# SMD Reflow Soldering Oven

## SMD boards hot from the oven

**It's usually possible to solder 'ordinary' SMD components using a low-power soldering iron and small-gauge solder. However, it's a completely different story when you have to solder a component in a BGA, CSP or similar package. Such components can actually only be soldered using a reflow soldering oven. Here we describe how a normal, inexpensive oven can be transformed into a reflow oven.**

Paul Goossens

The designers in the *Elektor Electronics* lab do a lot of soldering. It's thus hardly surprising that they aren't put off by a difficult soldering task. Nevertheless, our esteemed designers were briefly perplexed when they had to solder a FPGA in a BGA package (**Figure 1**). It was clearly impossible to tackle that task using a soldering iron. After a bit of experimenting (see the LabTalk article elsewhere in this issue), they decided it was high time to equip the lab with a reflow oven. Naturally, as true *Elektor Electronics* adepts they'd rather come up with something on their own than buy a ready-made solution. As a result of their efforts, our beloved pizza oven has now been converted into a genuine reflow oven.

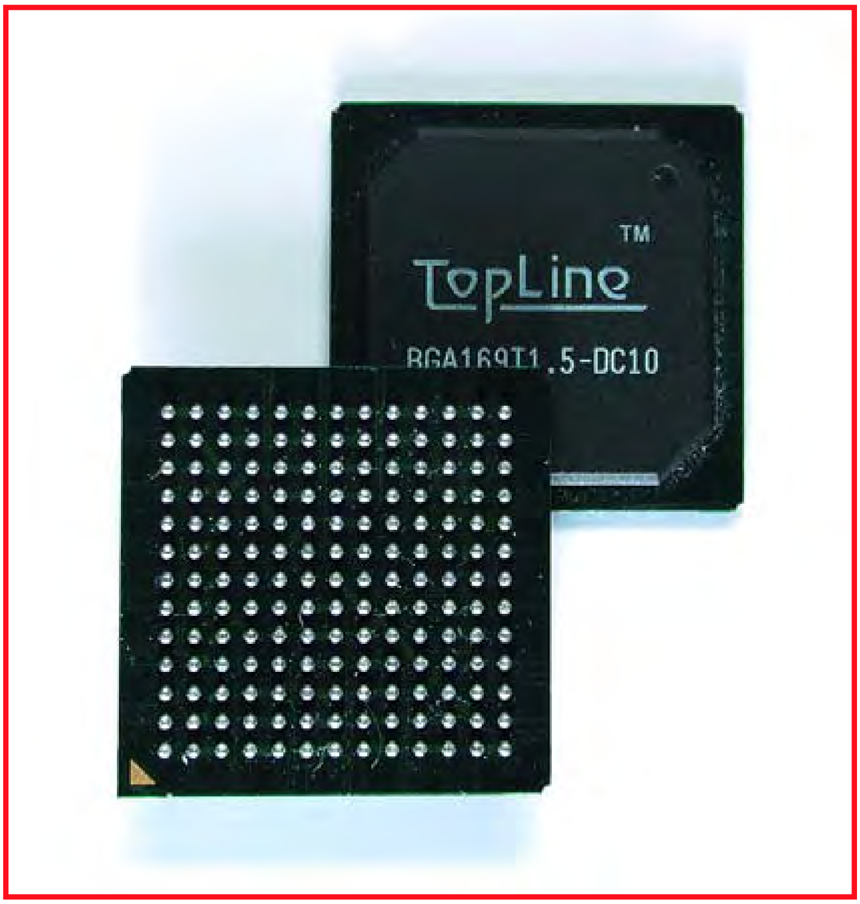



**Figure 1. A ball grid array (BGA). It hardly needs saying that it doesn't lend itself to hand soldering.**

## The reflow method

The reflow method can be used to solder components whose leads are inaccessible to a soldering iron. In the reflow method, a layer of solder paste is first applied to solder pads for the SMD components on the printed circuit board. The SMD components are then placed on the board with their leads in the solder paste. The actual soldering takes place in a reflow oven in five stages.

The soldering must be performed using a rather strict procedure. In the first stage, the temperature inside the reflow oven is raised to approximately 125 °C. This heating must be relatively gradual, as otherwise the solder paste will start bubbling and splatter tiny balls of solder over the circuit board. A rate of approximately 2 °C per second is fairly safe. This stage is called the 'pre-heat stage'.
The second stage is called the 'soak stage'. During the soak stage, the temperature is very slowly raised to approximately 175 °C. The purpose of this stage is to ensure that the circuit board and all the components on it are at nearly the same temperature. That prevents cracking or warping of the PCB or components during soldering. This stage also causes the flux to become activated. Activation of the flux means that it liquefies and coats the pads better.

Now that everything is nicely preheated, the actual soldering can start. The stage that comes next is called the 'reflow stage'. During this stage, the temperature in the oven is raised to the soldering temperature as quickly as possible. The 'dwell stage' starts when the soldering temperature is reached (guideline value 220–240 °C). The soldering temperature is maintained for several seconds during this stage. The balls of solder in the solder paste, which is now liquid, start melting and are drawn together by surface tension. The flux is forced outward by the surface tension, so only liquid solder is present between the SMD components and the circuit board, and the two layers of solder melt together. A pleasant side effect is that the components are usually pulled nicely flat against the pads by the surface tension of the solder. Any SMD components on the board that may be slightly tilted will thus level out during the soldering process. The final result is a trim, attractive circuit board with practically all of its components perfectly flat.

After the dwell stage, which lasts 10 to 15 seconds, it's time for the final, 'cool-down' stage. Not surprisingly, the temperature slowly decreases to room temperature during this stage. This must also take place fairly slowly, as otherwise there's a risk of cracks developing in the components and/or circuit board during this stage as well.

## DIY saves money

The price of a commercial reflow oven is outside the budget of most DIYers, but as we already mentioned, an ordinary oven can be converted into a reflow oven as explained below. For this purpose, you will need a (small) standalone oven dedicated to this use. Preparing food in an oven that is also used for reflow soldering is definitely a 'no-go'. When purchasing an oven, you should look for one with inner volume as small as possible but still large enough for the boards you plan to solder, and with the highest possible heating capacity. That makes it possible to raise the temperature inside the oven quickly, which is very important during the reflow stage.
The oven we used for this purpose has a volume of 18 litres and a rated power

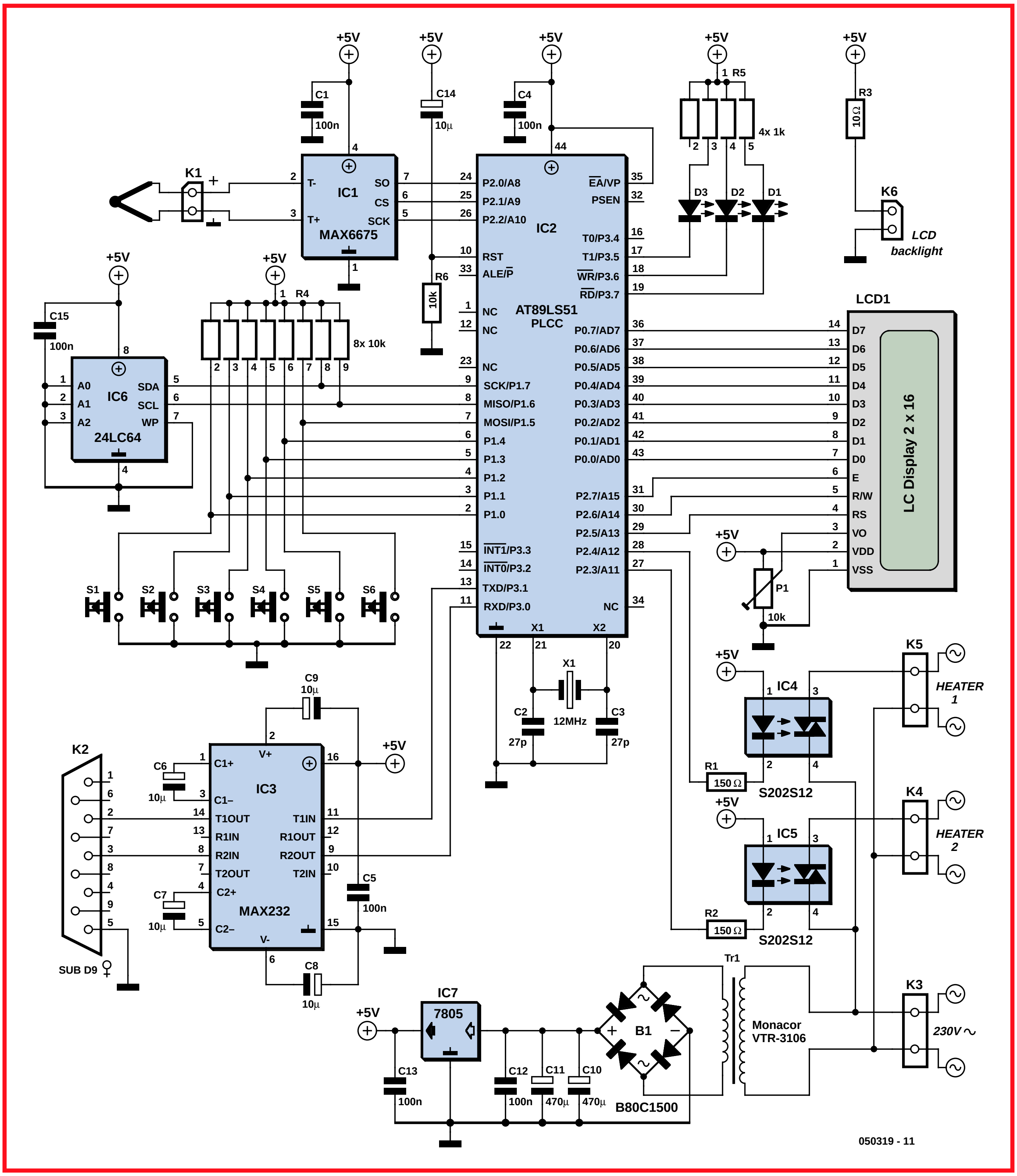



Figure 2. Our controller circuit is built around an Atmel AT80C52. It controls the entire process. The temperature can be read out on a PC via an RS232 interface.

of 1.8 kW. That amounts to a hefty 100 watts per litre. Such ovens are surprisingly inexpensive; our model cost only about 60 pounds.

The standard controller fitted in such ovens is totally useless for our purpose. It must be replaced by the circuit shown in **Figure 2.**

## A new controller

Although the controller we developed for the oven is relatively simple, it is highly effective and has several convenient extra features.

As usual, the circuit is built around a microcontroller – in this case a member of the 8051 family (Atmel AT89C52). This microcontroller contains 4 KB of flash memory for the firmware and the usual peripheral functions found in practically every 8051 derivative.

A thermocouple (see inset) is used as the temperature sensor. It is fitted in the oven with its tip located roughly in

# About the controller

Controlling the oven temperature may appear to be rather easy. To put it simply, you might think that all you have to do is switch on the heating elements when the temperature is too low. Otherwise the heating elements must be switched off. But as so often happens, there's more to it than meets the eye.

A simple 'on–off' control is far from ideal in actual practice. The heating elements do not immediately stop radiating heat when they are switched off. The internal temperature of the elements is higher than the air temperature, so they keep transferring thermal energy until both temperatures are the same. That causes the temperature in the oven to continue rising for a while. This undesired temperature increase is called 'overshoot'.

When the temperature subsequently drops below the set temperature, it takes a little while before the switched-on heating elements become hot enough to raise the temperature. The temperature thus continues to drop for a while before it heads back toward the set temperature. That is called 'undershoot'.

These phenomena are well known to measurement and control engineers. The most obvious way to deal with this sort of situation is to use what is called a 'PID controller'. Such a controller requires at least three parameters to adjust the control loop.

However, both of the above-mentioned types of control (simply switching the elements on and off or using a PID controller) are unsuitable for this application. The first approach causes excessive overshoot, with the result that the temperature cannot be controlled with sufficient accuracy. The second approach requires a certain amount of understanding of control circuits, because the user must provide the three parameter values needed to properly adjust the controller for his or her particular oven. That's not exactly what we call 'user-friendly'.

The solution we finally devised in our lab takes a different approach. It provides surprisingly good results without requiring the user to have any understanding of control systems. First, we measure the magnitude of the overshoot when the oven is heated from 50 °C to 100 °C. When a temperature of 100 °C is reached, we measure how fast the temperature is rising at that instant (the slope of the curve, in other words). The heating elements are switched off at that point, and we measure how much the temperature continues to increase. The amount of overshoot is divided by the slope of the curve at 100 °C, and the result is stored in memory. For readers who want to examine the source code, this value is found in the variable with the somewhat misleading name 'overshoot'.

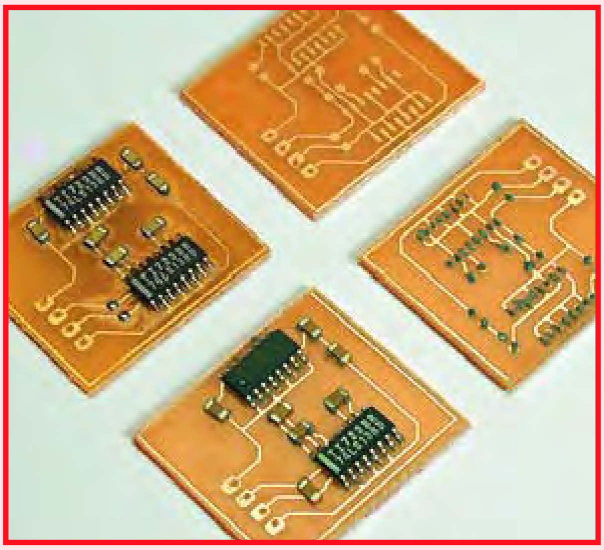
We also assume that if the temperature increases less quickly while the oven is heating up, which can occur for various reasons, the overshoot will be proportionally smaller.

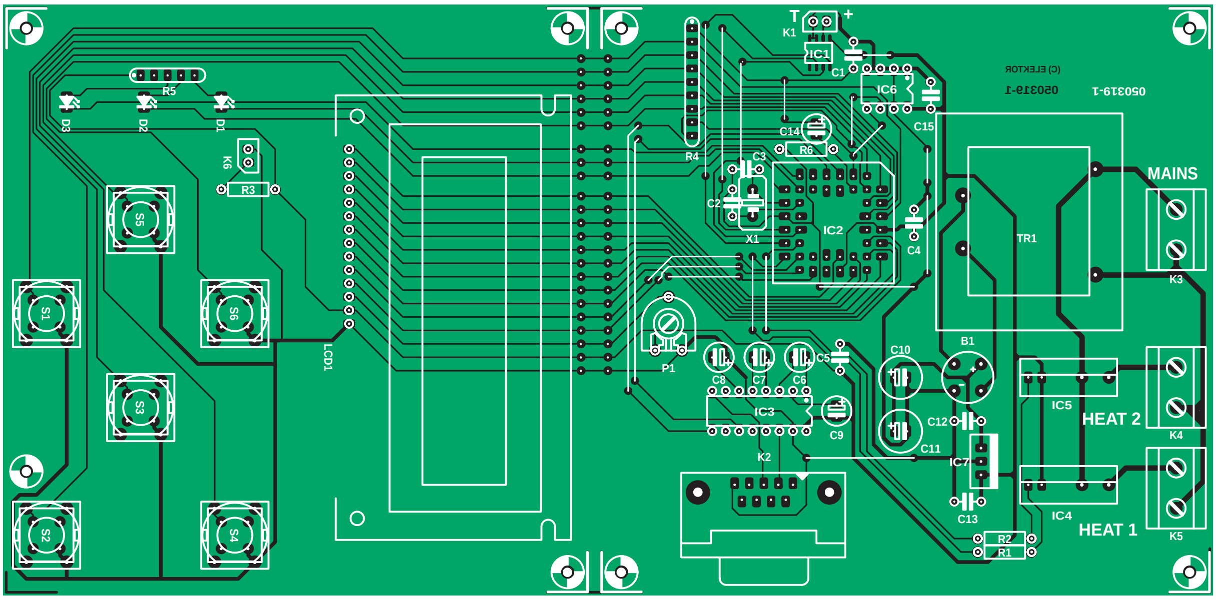
Once per second, our controller circuit attempts to estimate how much the temperature would continue to rise if the heating elements were switched off at that instant. For this purpose, we measure the rate of rise of the temperature using a simple digital filter. This rate of increase is stored in the variable 'deltaT'. The value of deltaT is then multiplied by the calibration value (held in 'overshoot'). That gives a reasonably accurate estimate of the anticipated overshoot at any given time. As soon as the anticipated final temperature is equal to or greater than the desired temperature, the heating elements are switched off. If the estimated final temperature is less than the desired temperature, the heating elements are switched on (or remain on).

Besides reducing or eliminating the overshoot of the oven, this technique also counteracts undershoot. That's because the value of deltaT becomes negative when the oven is cooling down, so the routine calculates the undershoot instead of the overshoot.

Devising a control technique is only half the battle; it still has to be tested to see whether it works properly in practice. In our tests, we measured a maximum overshoot of 2 °C, and in many cases it was only 1 °C. At relatively high temperatures (above 200 °C), the overshoot was actually less than 1 °C. That's more than adequate for our purposes.

Besides having good operational characteristics, this control technique makes it easy to automate the calibration process. It thus occurs fully automatically in our oven when the user selects and runs CALIBRATE in the main menu. And that's what we call 'user-friendly'






70%

## COMPONENTS LIST

**Resistors:**
R1,R2 = 150Ω
R3 = 10Ω
R4 = SIL array 8 x 10kΩ
R5 = SIL array 4 x 1kΩ
R6 = 10kΩ
P1 = 10kΩ

**Capacitors:**
C1,C4,C5,C12,C13,C15 = 100nF
C2,C3 = 27pF
C6-C9,C14 = 10µF 16V radial
C10,C11 = 470µF 16V radial

**Semiconductors:**
B1 = B80C1500 bridge rectifier, 80V piv, 1.5A
D1,D2,D3 = LED, red, low-current
IC1 = MAX6675
IC2 = AT89C52/24JI, programmed, order code **050319-41**
IC3 = MAX232
IC4,IC5 = S202S12
IC6 = 24LC64
IC7 = 7805

**Miscellaneous:**
K1 = connector for thermocouple
Thermocouple, K-type
K2 = 9-way sub-D socket (female), PCB mount
K3,K4,K5 = 2-way PCB terminal block, lead pitch 7.5mm
K6 = connection for LCD backlight LCD1 = LCD module, 2x16 characters, e.g. order code **050319-72** or PLED version **050319-73**
S1-S6 = pushbutton, ITT type D6-R
Tr1 = mains transformer, primary 230V, secondary. 6V (e.g. Monacor/Monarch VTR-3106)
X1 = 12 MHz quartz crystal
PCB, ref. 030519-1 from The PCBShop
Disk, source and hex code files, order code **030519-11**
17 wire links

the middle of the oven. Ensure that the thermocouple remains electrically isolated from the rest of the oven, in order to avoid creating a hazardous situation. The two leads of the thermocouple are connected to IC1, a MAX6675. This IC computes the temperature at the tip of the thermocouple based on the voltage generated by the thermocouple and the ambient temperature. The microcontroller can query the temperature via a serial interface.

IC4 and IC5 are connected to the microcontroller via resistors. These two ICs are optotriacs with integrated zero-crossing detection and snubber networks. That makes them very easy to drive from the controller. Power is applied to the two heating elements as necessary via these two ICs.

The controller contains several settings that must be stored in an EEPROM. IC6 is included for that purpose. Alert readers may quickly come to the conclusion that 64 KB is rather generous for the number of settings to be stored. That's certainly true, but the circuit does in fact need that much memory because it is also used to store the measured temperature once per second during the soldering process. That allows the operation of the oven and the corresponding temperature profile during soldering to be examined afterwards.

This information must be sent to a PC in some way or another. Here we use an old faithful: the RS232 serial port. As usual, the port is implemented using a MAX232 IC and associated components.

Pushbutton switches S1–S6 provide the operating controls for the oven controller. LEDs D1–D3 and LCD1 keep the operator informed while the oven is in use.

Not much needs be said about the power supply. It is very basic and perfectly ordinary. Note that no fuse is included on the circuit board. An external fuse **must** be used for the input voltage. Besides a normal fuse, which is usually located at the rear of the equipment, a thermal cutout is also necessary. It must be fitted such that it switches everything off if the temperature of the oven becomes too high. That prevents the oven from overheating if something goes wrong, which in turn prevents everything from catching fire.

The (fused) 230-V supply voltage is connected to K3. The two heating elements are connected to K4 and K5.

## Installation

Installation of the controller in the oven will be different for each type of oven.

That means we can't provide installation instructions along the line of 'first loosen the four screws on the bottom', etc. In this regard, you'll have to rely on your own talent for improvisation. That also applies to the printed circuit board. Of course, you can use our design (**Figure 3**), but it probably won't fit in oven models (and it's anyway rather large; its full-size artwork files can be downloaded free of charge from our website). That means you will most likely have to design your own circuit board to fit in your oven. We assume that anyone who wants to solder components in BGA packages and the like is also capable of designing a circuit board for the controller circuit. In any case, our board design can serve as a starting point.
As regards safety, this circuit operates at 230 VAC, which means it can pose a **fatal hazard** if it is built or used improperly. As already mentioned, the mains voltage must be fused before it enters the circuit, and the fuse must be selected based on the maximum rated power of the heating elements. A separate thermal cutout is also essential.

## Operation

We've kept the user operation aspects of our controller quite simple. When the unit starts up, a welcome message appears on the LCD and the microcontroller performs various checks. For instance, a warning message will be displayed if the EEPROM is missing, and a check is made to verify that a valid calibration value is stored in the EEPROM. If a valid calibration value is not found, the controller must first be calibrated. A message will be displayed to indicate that automatic calibration can be started by pressing ENTER.
If everything goes properly, the main menu will be displayed after calibration is finished. In the main menu, you can use the ↑ and ↓ buttons to select START, EDIT, LOG or CALIBRATE. Press the ENTER button to start the selected function.

## Calibration

The CALIBRATE function performs a fully automatic measurement of the most important characteristic of the oven: its overshoot (see the 'About the controller' inset). Before you start a calibration, make sure the oven door is tightly closed and nothing is inside the oven. The controller will heat the oven

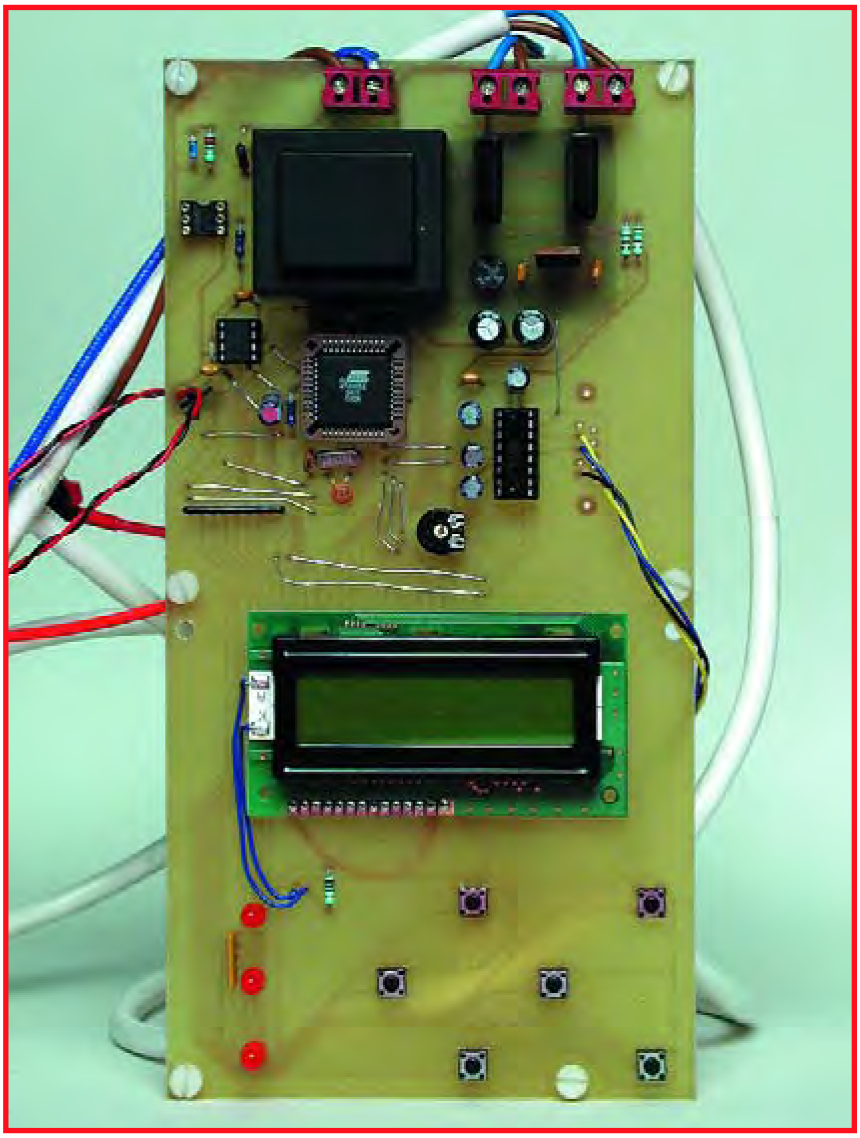


Figure 3. Our prototype circuit board isn't likely to win a prize for pretty design. The true-size artwork (pdf) files are available free of charge from our website.

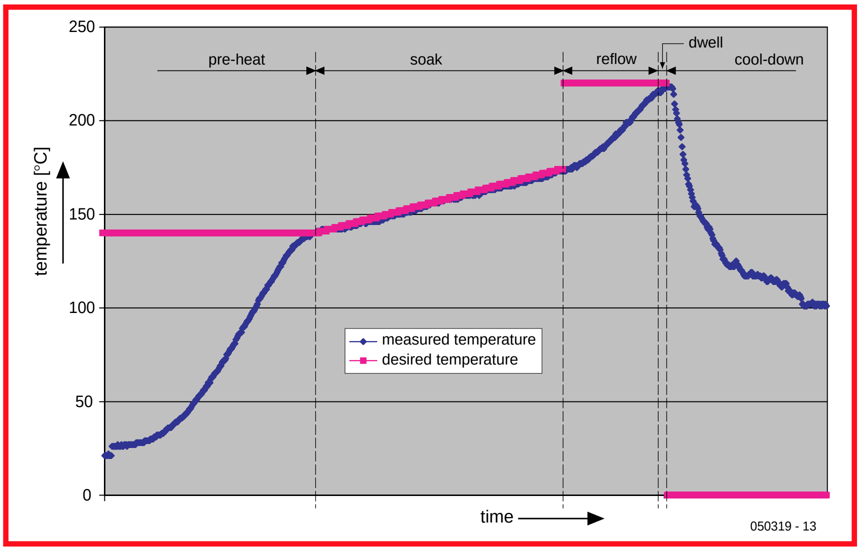



Figure 4. The actual temperature curve (blue) and the desired temperature (red). We added extra rock-wool insulation to the oven to increase its heating rate.

## Thermocouples

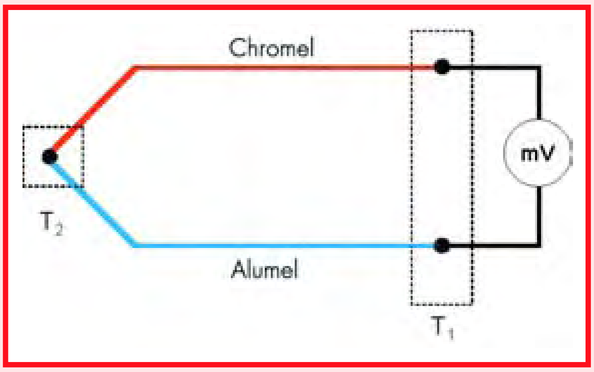
A thermocouple is a sensor that generates an electrical potential related to the temperature. The operating principle of the sensor is base on the fact that any electrical junction between two different metals generates an electrical potential that depends on the temperature and the metals that are used. The principle applies equally well if three metals are used. In that case, there are two junctions in series, and the net potential results from the series addition of the two individual potentials. For example, if a copper–iron junction is in series with an iron–tin junction, the net potential is the same as for a copper– tin junction. However, that is only true if both junctions are at the same temperature.

With a 'K-type' thermocouple, which is what we used for this design, the metals forming the thermocouple are always Chromel (on the positive side) and Alumel (on the negative side). The voltage generated by this combination is approximately 4 mV / 100 °C. K-type thermocouples can withstand temperatures of 1000 °C or more without suffering any damage.

However, thermocouples also have an inherent drawback: connecting a thermocouple to a circuit inevitably creates additional junctions between different metals, which naturally generate their own thermal potentials. Strictly speaking, the potential at the output of a thermocouple is not a function of the absolute temperature at the measurement point, but instead a function of the difference between the temperature at the measurement point (the hot junction) and the temperature at the connection point (the cold junction).

In our circuit, the cold junction is located on the circuit board. If we measure the temperature of the circuit board, we can calculate the actual temperature at the measurement point from the voltage generated by the thermocouple and the temperature of the circuit board.

Fortunately, the MAX6675 IC used in the circuit automatically looks after all that for us. It measures the voltage from the thermocouple and converts it into a temperature difference between the hot junction and the cold junction. We assume that the temperature inside the IC is essentially the same as the temperature of the circuit board (the cold junction). If we add the chip temperature to the computed difference




between the temperatures of the hot junction and the cold junction, we obtain the temperature of the hot junction, which it the temperature at the point we want to measure.

This assumption regarding the cold-junction temperature is why it's important for the thermocouple leads (which are made from Chromel and Alumel) to be soldered to the circuit board as close to the IC as possible. That also means that if you have to extend the thermocouple leads, you must use Chromel wire for the positive side and Alumel wire for the negative side. If you use normal copper wire, the junctions with the copper wire will form additional measurement points. That will create a measurement error if the temperatures of those junctions are not the same as the temperature of the circuit board. It's thus best to buy a thermocouple with sufficiently long leads.

It shouldn't be necessary to say this, but we'll say it anyhow for good measure: make sure your thermocouple is made from Chromel and Alumel. In other words, ensure that it is a K-type thermocouple. Other types of thermocouples generate different potentials, which will result in incorrect measurements.

In case of doubt, you can easily check the measurements. If you put the thermocouple in ice water, the circuit should indicate a temperature of approximately 0 °C. If you put the thermocouple in boiling water, the circuit should indicate a temperature of approximately 100 °C. If you have any doubt, we definitely recommend making this test.

to 100 °C and switch off the heating elements. The temperature inside the oven will continue rising for a short while until it reaches some maximum value. When the temperature has just about stopped rising, the microcontroller calculates the associated overshoot value. That value is stored in the EEPROM, so the calibration routine does not have to be repeated every time the oven is used.

### Edit

Different types of solder paste may have different rated soldering temperatures. The melting temperature and the temperature needed to activate the flux depend on the composition of the solder paste. You also have to consult the data sheets of the components you use to determine the requirements for the temperature profile (in other words, the settings). You can deviate from the

```
STAGE, TEMPERATURE, DESIRED TEMPERATURE, HEATER 1, HEATER 2
```

| | | | | | | |
|---|---|---|---|---|---|---|
| **STAGE :** | **0 = COOL** | **1 = PREHEAT** | **2 = SOAK** | **3 = REFLOW** | **4 = DWELL** | **5 = COOL** |
| **HEATING :** | **0 = OFF** | **50 = ON** | | | | |

**Figure 5. The format of the data sent to the PC. 'Stage' indicates the progress of the process, 'Temperature' indicates the current temperature in the oven, 'Desired temperature' is self-explanatory, and 'Heater 1' and 'Heater 2' indicate whether the heating elements in question are switched on.**

manufacturer's requirements in actual practice, but if you do, there's no guarantee that the components will still be intact after soldering.
The temperature profile generated with the settings we used for our controller is shown in **Figure 4.** The values shown in the figure are guidelines; we have achieved good results with them on our lab.

## Log

The LOG menu lets you enable or disable the built-in temperature logging function. The logging function is disabled by default when the unit is switched on. That avoids rewriting the contents of the EEPROM any more often than necessary, which helps prolong the useful life of the EEPROM.
You can enable logging with the ↑ button and disable it with the ↓ button. If you press the → button, the content of the temperature log for the most recently logged soldering cycle will be transmitted via the serial port. The settings used for the serial port are 4800, 8, N and 1 for the baud rate, number of data bits, parity, and number of stop bits.
The stored values are separated by [RETURN] codes. You can use HyperTerminal or any other suitable communications program to store the data stream in a file. It's a good idea to assign the extension '.csv' to the file name. That makes it possible to open the file using a spreadsheet program. In a program such as Excel, you can then generate a chart from the data to show the actual temperature plot (temperature profile).
Press [ESC] to return to the main menu.

## Start

The START function does exactly what it says: it starts the soldering process. The display provides a convenient indication of the progress of the process. If anything goes wrong, you can always stop the soldering process by pressing the [ESC] button.
Another way to modify the process is to use the ↑ and ↓ buttons. You can use these buttons to increase or decrease the desired temperature while the oven is operating.
Another handy feature is that the most significant data is transmitted via the serial port during the soldering process. The same settings are used here as for reading out the EEPROM, namely 4800, 8, N and 1 for the baud rate, number of data bits, parity, and number of stop bits.
The format of this data is shown in **Figure 5.** The data can also be stored on hard disk using HyperTerminal and then processed in a spreadsheet program. The temperature profile shown in **Figure 4** was generated in this manner.

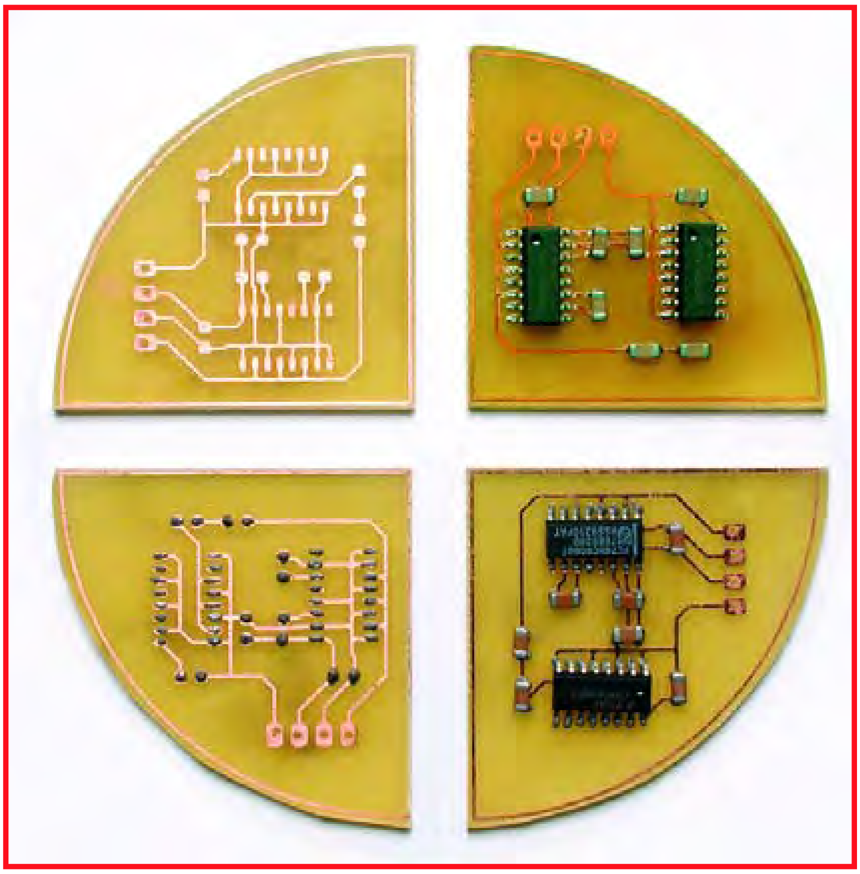


**Figure 6. The four stages in PCB form: without solder paste, with solder paste, components not yet soldered, and finished circuit board.**

## Practical experience

We've used our SMD oven successfully several times already for soldering prototypes. However, each time we had to open the door at the end of the soldering process to reduce the cool-down time.
Most inexpensive ovens don't have a fan to help cool down the oven. There's also no provision in our circuit for a fan. For people who only want to solder the occasional circuit board, it shouldn't cause a big problem if someone has to keep an eye on the oven and open the door when the process is finished.
It's a good idea to make sure the solder paste you buy is suitable for use at the lowest possible temperature. New solder paste compounds comply with the RoHS requirements, which among other things means they do not contain any lead. That makes the melting point of the solder somewhat higher. The situation in this area is currently rather dynamic, so you should remain on the lookout for solder paste compounds with relatively low melting points.
And while we're on the subject of solder paste, it's recommended to store solder paste in a refrigerator to prolong its useful life.

(050319-1)

**This circuit operates at 230 VAC, which means it can pose a fatal hazard if it is built or used improperly.**