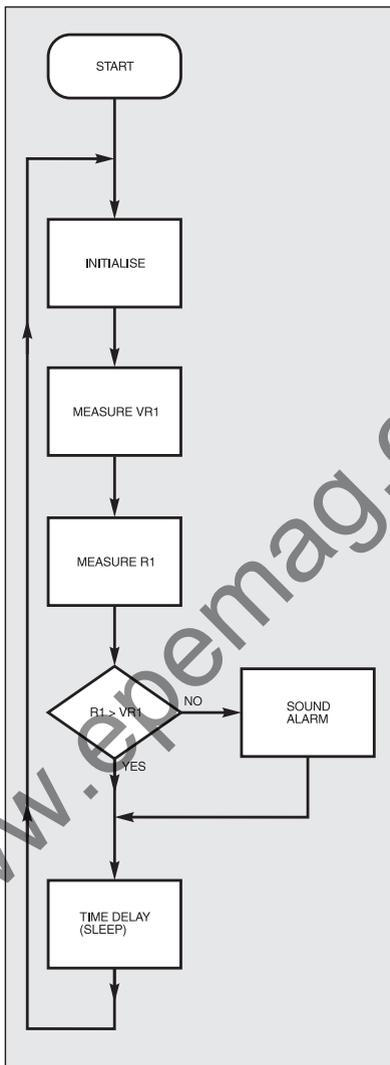
The mathematics of this charging process produces a very simple relationship, namely that the time taken to reach a certain voltage is directly proportional to the value of R, so the time we have counted out until the voltage rises to 3.0V is an accurate representation of the value of the resistor. So all we need to do is to time how long it takes for the capacitor to charge via the thermistor, then time how long it takes to charge via the preset resistor. If the first time is shorter than the second, the thermistor is too warm and we ought to sound the alarm!



**Fig.3. Simple circuit to demonstrate capacitor charging.**



**Fig.4. Capacitor charging curve.**



**Fig.5. Flow chart for the main program.**

What makes it possible for the PIC to do this is that not only can we force pins to go high or low, but we can also change whether a pin is an output or an input as the program is running. Using this fact we can both switch between the two resistors, and discharge the capacitor between our readings. This is described in the section on programming.

### MAIN PROGRAM

The way the program operates is straightforward, as illustrated by the flow-chart in Fig.5. First the PIC is initialised in that pins are set up either as outputs or inputs. We then have to measure the charging times of the capacitor first for the preset resistor VR1, then for the thermistor R1.

In order to give a reasonable number of steps to do the counting and improve the range of accuracy of the measurement, two registers are used together to form a 16-bit value.

The technique used for measuring resistance is shown in Fig.6, and involves the following:

1. First the capacitor needs to be completely discharged. To do this, pin 5 is set as an output, and then set at logic 0 (zero volts). This effectively shorts the capacitor. A delay of approximately 5ms makes sure it is fully discharged.

2. Next the two registers used to store the count time for the thermistor are set to zero.

3. Then pin 6 is set as an output, and reset to logic 0. Pin 5 is swapped to being an input pin, and then pin 6 taken to logic 1. At this moment the capacitor begins to charge up.

4. The program now starts to loop, incrementing the counter registers as it goes. Each time around the loop pin 5 is checked for having reached the voltage threshold at which it is considered to be at logic 1.

5. As soon as this happens, the routine ends, and the counter registers then contain the final time count. This process is repeated for the variable resistor VR1 using a different set of counter registers and pin 7 instead of pin 6.

After both measurements have been made, the two answers are compared: all we need to do is to subtract one number from another – we just want to know which process took the longer. Depending upon the outcome, we either go directly into the Sleep mode, or sound the alarm for a short time before again going to sleep.

### ALARMING

Generally speaking, a piezo device is not a buzzer – it has to be driven by an oscillating signal to make the alarm

## COMPONENTS

<b>Resistor</b>		See <b>SHOP</b> <b>TALK</b> page
R1	n.t.c. disc thermistor, 10k (1k at room temperature)	
<b>Potentiometer</b>		
VR1	10k min. enclosed carbon preset, horiz	
<b>Capacitor</b>		
C1	100n ceramic disc	
<b>Semiconductor</b>		
IC1	PIC12C508 preprogrammed microcontroller (see text)	
<b>Miscellaneous</b>		
WD1	piezo sounder, 3V to 24V	

Stripboard, 10 strips x 17 holes; plastic case, size and type to choice; battery holder; AA-size batteries (4 off); battery clip; 8-pin d.i.l. socket; connecting wire; solder, etc.

Approx. Cost  
Guidance Only  
**£9**  
excluding case & batts.

sounds. There are two reasons for using a “swept frequency” alarm sound using the software:

a. Changing sounds such as beepers, sirens, etc. stand out much better from background noises, and

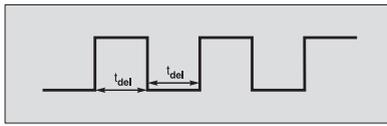


Fig. 7. Output waveform.

b. the piezo device used here has a very sharp resonance – i.e. the sound output is very much higher at a specific frequency than at others. Different devices have different resonances, and even those tested of the same type had frequencies which varied quite a bit. By sweeping the frequency through the resonance one can be sure of making the most irritating noise! In this design, the frequency sweeps from approximately 2kHz to 8kHz.

To make a PIC generate an oscillating signal is quite simple. Referring to the waveform in Fig. 7, you take the output high, wait for a short time ( $t_{del}$ ), take the output low, wait for a short time ( $t_{del}$ ), and then repeat the process.

The result is a square wave driving the piezo sounder. If, for instance, we made  $t_{del} = 0.5ms$ , the period would be  $2 \times t_{del} = 1.0ms$ , and the sound would have a frequency of 1kHz.

Sweeping the frequency may be accomplished by gradually changing the value of  $t_{del}$  which can be done using the fact that delays in microcontroller systems are often made by “time-waster” loops where a counter counts down to zero.

By changing the value loaded into a register in the first place, the time taken can be varied. Fig. 8 shows the program segment responsible. For each specific frequency, eight cycles are generated. The whole process takes approximately 130ms.

### WATCHDOG

Not many published projects seem to use a PIC’s Watchdog Timer (WDT), but when it comes to conserving power using the Sleep mode, it is very useful. Information can be found in Microchip’s data sheets and other publications.

When a PIC is given the Sleep command, it shuts down nearly all its functions and as a result consumes a very tiny amount of power – it only draws about  $4\mu A$  from the supply.

There are three ways in which it can wake up again:

- by a logic change on a pre-defined pin
- when the WDT times-out, and
- by an external master Reset

It is the second of these options that is used in this project. Inside the PIC12C508, the WDT function has its own internal oscillator and counter, and these continue to run even if the PIC is asleep. When the WDT times out, a system Reset is generated, and the program restarts.

It should be noted that one bit of the configuration word of the PIC enables or disables the WDT, and this bit must be set when the chip is programmed.

So, if we reset the WDT, it will then start counting and when it times-out (18ms later), the program goes through a Reset and starts again. That is not a very long time, but fortunately we can use the internal pre-scaler which can be used in conjunction with the WDT. To do this, we use the OPTION register to extend the time-out period to approximately 2-3 seconds.

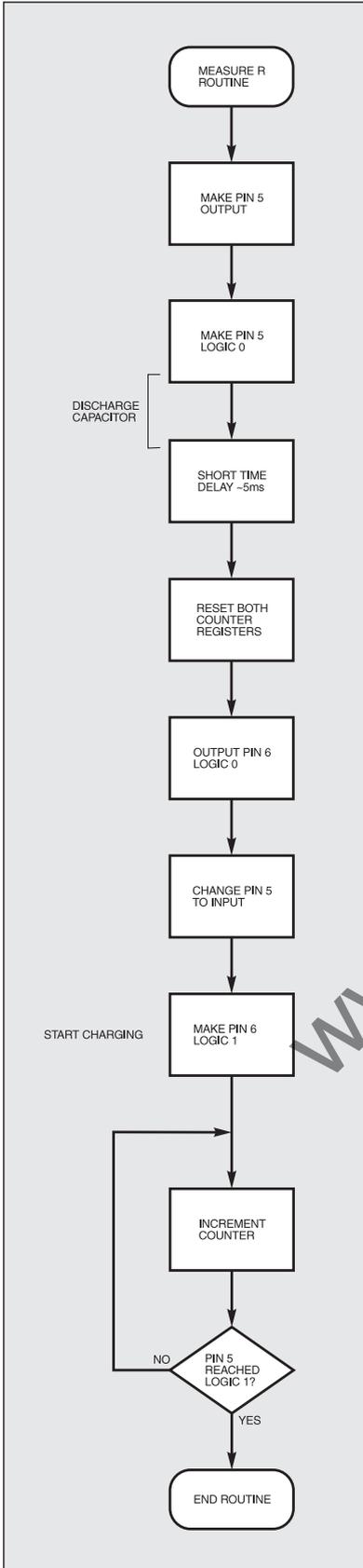


Fig. 6. Measuring the resistance of the thermistor.

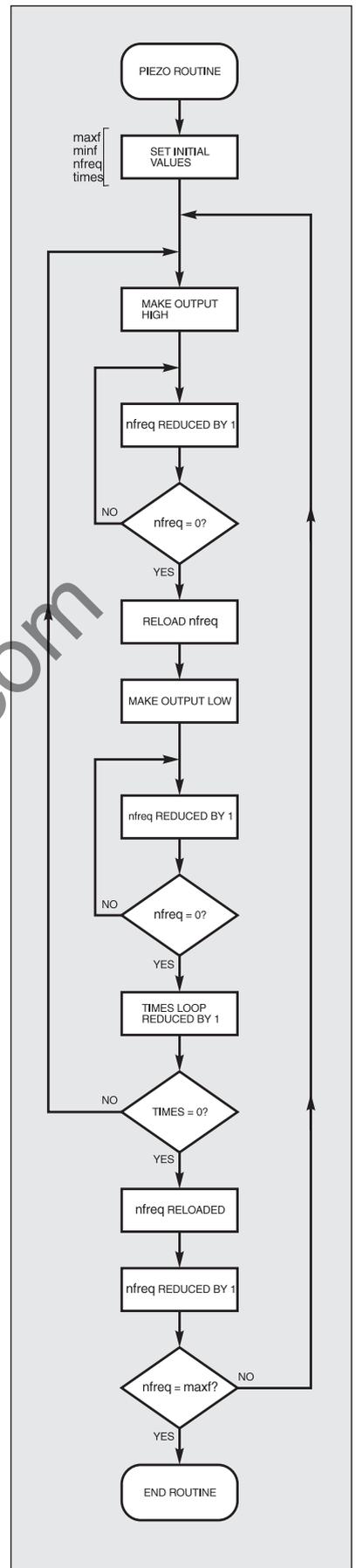


Fig. 8. Flowchart for the piezo sweeper. The whole sequence takes approximately 130ms.

This may not seem very long, but what matters in conserving power is the ratio of the time taken to run the measurements compared and the time remaining asleep. If, for instance, it takes 50ms to take the measurements, the circuit is asleep for 98 per cent of the time with a resulting extension of battery life.

The OPTION register has eight bits which must be set up in order to control the WDT and prescaler. The functions are shown in Table 2. For this application, the OPTION register must be set to contain the binary byte 11001111. The short main program section is shown in Fig.9.

Note that during a Sleep period, the I/O ports maintain the same conditions that they had immediately beforehand. Therefore, to minimise the current drawn, all the pins are made inputs (high impedance) before the Sleep command.

## CONSTRUCTION

Construction is very simple. The suggested stripboard component layout and track cut details are shown in Fig.10.

The thermistor can be soldered to a short length of wire such as thin audio coax. An improvement would be to waterproof the thermistor connections by dunking it in polyurethane varnish. The wire can be fed into the freezer via the door seal.

It is important to resist the temptation to add a light emitting diode as a battery indicator – the I.e.d. would take about a thousand times more power than the rest of the circuit!

The PIC should be plugged into the board via an 8-pin d.i.l. socket. The circuit and batteries can be housed in a plastic box to sit outside the freezer, a small hole being provided to glue the piezo sounder behind. You should not need to replace batteries very often.

Software and pre-programmed PICs are available as stated in this month's *Shoptalk*.

## TESTING

The circuit will work quite happily at room temperature. Once the batteries are connected (it seems to work well on 6V although this is higher than the maximum recommended).

Gently rotate preset VR1 until the threshold is found between the alarm bleating or not. Set it so that the alarm is just off. Then hold the thermistor in your fingers to warm it up, and the alarm

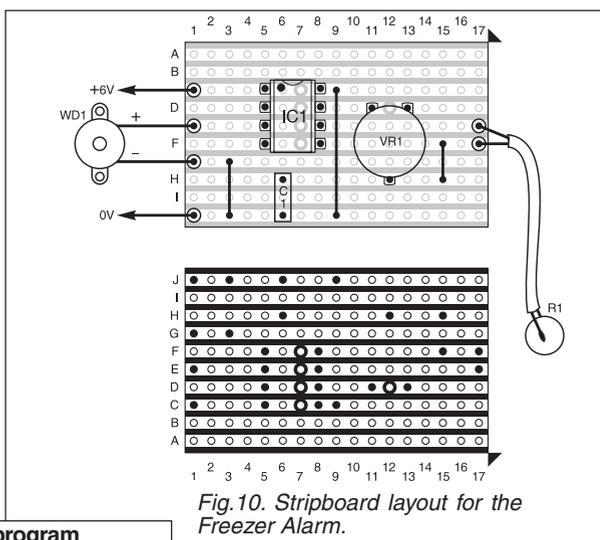


Fig.10. Stripboard layout for the Freezer Alarm.

Fig.9. Main program

```

; Main program start
start
  clrw dt          ;initialise I/O
  clrf  gpio
  movlw b'00000111'
  tris  gpio
  movlw b'11001111'
  option          ;wdt and prescaler
                    setup

  call  delay      ;settle down
  call  measureR
  call  measureT
  call  compare
  xorlw 0FFH      ;check if resT <
                    resR
                    ;if so, set off alarm!

  btfsc z          ;if so, set off alarm!
  call  alarm
  movlw b'11111111'
  tris  gpio      ;all pins high
                    impedance

  clrw dt
  sleep          ;until reawakened
                    by wdt
                    ;with a reset kiss

  goto start     ;should never reach
                    here!

```

Table 2. Option Register Bits

Bit 7	GPWU	Enable (0) or disable (1) wave-up on pin change
Bit 6	GPPU	Enable (0) or disable (1) weak pull-up resistors
Bit 5	TOCS	Timer 0 Clock source
Bit 4	T0SE	Timer 0 edge select
Bit 3	PSA	Prescaler assignment: Assigned to WDT (1), or to Timer 0 (0)
Bits 2-0		Prescaler rate – division ratio (111 for WDT ratio of 1:128)

should sound; let go to allow the thermistor to cool again to room temperature, and the alarm should stop.

Once you are convinced all is well, put the thermistor in the freezer, and after allowing time for the temperature to stabilise, increase the resistance on the preset so that the alarm threshold is set where you would like it.

In fact, the best way to find out if the batteries are OK is to let the thermistor warm up a bit when you open the freezer – if it is working and the alarm sounds, the batteries are fine!

## MODIFICATIONS

As mentioned earlier, the techniques explained allow simple modifications to change how the circuit operates:

a. To make the alarm work if something is too cold rather than too warm (for example as a greenhouse frost alarm), simply swap over the connections to pins 6 and 7. Alternatively, make the appropriate swaps in the software.

b. The method can be adapted for any resistance changes – for example, the alarm could be made light-sensitive by using an I.d.r. (light dependent resistor) instead of a thermistor. Whatever thermistor or other device is used, VR1 needs to have a resistance which can be set to the same that the device has at its operating threshold.

c. An important design consideration concerns the value of the capacitor C1. Once the resistance of the appropriate sensor is known, the variable preset resistor needs to have the same value. However, the counter which waits for the capacitor to charge to its threshold must not overflow. This will happen after 320ms with this design's value for C1. Therefore, ensure that  $C(\mu F) \times R(k\Omega) < 320ms$ .

d. The circuit is remarkably accurate and stable. Because of the timing method of comparing the two resistors, any changes in the supply voltage (within the parameters above), or changes in the capacitance value caused by temperature or ageing, have virtually no effect.

## ACKNOWLEDGEMENTS

The author expresses his thanks to Mrs Jan Edwards for her help in this project. □

