Meet

Hidden cos

Dr. Erik Lins and Steffen Fuchs (Germany)

Longwinded discussions are costing commerce millions. Our meeting cost timer reveals the true cost of 'free speech' at business gatherings. The large seven-segment display lets you watch your money running away, whilst a single turn-and-push control switch makes operation of this attractive timer child's play.

The credit crunch affecting most firms currently is not the only reason why they should be saving money. Cutting down on direct costs often also leads to significant productivity improvements. Frequently the actual outlay on individual processes is not known in any great detail, meaning the potential savings of some new measure may not be easy to assess. This applies in particularly to purely organisational procedures, in which we must include those beloved meetings.

A distinct and ingrained meetings culture exists in most large organisations. Participating in all this are many (expensive) executives and not infrequently also plenty of (more or less expensive) fellow workers also play roles, even though they are generally involved in only a few points on the agenda.

In itself the duration of a meeting provides only a sketchy idea of the costs involved. Good intentions for holding productive and time-efficient meetings are all too often defeated by the time wasted afterwards on debriefing and reappraisal. How effective it would be if everyone could see the true cost of time wasted in meetings!

Practicalities

Even though notebook computers and data projectors are nearly always to hand in conference rooms and could be used to display the costing we have been discussing, the fact is that the stark reality of a large bright-red seven-segment display device would have significantly greater impact. In any case speakers would not take kindly to having a cost-clock display overlaid onto their carefully crafted PowerPoint presentations! A standalone solution would make the choice of a prominent location for displaying the cost of meetings significantly simpler.

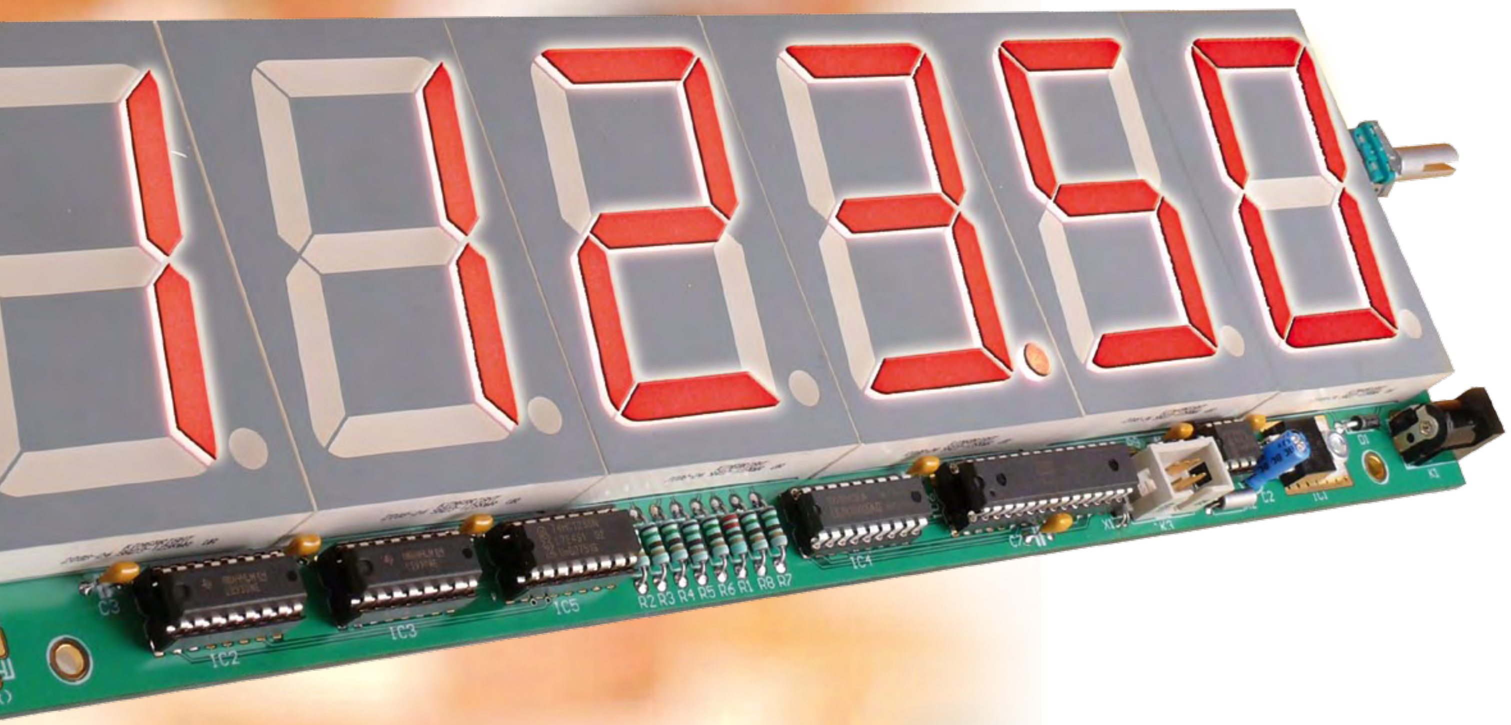
The device would have to be easy to operate and confine itself to the bare essentials. Parameters to preset would include the number of participants and their average hourly rate of pay. Anything more than a start, stop and reset function would be unnecessary. The number of digits for displaying the cost has been defined as six. With two digits behind the decimal point we can show up to 9999.99 pounds (or euro, or whatever currency you choose). If this total is exceeded regularly, things will presumably not be looking very good for your firm in future. That said, digits after the decimal point are not absolutely necessary but then you'd lose the effect of the decimal point flashing every second, which certainly underlines the impression of time flying by (and money running away).

Operation

At switch-on the meeting cost timer announces itself briefly with the word HELLO and then invites you to enter the two vital parameters. These are the number of participants (display prompt shows PAR) and average salary cost (display shows EUR but of course your figure can be in any currency of your choice). Data is entered using the combined rotary switch

ing Cost Timer

ts become totally transparent!



The recipe for productive meetings

by Markus Söhngen

Office jokes like "Feeling bored? Then call a meeting!" or "How about coffee at the firm's expense? Set up a meeting!" are no longer amusing for most organisations. Meetings can of course have a serious purpose: to agree a new policy or an important decision, for conflict resolution or a team celebration after achieving an important goal.

Here are the ten most important rules for meetings:

1. Spend some time on the invitation. Give people precise information on the date, time, duration, agenda and targets but keep the fine detail under wraps until the meeting.
2. The place you choose can determine how successful the meeting turns out. The 'right' location may work wonders!
3. The time of day should be selected carefully. Making crucial strategy discussions as soon as people have come into work may be unwise.
4. Invitees should feel their participation is crucial. 'The fewer the better' is only a half-truth and what you need is the ***right*** number and the ***right*** people.
5. Speak in a clear and engaging way, setting out background issues, the sequence of events to date, successes so far and the desired outcome. Bring any fruitless discussion to a rapid end.
6. If the meeting runs out of steam you can use this hiatus to offer a preliminary conclusion. Stick to your planned remit, time and targets.
7. Summing up when you reach intermediate conclusions is vital.
8. Motivation must continue until the very end of the meeting. Exploit opportunities for humour — but with care!
9. Always keep an eye on the time. Are you leading a constructive meeting or is it developing into just an expensive hot air session?
10. When you come to wind up, recapitulate all the solutions that you have succeeded in finding. Compare your targets and the results you have achieved. Give the participants plenty of feedback. Always close your productive meeting with thanks and motivate the participants for the next meeting!

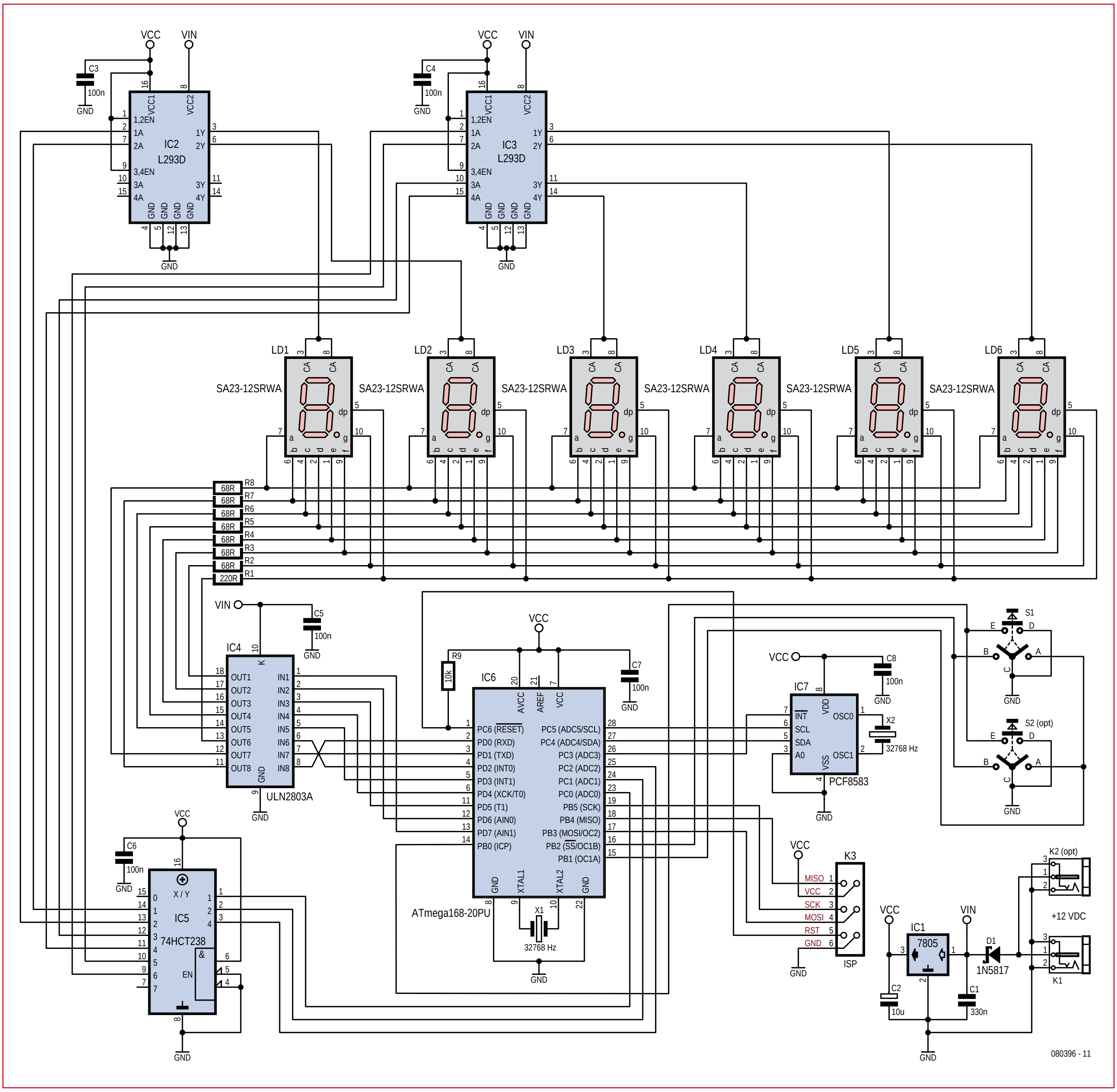


Figure 1. The ATmega168 controls six large 7-segment display characters.

(encoder) and pushbutton. Turning the knob increments or decreases the figure and pressing it enters the amount, ready for entering the second parameter. Following entry the display first pauses at 0.00 currency units. Pressing the switch once more triggers the timer, which will then increment until the clock is stopped by pressing a second time. A third press on the switch resets the timer to 0.00. Pressing the switch at any time for more than two seconds reverts to parameter entry, enabling you to re-enter the parameters to change the number of participants and salary cost.

The circuit

When we selected the components for this project we kept in mind constantly the need to choose ones that were readily available from the more popular catalogue distributors. All parts used have straightforward wire leads or solder pins, to make replicating the design as easy as possible.

Figure 1 shows the circuit of the meeting cost timer, where besides the 70 mm-tall 7-segment display you will spot many familiar 'old friends'. A coaxial-type power supply connector means that many common 'wall-wart' mains adapters will suit without further ado. A Schottky diode protects the circuitry against accidental polarity reversal of the external voltage on its way to the 7805 linear voltage regulator. As the 7-segment displays are connected directly to the external supply voltage, this latter must not exceed 12 V. If you intend to use a higher voltage then the series resistors of the cathodes must be changed (see below).

The 'brain' of the device is an ATmega168 microcontroller from the Atmel AVR range, which on the one hand takes care of setting the base parameters plus the start, stop and reset functions and on the other sees to controlling the multiplexing for the 7-segment LED displays. The controller is programmed using an ISP connector wired to the standard 6-way pinout, meaning that any common AVR-ISP programmer can be used for this.
An optional I^2C real-time clock PCF8583 is not required at this stage but could be retrofitted subsequently if you felt the need to add a conventional time display option to the clock.

Data entry is by means of a rotary coder with additional press-switch function (S1 or S2). Twin output signals from the encoder provide two phase-shifted square-wave signals. The sequence of the rising and falling edges provides the number of steps and the directional information. Turning and pressing the encoder triggers interrupts on the I/O pins connected to the microcontroller. The corresponding interrupt service routine evaluates the pins and controls the onward flow of the program.

Reference timing for the display is provided by a 32768-Hz clock crystal, to which the asynchronous oscillator (Timer 2) of the ATmega168 is slaved. This provides a timebase accurate to a second for calculating the cost. The timebase for multiplexing the 7-segment display modules relies on the internal 8-MHz R-C oscillator of the microcontroller. Corresponding prescalers create a timebase of 488 Hz, so that the human eye is not aware of the time-displaced presentation caused by the fact that each segment is activated at more or less 80 Hz (= 488 Hz / 6 displays).

Multiplexing

Were we to attempt to drive the six 7-segment devices in parallel without resorting to multiplexing, this would require a microcontroller with at least 42 I/O pins. Furthermore, large high-brightness 7-segment displays generally use not just one LED per segment but four LEDs in a row (as in the type used here). This demands four times the voltage, in other words a good 8 V. Each I/O pin would then need a low-side (open collector/open drain) driver too. That would mean 42 drivers plus 6 for the decimal points.

From a circuitry point of view it is far simpler to control the displays by time displacement. The displays are provided with a common anode and separately connected cathodes. We then employ multiplexing to connect a positive voltage to each of the 7-segment display devices in sequence, using one high-side driver each. The respective cathodes of the 7-segment displays are connected to one another so that after each time they are advanced, the appropriate bit pattern is applied to the cathodes of the active displays. In this way our need for driver devices is reduced from 48 low-side drivers to just eight low-side drivers and six high-side drivers and correspondingly only 14 I/O pins on the microcontroller.
We can save three more I/O pins by using a 3:8 demultiplexer to select the high-side driver, as then no more than one display is ever active at the same time. Selection is now performed by the bit pattern of the three I/O pins. The output Y0 belonging to bit pattern 000 is not used, to prevent one of the displays being activated by an unprogrammed controller (all I/O pins are high-impedance). The last output Y7 is also unused, since only six display devices are in operation.
For the high-side drivers we are employing two L293D H-bridge drivers. Designed primarily for controlling DC motors, each L293D gives us four push-pull outputs (to enable their use as an H-bridge) that can handle currents of up to 600 mA. For our purpose with six displays we need one and a half L293D chips (the two remaining outputs are left open). As the L293D has active-high inputs the demultiplexer has got to be a 74HCT238 with positive output logic rather than the more commonly used 74HCT138 with negated outputs.

A ULN2803A Darlington transistor array with eight outputs, used as a low-side driver, takes care of controlling the eight cathodes (seven segments plus decimal point). The emitters of the eight transistors are commoned internally and are connected to GND externally, so that each input can switch one of the outputs to GND. At this point we cannot save any more I/O pins (by using a demultiplexer for instance), as this might lead to several segments lighting at once.
The series resistors for the segments are configured for the widely used supply voltage of 12 V. Each segment accounts for four times the voltage requirement of a single LED (about 1.85 V), meaning that around 4.6 V is unaccounted for. The 68 ohm series resistors limit the current flowing through each segment to around 67 mA / 6 = 11 mA (average value). This is slightly below the nominal value according to the data sheet but should nevertheless be perfectly bright enough. The decimal point employs only two LEDs connected in series, so here a different series resistor has to be used. In theory the 220 ohms used is too high and limits the current to just 6 mA or so, but in practice this provides the same 'perceived' brightness as the other segments to users.

Quick project specs

- 70 mm tall 7-segment displays
- Maximum total displayed: 9999.99
- Single switch operation using rotary encoder with press-button function
- Connections and operating control can be mounted on either the right or left-hand side of the PCB
- Controller: ATMega168-20PU
- ISP adapter for programming
- Power supply using 12 V / 800 mA mains adapter

Meetings run up vast bills

In The Netherlands, a study was made on this controversial topic. At the end of 2007 their first 'conference barometer' indicated that meetings in the Netherlands alone (population around 16 million) cost the country around 60 billion euro (£ 51 billion) a year!

The average Dutch employee spent some three and a half hours a week in meetings, the study revealed. The larger the organisation and the higher the employee's function, the more time was spent in discussions. Of the annual cost mentioned, around half was accounted for by the meetings themselves. Another 27 per cent was spent on preparations and the remainder went on travel costs.

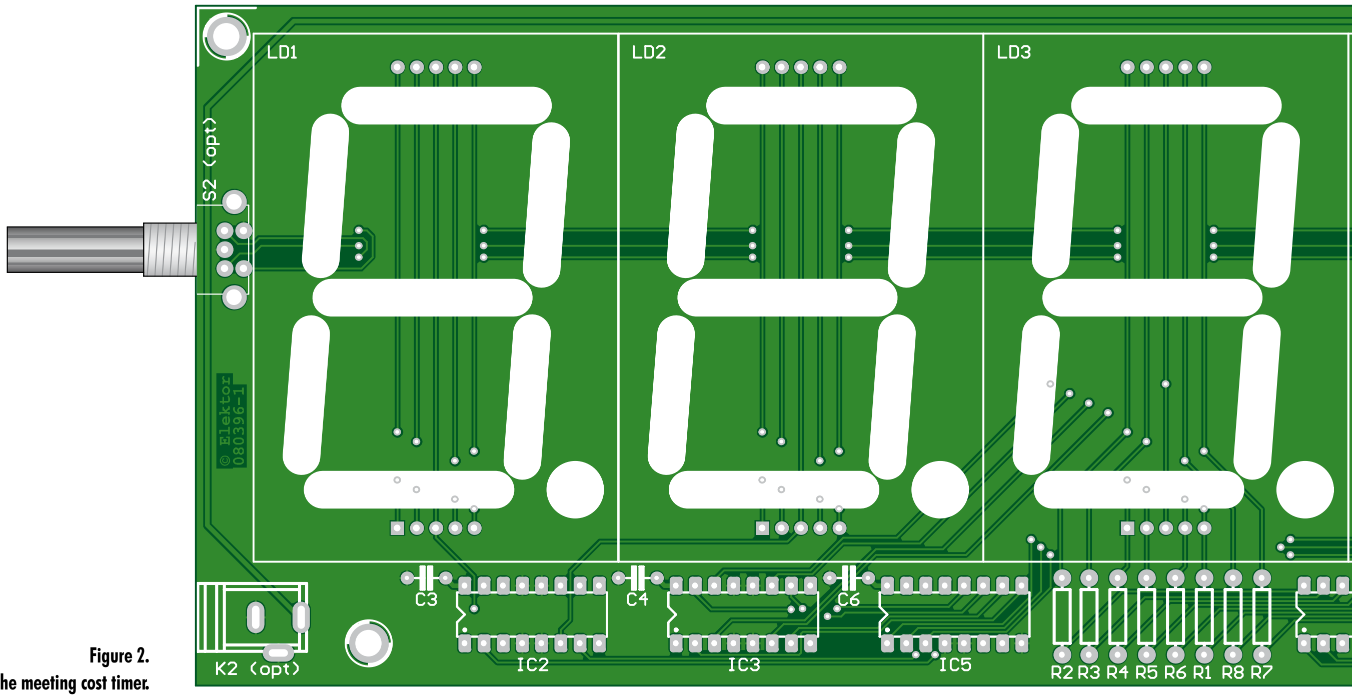


Figure 2.
The PCB of the meeting cost timer.

Software

Saving drivers for multiplexing is balanced by greater complexity in the software, on account of the constant need to shuffle the 7-segment displays and apply the correct bit patterns to the cathodes in order to give the impression of an always-on display.

The software of the meeting cost timer was developed using the free WinAVR C compiler along with Atmel's gratis AVR Studio (as a development environment and for in-system programming of the microcontroller).

For multiplexing the 7-segment displays the Timer 0 interrupt routine is responsible. Prescalers and the Compare Match Register are used to set the timer to 488 Hz. At each call a display counter is counted up from zero to five and matching values for the demultiplexer are output on port C to the corresponding display. The cathode bit pattern is made ready correspondingly at port D.

Also determined in the Timer 0 interrupt routine is how long the switch has been pressed. This must be for a minimum of around 120 ms (to debounce the switch) for the program to react. If the button is depressed for more than a second, the program changes into parameter input mode.Recognition of switch depression and rotation of the encoder is handled by the Pin Change 0 interrupt routine. This interrupt routine is invoked automatically by pressing or rotating the control switch; it sets corresponding flags that are then evaluated in the Timer 0 interrupt routine and in the main program.

The program needs to dwell as briefly as possible in the interrupt routines to avoid delaying the rest of the program flow. For this reason the interrupt routines generally do no more than poll statuses and set flags correspondingly, leaving the main program to handle the actual reaction to status changes and the consequent execution.

The Timer 2 interrupt routine is invoked exactly once a second. All it does is to increment the seconds counter of the cost calculation.

The first operation of the main program is generating the welcome message and inviting initial parameter entry. Next comes an endless loop of evaluating the flags for the press switch. A short depression starts and stops the cost display or resets this to zero.

COMPONENT LIST

Resistors

(all 0.25W, 5%)
R1 = 220Ω
R2-R8 = 150Ω
R9 = 10kΩ

Capacitors

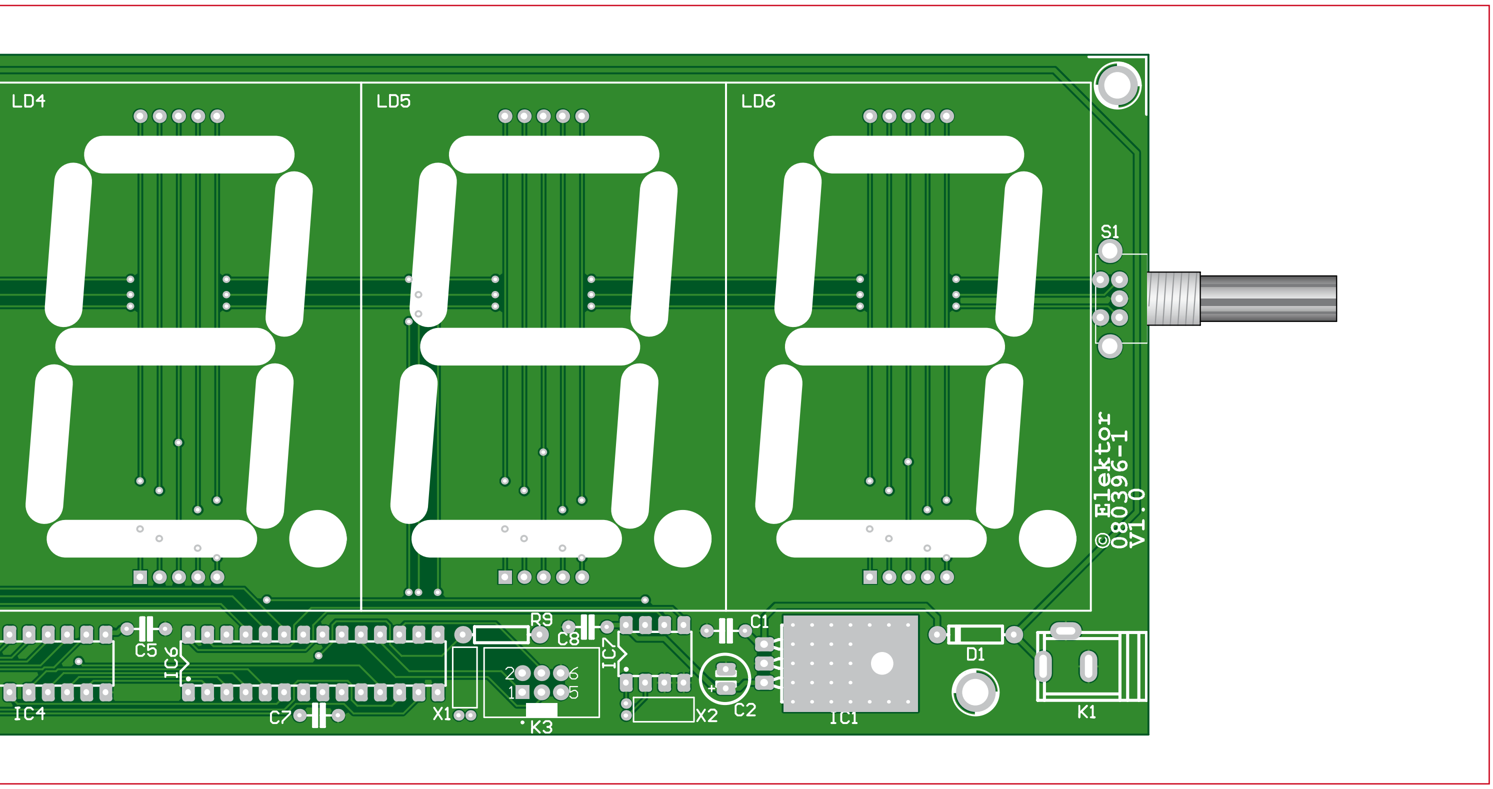
C1 = 330nF, ceramic, lead pitch 5mm
C2 = 10µF 16V
C3-C8 = 100nF, ceramic, lead pitch 5mm

Semiconductors

D1 = 1N5817 (Schottky diode, 1A)
IC1 = LM7805 (TO220 case)
IC2,IC3 = L293DNE, DIP16 case (Farnell # 1470423)
IC4 = ULN2803A, DIP18 case (Farnell # 1047761)
IC5 = 74HCT238, DIP16 case (Farnell # 382231)
IC6 = ATmega168-20PU, DIP28 case, programmed, Elektor SHOP # **080396-41**
IC7 = PCF8583P, DIP8 case (Farnell # 403908)
LD1-LD6 = SA23-12SRWA (Kingbright), 7-segment LED display, red, common anode (Farnell # 1168639)

Miscellaneous

X1,X2 = 32768 Hz quartz crystal, cylindrical housing 3x8mm
K1,K2 (optional) = mains adapter connectors for PCB mounting
K3 = 6-way DIL pinheader, lead pitch 2.54 mm (Farnell # 1096984)
S1,S2 (optional) = EC11B15242 (Alps), rotary encoder for front panel mounting, 11mm (Farnell #1191733)
PCB no. **080396-1** from the Elektor SHOP www.elektor.com



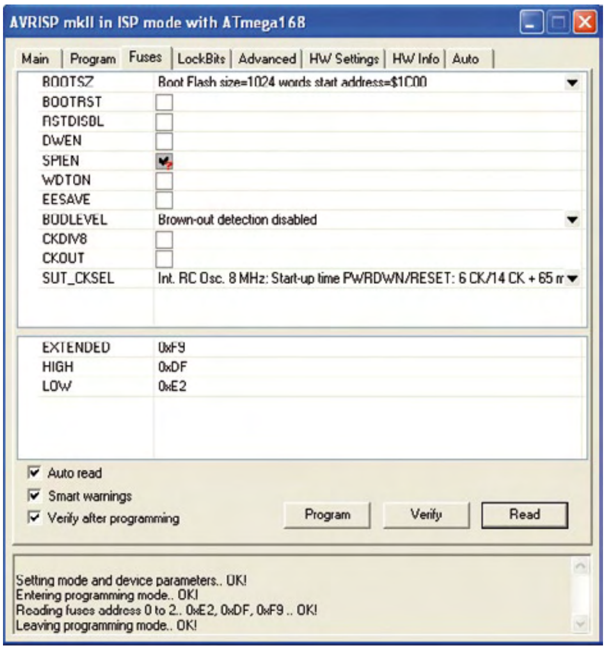
Pressing it down longer switches back to re-entering the parameters.

After polling the flags the seconds counter of Timer 2 continually calculates the cost amount.

Construction and commissioning

Because we avoided the use of SMD components entirely in this design, construction of the meeting cost timer (**Figure 2**) should be problem-free. Components should be fitted and soldered from the bottom upwards, in other words starting with low-lying parts such as resistors, diodes and capacitors, afterwards fitting things like IC sockets and the voltage regulator, then ISP and power supply connectors and finally the 7-segment displays and the rotary encoder.

For maximum freedom in choice of construction and housing we have designed the project so that power connectors (K1 or K2) and the control switch (S1 or S2) can be located on the right or left-hand side at the user's choice. You could consider painting a large Pound (or Euro) symbol on the clock case beside the display read-out. This might make the control switch awkward to use, so you could then relocate it on the left side. There are M4-sized holes drilled at each corner of the PCB and these can be used to fix the module in its case. Before fitting ICs into their sockets it's worth connecting a mains adapter and confirming the 5 V supply voltage following the 7805 voltage regulator, for example at Pin 2 (5 V) and Pin 6 (GND) on the ISP connector. If all's well in the voltage department then you can plug in those ICs.



BOOTSZ	Boot Flash size=1024 words start address=$1C00
BOOTRST	□
RSTDISBL	□
DWEN	□
SPIEN	☑
WDTON	□
EESAVE	□
BODLEVEL	Brown-out detection disabled
CKDIV8	□
CKOUT	□
SUT_CKSEL	Int. RC Osc. 8 MHz: Start-up time PWRDWN/RESET: 6 CK/14 CK + 65 m

EXTENDED	0xF9
HIGH	0xDF
LOW	0xE2

☑ Auto read
☑ Smart warnings
☑ Verify after programming

```
Setting mode and device parameters.. OK!
Entering programming mode.. OK!
Reading fuses address 0 to 2.. 0xE2, 0xDF, 0xF9 .. OK!
Leaving programming mode.. OK!
```

Figure 3. This screen shot illustrates the fuse bit settings in AVR Studio.

Next on the list comes programming the ATmega168. For this we used the AVR Studio software available gratis from Atmel and downloadable from the manufacturer's website. The same function can be carried out using any of the current STK500-compatible ISP adapters.

Another task is setting the fuse bits for the base configuration of the controller. **Figure 3** shows the ISP dialogue in AVR Studio highlighting the selected fuse bit settings. They correspond substantially with the default values as far as the CKDIV8 fuse bit, which must be erased. Otherwise the processor would work at a clock speed of only 1 MHz instead of 8 MHz! The software (source code) can be downloaded as an AVR Studio project on the Elektor website for this project (www.elektor.com/080396) with a precompiled HEX file in the default sub-folder.

(080396-I)