Minimalistic Time Switch

A lot of features in a small package

By Fons Janssen and Mark Vermeulen (The Netherlands)

A time switch lets you save a considerable amount of energy by switching off equipment when it is not being used. However, time clocks from building merchants tend to be bulky and not especially easy to use. The circuit described here provides a distinctly larger set of features and is very compact, so it can easily be built into an existing piece of equipment.

The hardware (**Figure 1**) can certainly be dubbed 'minimalistic'. The circuit consists of only 24 components, including the connectors and pushbutton switch. It is built around a Maxim DS1337 (IC2), which is a compact real-time clock IC in an SOIC-18 package. This IC is designed to work with a 32.768-kHz crystal (X1). However, crystals that operate at this frequency are available in two different types. One type works with a 12.5-pF load capacitance, while the other works with a 6-pF load capacitance. The DS1337 only works properly with a 6-pF crystal. If a 12.5-pF type is used, the clock will be highly inaccurate.

For the microcontroller we chose Microchip's most powerful eight-pin type, the PIC12F683 (IC3), which also comes in an SOIC-8 package. Backup power in case of a power grid outage is provided by a 5.5-V GoldCap capacitor (C3), with a choice of capacitance values. Resistor R7 is included to keep the charging current of the capacitor within reasonable bounds. It ensures that the capacitor charging current never exceeds 30 mA, which is the maximum rated output current of the MAX1615 voltage regulator (IC1). In addition to these components, a small handful of SMD resistors and semiconductor devices are necessary for proper operation of the circuit.

The circuit employs an extremely minimalistic user interface in the form of a single pushbutton and a single 3-mm LED (D2). Nevertheless, the time clock is relatively easy to use. Setting the time and date is fairly intuitive, and programming the switching times for weekends and weekdays are just as easy.

A few extra components around the microcontroller are included to enable in-circuit programming. For instance, T1 and R5 are included to drive the LED. Here R5 provides sufficient isolation when pin 7 is used for the data signal during in-circuit programming. Resistor R6 is included for a similar reason; it allows the clock signal to be applied to pin 6 of IC3 during in-circuit programming. The programming voltage is supplied via the MCLR line (pin 4). Diode D1 allows power to be supplied to the microcontroller during in-circuit programming without powering the rest of the circuit at the same time.

As the circuit must be able to switch a fairly significant load (for example, a WiFi router can easily draw 1 A), we chose a type PMV45EN in an SOT23 package for the FET switch (T2). This MOSFET can handle well over 5 A, and it has a low on resistance (R_{DSon}). It can effectively switch a continuous current of around 2 A, which is more than adequate for most applications.

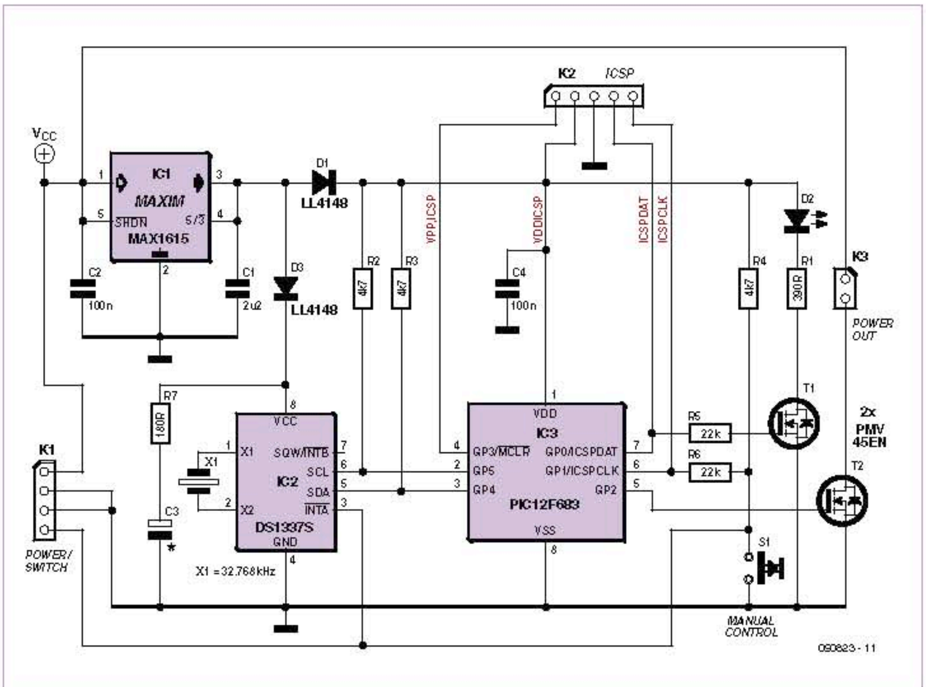

Figure 1. The entire circuit of the time clock consists of only 24 components, including the connectors and pushbutton switch.

Minimalistic software

Now let's have a look at how the time clock works.

Features of the minimalistic time switch

- 'Fit and forget' functionality
  - Automatic day of week determination
  - Automatic adjustment to summer and winter time
  - Long-term backup in case of power outage
- Easy to use
  - Manual override for 'always on' and 'always off' (regardless of the programmed switching times)
  - Single-button user interface (with a single LED for feedback) for all operating functions
  - Different switching times for weekdays and weekends (individual day of week programming possible if desired)
- Suitable for all DC-powered applications and battery-powered equipment
  - Input voltage range 6–28 V_{DC}
  - Very low current consumption (less than 10 µA)
  - Extremely compact design
  - Switches loads up to 2 A at 24 V_{DC}

Backup time

The circuit is designed to be used with a GoldCap capacitor rated at 1 F / 5.5 V. This capacitor must supply power to the DS1337 in the event of a power outage. With a supercapacitor of this type, the circuit can survive a power outage of more than three months. However, other types of capacitors can also be used. If you use a normal electrolytic capacitor rated at 470 µF / 6.3 V, the circuit will be able to handle a power outage with a duration of one hour.

Current consumption

The current consumption of the time clock is very low. The microcontroller is constantly in sleep mode under normal conditions, with a current consumption of only 350 nA. The corresponding current consumption of the DS1337 is somewhat larger at 600 nA. However, the current consumption of these two components is negligible compared with the current consumption of the voltage regulator. The MAX1615 used here has a maximum quiescent current of 8 µA, with the result that the maximum current consumption of the overall circuit is less than 10 µA.

Initial start-up

When you start up the clock the first time, the time setting is of course incorrect. This is indicated by the rapid blinking of the LED. You can set the time by first pressing the button within 30 seconds. If you don't do anything within this time, the clock automatically enters sleep mode. The only way to exit the sleep mode is to switch off power to the circuit. The circuit must be left without power for a few seconds to give the capacitors on the circuit board time to discharge, as otherwise the microcontroller will not reset.

Setting the time is relatively easy, despite the single-button user interface. A number from 0 to 9 can be entered by briefly pressing the button a corresponding number of times and then pressing it again for an extended length of time, which means holding the button pressed until the LED goes on. For example, you can enter the number '4' by pressing the button four times short and one time long.

The time clock expects you to enter ten numbers in the sequence DD–MM–YY–HH–MM (day, month, year, hours, minutes).

After a valid date and time have been entered, the time clock automatically determines the corresponding day of the week. This is important because you can set different switching times for weekends (Saturday and Sunday) and weekdays (Monday through Friday). Although the DS1337 has a register to store the day of the week, it is not able to determine the day of the week from the date, so this task is handled by the software. The entered information is stored in the registers of the DS1337.

Normal start-up

When the time clock starts up in normal use, it retrieves the date and time from the DS1337. Using this data, the software determines whether it is summer time or winter time. Based on the day of the week and the current time (both taken from the DS1337) as well as the switching times for the current day (taken from the internal EEPROM of the microcontroller), the software sets the output to the desired state. In addition, the software stores the next switching time in the ALM1 register of the DS1337. Once all of this has been done, the LED blinks once to indicate that the time clock has started up correctly and is ready for use.

After this, the microcontroller enables an interrupt on a change of signal level on pin 6 (GP1) and enters sleep mode. The microcontroller remains in this mode until the status of GP1 changes. When this happens, it carries on with execution of the program code. Consequently, there is no need for an interrupt service routine. Here the sleep mode is used as a low-power standby mode.

There are three possible sources of an interrupt via GP1: a button press, an interrupt signal from ALM1 (used for the switching times), or an interrupt signal from ALM2 (used for switching back and forth between summer time and winter time). As all three types of interrupts enter via GP1, the software must first determine the cause of the interrupt. It does this by reading the alarm flags of the DS1337. If one or both of the alarm flags is/are set, it or they is/are the cause of the interrupt. If no alarm flag is set, a button press is the cause of the interrupt. Here again, there are several options with the button:

- A **short** button press activates the override function.
- A **long** button press causes the time clock to enter switching time entry mode. In this mode, the time clock expects to receive the following data in the order listed:

HH–MM (switch-on time, weekend)
HH–MM (switch-off time, weekend)
HH–MM (switch-on time, weekday)
HH–MM (switch-off time, weekday)

The switching times are stored in the microcontroller's EEPROM. The switching times are stored separately for each day of the week, starting with Saturday and proceeding until Friday. This means that you can

COMPONENT LIST

Resistors
all SMD 0805
R1 = 390Ω
R2,R3,R4 = 4.7kΩ
R5,R6 = 22kΩ
R7 = 180Ω

Capacitors
C1 = 2.2µF, SMD 0805
C2,C4 = 100nF, SMD 0805
C3 = 470µF 6.3V, or GoldCap 0.1F or 1F 5.5 V (see text)

Semiconductors
all SMD, except D2
D1,D3 = LL4148
D2 = LED, red, 3mm
T1,T2 = PMV45EN
IC1 = MAX1615EUK+T
IC2 = DS1337S+
IC3 = PIC12F683-I/SN, programmed, Elektor Shop # 090823-41

Miscellaneous
X1 = 32.768kHz (CL = 6pF!, e.g. Farnell # 1216227)
S1 = pushbutton SPNO, 6mm (MCDTS6-5N)
K1 = 4-pin SIL pinheader
K2 = 5-way SIL socket
K3 = 2-pin SIL pinheader
PCB, # 090823-1
Project software # 090823-11.zip (source- and hex code files) from www.elektor.com/090823

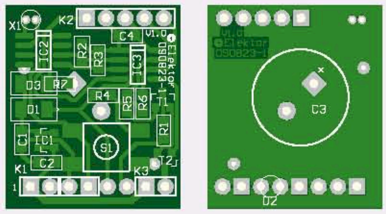

Figure 2. The accompanying PCB is also very compact. The largest component is the GoldCap capacitor on the bottom.

use a programmer to set different switching times for the individual days of the week if you so wish. We decided not to support this via manual entry because it would make setting the times significantly more complicated, and for the simple reason that there wasn't enough memory left to do so.

Override function

You can use this function to manually bypass the switching process. This works as follows:

- The first button press sets the output to **always on**.
- The second button press sets the output to **always off**.
- After the third button press, the output is again controlled by the programmed switching times.

The state of the override function is also stored in the microcontroller's EEPROM to ensure that the time clock continues to operate in the same way as before after a power outage.

Mini PCB

Figure 2 shows the printed circuit board for this minimalistic time clock. To keep everything as small as possible, SMDs are used almost everywhere. Most of the components are fitted on the copper side of the board, with only the backup capacitor and LED D2 fitted on the other side. Two headers (K1 and K3) provide the input and output connections, while K2 is the programming connector. Be sure to use the right type of crystal (6-pF capacitance) when assembling the board, as otherwise the clock IC will not work properly.

Readers who don't want to program the PIC themselves can order a pre-programmed microcontroller from the Elektor Shop (item number **090823-41**). The source code and hex code are available on the Elektor website.

External connections

The PCB has a pair of pin headers for connecting an AC powerline adapter and the switched equipment. In addition to the supply voltage pins, connector K1 has two spare pins that can be used if desired to connect a second pushbutton. This button can be located in a more readily accessible location on the selected enclosure. The switched equipment is connected to connector K3 (see the wiring diagram in **Figure 3**).

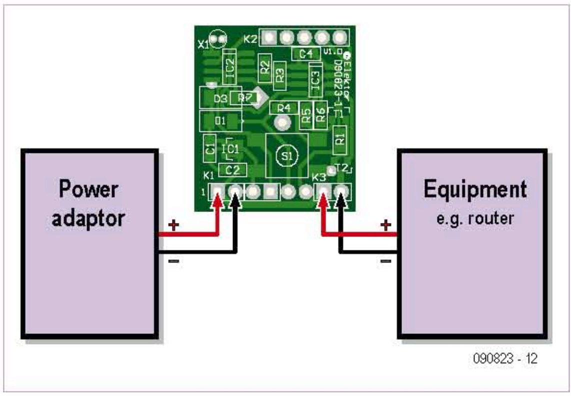

Figure 3. Follow this wiring diagram to ensure that you get the connections right.

The remaining connector on the PCB, K2, can be used for in-circuit programming of the microcontroller. It is intended to be used with a Velleman K8049 PIC programmer. Be sure to use the most recent version of the software for this programmer, as older versions do not support the PIC12F683.

The software is written in Pascal based on Mikropascal 8.0.0.1 from Mikroelektronika. This compiler is ideal for this application because the free version can be used to generate programs with code sizes up to 2 KB, which is exactly the amount of memory available in the PIC12F683.

Other applications

Switching a WiFi router on an off is of course only one example of the potential applications. Thanks to the extremely compact size of this time clock, there's virtually no limit to where it can be used. For example, you could use it to power equipment on and off in a caravan when you're camping, or use it as a time clock in combination with an inexpensive photo frame that doesn't have a built-in clock function. However, bear in mind that due to the choice of voltage regulator, the supply voltage from the switched equipment must lie in the range of 6 to 28 V_{DC}. The MOSFET selected for the switch limits the maximum switching current to 2 A continuous or 5.4 A peak. This is sufficient for practically all equipment that is powered by an AC power adapter.

(090823-I)

Fons Janssen is a Senior Field Applications Engineer at Maxim Benelux (www.maxim-ic.com)

Mark Vermeulen is a Managing Director and founder of Smart Sustainable Electronics (www.s2e.nl)