What-Remains-of-the-Day Clock

Analog meets digital

According to Wikipedia, 'A gadget is a small tool such as a machine that has a particular function, but is often thought of as a novelty.' German Wikipedia goes further, remarkably stressing the fun factor, which it says 'can play a major role, as these devices often cross the boundary between functionality and frivolity.' For positive proof of this statement, just read on!

By
Jörg Trautmann
(Germany)

People buy things because they are either useful, convenient, amusing, or else make the neighbors envious. This fourth gap in the market is the one we shall now set out to fill! [1] Nobody actually needs one of these but everybody will want to own one. Our What-Remains-of-the-Day Clock—actually it's a Day Remaining Clock—fills this niche very adequately.

Anyone who can remain asleep until 12 noon may well have not have realized that they have already dozed through half of the day. They glance at the clock and just think: "12 o' clock—so what!?!" But the point of perception is that it should cause us to react and do something. This project aims to make people more aware of the passing of time during the day.

The fuel tank indicator in our cars is a good example of action based on perception. If it shows more than 50%, we feel relaxed. If it indicates a value of 49 %, we immediately begin to calculate how soon we need to fill up again. The pointer position on an analog display can be grasped in a flash and levels like 'less than half' or 'almost full' are recognized without major mental acrobatics.

The same process could apply to the time of day if only we had an obvious means of displaying it that was familiar from our daily experience of life. An analog display instrument is practically tailor-made for a task of this kind. Let's say the time is 14.00. Subjectively we probably think that the day is yet young, with plenty of time remaining. On the other hand, this perception might alter if the message told us that there was barely 40 % of the day remaining. But enough of philosophical ruminations; it's time to put this project into practice!

The essential requirements of a Day Remaining Clock are:

- Analog display of time remaining, using a moving-coil meter
- Digital display of the current clock time
- Digital display of time remaining as a percentage
- Highest possible accuracy
- Emergency power backup in case of AC power failure
- Few external components
- Simple construction

Circuit

First let's clarify how a moving-coil meter can indicate the percentage of the day remaining. If we assume that at 00:00:00 (midnight) precisely 100 % of the day is available and that 0 % remains at 23:59:59, pulse width modulation comes rapidly to mind. We need to be able to set the pulse width ratio between 0 and 100 %, also display the time already elapsed during the day as a percentage on a moving-coil meter. A microcontroller of the type ATmega8 by Atmel, seen in **Figure 1**, is best suited for this purpose, as it has a pulse width modulator built in.
To show one day's time on a scale from 0 to 100 requires some calculation. A 24-hour day has 1440 minutes. Accordingly one percent corresponds to 14.4 minutes. The pulse width ratio of the PWM can be altered in 8-bit resolution (0 – 255), making a resolution of

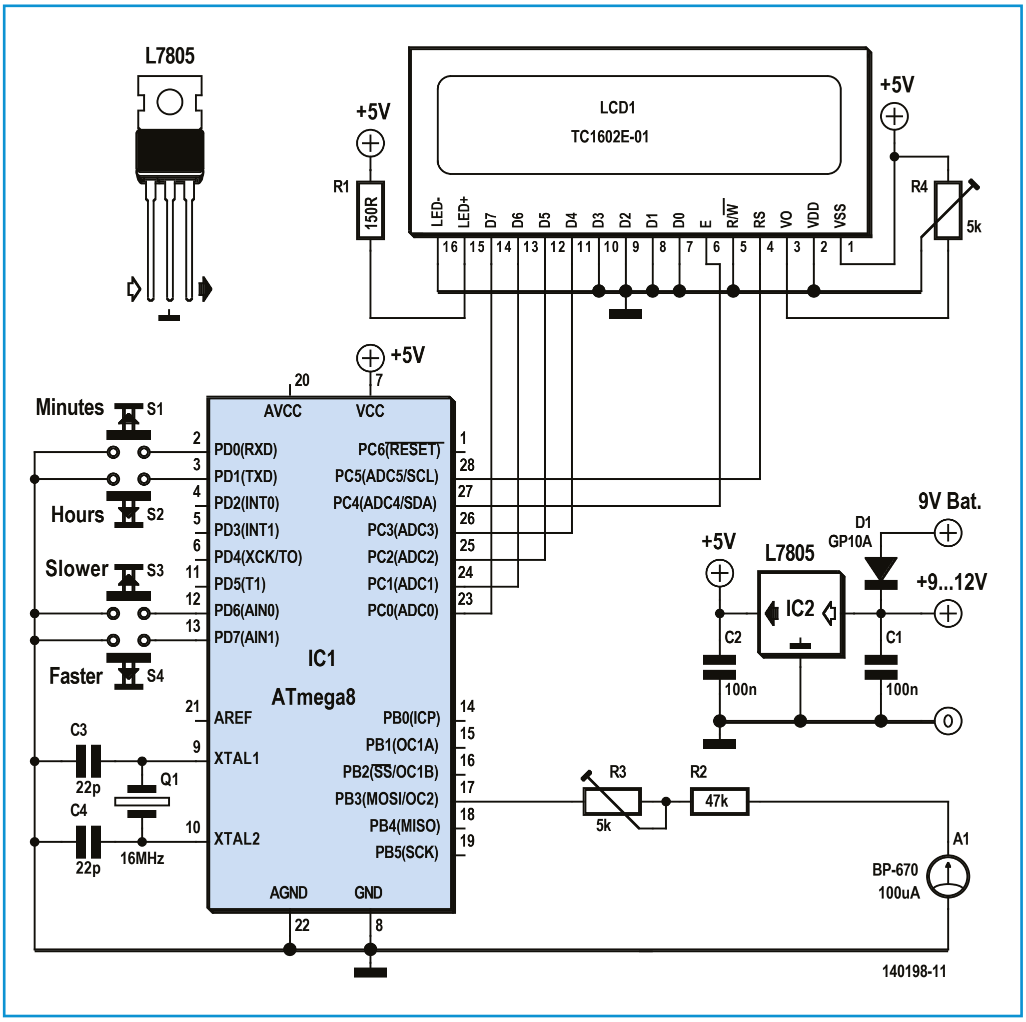

Figure 1.
The circuit of the Day Remaining Clock, centered on the ATmega8 microcontroller.

0.39 % achievable. This value is more than adequate, particularly as this level of detail would be barely distinguishable on a scale of 0 to 100. To produce full deflection of a 100 µA moving-coil meter with a supply voltage of 5 V we need a series resistance of 50 kΩ. Precise adjustment is achieved using trimpot R3 wired in series with R2.

The ATmega8 microcontroller is driven by a 16 MHz crystal. This high clock frequency makes it possible to achieve a superior level of accuracy for the timepiece using fine tuning. A two-line LCD readout is employed for displaying the time. In the event of AC supply failure the clock is powered by a 9 V 'block' battery (6LR61/1604A/PP3 size, the type used in smoke alarms) via D1.

Software

Programming a clock is not a very demanding task. I deliberately avoided using a WWV, MSF or DCF77 radio time code receiver; instead I implemented a tuning function for adjusting the accuracy and achieving a kind of 'analog clock' feeling. Completely mechanical clocks, after all, require fine adjustment over several days before they satisfy impressive accuracy. To this end an Offset has been added to the Timer Constants `Ctimer_value` in the BASCOM program code, available to download at [2] either as source code or in compiled form. As the crystal oscillates at 16 MHz, a respectably high Timer value of 49905 arises by default. This is a prerequisite for high tuning resolution and we have the ability to adjust the Timer value with very high precision. The frequency of oscillation of a crystal depends also on the ambient temperature and the external capacitors. We need to compensate for these influences. Counting of the seconds is interrupt-driven, in order to avoid the program code having any influence on the run time.

The individual functions of the program code are commented and should be self-explanatory.

For people who program the microcontroller themselves it's easy to fall into a trap (if you are inexperienced). Because the ATmega8 uses an external 16 MHz crystal in this circuit, it is necessary to set two fuses. `CKOPT` should be activated to ensure stable operation of the oscillator. This process raises the current consumption somewhat but it does ensure stable oscillator performance. It is vital that you set `SUT_CKSEL` to `Ext. Crystal/Resonator High Freq.: Start-up time: 1K CK + 64 ms`. The list of options provides a wide selection of settings. If you slip up here, you may accidentally destroy your ATmega8 following the programming process, rendering the microcontroller unable to get onto its feet unaided. Far better to double check (read twice) which of the settings you have selected. **Figure 2** shows the fuse settings in Atmel's AVR-Studio STK500.

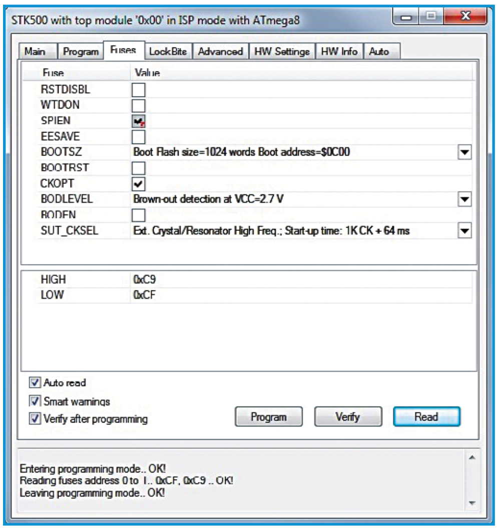

Figure 2.
Fuse settings in AVR-Studio.

Construction and commissioning

Given the small number of components used, the construction process is pretty straightforward. The greatest effort should go into making the case for the project. For my clock I cannibalized a real wood cabinet that previously housed a thermometer and hygrometer combo in the sauna. Made-to-measure laser-cut Plexiglas transparent acrylic panels can be ordered on the Internet for a couple of dollars. How you design the front panel is entirely up to you.

With the programmed ATmega8 inserted, all construction fully checked and the power connected, the background light of the LCD should illuminate and display 00:00:00. The display brightness can be adjusted with the contrast controller R4.

If the program is working properly, the seconds should duly count upwards. Your next task is to adjust the moving-coil meter correctly. Because the clock boots up showing 00:00, you should set the pointer on the meter to exactly 100 % (full deflection) using trimmer R3. You can now set the current time with the *Hour* and *Minute* buttons.

As the crystal's oscillator frequency can be influenced by manufacturing tolerances, temperature effects and the tuning capacitor used, the clock needs to be calibrated properly. As with a mechanical clock, this process can extend over several days.
If you want to make your clock accurate to the very last second, press the Minute button at the exact moment at which the minute changes on the reference timepiece. At this moment you also need to reset the seconds. It is recommended that you carry out observation for at least 12 hours. Following this, readjustments can be made with the buttons *Slower* and *Faster*. One press of the button corresponds to around two seconds over 24 hours. The final offset value is always saved in the EEPROM, so that following any interruption to the AC supply the last calibration settings are reloaded.
Because the AC supply can fail at any time without warning, it's advisable to provide a 9-V lithium battery. This type of battery, with a life of up to 10 years, is very reliable and has an impressive capacity of 1200 mAh.

Perspectives
Inspired by this project, we can envisage some further developments: Week Remaining, Month Remaining, Year Remaining, etc. Fortunately mankind has not yet reached the stage where we can display how much of our life is remaining ;-). Perhaps you need a Time Remaining Clock for Building a Day Remaining Clock and see what remains of the day, or watch the film [3]!

(140198)

Web Links

[1] The character Preposterus in Obelix and Co. (the twenty-third volume of the Asterix comic book series). http://en.wikipedia.org/wiki/Obelix_and_Co.

[2] www.elektor-magazine.com/post

[3] The Remains of the Day, from the novel by Kazuo Ishiguro