



## Application Note

AN2260

### ***Rapid NiCd/NiMH Battery Charger and DC Brushed Motor Controller for Autonomous Appliances***

**Author:** Victor Kremin

**Associated Project:** Yes

**Associated Part Family:** CY8C24xxxA

**PSoC Designer Version:** 4.2 + SP2

**Associated Application Notes:** AN2041, AN2107, AN2168, AN2246

#### **Abstract**

The proposed Application Note describes an electronic controller for autonomous appliances such as screwdrivers, drills, electric shavers, and brushing machines. The controller provides a rapid built-in NiCd/NiMH battery charge and motor powered by batteries or an external DC supply.

#### **Introduction**

Small autonomous electric appliances such as shavers, screwdrivers, portable drills, and brushing and haircutting machines are widely used. In spite of the inherent differences, all these devices have a motor, most typically a "classic" brushed DC motor. They also have batteries and a built-in or external battery charger to restore battery energy after use. Many of these devices allow the motor to be powered directly by an external DC supply when the battery is depleted or external power is available.

The goal for this project is to develop a simple low-cost design that combines a brushed DC motor controller and a rapid NiCd/NiMH battery charger. The user interface was kept to a minimum: there is only one push button to turn the motor on or off and a single LED to display the battery charge status. The controller rapidly charges batteries and allows the motor to be powered from batteries or an external DC supply. Controller technical specifications are given in Table 1.

**Table 1. Controller Specifications**

Item	Value
Battery Cells	2
Capacity	0.75-2.3 A/h
<b>Battery Charge Current:</b>	
Rapid	1 A
Trickle	0.15 A
Motor Supply Voltage and Power	2.5V, 3W
External DC Supply Specifications	15-20V, 0.35A
No Use Current Consumption	40 $\mu$ A

#### **Device Schematic**

The simplified device schematic is shown in Figure 1.

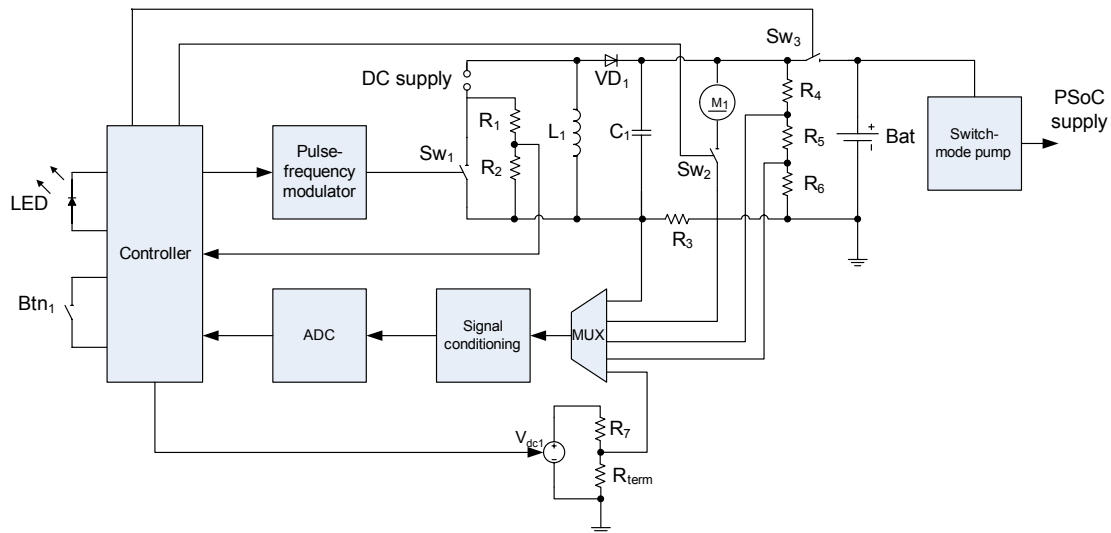


Figure 1. Simplified Device Schematic

The device consists of:

- Controller to implement battery charging and user-interface algorithms.
- Spread spectrum pulse-frequency modulator.
- Switch regulator,  $Sw_1$ ,  $L_1$ ,  $C_1$ ,  $VD_1$ .
- External DC supply inclusion-detection circuit,  $R_1$ ,  $R_2$ .
- Current sense resistor,  $R_3$ .
- Motor switch,  $M_1$  and  $Sw_2$ , respectively.
- Resistive divider,  $R_4$ - $R_6$ , for motor or battery,  $Bat_1$ , voltage measurement.
- Battery switch,  $Sw_3$ .
- Switch mode pump (SMP) to increase battery voltage to acceptable levels for the PSoC™ device.
- Analog multiplexer MUX to commutate different analog signal sources.
- Signal conditioning circuit for transforming input signals into valuable analog-to-digital conversion (ADC) levels.
- Circuit for thermistor,  $R_{term}$ , resistance measurement:  $V_{dc}$ ,  $R_7$ .
- User interface elements of push button,  $Btn_1$ , and LED.

The external DC supply voltage can be several times higher than nominal motor or battery voltage. This provides two significant advantages. It reduces the converter input current, which lowers the cost and weight of the external DC supply. It also allows the use of thinner connection wires and less expensive connectors. In this mode, the step-down converter switch duty cycle is small and the use of fixed-pulse width variable frequency control achieves a smaller duty cycle step.

The step-down switch regulator can be built around different configurations with different switch placement. The inverting regulator configuration is used in this project because of the following advantages:

- Only one N-channel MOSFET is required, which can be controlled directly by the device without any translators.
- The on-resistance for N-Channel transistor is less than for P-Channel in the same price range.
- The inverting converter configuration is inherently protected from load overload due to limited energy quantity, which is transferred by inductor charge/discharge cycles.

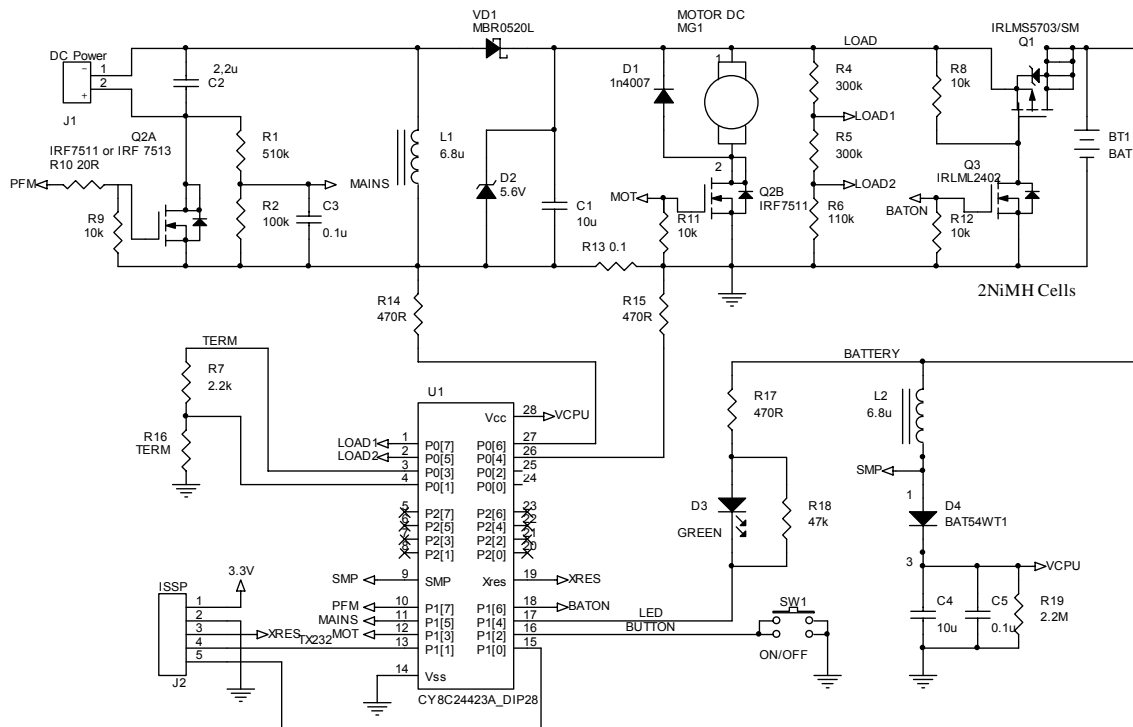


Figure 2. Complete Device Schematic

The complete device schematic is shown in Figure 2. Because most of the analog signal processing is implemented inside the PSoC device, only a few external components are required. Q2A is the regulator switch. R1R2C3 is the external power-supply detection circuit. D2 suppresses possible high voltage spikes when the switch regulator has no load (both Q2B and Q1 are open.) Q2B is the motor control switch. Q1 and Q3 are battery switches. They can be single load switch (FDC6331L, FDR8521L) from Fairchild Semiconductor.

The SMP circuit consists of D4, L2, C4 and C5 elements. R19 completely discharges storage capacitors when the battery is disconnected. LED D3 is directly powered by a battery to offload the SMP when it is activated. The open drain low output drive mode at P1[4] is used. R18 provides logic 1 for port read-modify-write instructions when the LED is in off state. Button SW1 uses the internal pull-up resistor, so no external components are required. J2 is the ISSP/debug connector.

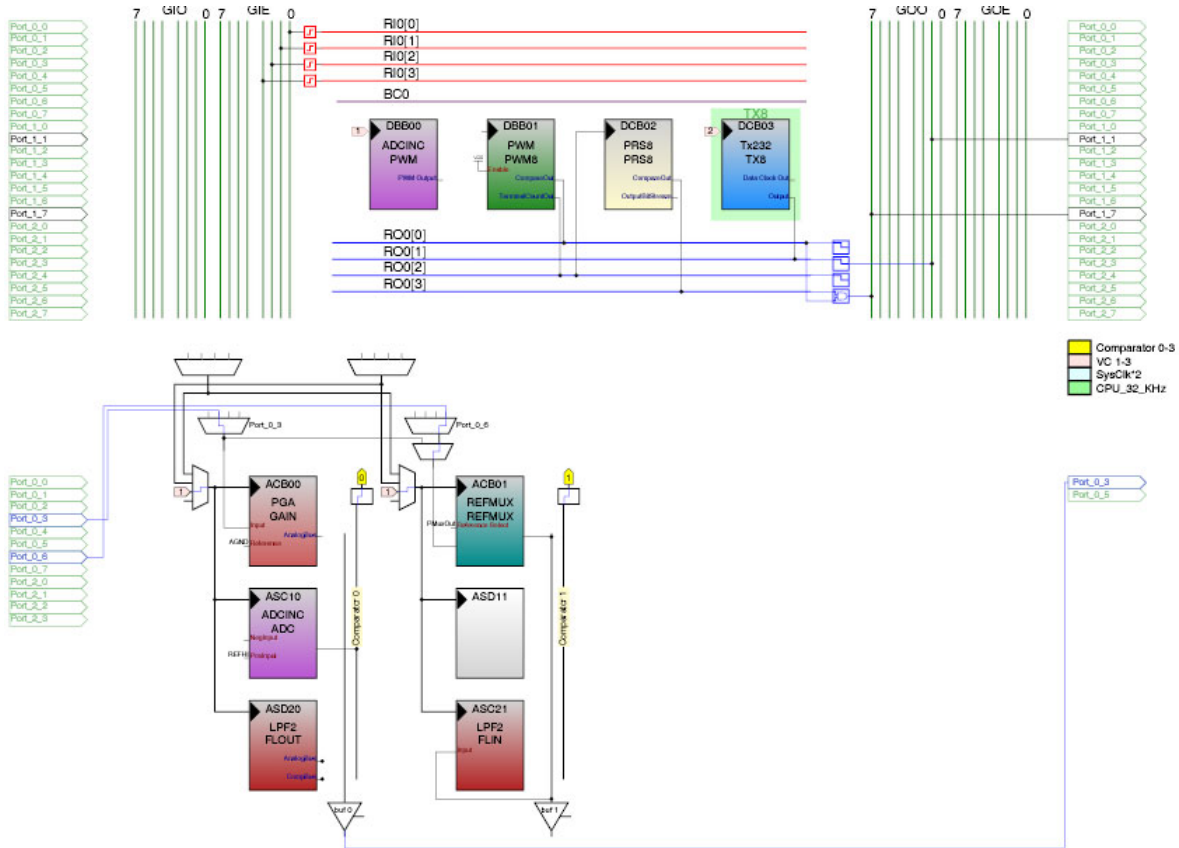


Figure 3. PSoC Device Internals

### PSoC Device Internals

The PSoC device's digital and analog user module placement is shown in Figure 3. The incremental ADC is placed in DBB00/ASC10. The resolution is set to 11 bits and conversion time is 4 ms. The spread-spectrum, pulse-frequency modulator (PFM) consists of a conventional fixed-pulse width PWM source placed at DBB01. An 8-bit PRS generator is placed at DCB02. The PWM and PRS user modules compare outputs and are AND-ed using the row LUT. The idea for this source was suggested in Application Note AN2246 "PWM Source - High Frequency, High Resolution." Figure 4 shows the PFM internal interconnection.

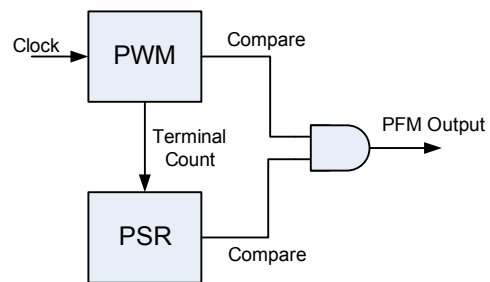
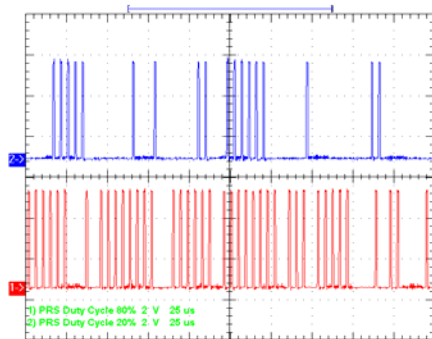


Figure 4. PFM Generator Interconnection

The source forms variable number, fixed-width pulses at the PRS repeat period, which is equal to 255 PWM output period cycles due to the use of maximum length polynomials. The PWM pulse duty cycle is set at 0.23. By varying the PRS compare value in range 0...254, the duty cycle varies from 0 to the PWM generator duty cycle, or 0.23 in this design. The pulse width duration is 1  $\mu$ s, and the minimum pulse repeat interval is 4.3  $\mu$ s. Figure 5 shows the PFM output signal scope images for different duty cycle values.



**Figure 5. PFM Output Signal at Different Duty Cycles**

The signal conditioning circuit consists of:

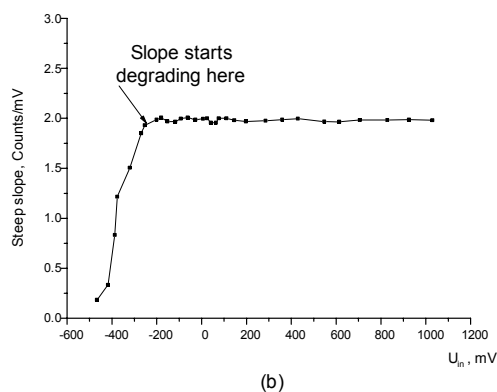
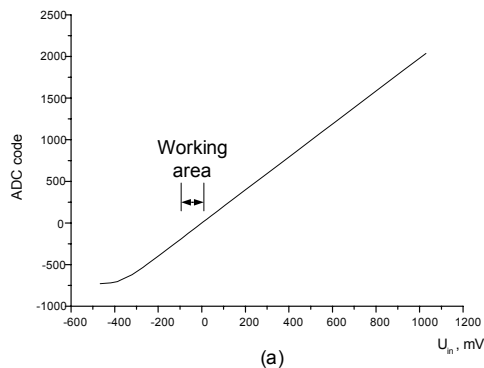
- PGA with unity gain, which is used to buffer input signals during the load (battery or motor) voltage.
- Thermistor resistance measurements.
- Level translator.
- Low-pass filter (LPF) to handle the signal for the current sense resistor.

The ADC input is multiplexed in the firmware. Depending on the required signal source, the signal comes from either the PGA or the LPF outputs.

This type of sense resistor signal is relatively low (maximum 100 mV in this design). It cannot be accurately digitized by directly using the ADC. Also, the PSoC device's amplifiers are not rail-to-rail on output, which limits the usefulness of the PGA for amplification purposes. This design uses the following technique to remove the hardware limitations: the input signals from port 0 analog inputs are directly passed to column analog bus via TestMux and fed to a low-pass filter input. The RefMux of ASC21 is set to RefLo, and RefLo level is Vss (reference is BandGap  $\pm$  BandGap). It transforms the filter input stage in the bias-free level translator that shifts Vss-related signals to AGND-related, which are easy to process using the LPF as gain stage.

In this design, the filter gain is near 20.5 dB, permitting effective utilization of the ADC dynamic range. This aspect of switching the capacitor operation is discussed in further detail in Application Notes AN2041 "Understanding Switched Capacitor Analog Blocks" and AN2168 "Understanding Switched Capacitor Filters."

The user may be surprised by the current sense resistor connection: during charge or motor powering from an external supply, the P0[6] potential is negative relative to Vss ground. This is not a problem because the level translator works linearly to 300 mV for negative input signals, which is much larger than maximum current sense resistor voltage drop (100 mV). To confirm this, Figure 6 illustrates the transfer characteristic for a bipolar input signal (measurements were taken with a 12-bit ADC with reduced filter gain). Note that the user can move the Vss connection point to the left of the R11 pin, which eliminates any negative input voltage sent to the PSoC device pins.



**Figure 6. Negative Voltage Measurement Transfer Characteristic (a) and Slope Estimation (b)**

It is useful to observe the dynamic behavior of the internal variables (such as battery voltage and current, and state machine state) during controller firmware debugging. The serial transmitter, placed at DCB03, is used for this purpose. The dedicated PC software allows users to monitor details of the operation, which greatly simplifies firmware debugging. The communication speed is set to 19200 bps and the transmitter is clocked by a VC2 source.

The slow mode main oscillator (6 MHz) in this design reduces power consumption at runtime. The SMP trip voltage determines the device's supply level. It is set to 2.55V in sleep state when the device is not used and boosted to 4.64V when external power is available. This reduces switch regulator MOSFET resistive losses by providing a higher gate voltage. The user can set another power level (such as preset 3.25V supply when MOSFET with smaller threshold voltage is used).

Also, the Power Setting field under Global Resources needs to be set according to the selected supply voltage in order to correctly load oscillator and BandGap trim values. The CPU clock is 3 MHz, which allows the system to run from a 2.55V supply.

## The Firmware

The controller firmware consists of the following:

- Control state machine
- Timekeeping services
- Electrical values measurement routines
- PFM duty cycle regulator
- Battery charge algorithm
- Sleep and Watchdog Timer handling
- Runtime debug support

The control state machine provides device operation in different states, such as powering the motor from battery and external supply, battery charging support, low-supply sleep mode, and so on. Figure 7 shows the machine state diagram.

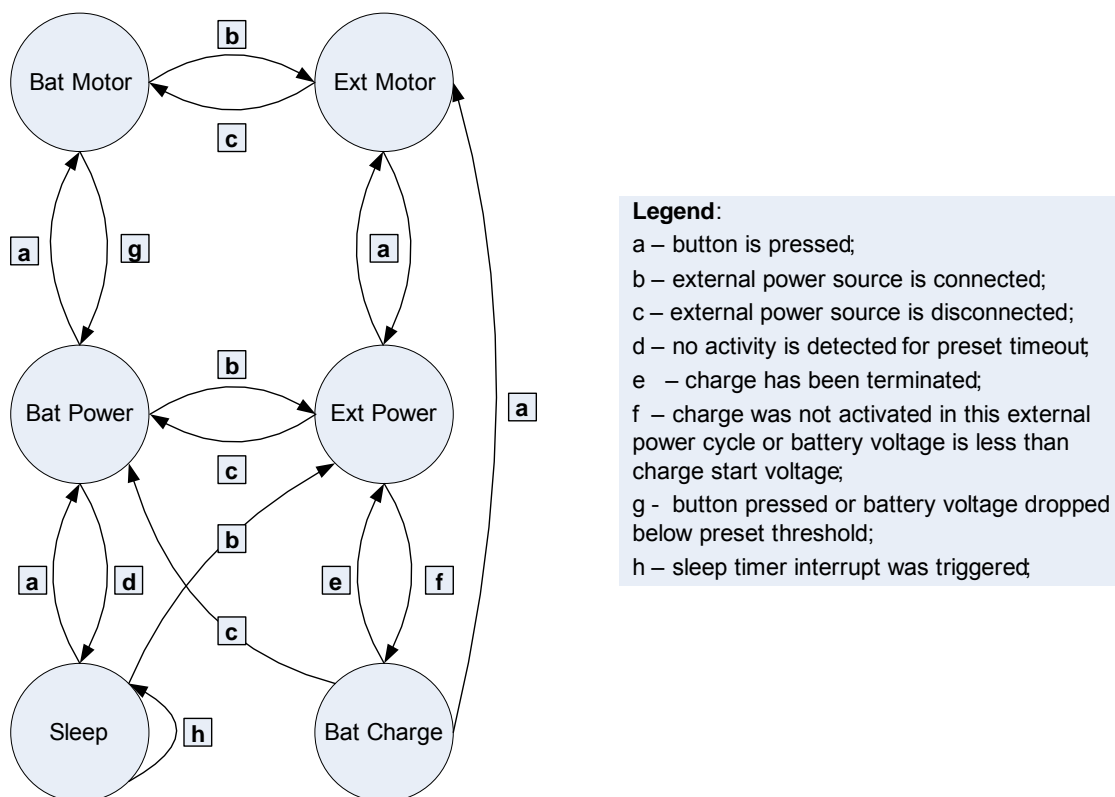


Figure 7. State Machine Diagram

Table 2. Controller State Descriptions

State	Description	LED Status	State Value
Bat Power	Device is battery powered	Off	0
Bat Motor	Motor is powered from battery	Off	1
Sleep	Device is in low-power sleep state	Off	8
Ext Motor	Motor is powered from external DC supply, no battery charge	Off	2
Ext Power	External DC supply is applied, motor is off	On	3
Bat Charge	Battery is charged, motor is off	Blinking	6

There are 4 working states (**Bat Motor**, **Ext Motor**, **Battery Charge**, **Sleep**) and two supplemental (**Bat Power**, **Ext Power**), all of which are described in Table 2.

When the device is turned on, it enters the **Bat Power** state. If the user does not press the button or if the external power is not connected within a predefined timeout, the **Sleep** state is activated. This state can be changed if the power button is pressed or if an external supply connection is detected. If the device is powered from batteries and the user presses the button, the **Bat Motor** state is activated and the motor is turned on. When the external DC supply is connected, the **Ext Motor** state is activated and the motor continues to run from an external supply, preserving battery energy and maintaining constant voltage to the motor via the switch regulator. When the external power supply is removed, the device returns to **Bat Motor** state and the motor runs continuously. Note that the battery voltage is controlled when the motor is powered from battery. When the battery voltage drops below a predefined value, the controller turns off the motor and switches to the **Bat Power** state to prevent extreme battery discharging.

While the controller is in the **Bat Power** state and an external supply connection is detected, it switches to **Ext Power** state so it can determine if the battery is being charged during this external supply connection cycle. If not, it starts charging the battery by jumping to **Bat Charge** state. **Bat Charge** state is maintained until the battery voltage slope equals zero. Also, the controller leaves this state when the external supply is disconnected or if the power button is pressed, which changes the state from **Bat Power** to **Ext Motor**.

The next charge cycle can be initiated when the power supply is disconnected and reconnected or if the battery voltage drops below a predefined value.

At first glance, the state machine looks too complicated due to high branch count to implement compact code. This can be resolved by assigning numerical values to each state variable (by applying bit-based state representation) and using direct bit-wise operations to handle the most frequently used branches. Using the state variable values from Table 2, the branches (a), (b), (c) and (g) shown in Figure 7 are handled by several lines of 'C' code as shown below:

```

if(CHECK_MAIN_SUPPLY())
{
    new_state &= 0x07;
    new_state |= 0x02;
}
else new_state &= ~0x06;

if (check_button_pressing())
{
    new_state &= 0x03;
    new_state ^= 0x01;
}

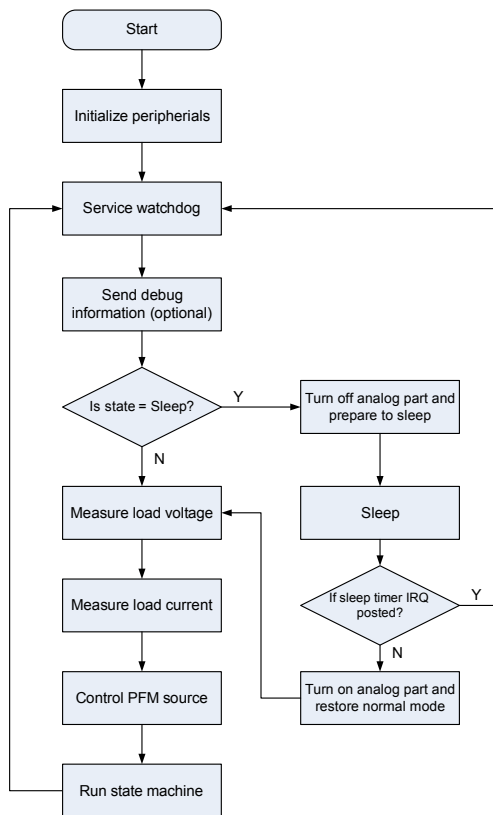
```

Code 1. State Variables in 'C'

The firmware switches the controller into sleep mode to conserve current consumption when the device is not used for 1 or 2 seconds. The device can wake up with a 1 Hz Sleep Timer interrupt or by GPIO interrupts. The Sleep Timer interrupt is used only to clear the Watchdog Timer; no other action is taken. The analog user modules are turned off in such state to minimize power consumption.

The GPIO interrupt occurs when the power button is pressed or an external power supply is connected. This event completely wakes up the controller by turning on the analog user modules.

Wakeup is handled by a state machine. It jumps to a new state depending upon the reason to wake up. After measuring the current and voltage, the PFM duty cycle control routines are executed in a loop. Figure 8 shows the controller operation in sleep and non-sleep states.



**Figure 8. Controller Operation in Different States**

As previously mentioned, the device uses dynamic power supply management to reduce current consumption. The power supply is set to a minimum value when in sleep mode and increases to higher a value when in normal mode.

The controller firmware (battery charging algorithms, electrical values measurement routines, PFM duty cycle regulator, etc.) includes timekeeping functions. There are state and interval timers with 1s resolution obtained by division VC3 interrupts at 600 Hz frequency. The precision of these timers is determined by the accuracy of the IMOs. The timers are primarily used by battery charging algorithms.

These timers can also be used to limit the duration of the motor-on states (**Bat Motor**, **Ext Motor**). This prevents overheating, as required in some applications such as power drills. The blinking LED, debounce push button switch, event generation, sleep periodic wakeup and debug information transmission are implemented using a low-accuracy Sleep Timer interrupt. The nominal interrupt rate is 1 or 8 Hz, depending on the present state.

The controller measures battery or motor voltage, charge/motor current and battery temperature. These values are used during rapid battery charge and to regulate the motor supply voltage when powered from an external DC supply. Voltage measurement is implemented by measuring the voltage drop on resistor R5 (see Figure 1) using a unity-gain PGA and an ADC. The value is measured for upper and lower R5 pins in sequence and the difference is calculated.

$$N_{v1} = N_{fs} \frac{R_5 + R_6}{R_5 + R_6 + R_4} \frac{V_l}{V_{ref}} + N_{fs} \frac{V_{off1}}{V_{ref}};$$

$$N_{v2} = N_{fs} \frac{R_6}{R_5 + R_6 + R_4} \frac{V_l}{V_{ref}} + N_{fs} \frac{V_{off1}}{V_{ref}}; \quad (1)$$

$$\Delta N_v = N_{v1} - N_{v2} = N_{fs} \frac{R_5}{R_5 + R_6 + R_4} \frac{V_l}{V_{ref}}.$$

- o  $V_l$  and  $V_{ref}$  are load and reference voltages
- o  $V_{off1}$  is ADC/PGA offset voltage
- o  $N_{fs}$  is full-scale ADC code range
- o  $\Delta N_v$  is ADC code difference, that is proportional to the measured voltage without the influence of offset voltage

Current measurement is implemented by measuring the voltage drop at R3 (Figure 1), using the modified LPF as a level translator and gain stage. The voltage levels for the right and left R3 pins are measured and the difference is calculated to eliminate the influence of the LPF/ADC/supply-pin offset on measurement accuracy.

$$N_{c1} = N_{fs} \frac{-I \cdot R_3 \cdot G_{LPF} + V_{off2}}{V_{ref}};$$

$$N_{c2} = N_{fs} \frac{V_{off2}}{V_{ref}}; \quad (2)$$

$$\Delta N_c = N_{c2} - N_{c1} = N_{fs} \frac{I \cdot R_3 \cdot G_{LPF}}{V_{ref}}.$$

- $G_{LPF}$  is LPF gain.
- $\Delta N_c$  is ADC code difference, which is proportional to the R3 current.
- $V_{off2}$  is cumulative offset voltage, which consists of the following components:
  - ADC offset - 10 mV maximum.
  - LPF offset -  $10 \text{ mV} \cdot G_{LPF}$ , typically.
  - Vss pin supply offset, which is caused by voltage drop on internal die resistance and is proportional to PSoC device supply current via Vss pin. Typical values are less than 20 mV.

Temperature measurement is implemented by measuring the voltage drop on R7 (Figure 9) and applying, in sequence, two different voltages to the upper R7 pin and subtracting the results. The signal from the thermistor comes to the ADC via the PGA with unity gain. Bias voltages are formed using the continuous time TestMux to apply RefHi (2.6V) level at the first step and AGND (1.3V) at the next. This technique compensates for both the ADC/PGA offset and the possible offset caused by potential differences between the Vss internal die connection and the lower thermistor pin. (The sources of potential differences between the Vss internal die connection and the lower thermistor pin are the device's internal die resistance, PCB voltage drop, battery pack voltage drop upon negative lead during charge, and so on.) Figure 9 and the following formulas illustrate how this works.

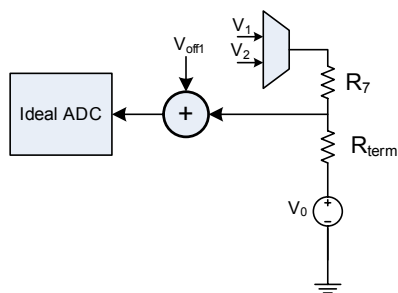


Figure 9. The Temperature Measurement

$$N_{t1} = \frac{N_{fs}}{V_{ref}} \left[ V_0 + V_{off1} + \frac{R_{term}}{R_7 + R_{term}} (V_1 - V_0) \right];$$

$$N_{t2} = \frac{N_{fs}}{V_{ref}} \left[ V_0 + V_{off1} + \frac{R_{term}}{R_7 + R_{term}} (V_2 - V_0) \right]; \quad (3)$$

$$\Delta N_t = N_2 - N_1 = \frac{N_{fs}}{V_{ref}} (V_2 - V_1) \frac{R_{term}}{R_7 + R_{term}}.$$

- $N_{t1}$  and  $N_{t2}$  are ADC codes for resistive divider bias voltages  $V_1$  and  $V_2$  (their difference is  $V_{bandgap}$  or 1.3V)
- $V_{off1}$  and  $V_0$  are offset voltages ( $V_{off1}$  is the same as in Equation (1))
- $\Delta N_t$  is ADC code difference

The thermistor transfer function is non-linear, but we do not need to obtain the temperature value in linear units for this design because the ratio of temperature increases is not used as rapid-charge termination criterion. Therefore, only the temperature thresholds during the charge need to be checked. This is done by analyzing the ADC code difference ( $\Delta N_t$ ). A hysteresis is added for the bottom and upper bounds of the in/out temperature range to prevent multiple triggers when the temperature is close to the preset range (see Figure 10).

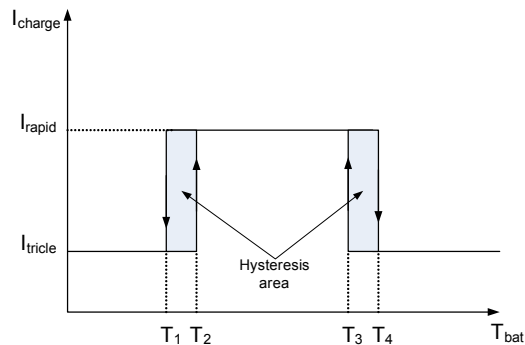


Figure 10. Temperature In-Range Checking

The device implements the rapid charge algorithm for NiCd/NiMH batteries. The batteries are charged by a constant current in rapid charge mode with 1 C rate (the C rate is the hour capacity of the battery at 600 mA) when the battery temperature is in the allowed range. When the battery is too hot or cold, the controller reduces the charge current using trickle charge mode. Note that the battery temperature is measured once per 4s using an interval timer timeout.

The charge is normally terminated when the battery voltage stops increasing; this termination criterion is suitable for either NiCd or NiMH batteries. The integration criterion is used for battery voltage slope analysis to get the best noise immunity. During the first step, the measured battery voltage is filtered using a simple IIR LPF filter.

$$y_j = (1 - \gamma)y_{j-1} + v_j \quad (4)$$

- $y_j$  is filter output signal j sample
- $v_j$  is input voltage sample
- $\gamma$ ,  $\gamma < 1$  is filter coefficient

This filter is characterized by the gain  $\gamma^{-1}$ . Implementation minimizes the influence of fixed-point rounding errors and improves the accuracy of small voltage variation detection. Instead of dividing/multiplying the binary shift,  $\gamma = 2^{-K}$  was used, where  $K=2$ . Each voltage measurement result passes through this filter.

Battery voltage slope estimation is implemented using an integral sum calculation. This technique is used in Application Note AN2107 “A Multi-Chemistry Battery Charger” and has been adapted for this application. It calculates the rate of increase of battery temperature. (Please refer to AN2107 for math details.) The idea is to calculate two integral sums from the filtered battery voltage samples and compare them to each other. When the new integral sum is smaller than the previous sum, the zero/negative voltage slope is considered achieved and rapid charge is terminated. Each integral sum consists of the 16 samples acquired with sample intervals of 4 s. Therefore, the battery slope is evaluated once every 64 s. Figure 11 graphically illustrates this method. This method provides better noise immunity because it is less dependent on a single sample measurement result.

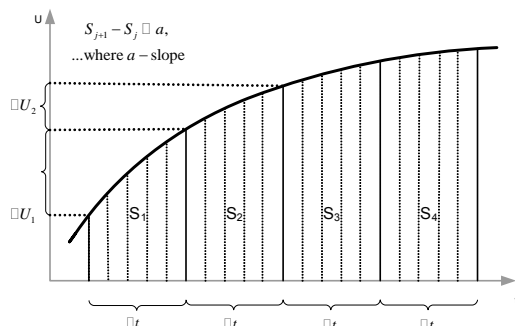


Figure 11. Integral Slope Estimation Method

Note that the battery scope voltage check is locked for five minutes after starting the rapid charge, if the battery was not used for an extended period of time. The battery voltage can temporarily drop after the charge starts and can be prematurely terminated.

There are two backup termination checks. The total rapid charge time is limited by a timer. This is to prevent overcharging when the slope voltage check, for some reason, does not trigger. Also, the rapid charge is instantly terminated when battery voltage reaches the predefined maximum value. With normal use, backup checks should not trigger unless there is a problem with the battery. The user can add bi-color LEDs to display this occurrence. The charge algorithm is shown in Figure 12.

Note that when the charge is finished, the controller jumps to the **Ext Power** state. The regulator provides low-battery charge current to compensate for any leakage caused by an activated LED and the PSoC device in active state. Sleep mode can be activated to assist.

The device employs the PFM duty cycle regulator to maintain the charge current or motor voltage, depending on the controller state (**Bat Charge** or **Ext Motor**). The regulator is an adaptive step integrator. Figure 13 shows PFM duty cycle regulator dataflow.

The regulator analyzes the difference between the preset and the actual values (current or voltage) and tries to minimize this difference. Also, the limiting flag is analyzed to prevent outranging for an auxiliary parameter. For example, the regulator allows reduced power on the motor by limiting the windings' current to safe values.

There is runtime debug support in the controller firmware that can be transmitted via serial port to the PC. Runtime debug support can monitor the state of the battery or motor voltage, the current, the temperature, the state of the machine and the error variable. The supplemental Power On state with unconditional branch to **Bat Power** is used to transmit the timestamp-reset command to the PC software when the device is powered.

This state is not covered in Table 2 or Figure 7 because it is only used for debug purposes. The user can disable the transfer of debug information by setting a conditional compilation directive `DEBUG` to 0 in the controller firmware `globdefs.h` file.

The debug information is sent via serial port using the PSoC device's P0[11] pin. An external EIA-232 (RS-232) level translator is required (such as MAX3221E).

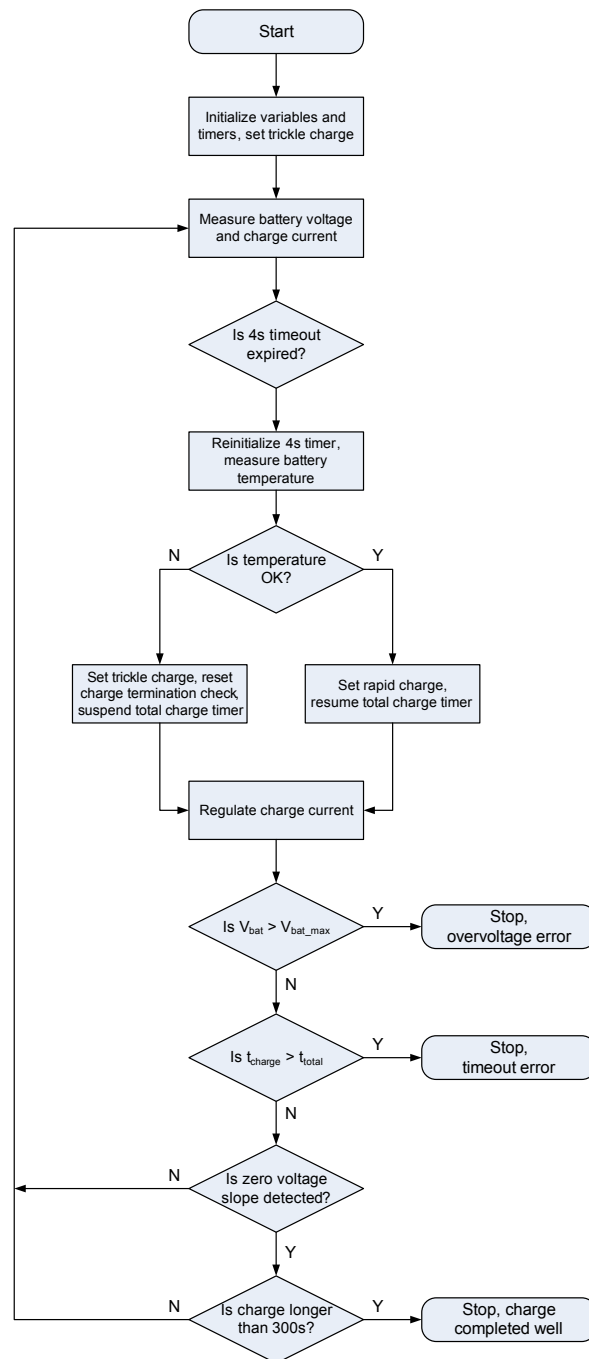


Figure 12. Battery Charge Algorithm

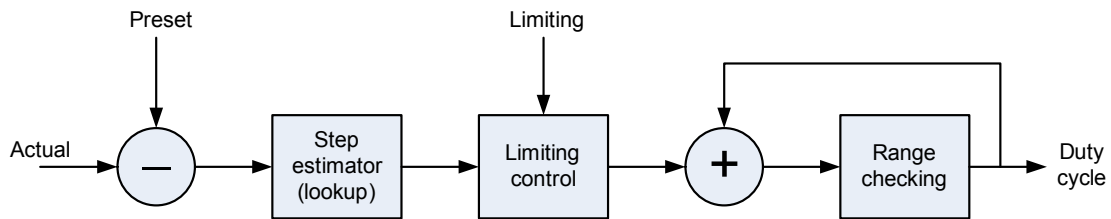


Figure 13. Duty Cycle Regulator

The PC software that monitors the system is a multithread Windows application. It is simplified in Application Note AN2107 “A Multi-Chemistry Battery Charger” and allows users to view graphs displaying battery or motor supply voltage, charge/motor current, and temperature.

State and error conditions can also be viewed. The software allows users to store and compare collected data for further analysis. Figure 14 shows examples of the graphs. The software was written using Borland® Delphi™ 7.



Figure 14. Example of Software Interface

Note: Battery was Slightly Discharged Before Charging Began

## Conclusion

This Application Note describes a simple, low-cost battery charger with a DC motor controller. The hardware and firmware can be easily adapted to meet end application demands. The following modifications can be implemented:

- Charge algorithm can be adapted to support Lilon batteries.
- Current sense resistor can be eliminated. Battery charge current can be found mathematically by measuring the switch regulator input, the output voltages and using the switch regulator look-up table. NiCd/NiMH batteries are not very sensitive to the absolute value of the charge current, and 10-15 percent can be tolerated without problems.

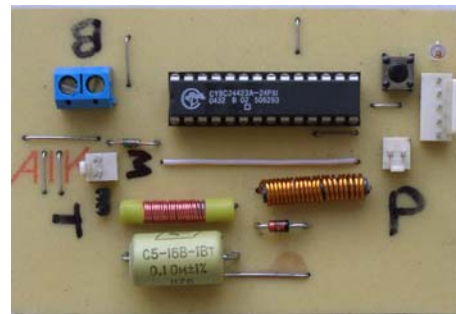


Figure 15. Device Photograph

---

## About the Author

**Name:** Victor Kremin

**Title:** Associate Professor

**Background:** Victor earned radiophysics diploma in 1996 from Ivan Franko National Lviv University, PhD degree in Computer Aided Design Systems in 2000, and is presently working as Associate Professor at National University "Lvivska Polytechnika" (Lviv, Ukraine). His interests involve the full cycle of embedded system design including various processors, operation systems, and target applications.

**Contact:** [vkremin@cypressmicro.com](mailto:vkremin@cypressmicro.com)

---

Cypress Semiconductor  
2700 162<sup>nd</sup> Street SW, Building D  
Lynnwood, WA 98037  
Phone: 800.669.0557  
Fax: 425.787.4641

<http://www.cypress.com/>

Copyright © 2005 Cypress Semiconductor Corporation. All rights reserved.  
PSoC™, Programmable System-on-Chip™, and PSoC Designer™ are PSoC-related trademarks of Cypress.  
All other trademarks or registered trademarks referenced herein are the property of their respective owners.  
The information contained herein is subject to change without notice. Made in the U.S.A.