

# The Rechargeable Battery “Cycler”

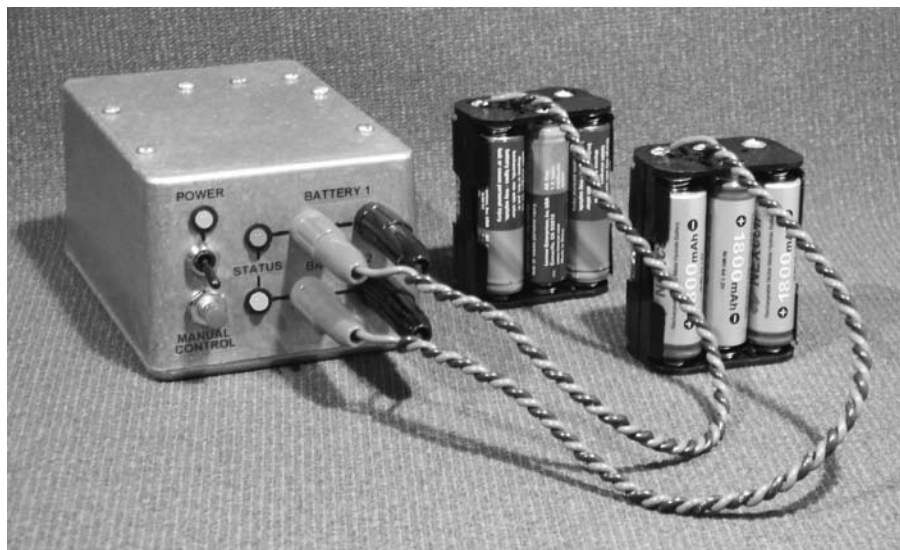
*With this computerized periodic discharger/charger, you will always be ready for action...and for an emergency!*

I am a sporadic handheld radio user. I find my dual-band radio mostly useful during the few hamfests I attend every year. I used to recharge the batteries the week prior to an event. I have, however, been caught far too often with my handheld radio battery packs in a depleted state when I needed them for unplanned activities. Even worse, I was never truly ready for an emergency situation requiring portable Amateur Radio communications. Are you ready to deploy? Always?

Nickel cadmium (NiCd) or Nickel metal hydride (NiMH) rechargeable cells are very economical and provide a lot of energy per weight, but they have drawbacks. The biggest issue is shelf life. These cells can lose up to 50% of their capacity within a month after recharging. Another concern is the now infamous “memory effect.” Whether the naming is appropriate or not, whether the effect is real or exaggerated, most manufacturers still recommend a regular discharge/charge cycle to maintain the energy storage capabilities. These rechargeable battery characteristics are well documented, and are worth reading about.<sup>1</sup>

Looking around on the market, I could not find a device that would do what I needed: an autonomous and periodic discharger/charger for my battery packs. Once again, if you can't find it, design it! So I have designed and constructed a discharger/charger circuit that I call the Rechargeable Battery “Cycler.” The Cycler will periodically discharge and recharge nickel cadmium (NiCd) or Nickel metal hydride (NiMH) battery packs so that they are always fresh and ready to deliver. A few years ago, I elected to use AA-size cell holder packs made specifically for my radio. AA-size rechargeable batteries are easier and cheaper to replace than custom-made battery packs. Whether you use sealed custom made battery packs that fit your handheld radio or regular rechargeable cells, this charger will do the job right.

<sup>1</sup>Notes appear on page 21.



Here is a list of key features of the design:

- It is a microcontroller-based approach, which brings a lot of features and flexibility into the unit. The device can easily be reconfigured when the battery specifications change.
- The Cycler supports one or two battery packs with distinctive configuration settings for either pack.
- The Cycler is designed to accommodate NiCd or NiMH battery cells of capacity C from 500 mAh to 5000 mAh, assuming the recommended C/10 charging current for 10 to 12 hours is used.
- The cycler can accommodate one to ten cells per battery pack.
- The firmware also offers a battery restoration mode, during which the batteries are discharged and recharged a number of times without any idle period between the cycles.
- A Windows-based software tool allows the user to configure the Cycler in a matter of minutes via an RS-232 link. After con-

figuration, the personal computer (PC) can be disconnected. All settings are saved in nonvolatile Flash memory.

- When left connected to the Cycler, the Windows tool provides discharge and charge data logging capabilities. This allows voltage-curve plotting using the built-in graph engine or using a third-party spreadsheet program.

- The following parameters can be configured via the Windows tool for both batteries: number of cells, battery capacity, end-of-discharge voltage, discharge current, charge current, charge duration, overall cycle period and number of conditioning cycles.

- All the electronics are integrated onto a small double-sided circuit board.

I could go on with the list of features, but instead let me explain how the battery cycler system works.

## System Description

Figure 1 shows the system's circuit schematic. The circuit is fairly simple and easy

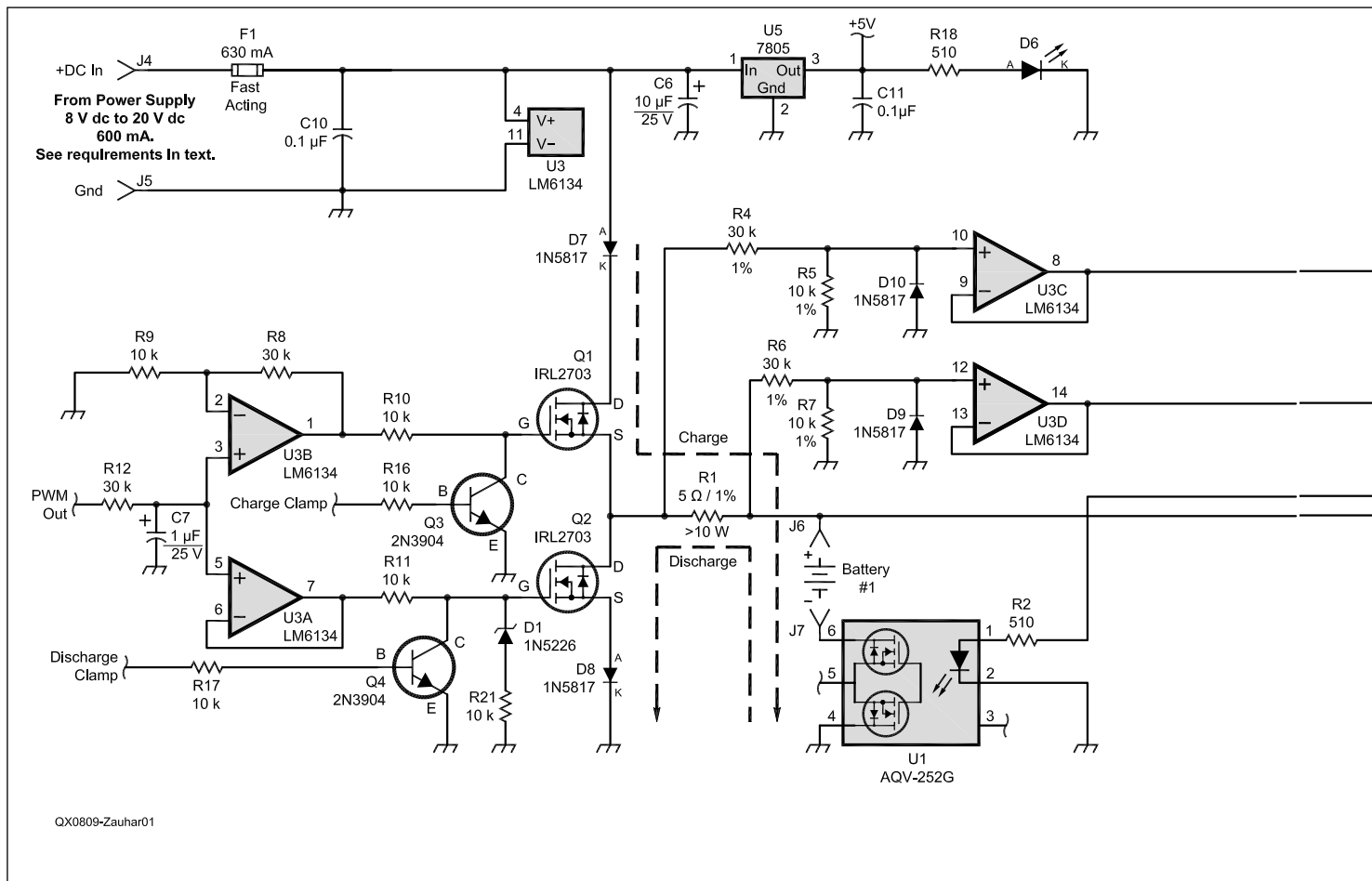


Figure 1 — Circuit schematic of the Rechargeable Battery “Cycler.”

## Parts List

### Component Reference

C1, C8, C9, C10, C11  
 C2, C3, C4, C5, C7  
 C6  
 D1  
 D2, D3, D6  
 D4, D5, D7, D8, D9, D10  
 F1  
 PB1  
 Q1, Q2  
 Q3, Q4  
 R1  
 R5, R7, R9, R10, R11, R16, R17, R21  
 R4, R6, R8, R12, R13  
 R2, R3, R14, R15, R18, R19, R20  
 U1, U2

U3  
 U4  
 U5  
 U6  
 Sockets for U1, U2

Socket for U3

Socket for U4

Socket for U6

Insulation kits for Q1 and Q2

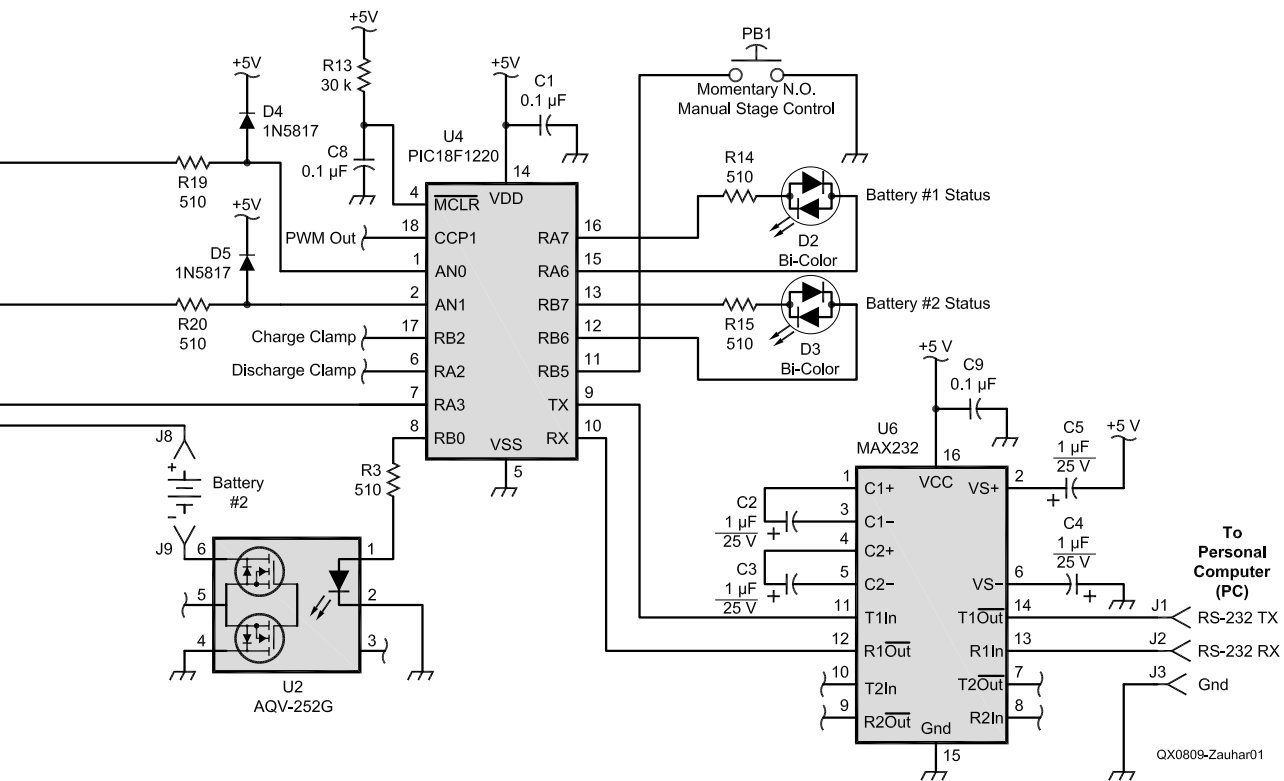
Enclosure

### Description

0.1  $\mu\text{F}$ , 25V or higher, radial  
 1  $\mu\text{F}$ , 16V or higher, electrolytic, radial  
 10  $\mu\text{F}$ , 25V or higher, electrolytic, radial  
 C6  
 1N5226B, 3.3 V Zener Diode, DO-35 package  
 Bi-color (Green-Red) LED, 5 mm, two leads.  
 1N5817 Schottky diode, DO-41 package  
 0.630 A Fast Acting fuse. “Pico II”, Axial  
 Momentary push button, SPST, Normally Open  
 IRL2703 MOSFET Transistor, TO-220 package  
 2N3904 NPN Transistor, or equivalent  
 5  $\Omega$ , 1%, 15 W, TO-220 package  
 10 k $\Omega$ , 1%, 1/4 W, axial  
 30 k $\Omega$ , 1%, 1/4 W, axial  
 510  $\Omega$ , 1/4 W, axial  
 Panasonic AQV-252G family Solid-State Relay,  
 DIP-6 package.  
 LM6134 Quad Operational Amplifier, DIP-14  
 Microchip PIC18F1220 micro-controller,  
 DIP-18 package, programmed part. See Note 3.  
 7805 Voltage Regulator, TO-220 package  
 MAX232 Conversion chip, DIP-16 package  
 IC Socket, 6-pin, 0.3” spacing, low profile  
 Shorten socket with heated hobby knife blade.  
 IC Socket, 14-pin, 0.3” spacing, low profile,  
 optional  
 IC Socket, 18-pin, 0.3” spacing, low profile,  
 recommended  
 IC Socket, 16-pin, 0.3” spacing, low profile,  
 optional  
 Thermally conductive, electrically insulated pad,  
 plus hardware.  
 Hammond 1590T Aluminum Enclosure,  
 suggested

### Part Number

Digikey: 399-4329-ND or equiv.  
 Digikey: P1196-ND or equiv.  
 Digikey: P1176-ND or equiv.  
 Digikey: 1N5226B-TPCT-ND or equiv.  
 Digikey: 160-1038-ND or equiv.  
 Digikey: 1N5817-TPCT-ND (lot of 10) or equiv.  
 Digikey: F2312-ND or equiv.  
 Digikey: EG2015-ND or equiv.  
 Digikey: IRL2703PBF-ND  
 Digikey: 2N3904-APCT-ND (lot of 10) or equiv.  
 Digikey: MP930-5.00F-ND  
 Digikey: P10.0KCACT-ND (lot of 10) or equiv.  
 Digikey: P30.0KCACT-ND (lot of 10) or equiv.  
 Digikey: P510CACT-ND (lot of 10) or equiv.  
 Digikey: 255-1791-5-ND  
 Digikey: LM6134BIN-ND  
 Digikey: PIC18F1220-I/P-ND  
 Digikey: LM7805CT-ND or equiv.  
 Digikey: 296-1402-5-ND or equiv.  
 Use Digikey ED3108-ND or equiv.  
 Digikey: ED3114-ND or equiv.  
 Digikey: ED3118-ND or equiv.  
 Digikey: ED3116-ND or equiv.  
 Digikey: 4724K-ND or equiv.  
 Digikey: HM159-ND or equiv.



to describe. As indicated on the schematic near battery no. 1, there are two main current paths in this circuit: the discharge and charge paths. Both are built similarly: a battery-selection solid-state relay, a current-controlling MOSFET and a current-sense resistor. These are monitored and controlled by the PIC18F1220 microcontroller firmware (its burned-in software).

### Current Control

Whenever a charge or discharge stage is about to begin, one of the batteries is first selected by the PIC microcontroller (U4) firmware via the corresponding solid state relay (U1 or U2). The amount of charge or discharge current is also controlled by the firmware. The PIC senses the voltage drop resulting from the current flowing across power resistor R1. Two built-in 10-bit analog-to-digital converter (ADC) inputs are used for this. A simple Ohm's law calculation is performed to determine whether to increase or decrease the current in order to match the user settings. The actual current control is performed by varying the gate-to-source dc bias ( $V_{gs}$ ) on power MOSFET Q2 for the discharge path or on Q1 for the charge path. The power MOSFETs operate as variable resistors and control the amount

of current flowing to and from the battery. The PIC microcontroller also has a built-in pulse-width modulator (PWM) output, from which a variable duty cycle square wave is produced. A downstream low pass filter made of R12 and C7 derives a corresponding "average" dc voltage. The higher the PWM duty cycle is, the higher the corresponding average dc voltage will be. The resulting action is similar to controlling a digital-to-analog converter (DAC) output.

Because there is only one PWM output on this microcontroller, both current control MOSFETs share the same dc control line. Since we don't want to enable the charge and discharge paths simultaneously (!), clamping switches made of NPN transistors Q3 and Q4 are added to shut off the unused MOSFET.

Operational amplifiers within U3 are used in 4 locations to buffer and/or scale the dc voltages. This is required, for example, for the ADC inputs sensing the power resistor via U3C and U3D. The driving impedance into the ADC inputs must be less than 2 k $\Omega$  in order for the ADC channels to sample correctly.

Zener diode D1 provides a smoother voltage-to-current response on MOSFET Q2. It also doubles up as protection to prevent the

gate voltage from exceeding the MOSFET's limit if operational amplifier U3 is ever to rail to its positive supply.

### Peripherals

Two bi-color LEDs (D2, D3) show the present stage of the two batteries. An optional LED, D6, indicates whether the Cyclor is powered on. A Manual Stage Control push-button, PB1, allows you to manually jump to the desired cycle stage. RS-232 chip U6 does the conversions between TTL and RS-232 signal levels for PC connectivity through a 3-wire TXD-RXD-GND connection. Finally, in order to guarantee a clean and more stable voltage supply to the board, a separate +5 V dc fixed voltage regulator, U5, is included.

### Firmware

The firmware running on the PIC microcontroller was written in assembly language. It is well documented in the source code file. Both the firmware source code and the assembled hex code can be downloaded from my Web site.<sup>2</sup> I can also supply pre-programmed PIC microcontrollers to those who don't have the ability to program PICs.<sup>3</sup> Once assembled, the hex code takes about 3 kilobytes of flash memory space.

## The Hardware

### Printed Circuit Board

I designed a double-sided circuit board to integrate the hardware.<sup>4</sup> The dimensions are  $2.3 \times 2.8$  inches ( $5.8 \times 7.1$  cm). Integrated circuit (IC) sockets are optional but recommended; this is especially applicable to the microcontroller and the solid-state relays. A fine tip soldering iron should be used for all components. Take all electrostatic discharge (ESD) precautions. The ICs and the MOSFETs are ESD sensitive devices, so care should be taken to guarantee a static-free environment when assembling the unit.

You may decide to build the circuit using other techniques such as breadboard and point to point wiring. The layout is not critical since there are no high-speed signals carried. Take care in proper dc supply decoupling using the  $0.1 \mu\text{F}$  capacitors near the ICs. Keep in mind the current to be carried on the outputs when selecting wire size for the solid-state relays, the MOSFETs and the power resistor.

### Enclosure

Figure 2 shows the Hammond diecast aluminum project box I used to enclose my Cyclor board. This type of enclosure offers two advantages. It has a thick wall that dissipates the heat generated by the MOSFETs and the power resistor, and it helps reduce EMI radiation in the shack.

Other enclosures can be used as long as you keep power dissipation in mind. Under worst case conditions, the two MOSFETs (Q1 and Q2) and the power resistor (R1) could dissipate up to 6 W in total. These three devices must have their tabs mounted to the enclosure wall or a decent sized heat-sink for proper heat dissipation. Make sure to electrically insulate the MOSFET metallic tabs from the enclosure because these are not at ground potential. Insulation and heat conductive kits are available to do this. (See the parts list.)

There is more information on how to put everything together using the suggested enclosure on my Web site. (See Note 2.)

### Battery Wiring

Wiring to the batteries is straightforward. Like any battery charger, the positive circuit board pad (+) connects to the positive battery terminal; the negative (-) circuit board pad connects to the negative battery terminal. This is done independently for both batteries (though the two positive (+) pads actually connect to a common point in the circuit). Colored banana plugs and jacks are a good idea for interfacing at the enclosure end. {I would use polarized connectors, such as Anderson PowerPole connectors. — Ed.} I strongly recommend using colored wiring as

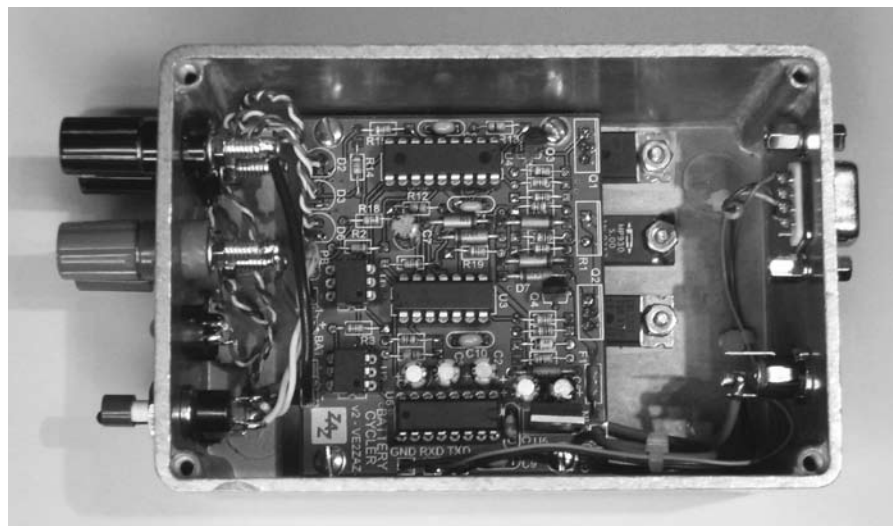


Figure 2 — The completed Battery Cyclor system in a Hammond diecast aluminum project box.

well, to help avoid battery reversal. Alligator clips or mini-grabbers will do a good job of connecting to the batteries. I use AA-size cell holders equipped with soldering tabs. A bit of creative thinking will likely be required to connect to sealed custom battery packs. One thing to understand here is that the wiring and terminals in the chain show some resistance. This will cause the voltage measured by the Cyclor to be different from the actual battery voltage, potentially by as much as a few tenths of a volt. So keep the wiring short, use wires of proper gauge and make sure you have good connections all the way back to the circuit board.

### DC Supply

A suitable external dc power supply for the Cyclor board is required. The supply voltage has to be at least 5 V higher than the expected fully charged battery voltage. This fully charged voltage turns out to be around 1.6 V per cell when still connected to the charger. A minimum dc supply of +8 V dc is required, but do not exceed +20 V dc. The maximum expected current is 100 mA more than the charge current set by the user. Assume a maximum current of 600 mA, to be on the safe side.

I have found that laptop power supplies make great power sources for this application. The output of these units typically ranges from +12 V dc to +18 V dc, with at least 1 A of available current.

### The Firmware and Software

I have created a *Windows* program to configure the Battery Cyclor; this makes the man-machine experience friendlier. The compiled *Windows* software is packaged in an installer program, and is available for download from

my Web site. The software will run under just about any 32-bit *Windows* version. Figure 3 shows screen shots of the *Windows* Battery Cyclor Configuration Tool, showing the system configuration, battery configuration and results log tabs highlighted. As seen in the figure, the tool is tab-based, with features distributed over several tabs.

The first tab provides the ability to set the number of batteries (one or two) connected. It also provides the ability to configure the type of discharge/charge cycle (periodic mode or battery restoration mode), the number of days in a cycle and the number of cycles during a battery restoration mode.

The second tab is where you set the battery parameters, such as the battery capacity, the number of cells, the end of discharge voltage threshold, the discharge current, the charge current and the charge duration. These parameters are individually set for both batteries. Charge and discharge parameters are filled in by the tool but can be changed if required. Note that the charge and discharge current ranges from 50 mA to 500 mA, which corresponds to a C/10 charge current for batteries ranging from 500 mAh to 5000 mAh. The combinations of voltages and currents are validated by the tool to ensure that the Cyclor can carry out what is requested. From the above entries, the cycle duration in days is also verified to ensure that the stages requested can be executed within the allocated time.

Another tab provides log and alarm download capabilities. This is useful to find out the current Cyclor stage, the duration of the previous stages, if there were issues detected during the entire cycle, the battery voltage and current values, and so on. More information on this is available in the

Operation section.

Finally, there is a settings transfer tab (not shown in Figure 3), where you will be able to read the current settings from the Cycler or write new settings to it. After selecting the computer COM port, reading from or writing to the Cycler is just a matter of pressing a button. A progress bar informs you of the status of the transfer. Everything completes in a matter of a few seconds. After reading the settings from the Cycler, all fields in the various tabs are updated with the currently configured settings. A change to only one field involves a read-modify-write process. There is nothing special to perform in order to save the data to non-volatile memory. A “Write Settings” command will permanently save the data inside the microcontroller. After configuration is complete, the user can disconnect the cable or leave it connected for log and alarm monitoring purpose.

## Operation

The best way to explain how the PIC firmware operates is by using flowcharts. Figure 4 shows the firmware flow for the two cycle modes available, namely the periodic mode and the battery restoration mode. The periodic mode (Figure 4A) is the regular mode to maintain optimum battery performance. At a repeating period (in days) set by the user, the batteries get discharged and then recharged, battery no. 1 first, followed by battery no. 2. If the Cycler is configured to service only one battery, battery no. 2 is skipped. These stages are followed by an idle stage (nothing happening) that runs until the period set by the user is reached. The Cycler then starts a new cycle as described above.

The battery restoration mode (Figure 4B) is used to repeatedly cycle the batteries with the intent of restoring their charge holding capability. In this mode, the batteries are discharged and then recharged, battery no. 1 first, followed by battery no. 2. This discharge/charge cycle repeats without idle period until the number of cycles set by the user is reached. These cycles are followed by the off stage, which turns off any activity on the Cycler. From there, pressing the Manual Stage Control button will initiate a regular periodic cycle with the previously configured settings. Note that the restoration cycles do not begin until the user depresses the button.

The first time the unit is powered up, the Cycler firmware is set to the off stage. Depressing the Manual Stage Control button advances the Cycler to the next stage in the cycle, as shown in the flow chart on Figure 4A. Repeatedly pressing the button, within less than 3 seconds, advances to the next stages in rotation. When the button is left alone for 3 seconds, the selected stage begins. Note that the currently running stage is still

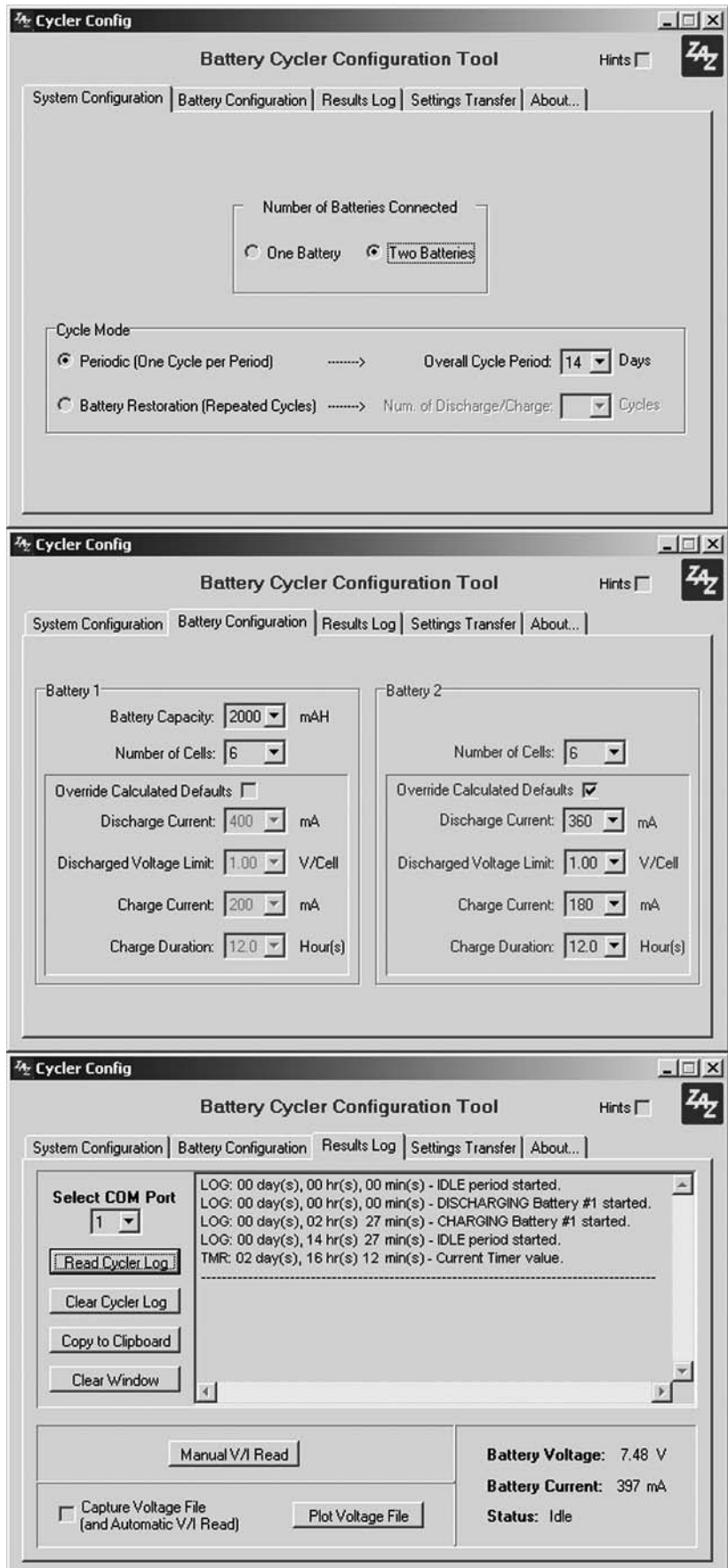


Figure 3 — Screenshots of the Battery Cycler Windows configuration tool, with the system configuration, battery configuration and results log tabs highlighted.

executed until the 3-second delay expires. Also note that the cycle timer is reset every time a stage is initiated via the Manual Stage Control button.

### LEDs

There are three light emitting diodes (LEDs) equipped on the Cycler. The first two (D2 and D3) are bi-color LEDs. Having a green/red LED pair in a single package enables the generation of three colors: green, red and amber. The latter is generated when both green and red LEDs alternate at a fast rate. In our application, amber indicates the discharge stage, red indicates the charge stage and green indicates the idle stage. Extinguished LEDs indicate the off stage. The battery no. 2 LED can also be seen permanently off, which indicates that the second battery channel is disabled.

In addition to solid colors, there are two LED flashing scenarios that may be seen. The first one happens when the user manually toggles between the stages, triggering the 3-second delay mentioned previously. The flashing rate will be about three Hertz. After the 3 second delay, the selected stage begins and the LEDs are solid colored. The second flashing scenario will occur when the Cycler detects a low battery voltage or is unable to generate the right amount of current. The corresponding battery LED will flash at a fast rate, indicating an alarm on that battery. More description of this feature is provided in the Alarm section below.

The third LED (D6) merely indicates the power on/off status of the cycler. A bi-color LED can be used there as well, but only one color will be shown.

### Logs, Readings and Alarm Messages

As seen on Figure 3, the Battery Cycler provides the user with information messages via the *Windows* tool. These messages can be consulted under the “Results Log” tab. The Log buffer has sufficient memory space to keep the logs and alarms of an entire cycle.

Whenever a new stage begins, a log message is generated, along with a timestamp generated by the internal timer. This allows the user to verify that the various stages occur as planned. The built-in timer starts at zero at power up and whenever a new cycle begins. It has a resolution of two minutes and an accuracy of  $\pm 2\%$ .

In addition to log messages, there are also alarm messages generated. Examples of such messages are: “Unable to generate the specified charge current” and “Insufficient battery voltage during idle stage.”

Another informative feature is the battery voltage and current display and logging. Pressing a button in the “Results Log” tab provides a reading of the instant battery voltage and current as seen by the PIC. This feature is available during the charge and discharge stages. This voltage and current reading can also be automated so that the results get logged onto the PC. When enabled, the automatic voltage/current read feature takes a reading every 30 seconds and saves the values along with a timestamp (in seconds) in a text file for further analysis. This “voltage file” can be plotted directly within the software using the built-in graph engine. An example plot is shown on Figure 5. The voltage file can also be analyzed and plotted using any third-party spreadsheet program

(Microsoft *Excel*, *Open Office Calc*, or other programs). When analyzing a discharge stage voltage file, the plot feature also calculates and displays the battery capacity in mAh. The calculation is rather simple: the discharge current is multiplied by the discharge time. This assumes that you have captured the whole discharge stage in the voltage file, of course. The calculation will show you the “real” capacity of your batteries, not the theoretical one. Be prepared for some disappointment here...

### Built-in Robustness

The Cycler will gracefully recover from a power outage or disconnection. If the power outage occurs during a charge or discharge stage, the Cycler will restart in the discharge stage. If the Cycler recovers from a power outage while it was in the idle stage or in the off stage, it will restart in that same stage.

The Cycler also has various anomaly detectors implemented in firmware. Whenever these trigger, an alarm log is generated. Here is a list of anomalies and actions.

- The Cycler will abort the charge stage if it is unable to generate the charge current requested by the user.
- The discharge stage will continue if the cycler is unable to generate the discharge current requested by the user, but an alarm log will be generated.
- The Cycler monitors the health of the batteries during the idle stage at a two-minute interval. If the battery voltage falls below the end-of-discharge threshold voltage configured by the user, the corresponding LED flashes at a fast rate and an alarm is gener-

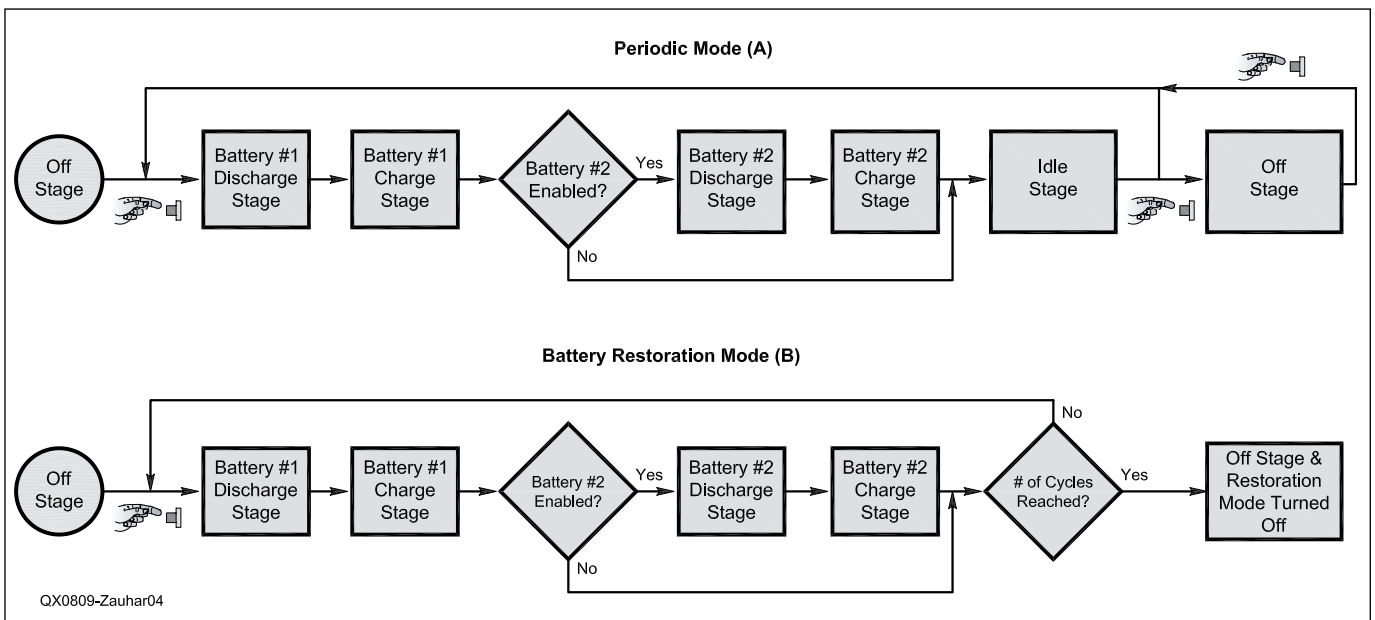


Figure 4 — Flowchart of the Cycler firmware operation for the two supported modes.



ated.

- The discharge stage will end if the battery voltage does not fall below the end-of-discharge threshold within a time limit calculated as 150% of the theoretical discharge time.

- The Cyclor is also equipped with a fuse (F1) that will protect the battery and the circuit if a fully charged battery is inadvertently connected with its polarity reversed. If the generated current is high enough, the fuse will blow. Otherwise, the Cyclor will still generate an alarm claiming that the requested discharge current cannot be produced, which should raise user suspicion about the wiring.

## Accuracy and Resolution

The Cyclor voltage and current reading accuracy and resolution call for some additional explanation. The design intent was to create a circuit that would have sufficient accuracy and resolution to properly charge and discharge batteries. In that sense, it is not meant to replace an accurate voltmeter or ammeter. The user must keep this in mind when analyzing the readings.

Assuming that the component tolerances meet those specified in the parts list, the Cyclor absolute worst case accuracy on the voltage readings is estimated to be  $\pm 6.5\%$ . The current reading has an absolute worst case accuracy of  $\pm 24\%$ . These accuracy errors are worst case scenarios assuming at-the-limit component tolerances, which is a very unlikely scenario. In fact, if the resistors of the same value come from the same batch, which is usually the case when you purchase them new, then the overall accuracy will be much better. This is what I measured on the two prototype assemblies I put together. The accuracy errors are mainly generated by (from the largest to the smallest contributor):

- Difference in resistance between R5 and R7.
- Difference in resistance between R4 and R6.
- The +5V supply rail accuracy (U5), which is used as the reference for the highest value (4095) read by the PIC microcontroller ADCs,
- Resistance tolerance of R1.

Assuming a more-or-less stable temperature, the offsets on the voltage and current reading can be measured manually using an accurate meter and taken into account if trying to derive a better accuracy.

When making a voltage/current reading via the *Windows* tool, the resulting voltage and current values represent an average of 30 readings. This is done by the *Windows* tool to null out any ripple caused by the firmware when maintaining the current to the user-

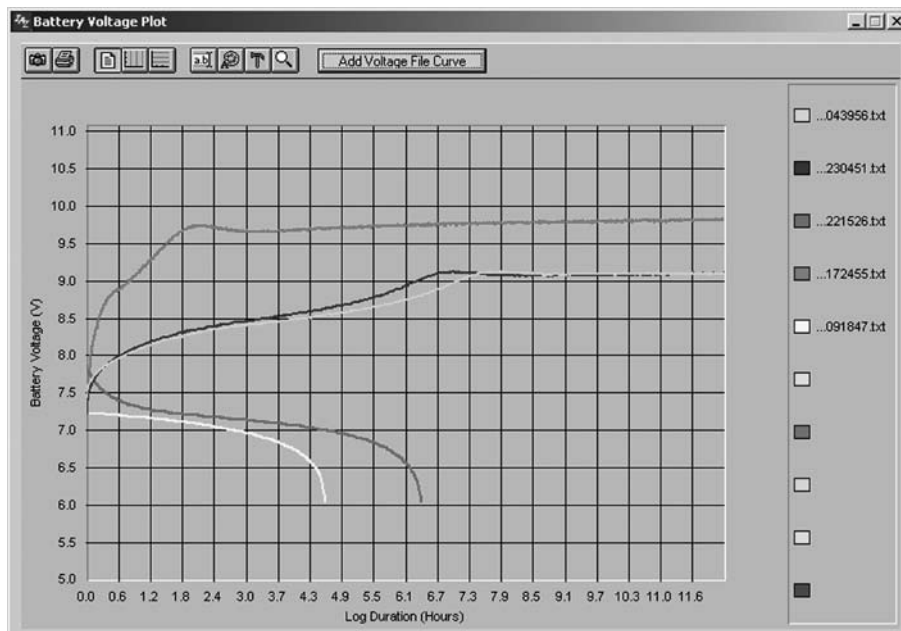


Figure 5 — Various charge and discharge curves can be displayed concurrently by the Battery Cyclor *Windows* configuration tool.

requested value. As a result, two decimals are provided on the voltage values. This additional resolution provides much better trend analysis and plotting clarity. The user must remember though that the accuracy study provided above still applies. In other words, better resolution does not translate into better accuracy.

## Conclusion

This Rechargeable Battery Cyclor fulfills what few other commercial chargers do — it keeps my rechargeable handheld radio batteries fresh and ready for action. Not only does it keep them charged up, but it cycles them at regular intervals so that their useful life is maximized.

Assuming a reasonably stuffed junkbox, you will be able to assemble and own the Cyclor for around \$60 US. This compares favorably with off-the-shelf high-tech universal chargers, which are not autonomous...

Now you will be able to claim you are always ready for emergency action! That by itself is reassuring, isn't it?

## Notes

<sup>1</sup>Wikipedia, the free online encyclopedia, provides a good overview of Nickel Cadmium and Nickel Metal Hydride battery technologies and how to optimize their performance. See [en.wikipedia.org/wiki/Ni-Cd](http://en.wikipedia.org/wiki/Ni-Cd) and [en.wikipedia.org/wiki/NiMH](http://en.wikipedia.org/wiki/NiMH) for more information.

<sup>2</sup>I maintain a Web site where I provide updates to this project, source files and additional comments. Please visit [ve2zaz.net](http://ve2zaz.net).

[net](http://ve2zaz.net) for more details. The PIC program files and *Windows* configuration software files are also available for download from the *QEX* Web site. Go to [www.arrrl.org/qexfiles](http://www.arrrl.org/qexfiles) and look for the file **09x08\_Zauhar.zip**. The files available on the *QEX* Web site are the latest versions as of press time, but **ve2zaz.net** may have more recent files.

<sup>3</sup>For those of you who cannot program Microchip PIC18F series micro-controllers, I make pre-programmed PIC microcontrollers available for purchase. Please contact me via e-mail for more details.

<sup>4</sup>I distribute high quality, fully etched, plated-through bare circuit boards for this project. Please contact me via e-mail if you are interested in obtaining a circuit board.

*Bertrand Zauhar has been a radio amateur as VE2ZAZ since 1984. He holds an advanced amateur license. Bertrand has designed for the hobby — among other things — a micro-controller-based radio T/R Sequencer (QST, March 2007), a GPS-Derived Frequency Standard (QEX Sept/Oct 2006), an L-band transmit converter (Amsat Journal, May/June 2003), a 1 to 12 GHz frequency counter prescaler, a microprocessor-based repeater controller, several amateur satellite antennas and an RF-sensing alarm (73 Amateur Radio, May 1998). Bertrand received his Electronics Engineering degree in 1989 from École Polytechnique de Montréal. Since then, his professional engineering career has been spent working for Nortel at the Montréal and Ottawa locations. In his current position, he is an electronics hardware design engineer on optical transmission equipment.*

