Application Note

AN2679/D
6/2004

Smart NiCd/NiMH Battery Charger Using MC68HC908QY4

By: Andre Vilas Boas
Marcus Espindola
Alfredo Olmos
Brazilian Semiconductor Technology Center – BSTC/SPS

Introduction

Many battery chargers do not provide efficient and reliable charging cycles. This means the charger may leave the battery improperly charged, which can reduce the battery's life and possibly damage the battery.

This application note presents an MC68HC908QY4 microcontroller-based charger for nickel-cadmium (NiCd) and nickel-metal-hydride (NiMH) batteries and battery packs. This "smart" charger is suitable for automatically charging a wide range of batteries with different capacities. It is designed to satisfy the demands of high current and fast charge applications such as cordless power tools and toys.

Features

- Flexibility to handle both NiCd and NiMH batteries with a broad range of capacities
- Intelligent charging algorithm based on MC68HC908QY4 MCU
- Sense circuits for charging voltage and temperature
- Reliable protection against overcharging
- Pre-discharging not required, battery is always charged to 100% of available capacity
- Two charging modes:
  - Fast charge (charge period for 100% charge)
  - Trickle charge (supplementary charge at the end of a fast charge cycle)
- Automatic switch from fast to trickle charge mode

NOTE: *With the exception of mask set errata documents, if any other Freescale document contains information that conflicts with the information in the device data sheet, the data sheet should be considered to have the most current and correct data.*

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

NiCd/NiMH Overview

Nickel-cadmium (NiCd) batteries may be recharged many times and have a relatively constant potential during discharge. They withstand more electrical and physical abuse than any other battery pack, have good low-temperature performance characteristics, and are inexpensive in terms of cost per hour of use. When used within their recommended ratings and in applications where the use of rechargeable battery packs is necessary, NiCd batteries will provide economical and trouble-free service.

The nickel-metal-hydride (NiMH) battery pack is used in high-end portable electronic products such as cellular phones and portable computers. NiMH battery packs are similar to sealed NiCd cell technology, but they use a hydrogen-absorbing electrode instead of cadmium-based negative electrode. Using the hydrogen-absorbing electrode increases the battery pack's electrical capacity (measured in ampere-hours) for a given weight and volume and avoids the toxicity concerns of cadmium.

Switching between the two battery pack types usually requires only slight changes to application parameters and few significant design issues. Table 1 compares key features between the two battery pack chemistries.

Table 1. Comparison of NiCd and NiMH Batteries

Main Features	Nickel-Metal Hydride vs. Nickel-Cadmium Batteries
Nominal voltage	Both 1.25 V
Discharge capacity	NiMH up to 40% greater than NiCd
Cycle life	Generally similar, but NiMH is more application dependent
Mechanical fit	Equivalent
High temperature (>35°C) discharge capability	NiMH slightly higher than standard NiCd battery packs
Discharge cutoff voltages	Equivalent
Charging process	Generally similar; multiple-step constant current with overcharge control recommended for fast charging NiMH
Charge termination techniques	Generally similar, but NiMH transitions are more subtle. Backup temperature termination recommended for both battery types.
Self-discharge	NiMH slightly higher than NiCd
Environmental concern	Less with NiMH due to toxicity concerns of cadmium

Battery Charging Systems

The operating life of rechargeable batteries is determined mainly by their rate of discharge, depth of discharge, operating and ambient temperature, and charging method. The equipment designer determines the first three features, but if an incorrect charging technique is used, the good design benefits are lost and the battery life and performance can be significantly degraded.

To obtain optimum performance and a longer life from any rechargeable battery, it is important to charge it correctly. NiCd chargers may vary greatly in level of sophistication—from the simplest unregulated dc output transformer to a fast intelligent charging system. In the first case, the charging current is limited only by the transformer's impedance and/or by series resistance. Fast charging requires intelligent monitoring of battery parameters at all stages of the charging cycle. Rapid chargers optimized for nickel-based battery packs are much more complex and typically consist of a current regulator, a voltage limiter, and a charge control. Charge control measures charging time, changes in battery pack temperature and/or voltage, and regulates or terminates charge current accordingly.

NiCd

NiCd batteries are typically charged with constant current. Most of them can be safely charged at rates up to C/3 (where C is the charge potential) without electronic control, but electronic control helps ensure reliability and efficiency. Charging at high rates up to 2C requires electronic monitoring of battery parameters to detect when the charge cycle is complete.

Charging/Termination Methods

- **Standard Charging (Overnight)** — Charging at rates C/10 and lower takes approximately 15 hours to fully charge the cell or battery. A limited amount of overcharging is acceptable, so it is not necessary to have an accurate end-of-charge detector. However, prolonged overcharging can damage the battery packs. The limitation of this method is the slow recharging time.
- **Controlled Charging Time** — The charging current is terminated at a specific time. This requires knowing the initial amount of charge in the battery, which is simple if the battery is discharged completely. The battery capacity must be known and set by the user. For this method, the battery pack capacity must be specified, but the capacity value is difficult to specify because it changes with age and other conditions. This method is commonly used as a fail-safe method for terminating any charging algorithm. If the charging algorithm does not complete within the predefined amount of time, the charge will terminate.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

- **ΔT/Δt Temperature Detection** — When the battery reaches full charge, the battery pack will experience a quick rise in temperature. This is due to an increase in the conversion of charging energy into thermal energy. The ΔT/Δt method uses a sensor to measure the battery temperature, and the MCU calculates the temperature rise rate with respect to time. The MCU will terminate the charge if the measured ΔT/Δt rate meets or exceeds the stored ΔT/Δt rate threshold. This method can be adversely affected by ambient temperature and may result in under charge conditions when charged in high ambient temperature environments, or overcharge in low ambient temperature environments.
- **Controlled Charge Voltage** — When the full charge point is reached, the battery does not accept charging current. Instead, it starts to turn the current into heat (see Figure 1). As soon as the battery pack temperature increases, the charging voltage stops rising, stabilizes at a certain level, and finally starts to decline at the onset of overcharging. This method uses the voltage drop to determine when to stop charging. The controlled charge voltage is useful when charging rates are greater than C/2, otherwise the voltage variation is too small to be detected.
- **Trickle Charge** — A very small amount of current is applied to the battery. This technique is used when a battery is continuously connected to the charger or as a supplementary charge at the end of a fast charge cycle to replace charge loss due to self-discharge. The recommended rate for trickle charging in most NiCd battery packs is between C/20 and C/200.

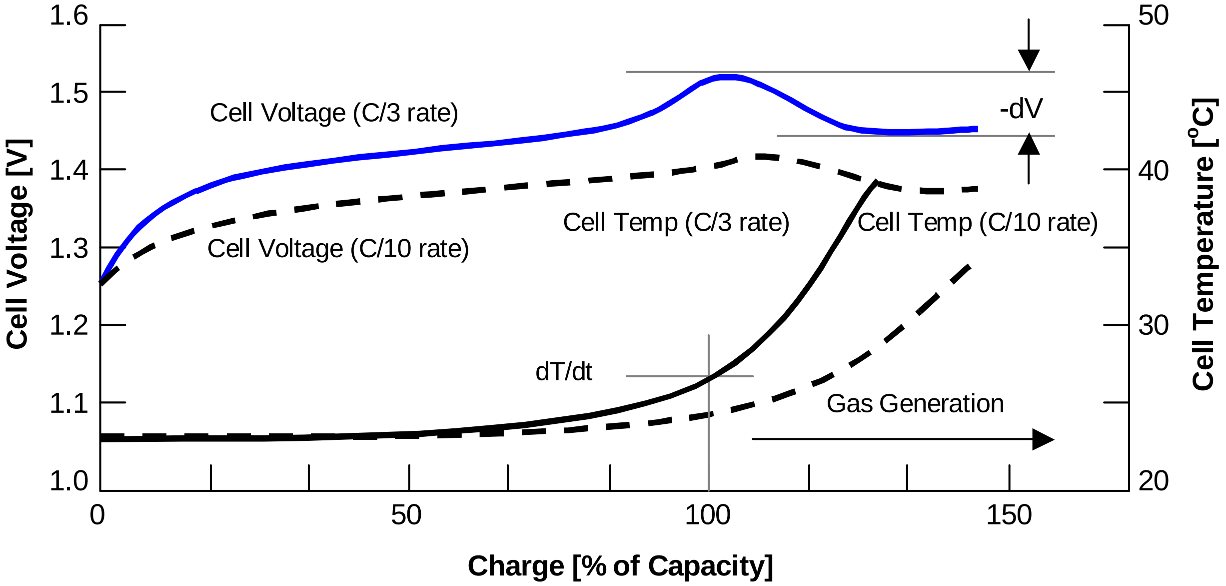


Figure 1. Typical NiCd Battery Pack Voltage and Temperature During Charge

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

NiMH

NiMH batteries are charged using techniques similar to NiCd batteries. However, NiMH requires more monitoring due to its greater sensitivity to overcharging. A NiMH battery is often charged with a constant current with the current limited to approximately C/2 rate to avoid excessive temperature rise. The charging characteristics of NiCd and NiMH battery packs are similar, but NiMH generates more heat during charge and peak voltage is less noticeable.

Charging Methods

- **Controlled Charging Time** — This technique is the same as for NiCd type battery packs and is typically used only as a way to complete the charge after using some other charge technique.
- **Absolute Temperature Detection** — This method uses a sensor to detect when the battery pack temperature reaches an absolute specified value. At that time, the charge is terminated. This method can be adversely affected by ambient temperature and may result in under charge conditions when charged in high ambient temperature environments, or overcharge in low ambient temperature environments.
- **ΔT/Δt Temperature Detection** — This is the preferred method of detecting end of charge for NiMH because it provides a long cycle life for the battery. When the battery reaches full charge, the battery pack will experience a quick rise in temperature. This is due to an increase in the conversion of charging energy into thermal energy. The $\Delta T/\Delta t$ method uses a sensor to measure the battery temperature, and the MCU calculates the temperature rise rate with respect to time. The MCU will terminate the charge if the measured $\Delta T/\Delta t$ rate meets or exceeds the stored $\Delta T/\Delta t$ rate threshold. This method can be adversely affected by ambient temperature and may result in under charge conditions when charged in high ambient temperature environments, or overcharge in low ambient temperature environments.
- **Controlled Charge Voltage** — Although this method is often used for NiCd batteries, it may not be effective for NiMH types. With NiMH, the voltage peak is not as noticeable for low charging rates and may not occur at all, especially at higher temperatures (see Figure 2). The voltage monitor circuit must have a resolution of a few millivolts to determine the end of charge. If the monitor circuit is too sensitive, noise and other conditions may cause an early end of charge. Also, the voltage curve as a function of charge condition varies between battery packs—even if they are the same age and type.
- **Voltage Flat** — This technique is similar to the controlled charge voltage method except that the voltage flat circuitry detects when the slope of the battery voltage curve (during the charge process) becomes zero ($\Delta V/\Delta t = 0$). Consequently, the risk of battery overcharge is small and trickle charge can be applied to complete a full charge operation.

Freescale Semiconductor, Inc.

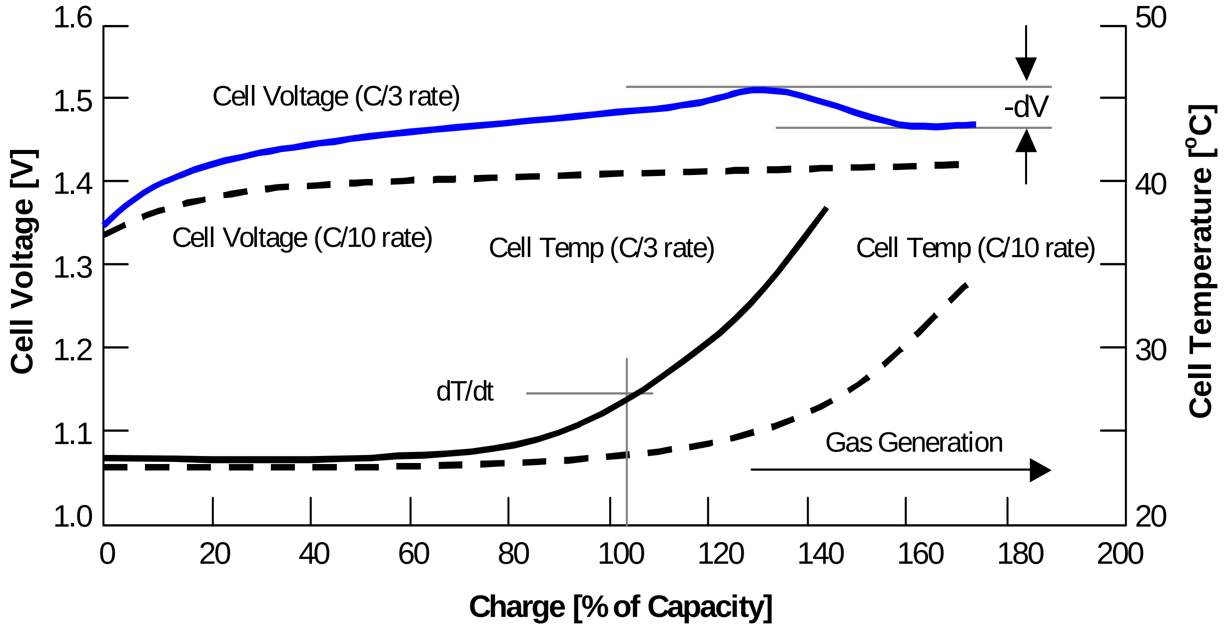


Figure 2. Typical NiMH Battery Pack Voltage and Temperature During Charge

Fast Charge

Various techniques are used to perform fast charging rates for both NiCd and NiMH battery pack types. These methods require a constant charge current that is typically greater than C/3 rate to induce significant rises in battery pack temperature or changes in battery pack voltage, which are used to indicate when the battery pack is fully charged.

Temperature

The exact recommended temperature range for charging varies among battery pack manufacturers. Typical ranges are summarized in Table 2. Fast charge rates may be applied between +10 to +40 degrees Celsius. Outside these limits, current must be reduced.

Table 2. Typical Temperature Range Recommended for NiCd/NiMH Battery Charge

Charge Rate	Typical Recommended Range
Fast charge (C to C/2)	+10 to +45°C
Standard charge (C/10)	0 to +45°C
Trickle charge (C/20 to C/200)	+10 to +35°C

For More Information On This Product,
Go to: www.freescale.com

Intelligent Battery Charger Design

The MC68HC908QT/QY Family is a member of the low-cost, high-performance M68HC08 Family of 8-bit FLASH MCUs. The M68HC08 Family is a complex instruction set computer (CISC) with a Von Neumann architecture. All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

The MC68HC908QT/QY Family allows designers to incorporate the benefits of FLASH technology into designs, which makes it possible to reduce overall system cost and speed time-to-market. FLASH-based systems offer ultra-fast programming with maximum flexibility and creativity. With FLASH, a design can be reprogrammed many times during the development cycle or even late in manufacturing. Upgrades can be made even in the field.

See the Freescale website, http://freescale.com, for complete details on the MC68HC908QT/QY Family.

MC68HC908QY/QT Features and Benefits

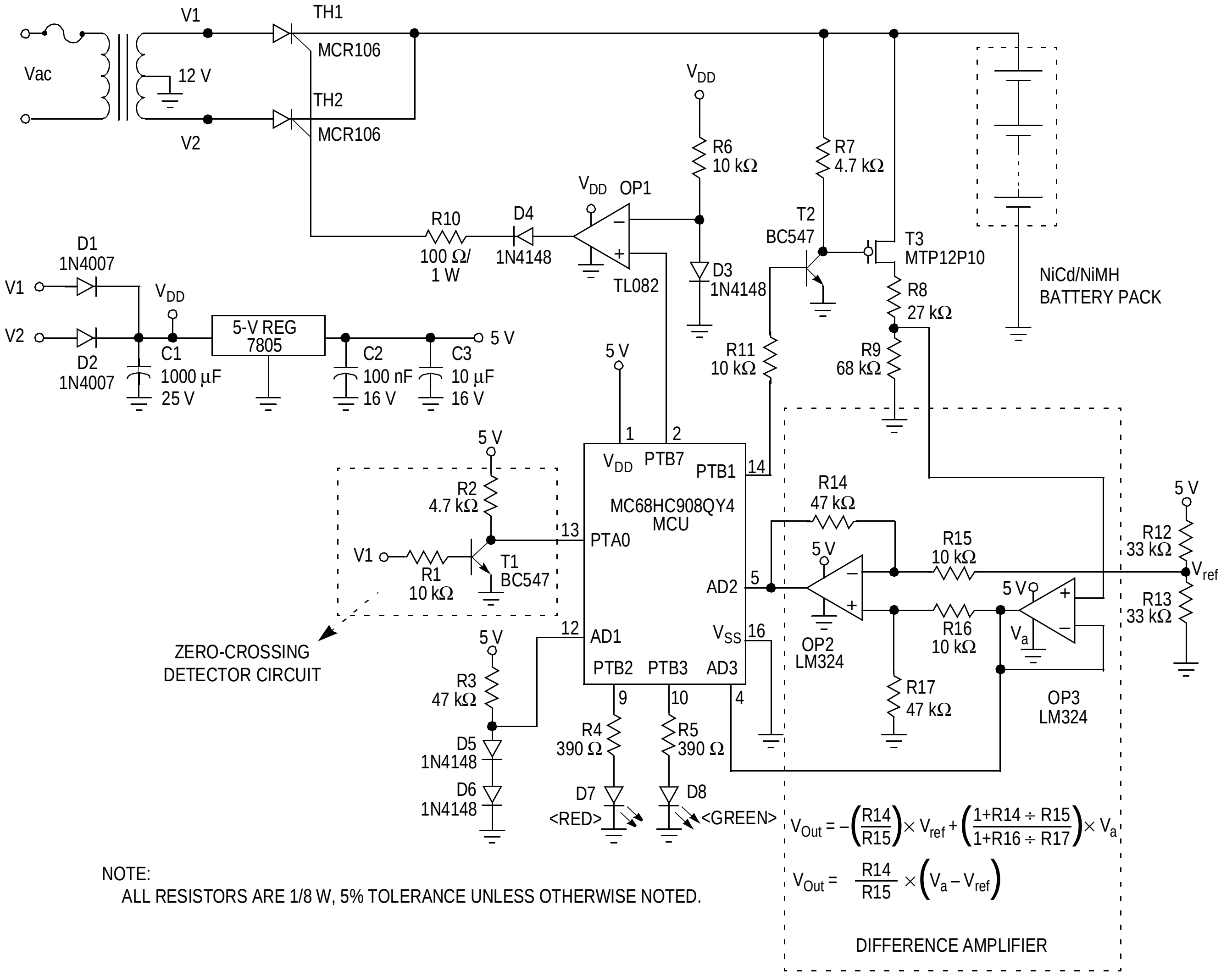
- High-performance 8-bit HC08 CPU
  - Fully upward-compatible object code with Motorola’s M68HC05 Family for easy migration
  - Enables the higher performance required of many 8-bit applications — as fast as 125 ns minimum instruction cycle time
  - Designed to allow efficient, compact modular coding in assembly or C with full 16-bit stack pointer and stack relative addressing
  - Efficient instruction set with multiply and divide that is easy to learn and use
- Memory
  - In-application, in-circuit reprogrammable FLASH memory (1.5K to 4K bytes)
  - 128 bytes of random access memory (RAM)
- Peripherals
  - Two-channel, 16-bit timer with selectable input capture, output compare, or PWM (pulse-width modulator)
  - Trimmable 5% accuracy internal oscillator
  - 4-channel 8-bit analog-to-digital converter (ADC) (on the MC68HC908QT2/QT4/QY2/QY4) – provides an easy interface to analog inputs such as sensors
  - Flexible I/Os allow direct drive of LED and other circuits without external drivers and help reduce system cost
  - System protection features, including watchdog timer and on-chip low-voltage detect/reset, to help reduce cost and increase reliability
  - Small packages — a variety of 8- and 16-pin packages, with more to come as the family develops

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

Application Description

Figure 3 is the smart NiCd/NiMH battery charger circuit diagram. The system combines the voltage flat and the absolute temperature detection techniques followed by the trickle charge method to charge a NiCd/NiMH battery pack with three cells.



$$V_{Out} = -\left(\frac{R14}{R15}\right)\times V_{ref} + \left(\frac{1+R14 \div R15}{1+R16 \div R17}\right)\times V_a$$

$$V_{Out} = \frac{R14}{R15}\times\left(V_a - V_{ref}\right)$$

NOTE:
ALL RESISTORS ARE 1/8 W, 5% TOLERANCE UNLESS OTHERWISE NOTED.

Figure 3. Battery Charger Circuit Diagram

Freescale Semiconductor, Inc.

Initialization

The battery charge procedure begins by determining whether a battery pack is available for charging by measuring for voltage.

1. MC68HC908QY4 MCU port PTB1 turns on the bipolar transistor T2.
2. The OP3 buffer allows the MCU ADC to read the battery pack voltage with the AD3 channel. An ADC reading greater than 500 mV indicates that a battery pack is available for charging. Discharging the battery set is not required before starting the charging process.

Controlled Rectifier

The MC68HC908QY4 MCU triggers the controlled rectifier, composed of two thyristors connected as a full-wave bridge rectifier, to provide the charge current for the battery pack.

3. The MCU activates the controlled rectifier after the zero-crossing of the ac sinusoidal signal within a user-defined time duration. The system uses the zero-crossing detector circuit to start the MCU timer counter by sensing any level transition in PTA0 and waits to generate the rectifier trigger signal via PTB7. For a 60 Hz ac line frequency, the time duration is set by default to approximately 4 ms to trigger the thyristors at the sinusoidal peak and provide 600 mA full charge current to the battery pack, as illustrated in Figure 4.

Charge Period

4. At the end of the user-defined charge period, the MCU pauses to turn on the controlled rectifier and measure the voltage variation of the battery packs. If the MC68HC908QY4 MCU is operating with the internal oscillator and it has been trimmed to obtain a 3.2-MHz bus clock frequency, the charge period is set by default to approximately 10 minutes.
5. To measure the battery voltage, the MCU asserts again the PTB1 pin to turn on the bipolar transistor T2 and apply the battery voltage to the resistor R9.

Freescale Semiconductor, Inc.

Overcharge Protection

The battery voltage variation measurement is intended to detect the decline of the battery pack voltage and the onset of the overcharging process.

6. The system detects when the slope of the battery voltage curve (during the charge procedure) becomes lower than zero ($\Delta V/\Delta t < 0$, see Figure 1 and Figure 2).
7. The system subtracts the reference voltage (V_{ref}) from the voltage on the battery packs using the difference amplifier (as shown in Figure 3). The MC68HC908QY4 MCU converts that data to a digital word using its internal ADC and the AD2 channel.
8. After the analog-to-digital conversion is complete, the MCU stores the subtraction result in an internal MCU variable. The initial value of this internal MCU variable is set by default to $00 and, assuming a difference amplifier gain of 4.7, the reference voltage for a battery pack of three cells is V_{ref} = 2.50 V.
    a. If the battery set voltage is lower than (or equal to) the reference voltage, the ADC reading is always larger than (or equal to) the value stored in the MCU variable at the end of the previous charge period. Therefore, the system continues the battery charge process during a subsequent charge period.
    b. If the battery set voltage begins to decline, the ADC reading will be lower than the previous value and the system stops the normal charge process and starts the trickle charge procedure. During trickle charge, current flowing into the batteries is reduced to 28 mA by adjusting the MCU timer counter to obtain a delay of 7 ms with regard to the zero-crossing of the sinusoidal ac signal, as shown in Figure 5. While in trickle charge, the system does not monitor the battery pack voltage.

Temperature Protection

9. To provide absolute temperature protection, the voltage drop across diodes D5 and D6 (located very near the battery pack) is measured to check for a user-defined battery temperature variation using the AD1 channel. Like the battery voltage variation measurement, the D5 and D6 voltage check is performed after a charge period. The initial D5 and D6 voltage drop is measured at the beginning of the normal charge process. For example, if the temperature increases by approximately 30^{o}C, the D5 and D6 voltage drop change would be equivalent to DV @ –125 mV (or roughly seven LSBs of the MC68HC908QY4 MCU 8-bit ADC). If the system detects a voltage-related temperature variation larger than the maximum allowed by user, it starts the trickle charge procedure.

Freescale Semiconductor, Inc.

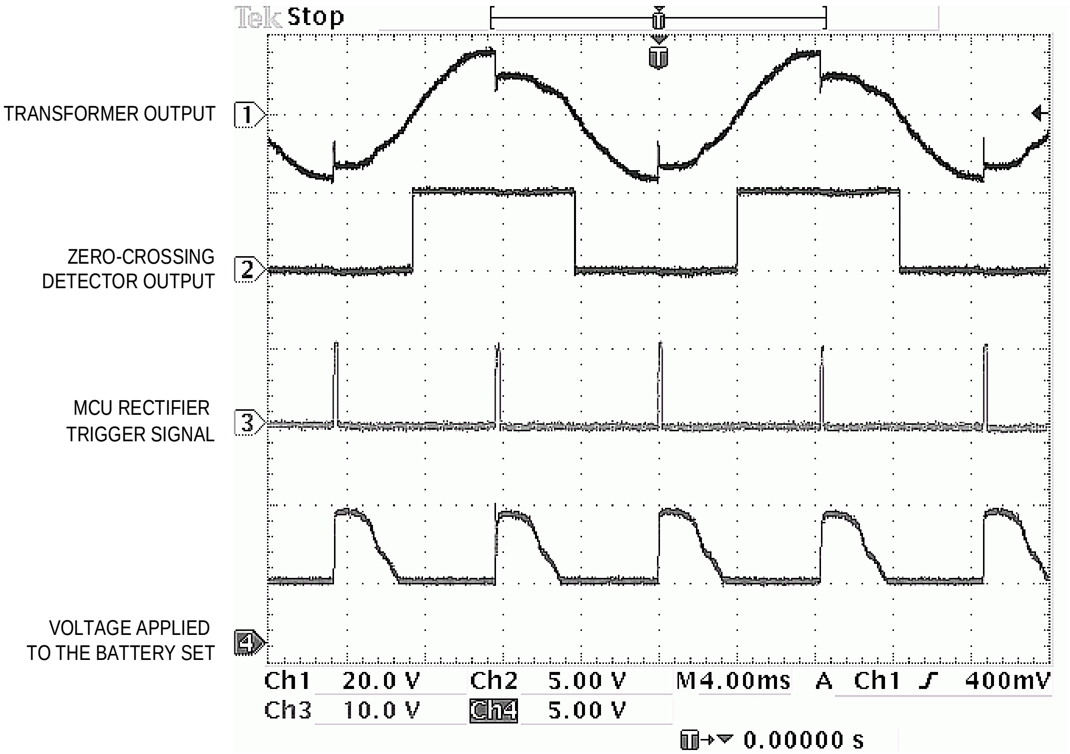


Figure 4. Normal NiCd/NiMH Battery Charge Waveforms

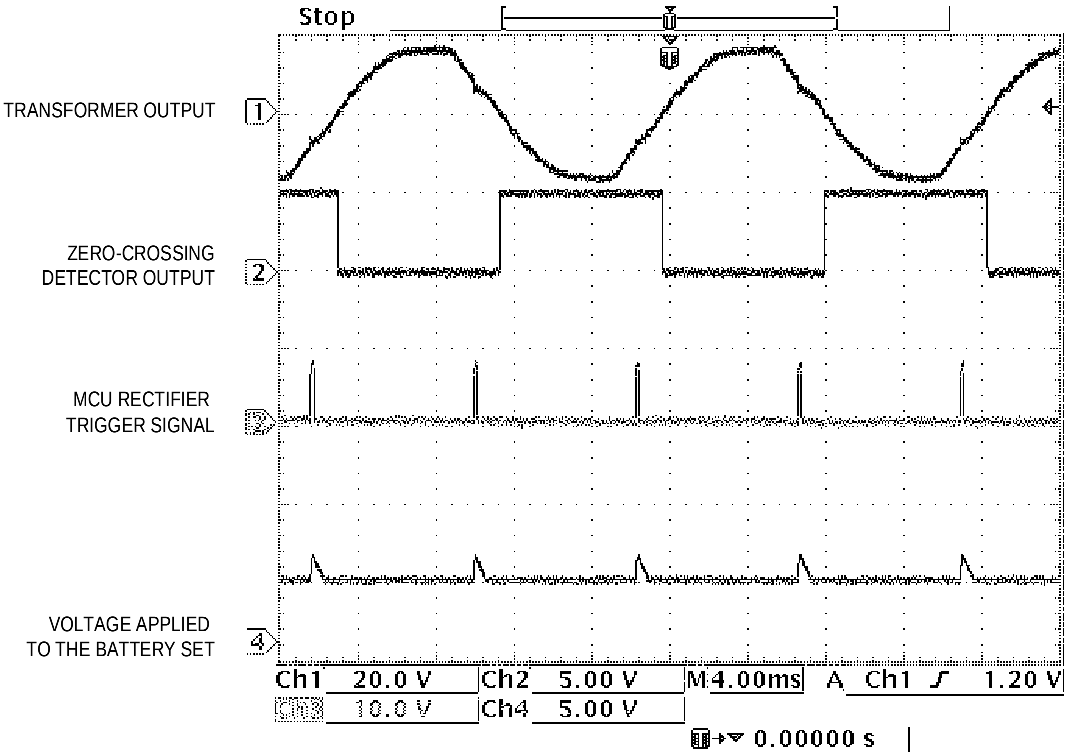


Figure 5. Trickle Charge Waveforms

Software Description

The following process is illustrated in Figure 6.

1. Initialization

The software starts configuring the I/O ports and registers properly, clearing variables and setting the timer to generate the appropriate thyristor conduction angle. In this application, timer registers are defined to overflow at about 4 ms. Constants *InitTMODH* and *InitTMODL* modify the values of the TMODH and TMODL registers on the timer module.

2. Monitoring Loop

After performing the initialization operations, the code enters a loop where the battery voltage is continually monitored through ADC channel 3 to determine whether a battery pack is connected. Constant *BatInit* sets the minimum value to check whether the battery pack is engaged.

As soon as a battery pack is connected to the charger, the ADC detects a voltage value larger than the one previously stored in *BatInit* and the charging process begins.

3. Begin Charging

Initially, the system measures the temperature of the battery packs. ADC channel 1 reads the voltage across the diodes D5 and D6. The value is converted and stored into the *FrstTmpRd* variable. This is the first value for temperature comparison. The red LED is turned on by setting PTB3, which indicates the beginning of the charging cycle.

4. Counter Loop

A loop is implemented and PTA0 waits for a level transition coming from the zero-crossing detector circuitry. After the transition is detected, the timer module is started and stays in a loop until it overflows. Then timer module is stopped and cleared and PTB7 triggers the thyristor.

5. Pause Charging

Counter variables are incremented. After the ten minute charging period (which can be changed by modifying the *StpChL* and *StpChH* constants), the charging process is paused. Counters are cleared and battery voltage is monitored again by setting PTB1 and reading ADC channel 2. A delay is needed to stabilize the voltage on the battery. The new ADR register value is subtracted from the last stored value.

- If the new value is lower than the previous value, the charging process is stopped and the trickle charging subroutine is defined.
- If the new voltage value is greater-than or equal-to the previous value, the charging cycle continues. At this time, the new voltage acquired by the ADC is stored in a variable to be compared with the next value that will be captured at the end of the next charge period.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Current temperature is captured (by reading the voltage over the diodes D5 and D6 by ADC channel 1) and compared with the first value. If difference in temperatures is greater than a predefined value on the *TmpSafe* constant, the routine goes to trickle charging. If the temperature is less than this value, the charging process continues. This is done to protect against battery overheat.

6. Trickle Charging

When trickle charging, PTB2 turns on the green LED and PTB3 turns off the red LED, which indicates the battery is fully charged. The timer module registers are changed to increment the overflow period. In this case, the timer overflows at about 7 ms after PTA0 detects the level transition. PTB7 pulses at the end of the power cycle line, reducing the current that charges the battery pack. It stays in a loop until the user disconnects the battery pack from the charger.

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

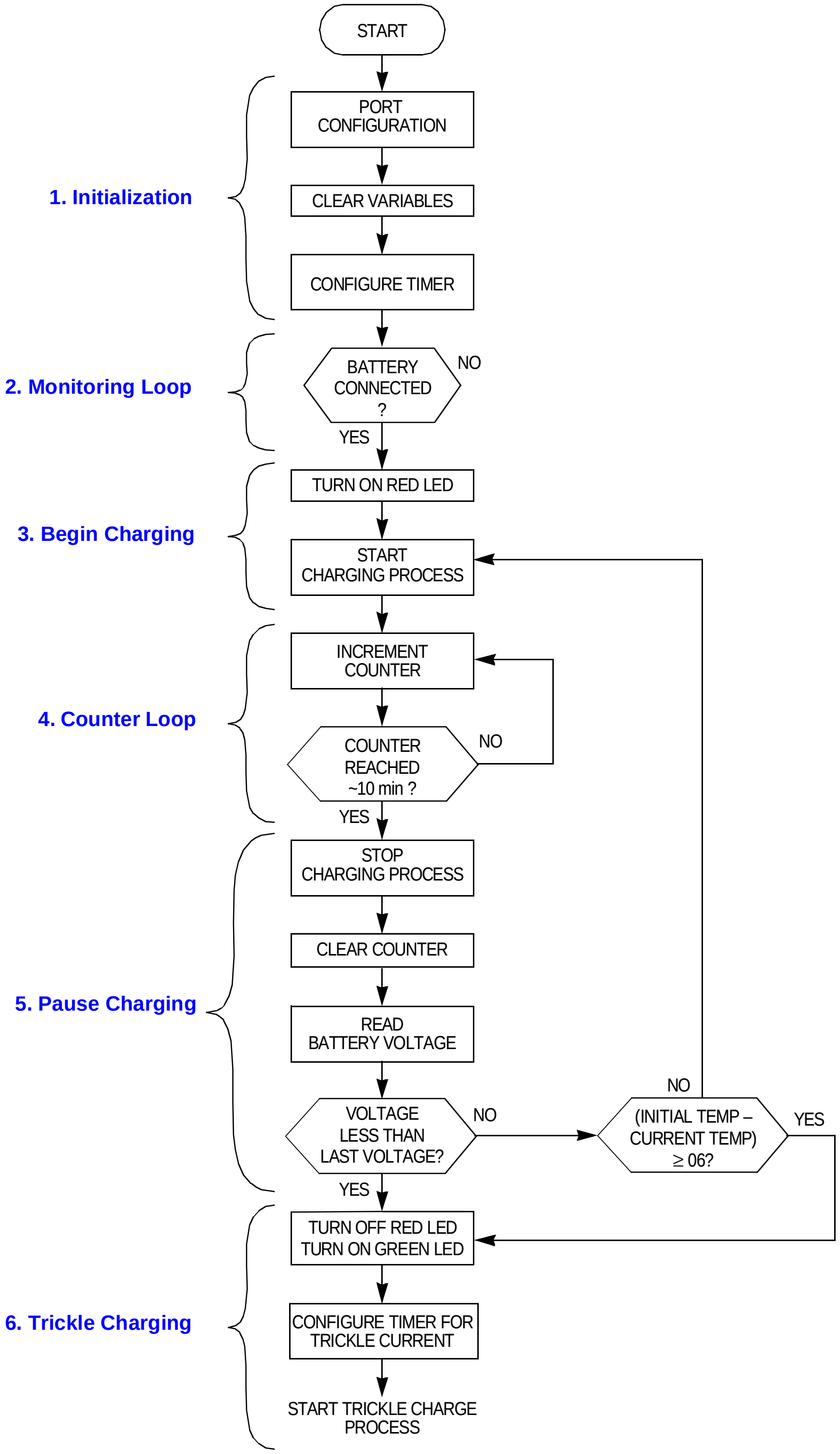


Figure 6. The NiCd/NiMH Battery Charger Software

Software Listing

```
;*******************************************************************************
;* Title: battery.asm                                           Copyright (c) 2003
;*******************************************************************************
;* Author: Marcus Espindola - Freescale  SPS/BSTC
;*
;* Description: Intelligent Battery Charger for QY family.
;*
;* Documentation: HC908QY4 Data Sheet (MC68HC908QY4/D) for register and bit explanations
;*
;* Include Files: battery.equ, MC68HC908QT4.equ
;*
;* Assembler: P&E Microcomputer Systems - CASM for HC08
;*            Metrowerks CodeWarrior Compiler for HC08 V-5.0.17
;*
;* Revision History:
;* Rev #      Date      Who         Comments
;* -----    -----------  ---------   -------------------------------------------------------
;* 0.3      04-Nov-03   Espindola   Placed constants into include file
;* 0.2      11-Sep-03   Espindola   Included timing before rd battery and transistor control
;* 0.1      12-Aug-03   Espindola   Initial data entry
;*******************************************************************************
;*******************************************************************************
;* Freescale reserves the right to make changes without further notice to any product
;* herein to improve reliability, function, or design. Freescale  does not assume any
;* liability arising out of the application or use of any product, circuit, or software
;* described herein; neither does it convey any license under its patent rights nor the
;* rights of others. Freescale products are not designed, intended, or authorized for
;* use as components in systems intended for surgical implant into the body, or other
;* applications intended to support life, or for any other application in which the
;* failure of the Freescale product could create a situation where personal injury or
;* death may occur. Should Buyer purchase or use Freescale products for any such
;* intended or unauthorized application, Buyer shall indemnify and hold Freescale  and
;* its officers, employees, subsidiaries, affiliates, and distributors harmless against
;* all claims, costs, damages, and expenses, and reasonable attorney fees arising out
;* of, directly or indirectly, any claim of personal injury or death associated with
;* such unintended or unauthorized use, even if such claim alleges that Freescale  was
;* negligent regarding the design or manufacture of the part.
;*
;* Freescale is a registered trademark of Freescale, Inc.
;*******************************************************************************

;                XDEF Entry,main,trimval
```

Freescale Semiconductor, Inc.

```
;*******************************************************************************************
;* Equates and Data Table Includes
;*******************************************************************************************

                         include 'MC68HC908QT4.equ'    ; For the QT1, QT2, QT4, QY1, QY2, QY4

        org $FFC0

;trimval:    DC.B $FF           ;here we set the FLASH trim to a default value.
                                ;DO NOT change this value, as the trim will not be
                                ;automatically calibrated by the programming interface if
                                ;this value is anything other than $FF

;DEFAULT_RAM         SECTION SHORT
        org   RamStart


;*******************************************************************************************
;* Constants and Variables for this file
;*******************************************************************************************

                include 'battery.equ'

;DEFAULT_ROM         SECTION
            org   FlashStart


;*******************************************************************************************
;* SUBROUTINES
;* This part includes subroutines
;*******************************************************************************************

;Subroutine for Timer

TimerHalfL: mov   #initTim,TSC ;Timer - Cleared + Stopped.

            mov   #InitTMODHL,TMODH ;Set Timer to ~ 4ms or 1/4 Power Cycle Line (PCL)
            mov   #InitTMODLL,TMODL ;after we start the timer

            jmp  Skip

TimerHalfH: mov   #initTim,TSC ;Timer - Cleared + Stopped.

            mov   #InitTMODHH,TMODH ;Set Timer to ~ 4ms or 1/4 Power Cycle Line (PCL)
            mov   #InitTMODLH,TMODL ;after we start the timer.

            jmp  Skip


Trickle:    mov   #initTim,TSC  ;Timer 1 - Cleared + Stopped.

            mov   #initTricH,TMODH ;Set Timer to low current
            mov   #initTricL,TMODL ;after we start the timer

            jmp  Skip
```

Freescale Semiconductor, Inc.

```
;Subroutine for Thyristor gate control

Gate:       lda   #GateVal    ;Gate pulse duration
loop:       bset  PTB7,PTB
            dbnza loop

            bclr  PTB7,PTB    ;PTB7 generates a pulse on Thyristor gate

            jmp  Skip

;Subroutine for Battery reading

BatRead:    bset  PTB1,PTB    ;Turn transistor on

            mov   #initADCH3,ADSCR ;Start Conversion, CH3 selected

            lda   ADR
            cmp   #BatInit    ;Keep in this loop while battery not connected
            blo   BatRead

            bclr  PTB1,PTB    ;Turn transistor off

            bra   Skip

;Subroutine for Timer Overflow

TOverflow:  nop
            nop
            brclr TOF,TSC,TOverflow ;Wait for Timer Overflow

            lda   TSC
            and   #TSCClr
            sta   TSC         ;Clear TOF bit

            mov   #initTim,TSC ;STOP and RESET Counter

            bra   Skip

;Subroutine to delay about 2s before reading battery voltage

Delay:      ldx   #Del

Delay1:     lda   #Dela1
            sta   del1

Delay2:     lda   #Dela2
            sta   del2

Delay3:     nop
            dbnz  del2,Delay3

            dbnz  del1,Delay2

            dbnzx Delay1

            bra   Skip
```

Freescale Semiconductor, Inc.

```
;Subroutine for battery voltage reading

VbattH:     mov   #initADCH2,ADSCR  ;Start Conversion, CH2 selected

Waitcoco:   nop
            nop
            brclr COCO,ADSCR,Waitcoco ;Wait for Conversion complete

            lda   ADR          ;Load AD value
            and   #MaskLSB     ;Mask LSB
            sta   VoltReadH    ;store value into variable

            bra   Skip

;Subroutine for first battery temperature reading

VFrsttemp:  mov   #initADCH1,ADSCR  ;Start Conversion, CH1 selected

Waitcoco1:  nop
            nop
            brclr COCO,ADSCR,Waitcoco1 ;Wait for Conversion complete

            lda   ADR          ;Load AD value
            sta   FrstTmpRd    ;store value into variable

            bra   Skip

;Subroutine for battery temperature reading

VActemp:    mov   #initADCH1,ADSCR  ;Start Conversion, CH1 selected

Waitcoco2:  nop
            nop
            brclr COCO,ADSCR,Waitcoco2 ;Wait for Conversion complete

            lda   ADR          ;Load AD value
            sta   AcTmpRd      ;store value into variable

Skip:       rts

;*********************************************************************************************
;* main    - This is the point where code starts executing
;*            after a RESET.
;*********************************************************************************************
Entry:
main:
            mov   #initCfg1,CONFIG1 ;Set config1 register
                               ;(LVI and COP disabled)

            mov   #initCfg2,CONFIG2 ;set MCU to internal oscillator

            clr   PTB
            mov   #InitDDRB,DDRB   ;PTB7 -> Pulses on Thyristor gate
                               ;PTB3 -> Red LED (Bat. Charging)
                               ;PTB2 -> Green LED (Bat. Charged)
                               ;PTB1 -> Transistor Control
```

Freescale Semiconductor, Inc.

```
            bclr  DDRA0,DDRA  ;Zero Crossing detection

            mov   #ADclkval,ADICLK ;ADC clock, bus clock/ 16

;Enable ADCH3

            mov   #initADCH3,ADSCR ;Start Conversion, CH3 selected


            lda   TRIMLOC     ;load the TRIM value stored in FLASH
            sta   OSCTRIM     ;use this stored value.

            rsp
            clra
            clrx

;Clear Variables

            clr   Counter0
            clr   Counter1

            clr   VoltReadL
            clr   VoltReadH

            clr   AcTmpRd
            clr   FrstTmpRd

            jsr   TimerHalfL   ;Go config Timer
            cli               ;Allow interrupts to happen

            jsr   BatRead     ;Go read battery

            jsr   VFrsttemp   ;Go read First temp value

            bset  PTB3,PTB    ;Turn on Red LED (Battery is charging)

Waitpta0:   nop
            brclr PTA0,PTA,Waitpta0  ;Wait for a positive edge on PTA0 (Zero crossing)

            jsr   TimerHalfH  ;Go config Timer

            mov   #StartTim,TSC  ;Start the timer

            jsr   TOverflow   ;Go to Timer Overflow subroutine

            jsr   Gate        ;Go to Gate subroutine

Waitpta:    nop
            brset  PTA0,PTA,Waitpta  ;Wait for a negative edge on PTA0 (Zero crossing)

            jsr   TimerHalfL   ;Go config Timer

            mov   #StartTim,TSC ; Start the timer

            jsr   TOverflow   ;Go to Timer Overflow subroutine
```

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

```
            jsr   Gate        ;Go to Gate subroutine

            inc   Counter0    ;Increment 1st byte Counter for charge time OVF period
            lda   #StpChL
            cbeq  Counter0,Count1  ;Go to Count1 if Counter0 > $FF

            bra   Waitpta0

Count1:     inc   Counter1    ;Increment 2nd byte Counter for charge time OVF period
            lda   #StpChH
            cbeq  Counter1,Vbat  ;Go to Vbat if Counter1 > $90

            bra   Waitpta0

Vbat:       mov   #initTim,TSC  ;Stop and reset counter

            clr   Counter0
            clr   Counter1

            bset  PTB1,PTB    ;Turn transistor on

            jsr   Delay       ;Go to Delay subroutine

            jsr   VbattH      ;Jump to subroutine that reads battery voltage

            jsr   Delay       ;Go to Delay subroutine

            bclr  PTB1,PTB    ;Turn transistor off

            lda   VoltReadH
            sub   VoltReadL   ;Compare last battery voltage with current one

            blo   Charged     ;Jump to Charged if last value < current value

            lda   VoltReadH
            sta   VoltReadL   ;load variable with last value

            jsr   VActemp     ;Jump to subroutine that reads temperature

            lda   FrstTmpRd
            sub   AcTmpRd     ;Compare last temperature with current one
            cmp   #TmpSafe

            bhs   Charged     ;Jump to Charged if temperature increases more than
                              ;25oC.

            bra   Waitpta0

; Battery fully charged

Charged:    bclr  PTB3,PTB    ;Turn off Red LED (Battery charging)
            bset  PTB2,PTB    ;Turn on Green LED (Battery Charged)

            jsr   Trickle     ;Go to trickle current subroutine

            bra   Waitpta0
```

Freescale Semiconductor, Inc.

```
Dummytc:
      RTI

*****************************************
* Vectors
*****************************************

     ORG $FFDE
     DW Dummytc                   ; ADC conversion complete vector
     ORG $FFE0
     DW Dummytc                   ; Keyboard vector
     ORG $FFF2
     DW Dummytc                   ; TIM overflow vector
     ORG $FFF4
     DW Dummytc                   ; TIM Channel 1 vector
     ORG $FFF6
     DW Dummytc                   ; TIM Channel 0 vector
     ORG $FFFA
     DW Dummytc                   ; IRQ vector
     ORG $FFFC
     DW Dummytc                   ; SWI vector
     ORG $FFFE
     DW   main                    ; Reset vector

END
```

Freescale Semiconductor, Inc.

```
;*******************************************************************************
;* Title: battery.equ                                       Copyright (c) 2003
;*******************************************************************************
;* Author: Marcus Espindola - Freescale BSTC
;*
;* Description: Constants and variables definitions for MC68HC908QY4 and MC68HC908QT4.
;*
;* Documentation: HC908QY4 Data Sheet (MC68HC908QY4/D) for register and bit explanations
;*
;* Include Files:
;*
;* Assembler: P&E Microcomputer Systems - CASM for HC08
;*            Metrowerks CodeWarrior Compiler for HC08 V-5.0.17
;*
;* Revision History:
;* Rev #      Date      Who         Comments
;* -----   -----------  ---------   -----------------------------------------
;* 0.2      04-Nov-03   Espindola   Included constants for source file
;* 0.1      12-Aug-03   Espindola   Initial data entry
;*******************************************************************************
;*******************************************************************************
;* Freescale reserves the right to make changes without further notice to any product
;* herein to improve reliability, function, or design. Freescale does not assume any
;* liability arising out of the application or use of any product, circuit, or software
;* described herein; neither does it convey any license under its patent rights nor the
;* rights of others. Freescale products are not designed, intended, or authorized for
;* use as components in systems intended for surgical implant into the body, or other
;* applications intended to support life, or for any other application in which the
;* failure of the Freescale product could create a situation where personal injury or
;* death may occur. Should Buyer purchase or use Freescale products for any such
;* intended or unauthorized application, Buyer shall indemnify and hold Freescale and
;* its officers, employees, subsidiaries, affiliates, and distributors harmless against
;* all claims, costs, damages, and expenses, and reasonable attorney fees arising out
;* of, directly or indirectly, any claim of personal injury or death associated with
;* such unintended or unauthorized use, even if such claim alleges that Freescale was
;* negligent regarding the design or manufacture of the part.
;*
;* Freescale is a registered trademark of Freescale, Inc.
;*******************************************************************************

;*******************************************************************************
;* Constants and Variables for this file
;*******************************************************************************

initCfg1:   equ   %00010001   ;Config1 Register value
;                  ||||||||   CONFIG1 is a write once register
;                  |||||||+-COPD     - 1 disable COP Watchdog
;                  ||||||+--STOP     - 0 disable STOP instruction
;                  |||||+---SSREC    - 0 4096 cycle STOP recovery
;                  ||||+----LVI5OR3 - 0 set LVI for 3V system
;                  |||+-----LVIPWRD - 1 disable power to LVI system
;                  ||+------LVIRSTD - 0 enable reset on LVI trip
;                  |+-------LVISTOP - 0 disable LVI in STOP mode
;                  +--------COPRS    - 0 long COP timeout
```

Freescale Semiconductor, Inc.

```
initCfg2:   equ   %00000000   ;Config2 Register value
;                  ||||||||   CONFIG2 is a write once register
;                  |||||||+-RSTEN   - 0 Reset function inactive in pin
;                  ||||||+--R       - 0 Reserved bit
;                  |||||+---R       - 0 Reserved bit
;                  ||||+----OSCOPT0 - 0 Set oscillator option as internal
;                  |||+-----OSCOPT1 - 0 Set oscillator option as internal
;                  ||+------R       - 0 Reserved bit
;                  |+-------IRQEN   - 0 disable IRQ function
;                  +--------IRQPUD  - 0 Internal pullup to connect IRQ and VDD

initADCH3:  equ   %00100011   ;AD configuration value
;                  ||||||||   ADC Status and Control Register
;                  |||||||+-CH0     - 1 Mux to select ADC channel
;                  ||||||+--CH1     - 1 Mux to select ADC channel
;                  |||||+---CH2     - 0 Mux to select ADC channel
;                  ||||+----CH3     - 0 Mux to select ADC channel
;                  |||+-----CH4     - 0 Channel 3 selected
;                  ||+------ADCO    - 1 Set ADC as continuous conversion
;                  |+-------AIEN    - 0 disable ADC interrupt
;                  +--------COCO    - 0 Conversions Complete Bit

initADCH2:  equ   %00000010   ;AD configuration value
;                  ||||||||   ADC Status and Control Register
;                  |||||||+-CH0     - 0 Mux to select ADC channel
;                  ||||||+--CH1     - 1 Mux to select ADC channel
;                  |||||+---CH2     - 0 Mux to select ADC channel
;                  ||||+----CH3     - 0 Mux to select ADC channel
;                  |||+-----CH4     - 0 Channel 2 selected
;                  ||+------ADCO    - 0 Set ADC as single conversion
;                  |+-------AIEN    - 0 disable ADC interrupt
;                  +--------COCO    - 0 Conversions Complete Bit

initADCH1:  equ   %00000001   ;AD configuration value
;                  ||||||||   ADC Status and Control Register
;                  |||||||+-CH0     - 1 Mux to select ADC channel
;                  ||||||+--CH1     - 0 Mux to select ADC channel
;                  |||||+---CH2     - 0 Mux to select ADC channel
;                  ||||+----CH3     - 0 Mux to select ADC channel
;                  |||+-----CH4     - 0 Channel 2 selected
;                  ||+------ADCO    - 0 Set ADC as single conversion
;                  |+-------AIEN    - 0 disable ADC interrupt
;                  +--------COCO    - 0 Conversions Complete Bit

initTim:    equ   %00110001   ;Timer Status and control Reg. value
;                  |||||||| TIM Status and Control Register
;                  |||||||+-PS0     - 1 Prescaler select bit
;                  ||||||+--PS1     - 0 Prescaler select bit
;                  |||||+---PS2     - 0 Tim clock source int. bus
;                  ||||+----0       - 0
;                  |||+-----TRST    - 1 TIM reset bit
;                  ||+------TSTOP   - 1 TIM counter stopped
;                  |+-------TOIE    - 0 disable TIM overflow interrupts
;                  +--------TOF     - 0 TIM overflow flag bit
```

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

```
StartTim:   equu   %00000001   ;Timer Status and control Reg. value
;                   ||||||||   TIM Status and Control Register
;                   |||||||+-PS0     - 1 Prescaler select bit
;                   ||||||+--PS1     - 0 Prescaler select bit
;                   |||||+---PS2     - 0 Tim clock source int. bus
;                   ||||+----0       - 0
;                   |||+-----TRST    - 0 TIM reset bit
;                   ||+------TSTOP   - 0 TIM counter started
;                   |+-------TOIE    - 0 disable TIM overflow interrupts
;                   +--------TOF     - 0 TIM overflow flag bit

InitDDRB:   equ    %10001110   ;PTB7 -> Pulses on Thyristor gate
                               ;PTB3 -> Red LED (Bat. Charging)
                               ;PTB2 -> Green LED (Bat. Charged)
                               ;PTB1 -> Transistor Control

InitTMODHL: equ    $1A         ;Set Timer to ~ 4ms or 1/4 Power Cycle Line (PCL)
InitTMODLL: equ    $1D         ;after we start the timer for negative edge.

InitTMODHH: equ    $0D         ;Set Timer to ~ 4ms or 1/4 Power Cycle Line (PCL)
InitTMODLH: equ    $0E         ;after we start the timer for positive edge.

initTricH:  equ    $2A         ;Set Timer to low current
initTricL:  equ    $6F         ;after we start the timer

ADclkval:   equ    %10000000   ;AD clock configuration
                               ;ADC Clock prescaler bit

GateVal:    equ    $50         ;Gate pulse duration

;Variables for counter for charge time overflow period
Counter0    rmb    1
Counter1    rmb    1           ;Time Counters

;Variables for voltage reading
VoltReadL   rmb    1
VoltReadH   rmb    1

;Variables for delay before reading battery voltage
del1        rmb    1
del2        rmb    1

;Variables for Temperature reading
AcTmpRd     rmb    1
FrstTmpRd   rmb    1

;Other Constants

BatInit     equ    $19         ;Value to identify if battery pack is connected

Del         equ    $10         ;First value for delay
Dela1       equ    $FF         ;Second value for delay
Dela2       equ    $FF         ;Third value for delay
```

```
MaskLSB     equ   $FE          ;Value to mask ADR LSB

StpChL      equ   $00          ;Low byte for stop charger period
StpChH      equ   $90          ;High byte for stop charger period

TmpSafe     equ   $06          ;Temperature value for backup

TSCClr      equ   $7F          ;Value to clear TOF bit on TSC register
```

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

This page is intentionally blank

Freescale Semiconductor, Inc.

This page is intentionally blank

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

[redacted]

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

freescale™
semiconductor