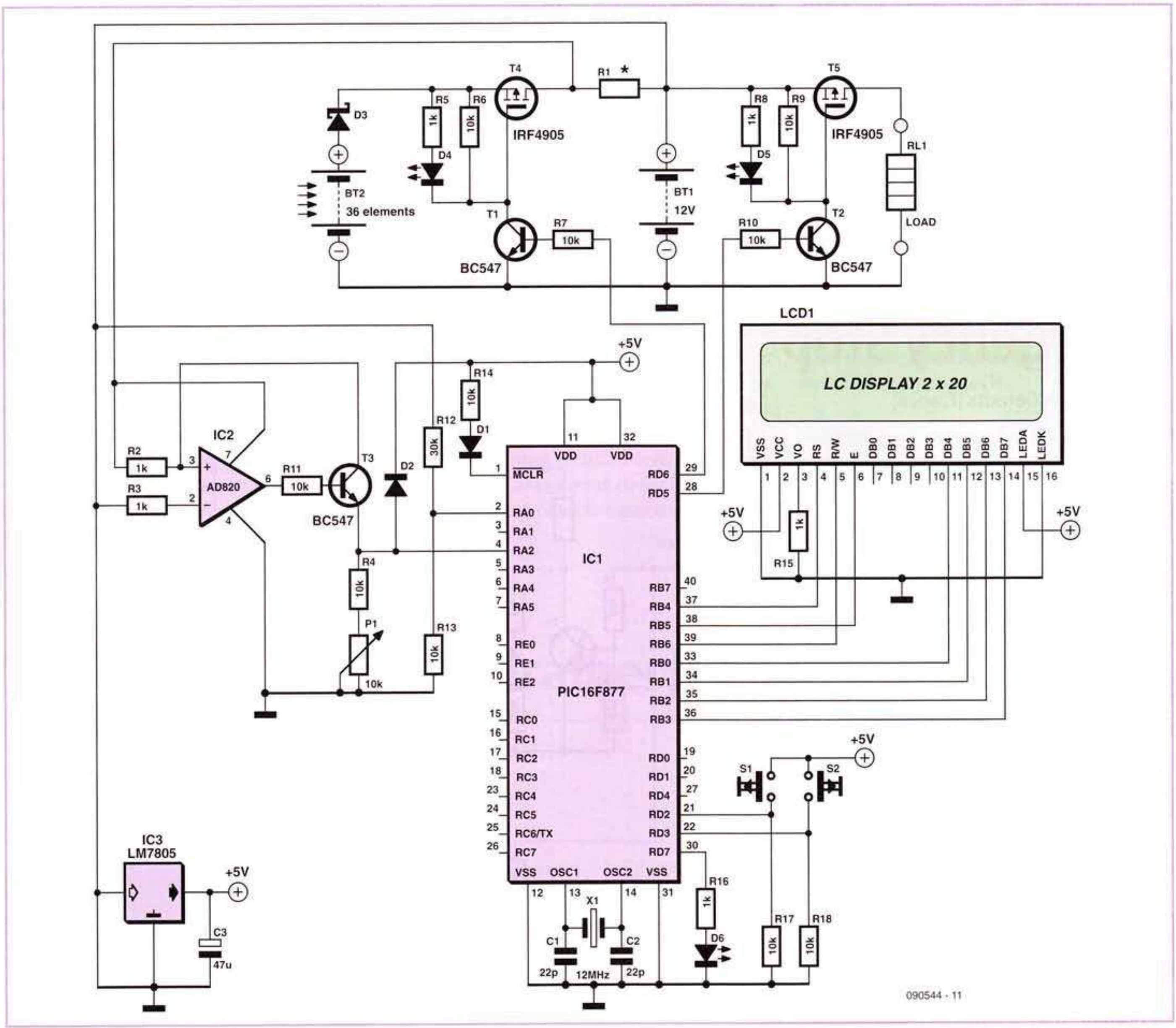
Solar Cell Battery Charger/Monitor



Matthijs Hajer (The Netherlands)

During the past year the author has built a standalone solar panel system, which included the construction of the panels themselves. Such a system stores the generated energy in a battery. This is in contrast to a mains-connected system where the excess energy generated is fed back into the national electricity grid.

A battery charger/monitor was designed in order to charge the battery correctly, protect it from deep discharges and to monitor its performance. Specifications for the solar panels: 150 watts maximum at 14.5 volts. With all the losses (glass, temperature, cables, etc.) taken into consideration, the measured current from the combined panels was about 7.5 A during sunny weather (the peak values stated by the manufacturer are rarely reached in practice).

This isn't a fast charger. This charger is intended to be used with solar panels and the like (wind and water energy), where the maximum charging current is much less than 0.1 of the battery capacity *C*.

The circuit is built around a PIC 16F877 microcontroller. The battery voltage is measured via input RA0 with the help of a 1:3 resistive divider. To measure the current, a 'high-side' reading is taken via R1 (with a value of about 0.03 Ω, using a number of resistors connected in parallel). IC2 amplifies the measured voltage across R1 and buffers it with T3. The resulting 350 mV/A signal is fed to input RA2 of the PIC. The opamp used for the current measurement needs to have a good rail-to-rail performance and a low input offset. The gain here is (R4+P1)/(R2) and the voltage across the resistor (R4+P1) is directly proportional to the measured current. The offset at the output, which is created by the opamp itself, is measured as soon as the 'info screen' is closed (pressing S1 or S2) and used as a 'null-offset' for the current measurement. D2 protects the PIC against too large a voltage at the input.

From the measured current and battery voltage the energy input and the capacity are calculated. This information is shown on the 4x16 LCD display.

FET T4 connects the solar panels to the battery to charge it and removes the connection once the battery is fully charged. FET T5 connects the load to the battery when the voltage is high enough and disconnects it when the battery voltage becomes too low. The Schottky diode prevents the battery from slowly discharging into the solar panel when it is dark. T1 and T2 are required to drive the FETs, which work at the battery voltage, with the 5 V outputs of the PIC. LED D4 and D5 indicate when the corresponding FET is turned on.

The Schottky diode, the current resistor and the FETs have to be provided with small heatsinks.

The code for the PIC is written in C and compiled using the HI-TECH C Pro (Lite mode) compiler included with MPLAB. The code uses very little memory and isn't very time-critical. The only thing you have to make sure of is that the firmware runs at a rate of about 10 times per second in order to obtain an accurate value for the capacity measurement [Ah].

Following a reset, the PIC loads the capacity values [Ah] & [mAh] from its EEPROM and an 'info screen' appears next. This shows the firmware version, the voltages at which the load is turned on and off, and the voltage at which the charger stops charging. When either of S1 or S2 is pressed, the PIC takes 10 measurements in order to determine the offset of the current measurement stage (IC2). The average is taken of these 10 measurements and that value is then used to correct all subsequent current measurements.

When S1 is pressed, the main program is started, where the battery voltage determines whether the load is turned on or off. When S2 is pressed, the load is immediately connected until the battery voltage drops below 11.5 V.

The main program is called 10 times per second; the LCD is refreshed at a rate of 2 Hz. In the main program the A/D converters are read first, after which the values for V, I, P and C are calculated. The results determine whether the charger and load are turned on. When the main screen is displayed, only S1 has a function: When this switch is pressed, the capacity [Ah] & [mAh] is stored in the EEPROM and the info screen is shown.

The watchdog function of the PIC has been enabled in this project. This way the PIC will be reset if the software crashes. In this case the info screen will appear again, and the charger and load are turned off, which is a safe state. In this way the battery is protected against over-charging or a complete discharge due to a crashed PIC micro. When programming the PIC you have to remember to set the configuration bits for the watchdog timer. At the start of the C code they are also set.

The limits for charging the battery were taken from the datasheet from Yuasa. This type of maintenance-free gel lead-acid battery is perfectly suitable for a small solar energy system. If you use a different type of battery you may have to adjust the voltages in the code somewhat. The values used here are:

14.5 V: Gassing voltage

13.6 V: Float voltage (small charging current)

12.7 V: No load, 100% charged voltage (no charging current)

11.5 V: 50% empty with small load ($I < 0.01$ C)

The charger turns on as soon as the battery voltage drops below 13.6 V. Should the voltage rise above 14.5 V during the charging, the charger will be turned off. Because the battery will be about 80% charged (according to the datasheet), the voltage will drop below 13.6 V again. When this happens, the charger will turn on again after 10 seconds and the battery voltage will rise. This process will repeat itself, but the 'charger-off' period will become longer the more the battery is charged. Overnight, a fully charged battery will slowly drop to 12.7 V.

Every 5 seconds the PIC transmits a text string via pin RC6/TX (2400 baud, 8n1), which shows the current state. This string could for example be sent to a web server or data-logger. An example string:

K_+12055|mV_+00826|mA_+00694|Ah_+00685|mAh–

The structure is as follows:

<Length>_<value>|<unit>_<value>|<unit>_<value>|<unit>>_<value>|<unit><CRC>

<Length> = length of the string incl. CRC (+ offset of 32 to stay within ASCII range)

_ = separator

<value> = field-value

| = separator

<unit> = units of the value

<CRC> = sum of the previous characters mod 256.

The source and hex code files for this project are available free from the Elektor website as archive file # 090544-11.zip. A programmed controlled is available under product number 090544-41.

(090544)

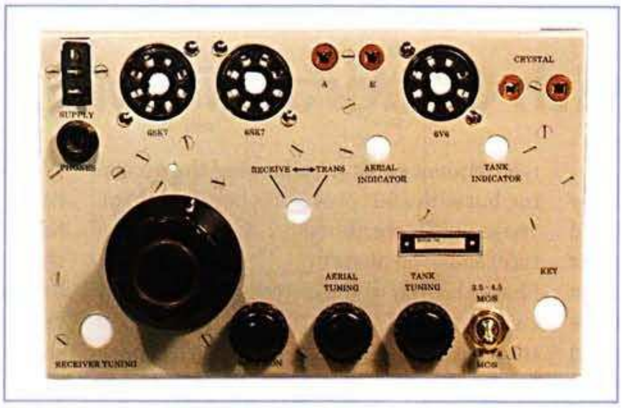
DIY Front Panels

Henk van Zwam
(The Netherlands)

This issue includes an article on a handy DIY front panel design program called Galva. Once your design is ready, the next question is how to convert it into a real front panel. One option for this is described here.

There is material available that you can print using your own printer. It is called waterslide transfer paper or waterslide decal paper, and it is the same as the decal material well known to many builders of model aeroplanes. You loosen the decals from the base material in water and then place them on the model aeroplane.

The material looks the same as photo paper, and it is available in two types: one for laser printers (including colour printers), and the other for inkjet printers. Transparent sheets and sheets with a background colour are available in both types. The letter colour is determined by the printer. A lot of information on this material, including demo videos, is available at a website we came across while researching this article [1].



It's just as easy to use as simply printing a normal sheet of paper. If you use a laser printer (colour or monochrome), the toner is melted into the material and is therefore well bonded. If you use an inkjet printer, you need to fix the ink to the material after printing. Spray cans with a special fixing agent are available for this purpose.

Now let's examine the process of applying the material to the front panel. After degreasing the aluminium front panel, coat it with several layers of matt grey undercoating for car paint (applied with a spray can). Cut the let-