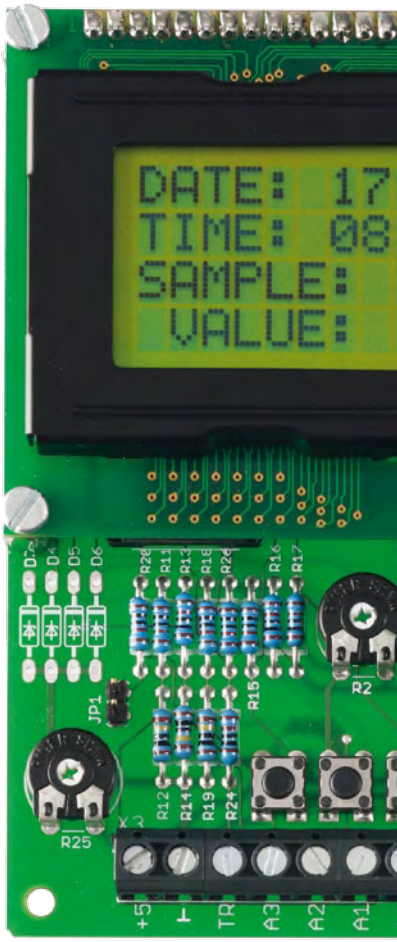
Data Logger "deLuxe"



We have had the pleasure of proposing various data acquisition units over the last few years. The one we describe here is a nice exercise in product development. It actually utilises an SD card as the media for data storage. The hardware design is compact and that makes the firmware and software features even more interesting.

Loïc Marty

This data logger is used to save the values of four analogue channels supplying any voltage ranging from 0 to 5 V onto a standard memory card (SD – Secure Digital).

Elektor has a long history when it comes to publishing data acquisition systems and loggers — the most recent project [1] proved very popular. The data logger we are proposing here is unique due to its simplicity and compact size; a microcontroller and a handful of common components are all it takes for hardware.

Our data acquisition unit has several operating modes available:

1. Triggering on the fly (internal trigger)... in other words, press the pushbutton!

2. Triggering from an external signal; this may come, for example, from a sensor operating at 12 or 24 V (say, a proximity detector); the signal input is protected by a 5.1 V zener diode.

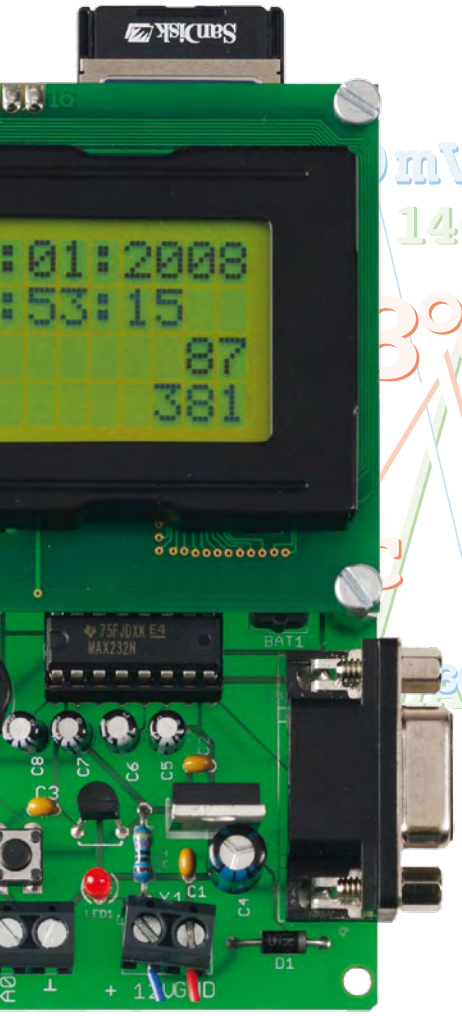
3. Saving data at 10-second intervals.

4. Saving data at 1-minute intervals (e.g. temperature sensor).

5. Saving data when a signal exceeds a certain maximum value; in this case, analogue channel 3 defines the reference level.

The electronics

As the schematic in **Figure 1** shows, there is nothing special or too complicated as far as the electronics are involved.
The core of the setup is IC4, a top-level 18F452 PIC micro [2], set to maximum speed in HS mode, i.e. 20 MHz. It incorporates different peripherals put to good use here: A/D converters, an SPI port (to communicate with the SD card) and an RS-232 port (for possible future software extensions).
This PIC device, the top-of-the-line in the 18Fxx2 family (32 kB of Flash memory, 1,536 bytes of RAM and 256 bytes of EEPROM) also has an I^2C port. Although this uses the same pins as the SPI port, it proved necessary to use two other pins for communication with the real-time clock (RTC) device, IC5. Here, this takes the physical form of a PCF8583 [3], an IC that's very simple to configure. Some of its key data are listed separately in the **inset**. Note the presence of a CR2032 3-V lithium battery; this is used to save the time and date information if the normal supply voltage disappears. D2 and D3, a pair of low forward-bias (Schottky) diodes make it possible to power the RTC permanently. Note the presence of a trimmer capacitor, C14, used to correct possible frequency drift in crystal X1, which is used as a clock (not very critical). The RTC has an open-drain interrupt output with a pull-up resistor, R23, allowing for an excellent 1 Hz timebase, simplifying time management by the 18F452.
LCD1, a large display with 4 lines × 16 characters driven in 4-bit mode, makes it possible to visualise the various data: date, hour, recording file number, as well as the value of the analogue channel AN0 (0 to 1024, given that the A/D converter has a resolution of 10 bits). Three pushbuttons, S1, S2 and S3, are used to select the operating mode.
Also note the presence of the RS-232 port, which may be implemented here. Future changes to the firmware might need this port in combination



The efficiency of simplicity

Technical specifications

- 5 data recording modes
  – on the fly (button press)
  – external trigger
  – timed: every 10 s, every 60 s
  – on exceeding a preset level
- Data saved in .txt file format
- Max. 99 files
- Direct reading of SD card in PC word processing program
- Formats FAT16 SD cards
- RTC circuit to timestamp the files
- Available as a kit from Elektor Shop

with a bootloader to load the PIC executable code.
The SD card requires a 3.3 V supply; this voltage comes from a TS2950-3.3 low drop-out regulator that only requires a small voltage difference to supply the necessary voltage (from 5 to 3.3 V). This device is capable of withstanding the (relatively) large peak current consumption of an SD card starting up.
The PIC-to-SD card communication is in SPI format with 5 V on the PIC side and 3.3 V on the memory card side. By contrast, in the direction PIC-to-card direction, resistors R5–R10 are used to obtain the required voltage level at the card inputs (CS, DATA IN, CLK); in the other direction, DATA OUT ® PIC, the 3.3 V level is sufficient for the PIC to recognize valid logic states.
One note concerning the analogue and external trigger inputs (the signal which triggers the recording of the values in 'External trigger' operating mode): these lines may be protected by 5.1 V zener diodes, D4–D7, but these should only be fitted if you know how to handle the inherent voltage drop they will cause. Since they offset the measurement, it follows that you will need to adapt the input stage according to your application — using a potential divider, for example, to measure variations above 5 V.
Note that a jumper can be fitted on connector JP1 to supply a fixed voltage (adjusted by the 10 kΩ potentiometer P2) to input RA3; this jumper should be fitted when the maximum signal value mode is being used (on AN0 only). This is very easy to adjust in practice: it defines the threshold above which the voltage peak is taken into account. It is worth noting that this project was originally designed with this prospect in mind.

The PCB

The project may be purchased a s a kit of parts # **070745-71** from the Elektor Shop. **Figure 2** shows the component overlay. If you want to make your own board, you can download the copper track pattern from the Elektor website.

Given the (small) number of components involved and the size of the printed board, fitting the components should not pose any problems. Remember to adjust the LCD contrast using the 10 kΩ preset P1.
Construction is easy, since the components are common and readily obtainable, like the PIC18F452. The display is a 4 line/16 character model in standard green; it is mounted using four plastic 10-mm pillars. The electrical connection is made via a 16-way male header soldered onto the PCB; a strip of female contacts will be soldered onto the display side. Doing this makes assembly / disassembly much easier.

Note that we have made provision for an RS-232 section which makes the setup even more flexible. The RS-232 port makes it possible to employ the bootloader option (see the **inset** with screenshots devoted to this topic).
Since it is difficult to obtain SD card connectors as spares, you can use one cannibalized from a PC external card reader (bash the cheapest one you can find!). The pins for the Lock/Unlock contact on the SD card are not soldered (like pins 8 & 9); the PIC program in fact ignores the data they carry.
If the hardware part is simple, that often means that the software is hugely complex, as is the case here.

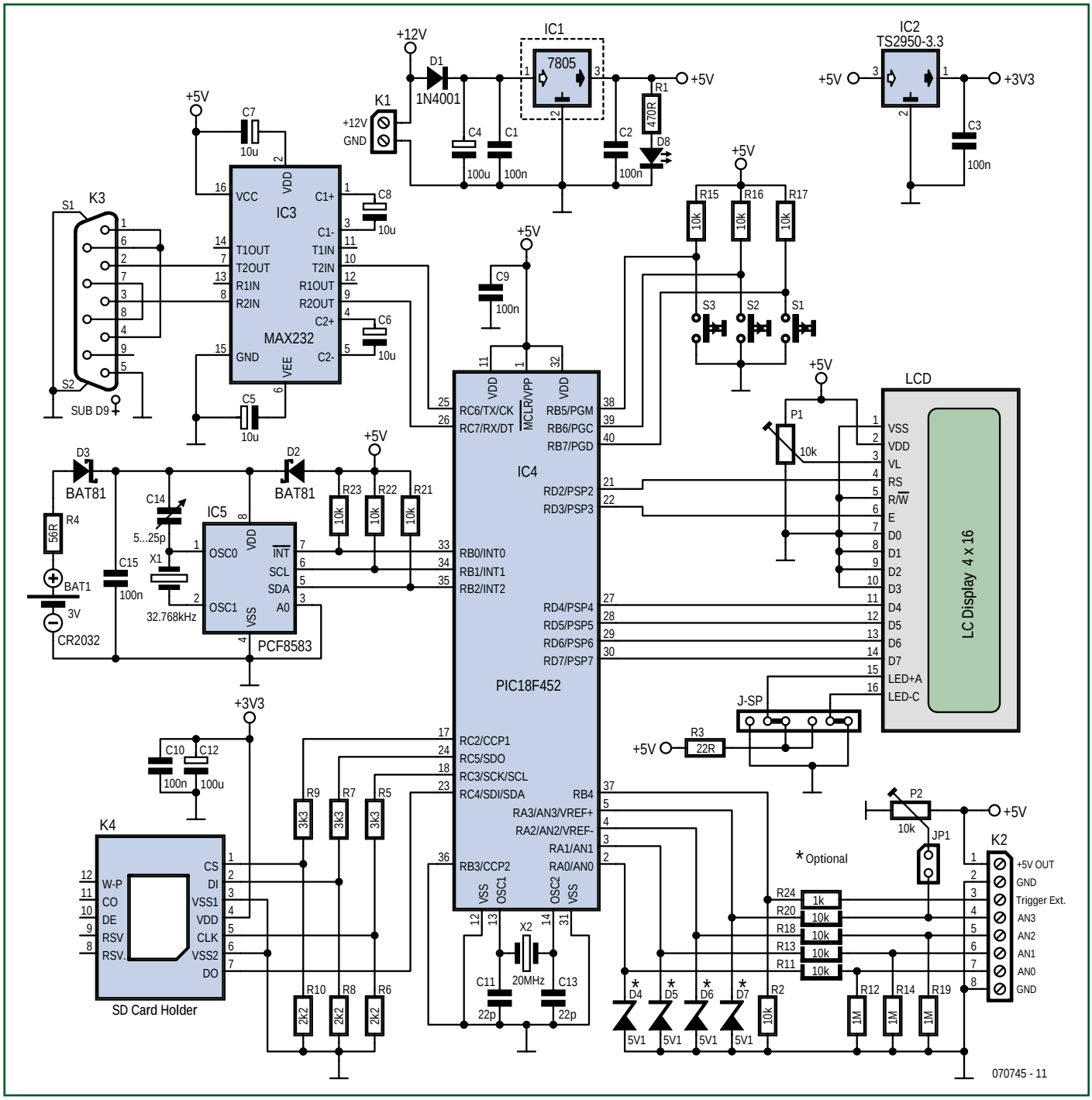


Figure 1. At the heart of the circuit, the PIC18F452, surrounded by (clockwise) LC display, SD card connector, the PCF8583 RTC, and the MAX232 level converter.

Keywords

LC Display – LC displays (LCDs) have replaced cumbersome, power-hungry, inflexible LED displays. There are many types, with characters (x lines of x characters) or graphic symbols. The present project employs an LCD with 4 lines of 16 characters.

I2C Bus – Inter-Integrated-Circuit bus. Developed by Philips in the early 80s for home automation and interconnectable home electronics applications. The I²C bus comprises 3 lines: SDA (Serial Data) for the data, SCL (Serial CLock) for the clock, and a ground line.

EEPROM – Electrically Erasable Programmable Read Only Memory. This type of rewritable memory has the advantage of not losing the data it contains during power loss.

FAT (16 or 32) – Even before Windows existed, Microsoft developed and patented (in part) an operating system for files, for floppy disks as well as for removable optical media called FAT (File Allocation Table). For more information, see Wiki sites.

IDE – Integrated Development Environment. More and more complex software sets propose a centralized environment from which one can access different programs.

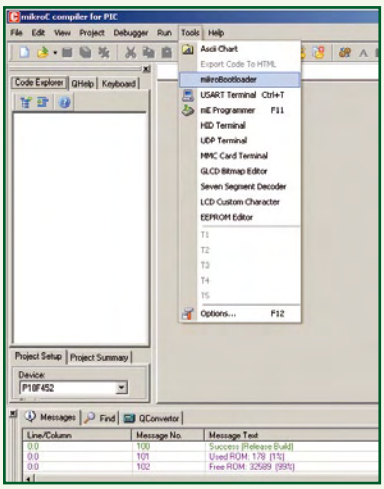
The software

The source code listing runs into 500+ lines and cannot possibly be printed in this article; you can download it from the Elektor website (www.elektor.com) under the filename **070745-11.zip**. This archive file also includes the .hex file you will need to program into the PIC18F452 if you have the means to program the controller yourself. Note that the device is also available ready-programmed from the Elektor Shop as item **070745-41**. There's a special **inset** to explain the programming procedure using the bootloader.

The program was written in C (a first for this author) and compiled using MikroC, the excellent compiler from Mikro Elektronika (very good technical support). The source code is fairly intuitive, even for those who do not know this language very well (in C, you start off with the function `main()`) There is a free version of this compiler; it supports code of less than 2 k words. For this project, which uses a lot of memory due to file management in FAT mode, you need to use the full version. The price is fair, considering the technical capacities of the compiler: IDE included, and especially the use of the built-in features. These were indispensable and put to good use for managing the files in FAT, their attributes (date/hour, etc.), as well as to manage communication with the RTC in implemented I^2C mode. The author encourages you to discover this surprising and powerful compiler — see the link at the end of the article. Mikro Elektronika are also valued advertisers in Elektor.

The LCD screen messages mostly speak for themselves, using established terms like *Sample*, *Value* and *Save*.

The bootloader



The PIC18F452 microcontroller used in this project can be programmed in the usual manner using the .HEX file [logger.hex]. To make it easier to develop and especially debug the application, the author has included a bootloader in the firmware.
The principle is simpler than it might appear at first. The bootloader is a tiny program that is executed whenever the microcontroller is reset. It scans the serial connection looking for a response (here, an 'r') to the message (here, a 'g') it sends. If it receives the agreed response within the time allotted (here, 5 s), it goes into firmware update mode (in other words, it flashes the part of the memory assigned to the firmware with the data carried by the serial connection, avoiding crashing). Otherwise, it sets the pointer to the firmware address, which then takes over control.

In this way, it is possible to reload software into the microcontroller at each start-up, as long as the serial connection is correctly set up (57,600 bps / 1 stop / 8 bits / no parity / software flow control) and the PC host software is ready to communicate.

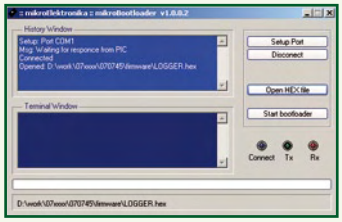
Now we will see how to take advantage of this function. First the PIC must be programmed with the bootloader in the usual way, in order to have software for bootloading.

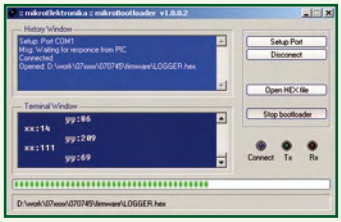
Next, just execute the software [mikroBootloader] available in the menu [Tools] in [mikroC]. Be careful to configure the serial connection correctly.

Connect the data logger and click on [connect] before the 5 second time delay elapses. Then you can relax: now the .HEX file must be loaded from the data logger firmware with [Open HEX file] and click on [Start bootloader] to flash the PIC.

Once this operation is over, just disconnect and reconnect the data logger power supply [Hard Reset], wait 5 seconds and the software should run. Do not forget to insert your SD before starting up again, reset takes place right at the start, otherwise you will need to reset again from off in order to use your module.

The screenshots shown here demonstrate the sequence of operations using the bootloader to update the software, from the first step to ...success (obviously!).





MikroC – C compiler intended for PIC12, PIC16 and PIC18. A demo version is available for download, its only limitation being the 2 kB maximum size of the .hex file. Not bad to start out with...

RTC – Real Time Clock. This IC came out with the first microprocessors in order to have an accurate time/date stamp. Among other places, they are found on every PC motherboard.

SD (Card) – Secure Digital (Card). Undoubtedly the most widely-sold removable Flash memory card in the world. Its format is a reference for many other cards whose minuscule dimensions require the use of an adaptor, like the microSD and miniSD.

SPI – Serial Programming Interface. This bus is used to establish a synchronous serial connection. It has 4 lines:
SCLK – Serial CLocK (Master Output)
MOSI/SIMO – Master Output, Slave Input
MISO/SOMI – Master Input, Slave Output
SS — Slave Select (active low). Often used to program modern processors.

Wiki – open-source online encyclopaedia at http://en.wikipedia.org/wiki/Main_Page. You can find information there on practically anything... and why not contribute to it...

Implementation

Let's begin by describing the three pushbuttons:
S1: Used to select the operating mode. Also lets you adjust hour and date: in this case, keep it pressed during start-up. S2 and S3 are then used to increment/decrement the values, S1 is used to move to the next field, in the logical order: day → month → year → hour → minute → second. Pressing S1 twice at the end of data entry is used to memorize the time clock.

S2: Used to record on the fly in 'internal trigger' mode. Also used to select the number of analogue channels to be recorded when the operating mode is set to the one that allows selection of the number of analogue inputs to be recorded: note that only one analogue input, AN0, is displayed on the LCD, due to code execution speed.

S3: Used to select a new recording file (in .txt format, so as to allow direct reading on a PC that has an SD card reader and a text editor). The values recorded may range from 0 to 1024 (from 0 to 5 V), with the possibility of recording from 1 to 4 channels simultaneously.

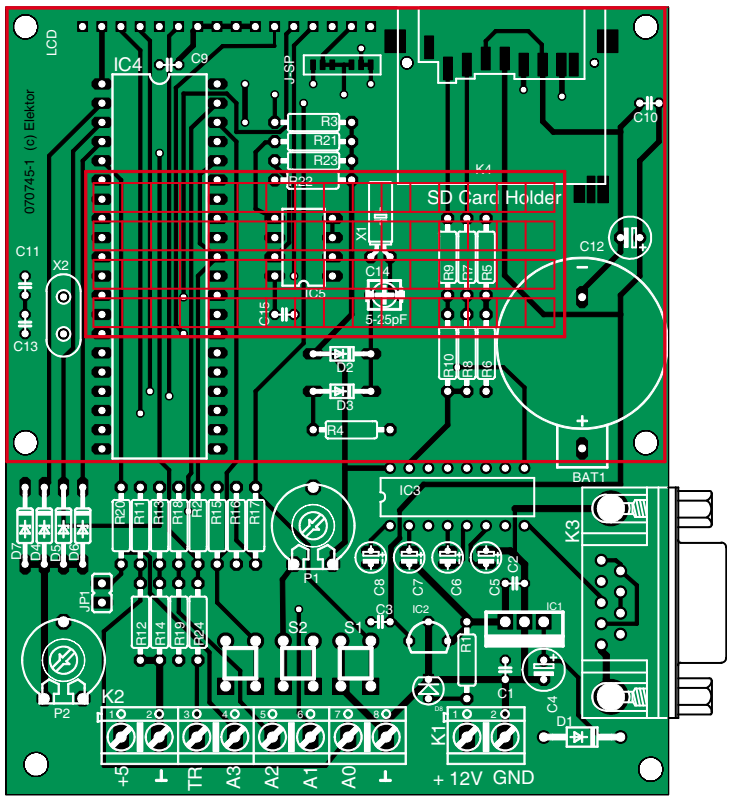


Figure 2. Component mounting plan for the compact PCB designed for this data acquisition system using an SD card.

Closing remarks

You will no doubt have spotted the jumper field J-SP. This is used to reverse the power supply for the LED display backlighting if you are using a different type of LCD display from the one mentioned in the component list, with different characteristics.
A word about the SD card: normally an SD utilizes its own protocol to communicate with its host. At start-up, it is possible to communicate in serial mode (SPI), as long as certain conditions are

Reference

[1] USB Data Acquisition Card, Elektor November 2007

Web Links

[2] www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en010296

[3] www.nxp.com/#/pip/cb=[type=product, path=50807/53497,final=PCF8583_5]|pip=[pip=PCF8583_5][0]

www.f1ubz.fr

www.mikroe.com (compiler and RTC diagram)

www.sdcard.org

COMPONENT LIST

Resistors
R1 = 470 Ω
R2,R11,R13,R15,R16,R17,R18,R20,R21,R22,R23 = 10kΩ
R3 = 22Ω
R4 = 56Ω
R5,R7,R9 = 3kΩ3
R6,R8,R10 = 2kΩ2
R12,R14,R19 = 1MΩ
R24 = 1kΩ
P1,P2 = 10kΩ preset

Capacitors
C1,C2,C3,C9,C10,C15 = 100nF
C4,C12 = 100µF
C5–C8 = 10µF
C11,C13 = 22pF
C14 = 5-25pF trimmer *

Semiconductors
D1 = 1N4001
D2,D3 = BAT81
D4-D7 = 5V1/400 mW zener diode
D8 = LED, 3 mm, red
IC1 = 7805
IC2 = TS2950-3.3
IC3 = MAX232
IC4 = PIC18F452, programmed, Elektor Shop # 070745-41
IC5 = PCF8583

Miscellaneous
S1,S2,S3 = miniature push button
JP1 = 2-way SIL pinheader with jumper
K1 = 2-way PCB-mount screw terminal block, 5mm lead pitch
K2 = 8-way SIL pinheader
K3 = 9-pin PCB-mount sub-D socket (female)
K4 = SD card connector
X1 = 32.768 kHz quartz crystal
X2 = 20 MHz quartz crystal
BAT1 = CR2032 Lithium battery
LCD = LCD display with 4 lines of 16 characters, e.g. DEM 16481
PCB, ref. 070745-1
Kit of parts, Elektor Shop # 070745-71
Project software (source code, hex file, PC program), file # 070745-11 from www.elektor.com

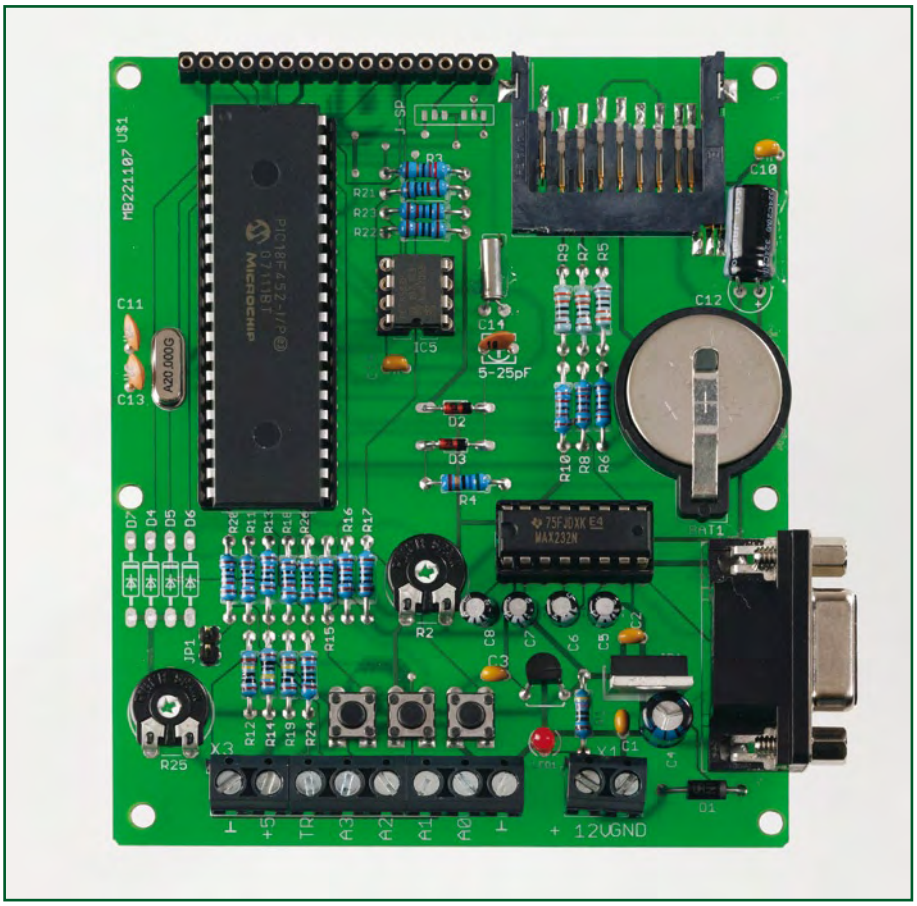


Figure 3. Completed example of our SD card data logger. The LCD display is neatly in place on the top of the main PCB. The real time clock capacitor in our prototype had a fixed value, determined by measurement.

fulfilled; this is certainly slower, but is easier to manage using a microcontroller. This facilitates interfacing with the 18F452 and, even more so, with the Built-ins from MikroC.

A 1 GB SD card is surprisingly affordable these days. At the time of writing, it's found for less than a tenner making it the cheapest card per MB. Who still remembers the £ 25+ introductory price of a 128 MB SD card just four years ago? The 1-GB card will work perfectly.

It is vital that this card is formatted in FAT16 mode (i.e. not FAT32). The files in which the data are recorded begin at 1.txt and go up to 9999.txt (incrementing automatically). Obviously, it will be necessary to reformat the card once it is full and you have transferred the contents to your PC (for your information, each file can record a maximum of about 65,000 discrete samples).

Now you are equipped with a brand new data acquisition system. Use it to your advantage. We are here to listen to your feedback.

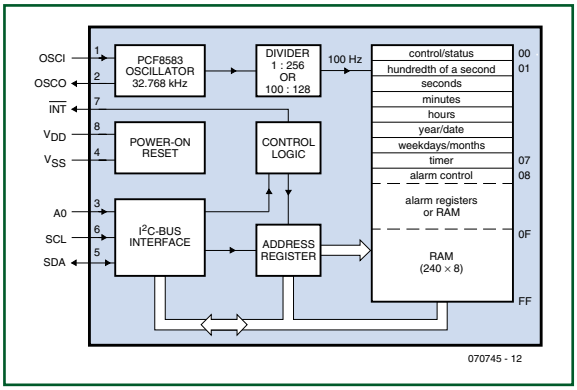
Loic.marty@neuf.fr (070745-I)

The PCF8583

The PCF8583 is a clock/calendar IC from Philips (now NXP) with a 2,048 bit static CMOS RAM arranged as 256 bytes. Address and data transfer takes place via the 2-line I²C bidirectional bus. Address line A0 is used to program the hardware address, thereby allowing connection of two components to the bus without requiring additional hardware.

The built-in oscillator operates at 32.768 kHz; the first 8 bytes of RAM perform the clock/calendar and counting functions. The next 8 bytes may be programmed as alarm registers or used as free RAM space, if not otherwise used. The other 240 bytes are available to the user.

Note that this Real Time Clock can also be synchronised using an external 50 Hz clock signal.



Block diagram of the PCF8583 (Source: NXP/Philips Semiconductors)

The author

Self-taught, a thirty-plus amateur radio operator, especially interested in everything that involves PIC programming, in assemby language (for video) and lately in C. Loic does not work in the electronics field at all, as he is a production agent (www.mapei.com); his website is www.f1ubz.fr