ThermIIC Series

Dr. Simon J. Melhuish

***Part** 1*

This is the first part in a series of articles exploring temperature logging with a RISC OS computer. In this part I will describe the hardware side of the project. Next time I will present the main software component, ***ThermIIC***. The construction of a sensor board and the computer connexion requires only basic tools, a little electronic assembly experience, and a very modest budget.

Why would one want to read temperatures with a computer? In an educational situation there are many uses, in science, maths and other areas. More on that in a later issue though. In the home it might just be for interest − what's the weather doing? Or you might want to see what your greenhouse temperature is doing, for example. In industry there are countless uses for temperature logging.

For myself, working at Jodrell Bank on the *Very Small Array* project, a new radio telescope for probing the *cosmic microwave background*, I found a need for temperature logging when I built a temperature control system for some electronics. I wanted to see if it worked as well as it should. I'd used I²C on an A440 and an A4000 in the past, monitoring ambient and cryogenic temperatures, so it was natural to apply these techniques to this new problem. Previously, to monitor computer temperature, I used the National Semiconductor, LM75 *Digital Temperature Sensor*. But there are a couple of snags...

Snag number one: the package. The LM75 comes only as a surface mount device (SMD). This meant some nifty work with a soldering iron. My eyesight wasn't up to that, and I had to get a technician to wire it up (He used fine wires to join each pin to the corresponding pin on an 8-pin DIL socket, which could then be plugged into another DIL socket − a useful tip if you have to do this). Trawling through the catalogues I came across an alternative device by Dallas Semiconductor − the DS1621. This is available as both SMD and a dual in-line package (DIP).

But then there was the second snag. My temperature control system works to a resolution of 0.1 C°. Both the LM75 and the DS1621 have a resolution of just 0.5° (actually you can get 0.1° resolution out of the DS1621 if you use some software trickery). This was way too poor. Then the DS1624 presented itself. Dallas's newer chip has a resolution of 0.03125°. Logging air temperature with so fine a resolution as that even shows when I go in and out of my lab, or if I switch the lights on. It looks smooth too − the digitization steps are swamped by the very small, but real, variations in air temperature.

The DS1624 so far has not appeared in the UK electronics catalogues. But fortunately Farnell Components have recently bought out a US competitor, who **do** supply it. If you 'phone Farnell with your credit card details they should be able to satisfy you. I've put a parts list for the project, along with stock numbers and contact details, in a file on the cover disc.

Now on to the job of construction. First, you'll need to get access to the I²C bus from your computer. There are four lines you need to connect to. They are +5 V, 0 V (ground) and the I²C clock (SCL) and data (SDA) lines. These are all on the podule bus connectors. If you have an A3000 you've got it easy, as the whole podule bus is available on the DIN 41612 connector at the back of the machine. All you need to do is make your connexions to a DIN 41612 plug (see figure), to go in the back.

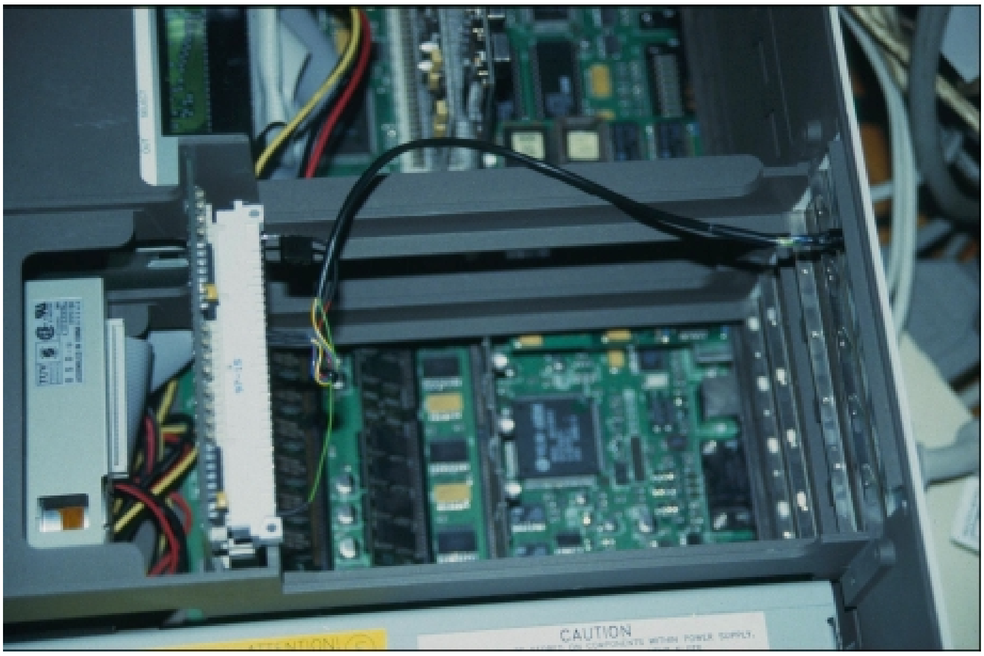
The DIN 41612 indirect edge connector has 3-rows. The rows are given the letters a, b and c (earlier A-series machines used only 2 rows). We do not need any connexions to row b, so an a+c style connector is fine. If you look very closely at the back of the connector you'll see the row letters and a number on each pin. Refer to the table for numbering.

I recommend that you cut off all the unused pins, so that there's no chance of them shorting to anything they shouldn't. Do double and triple check though. The first time I did this I cut off all the **wrong** pins! It is wise to sleeve your solder joints. Heat-shrink sleeving is very useful for this. Having cut it to length and put it in position, apply heat. A powerful hair drier might just about do it. If you use anything hotter be careful not to melt any insulation.

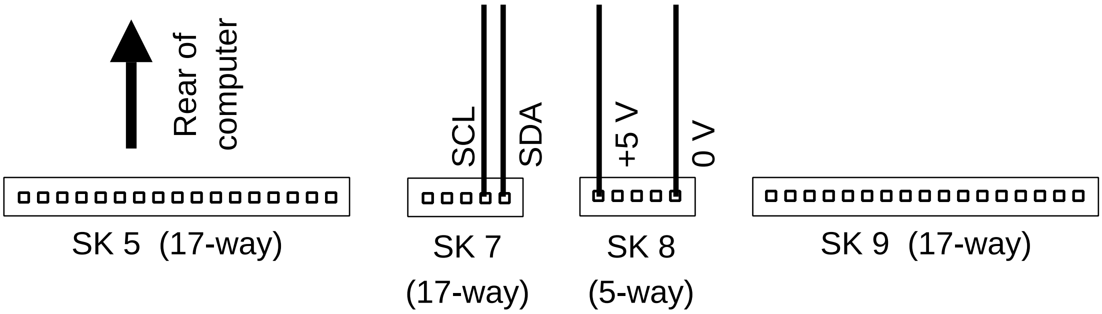
The diagram shows one extra component in series with the +5 V line. This is a fuse, to safeguard the computer power supply in case of “accident”. Rather than mess about with fuse holders I like to use the solder-in leaded type. After all, it really shouldn't ever blow so long as you're careful! You should protect the solder joints to the fuse with heat shrink, and a larger diameter piece can be used to cover the whole thing.

On models with a podule backplane (A5000, Risc PC, &c) you'll need to get inside to bring your connexions out through the case somehow. Refer to the Welcome Guide that came with your computer for instructions on removing the case. Remember that there can be dangerous mains voltages present. **On no account** should you do any work inside of your computer with the mains plugged in, even if the computer is switched off.

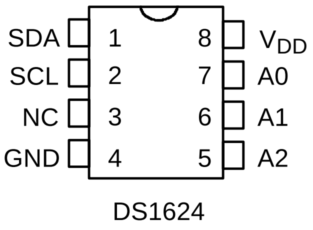
If you can sacrifice a podule blanking plate (or make yourself a new one - especially with a Risc PC which has only very flimsy things) it's not difficult to make a hole for your cable. Use a grommet around it. If you prefer you can break the cable with a pair of connectors. There is not really a standard, as such, for I²C connectors. A commercial device that I bought uses a connector similar to an RJ45 ethernet connector. I believe that the Baildon Electronics “oddules” used a Molex connector. Use whatever suits you best. I have a favourite 4-pin miniature circular connector, but they're a bit pricey (see parts list). I mount a bulkhead-type socket on the podule blanking plate (see photo'), which accepts a cable-mounted plug. Owners of A30x0 and A4000 models need not miss out. The four lines are available on the mini podule bus (see figure).



DIN41612 plug in place



A4000 Mini podule I²C connexions



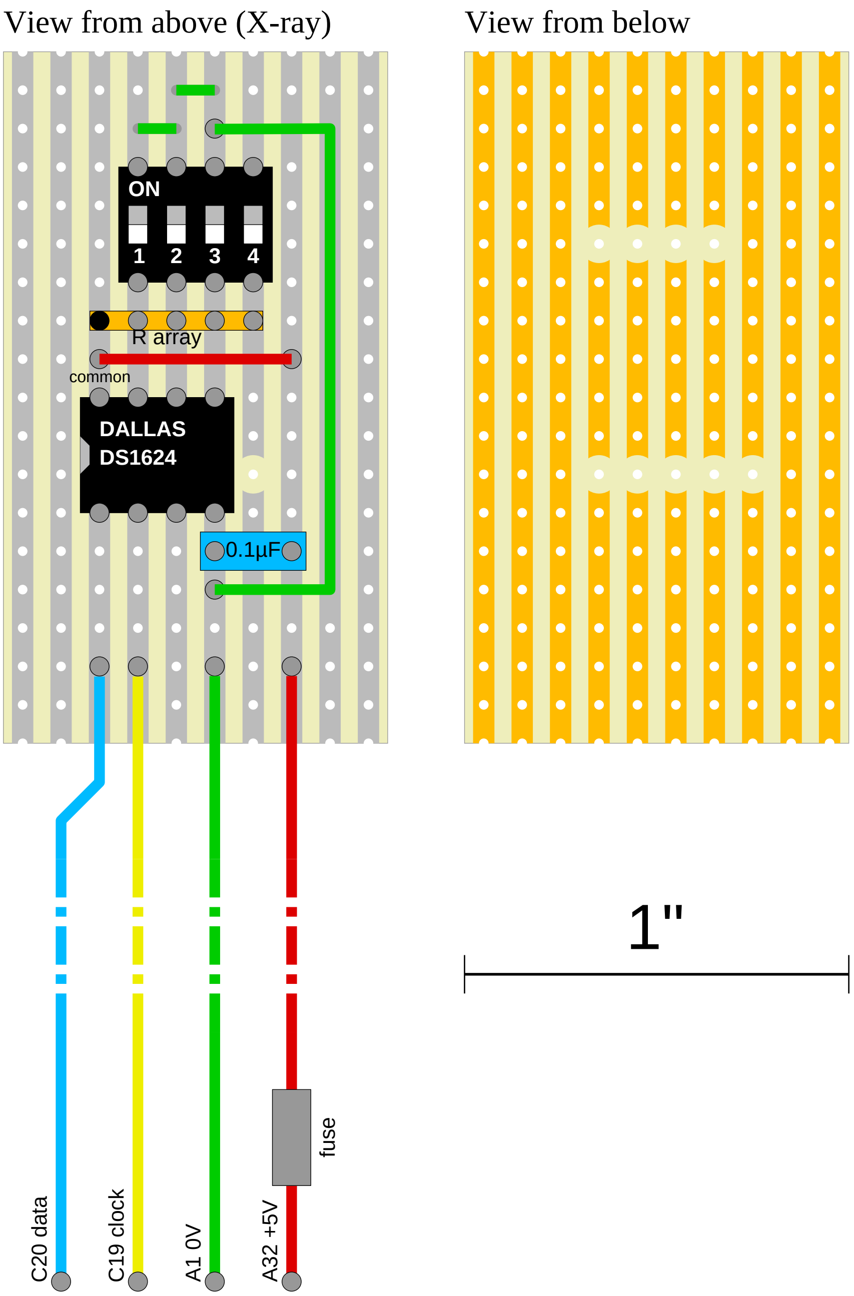
Pin	*Name*	*Function*	*DIN41612 Pin*
1	SDA	I^{2}C data	C 20
2	SCL	I^{2}C clock	C 19
3	NC	No connexion	
4	GND	Ground (0 V)	A 1
5	A2	Chip address	
6	A1	Chip address	
7	A0	Chip address	
8	V_{DD}	+5 V	A 32

DS1624 Pinout

Let's turn now to the sensor itself. As you can see from the pinout diagram there aren't many connexions to make. You will need to connect the four lines from the computer to the corresponding pins on each DS1624, connect a decoupling capacitor across the power pins, and set an address on the three address pins. In the diagram I show how to do this with a dual in-line switch array and pull-up resistors.

Probably the easiest way to get a sensor working is to use stripboard (Veroboard). You will need a piece similar in size to that shown in the diagram. Stripboard has conducting strips of copper on one side. In a few places these will have to be broken to form the correct circuit. There is a special tool for this, called a spot face cutter, but if you don't have one a drill bit (about 6mm or 1/4" diamter) used in a hand drill or just between your fingers, will do. To cut a break in the track turn the board over and place the cutter on the copper track (over a hole) and twist. Be ***sure*** that the track is fully broken. Even with the proper tool it is easy to leave some copper by mistake. A multi-meter will give a clear test of continuity. Don't drill so much that you go through the board! Cut the pattern of 9 breaks shown on the right half of the diagram.

Now turn the board back over to the component side. The left half of the diagram shows the relationship between the components and the cuts in the tracks underneath with an “x-ray vision” effect. Be sure to line everything up correctly. There should be a line of breaks between the two rows of pins on the DS1624 sensor and the pins on the DIP switch. The DIP switch is offset from the sensor by one column to the right. This places it over the three address pins, leaving the fourth switch spare. Rather than connecting the IC directly to the board you might prefer to use an IC socket.



DS1624 board from above (left) and below (right).

The resistor array has a spot on one end to indicate the pin connected to all four resistors. This goes to +5 V, above pin 8 of the sensor. The resistors then pull-up each of the address lines to +5 V, so long as the switches are not in the **on** position. If you can't get a resistor array individual resistors from each address pin to +5 V would be fine. The pins on the top row of the DIP switch should be linked with wire or just by adding solder between the pins on the copper side. They must also be connected to 0 V on pin 4 of the sensor. Thus, switching **on** will connect the address pin to 0 V. This is the reason for the resistors (in the array), so that they can pull down an address line without shorting the +5 V line. Current will flow through the resistor down to earth. I used 2k2 resistors, but with address 0 selected the temperature goes up slightly. I suspect heat dissipation from the resistor array, so a higher value might be better.

The final component is a 0.1 µF capacitor. Its purpose is to soak up high frequency *rubbish* on the power lines. Connect it as closely to the DS1624 as possible. For a really compact arrangement it could even go straight between the pins on the copper side. Using the layout in the diagram you need also a wire link to take +5 V from one side of the capacitor to pin 8 of the sensor.

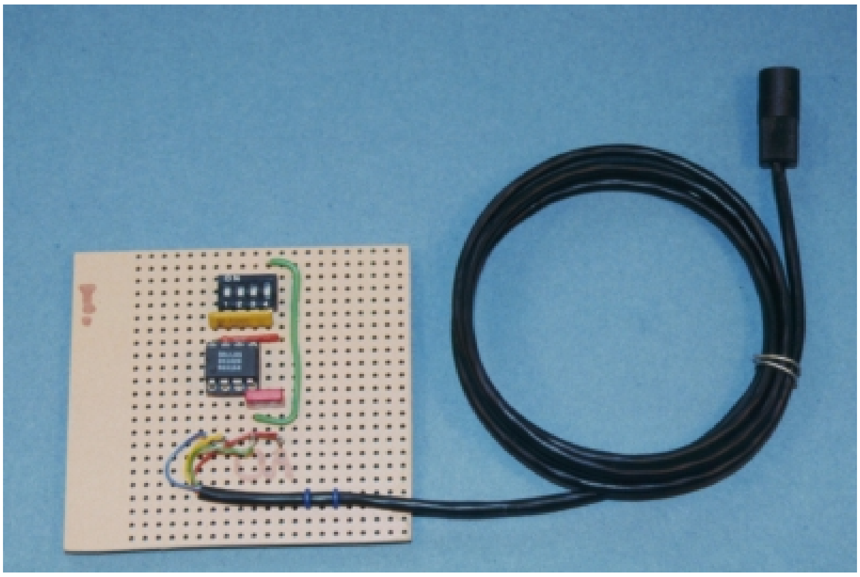
Finally, you should use terminal pins for the off-board connexions. Insert each pin from below and push home hard with, for example, the nose of a pair of long-nosed pliers. They take a fair bit of force to get them all the way in. Don't forget to solder them! The second photograph shows the finished board.

Before connecting to the computer backplane give everything a thorough visual inspection. Look for blobs of solder or joints that were larger than you intended, that may be shorting tracks together. If you have a multi-meter check for shorts between any of the four connecting wires (especially +5 V to 0 V). If everything checks out, connect the DIN connector to the podule backplane (with the power off).

If you prefer a more compact arrangement I have found that it's not hard to do the wiring on the back of an 8-pin DIL socket. This is especially so if you use address 7, as then you can bridge across the whole top row with one piece of wire connected to +5 V. Wire the decoupling capacitor from here to pin 4, along with the 0 V line. Connect SDA and SCL. You can cut away pin 3 of the socket. Wrap around with insulating tape, but leave space for air to circulate.

I you have the means to etch PCBs you could lay out a pattern with an etch resist pen or “Brady tape”. I have put a draw file of a set of patterns (one for each address) on the disc, which you could print out onto acetate sheet, if you have the necessary facilities.

Once you have a sensor(s) hooked-up, switch the computer back on, and run the test program. Double click it or run it from a command line. It will read the temperature from each connected device. I'll explain the working of this program in the next part.



Completed DS1624 board