ThermIIC Series

Dr. Simon J. Melhuish

Part 2

In Part 1 I described how to attach up to eight DS1624 I²C temperature sensors to your Acorn 32-bit computer. At the end of that article I left you running the test program, *1624Test*. Assuming nothing dramatic happened (involving smoke, perhaps!) you'll want me to explain how it works. So this time I will run through the operation of *1624Test*, before presenting a rather more sophisticated program, *ThermIIC*.

Information on the operation of the DS1624 is given in its data sheet, which can be obtained as a *pdf* file (you'll need *RiScript Pro* or *!PDF* to read it) from the Dallas web site, *http://www.dalsemi.com/*

RISC OS provides a *Software Interrupt* (SWI) call to address devices on the I²C bus. The program calling this *IIC_Control* SWI provides the I²C slave address, a memory address for reading from or writing to, and the number of bytes to read or write. A DS1624 chip can reside on any of eight I²C slave addresses. The first four bits of the address are factory set to 9. The next three bits are set by pins 5 to 7. The last bit of the address depends upon whether we want to perform a read (in which case it is a one) or a write (zero). Thus, for writes we address &90 + $N \times 2$, where N is the code set on the address pins. &91 + $N \times 2$ will perform a read.

In procedure *PROCinit_1624* each device (assuming it's attached) is initialized. Sending byte &AC (access configuration) followed by &4A (which sets bit 0) sets the device for continuous temperature conversion. Next, sending &EE triggers the start of a temperature conversion. The program pauses for 1 second to allow time for the first conversion.

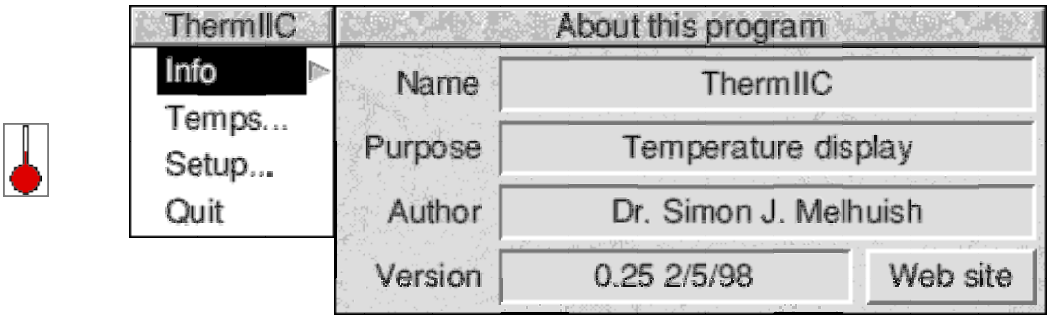
When we want to read a temperature we send &AA and then read two bytes. The first byte read is the integer part of the temperature, in °C. The next byte gives the part following the decimal point. To convert these bytes to a proper temperature value *1624Test* swaps the byte order (so that the low byte comes first) and appends two 0 or &FF bytes, depending upon whether the value is positive or negative. This gives a conventional 4-byte integer. Dividing by 256 gives the decimal temperature in °C.

For the addresses with no attached sensor the I²C SWI fails, in which case *1624Test* prints “No response”.

1624Test is, frankly, a bit dull. To get the most out of the temperature sensors I wrote *ThermIIC*, which is provided on the cover disc. Please carefully read the licence conditions and any other text files supplied on the disc. There are limitations on the free use of the software. As well as providing a much more “classy” display of temperatures *ThermIIC* can write log files to disc. These can be read by a spreadsheet program, for example, allowing analysis and presentation of temperature data.

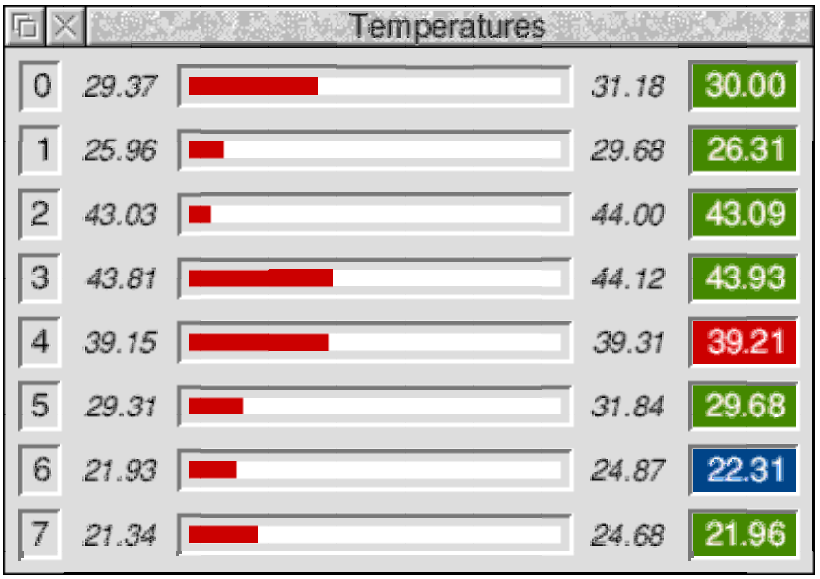
ThermIIC will run from the cover disc archive, but I suggest that you make your own copy elsewhere. If you have not yet installed the Acorn *Toolbox* modules on your computer (*ThermIIC* will complain if it can't find them) you will need to update your *!System* application with the resources provided.

Run *ThermIIC* by double-clicking its icon in the usual way. The *ThermIIC* icon will appear on the iconbar. Note that *ThermIIC* supports interactive help. To find out what any of the buttons, &c, do, you can run the *!Help* application and move the mouse pointer over the item in question.

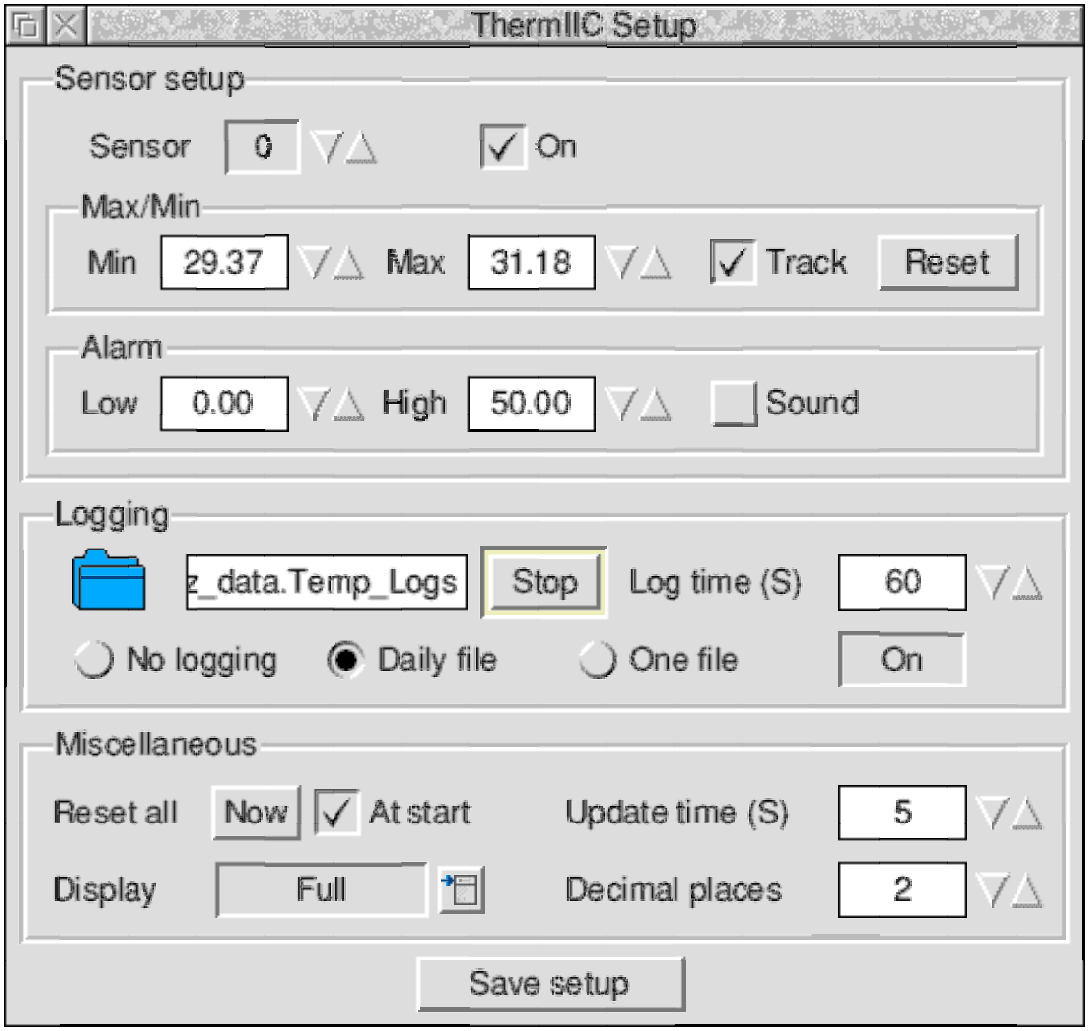


The *ThermIIC* icon and iconbar menu

If it hasn't appeared already, you can display the main temperature window by left-clicking the iconbar icon. The number of temperatures shown and the display format are determined by the program setup. Right-click the iconbar icon for the setup window. You can also access these windows from the iconbar menu. This has entries for program info and quitting the program as well.



The main temperature window



The setup window

The setup window is the most complicated bit of the program, so I'll take some time to explain it properly. Note that any changes you make are acted-upon immediately. However, no changes will be remembered for the next time the program is run unless you click the “Save setup” button. Note that this also records the state of the temperature display window − whether or not it is open, and where.

The are three sections to the setup window. The first of these is concerned with individual sensors. Find the sensor you want to set up by clicking the up and down arrowheads. The sensor is made active (or not) according to the “On” option. The “Max/Min” box relates to the maximum and minimum values used in some display formats − those with a bar (see below). “Track” determines whether or not these values are updated automatically when the actual temperature goes out of the set range. The “Reset” button forces them to the current temperature. The “Alarm” box sets low and high limits for the alarm. If the temperature drops below or rises above these values respectively, an alarm condition will result. In this case the temperature display value will turn blue or red (see the screen grab), and the iconbar icon will turn amber. If “Sound” is ticked the computer will beep.

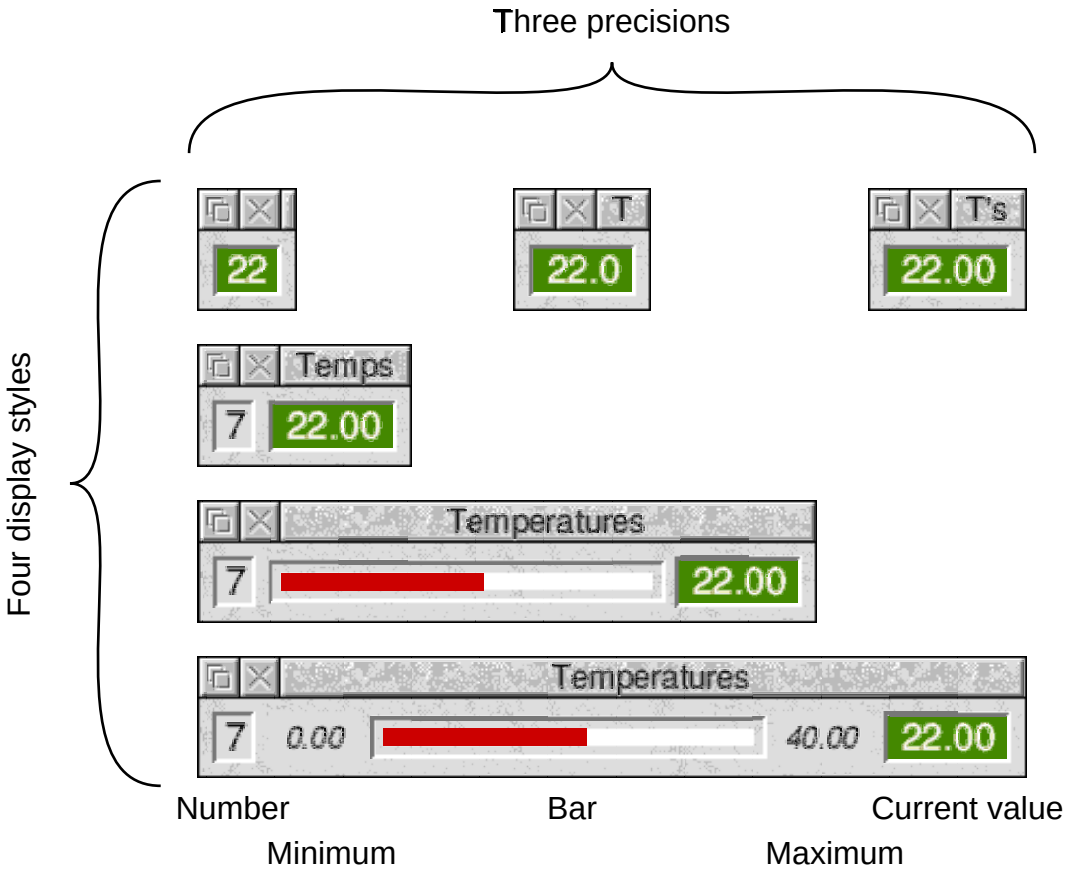
The second part of the setup window controls logging operations. Logs can be written in either of two ways. If “One file” is selected the log will be written to a single file, as long as the program runs. The file might grow rather large! To start logging drag the CSV file icon to a filer window, or enter the full filename specification at the icon and hit return. For many applications it makes sense to split the log into 24-hour chunks. You can do this by selecting “Daily file”. A new file is started every midnight. In this case you get to drag a directory icon (as in the screen grab). The directory thus specified is created if it does not exist already. A directory is created inside it each time a fresh month is started. This is named according to the form *yymm*. For example, June 1998 would be 9806. The CSV files created inside the month directory are

named according to the form *yymmdd*. For example, 14th June 1998 would be logged to 9806.980614

The “Log time” entry sets how often the log is written. Log entries are in *comma separated value* format, starting on each line with the date and time. Each temperature value follows, separated by commas. Entries for inactive sensors are left blank. Here is an example from the start of a file:

```
Date / Time      ,  Temp 0,  Temp 1,  Temp 2,  Temp 3,  Temp 4,  Temp 5,  Temp 6,  Temp 7
05/04/98 19:31:09,    5.33,    5.86,    5.06,    5.86,    5.20,    5.06,    5.25,    5.91
05/04/98 19:31:39,    4.18,    4.64,    4.02,    4.76,    4.02,    3.98,    4.18,    4.78
```

The third and final setup section is for miscellaneous items. “Update time” simply sets the time between temperature readings. “Reset all” resets the min/max fields of tracking sensors either at program start or when the “Now” button is clicked. “Display” and “Decimal places” control the appearance of the main temperature window. The effect of these options is illustrated by the figure. Note that changing the number of decimal places displayed does not affect logging or alarms, which remain at full precision. There are four display styles, switching on or off parts of the full display. Each line of the full display shows, from left to right, the sensor number (0 − 7), minimum value (used for the bar), a bar graphic, maximum value (used for the bar) and the current temperature. The background colour of the last icon is blue, green or red, depending on the alarm state of that sensor. Note that when *ThermIIC* receives no response from a sensor it greys-out its line in the display.



Temperature display options

I am setting up a *ThermIIC* web page. To get there run your browser and click on the “Web site” button on the *ThermIIC* program info window. If anyone reports any bugs I'll upload fixes there first. Also I hope to add some more features in future.

In part three I'll look at *ThermIIC* in use and talk about some of the problems of accurately measuring temperatures. For those wanting to get their sensors wet, I'll tell you what I got up to with a packet of potting compound…

Thanks to David Watkins for his help in testing *ThermIIC*.