SAMSUNG ELECTRONICS

APPLICATION NOTE

S3F84K4

LI-ION BATTERY CHARGER

December, 2008

REV 1.00

Trademark & Copyright Information

The information in this publication has been carefully checked and is believed to be entirely accurate at the time of publication. Samsung assumes no responsibility for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

No part of the information contained in this document maybe reproduced or used without the prior consent of Samsung Electronics Co., Ltd.

Samsung reserves the right to make any changes in its products or product specifications including documentation with the intent to improve function or design at any time without notice.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung 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 any consequential or incidental damages.

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu,
Yongin-City, Gyeonggi-Do, Korea
446-711

TEL: (82)-(31)-209-8344
FAX: (82)-(31)-209-6494

Revision History

Revision No.	Description	Author(s)	Date
0.00	- Initial Draft (Preliminary spec)	Fan. Zhang	July 2007
1.00	- Added a test result	Fan. Zhang	August 2007

Table of Contents

List of Figures

List of Tables

LI-ION BATTERY CHARGER

1 OVERVIEW

The battery charger reference design is based on Li-Ion batteries, which is widely used in portable electronic equipment today. It can also charge other batteries such as SLA, NiCd, NiMH batteries by little change of the software.

The battery charger reference design used Samsung's newly developed microcontroller S3F84K4, which is ideal for use in a wide range of electronic applications requiring simple timer/counter, PWM, ADC. The 12-bit PWM and 10-bit Analog-to-Digital converter plays an important part in making the optimum charging algorithms possible. Special features include a build-in reset circuit and an internal RC oscillator that reduces the using of external components.

1.1 FEATURES

- Advanced Charge Algorithms:
  - Automatic detection of shorted or damaged cells
  - Automatic detection of battery inversed input
  - Configurable over-current, over-voltage and over-temperature suspension
  - Charge termination at user-specified minimum current or time-out
- 10-bit ADC for Voltage, Current and Temperature Measurement
- 1 Bi-Colour LED (Red/Green) to indicate the charging status
- Optional power supply
  - DC +9V input with a LM7805 regulator
  - USB input to satisfy the new mobile phone battery charger standard of China
- Maximum Integration for Optimal PCB Size:
  - Build-in reset circuit (LVR)
  - Internal 8 MHz/1MHz clock oscillator

1.2 SYSTEM BLOCK DIAGRAM

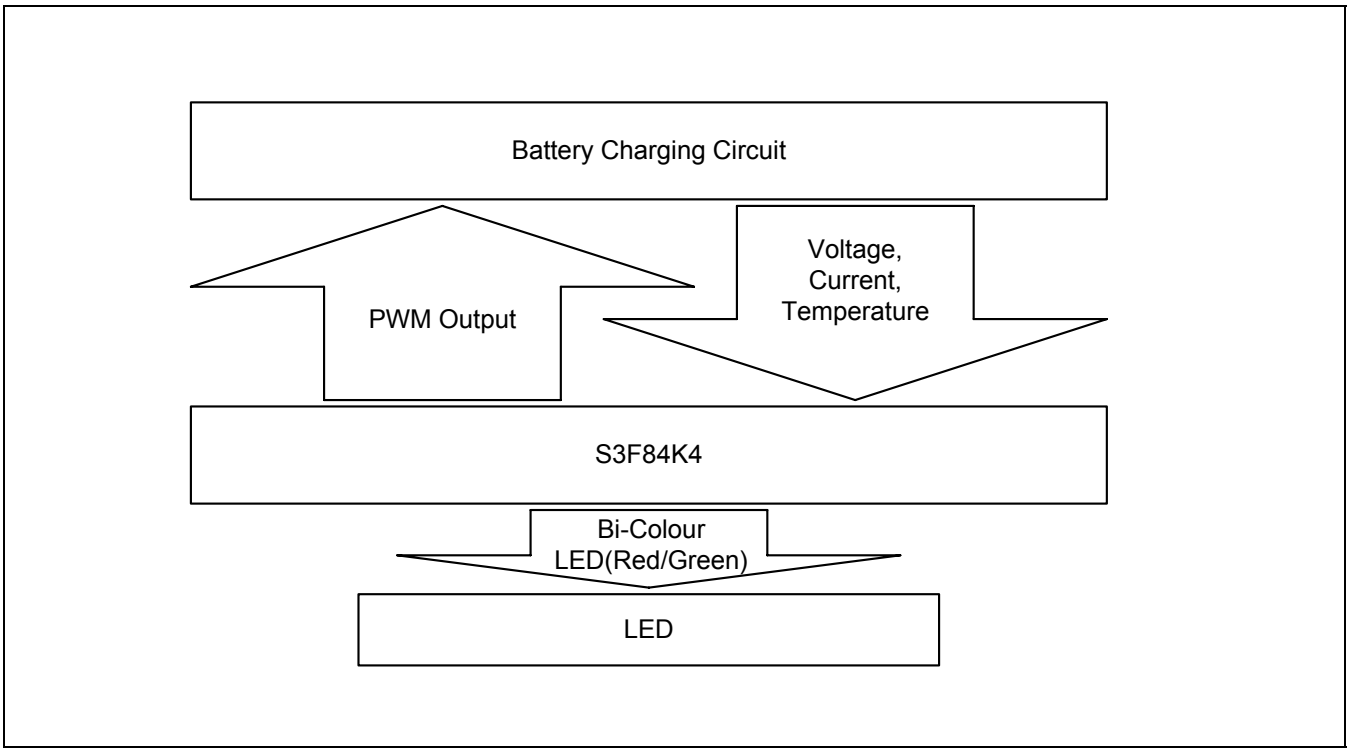

Figure 1. System Block Diagram

1.3 CHARGE THEORY

The portable electric equipments need large capacity batteries to keep the working time much longer. As far as we know, the charging of batteries is made by a reversible chemical reaction that restores energy in batteries. And the larger the capacity is, the longer it takes to charge. So we need fast charging method to reduce the charge time. The easiest way is to increase the charging current. But for some kind of batteries such as the Li-Ion batteries this method will make the battery overheating, makes it damaged or even exploded. A better way is to charge different kind of batteries in different way according to the chemical characters of each.

1.4 CHARGRE METHOD

For Li-Ion battery, the better way is to divide the whole charging process into 2 states, one is called the constant current state, and the other is called the constant voltage state. In the first constant current state, we need to keep the current stable, and the voltage of the battery will rising slowly. While in the second constant voltage state, we need to keep the voltage stable, and the charging current will falling down slowly, as showed in the Fig below.

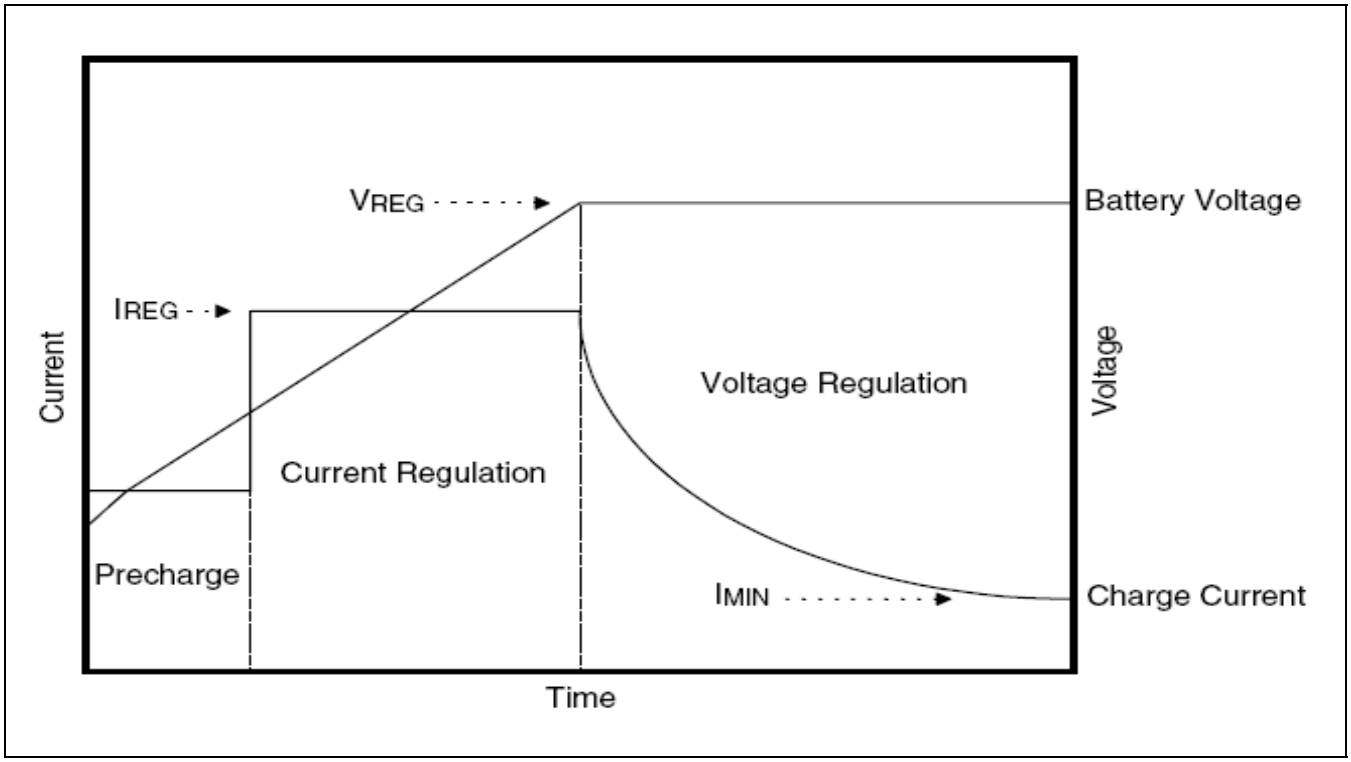

Figure 2. Charge Method

To fulfil this charging task, we need 2 ADC channels to monitor the battery voltage and the charging current. And for safe charging, we need an additional ADC channel to monitor the temperature of the battery.

1.5 TERMINATION METHOD

As we see in the Fig above, for Li-Ion battery, charging is terminated when the charge current drops below a preset value. This is the main termination method. Then check if the voltage is above a preset value, this is used as a backup termination method. And for safety, charging is terminated when temperature, voltage or current is above a preset threshold value. This is the emergent termination method.

2 HARDWARE IMPLEMENTATION

The reference design could be divided into 5 main blocks, power supply, buck converter; microcontroller, display and signal transform circuit.

2.1 POWER SUPPLY

There are two power supplies in the reference design. One is a +9V DC input, it can supply +5V DC for both microcontroller and buck converter through a LM7805 regulator. The other is a USB +5V input, directly supply the whole system. We can also supply the buck converter by the +9V DC input directly to get larger charge current. But in that way, we need to adjust the signal transform circuit a little for not exceeding the threshold of ADC input. The red led on the board indicates power on.

2.2 MICROCONTROLLER

The S3F84K4's high frequency PWM output (135 KHz maxim) makes it especially suitable for battery charger application. This high frequency PWM output can reduce the cost on the external coil and capacitors. The reference design use 3 ADC channels to monitor the battery voltage, charge current and temperature. The S3F84K4's 10-bit ADC module provides high accuracy measurement of the charge parameters.

2.3 DISPLAY

The reference design provides one bi-colour LED to indicate the state of charge. If there is no battery inserted, the LED is red. If the charge is in process, the LED is flicking red and green in turns. If the battery is fully charged, the LED is green. If there is some error detected, the LED is flicking red. In this way, all the states could be distinguished easily.

2.4 BUCK CONVERTER

The buck converter consists of one MOSFET switching transistor driven by the PWM output of microcontroller via a NPN transistor. The MOSFET is connected to an inductor, a diode and a capacitor.

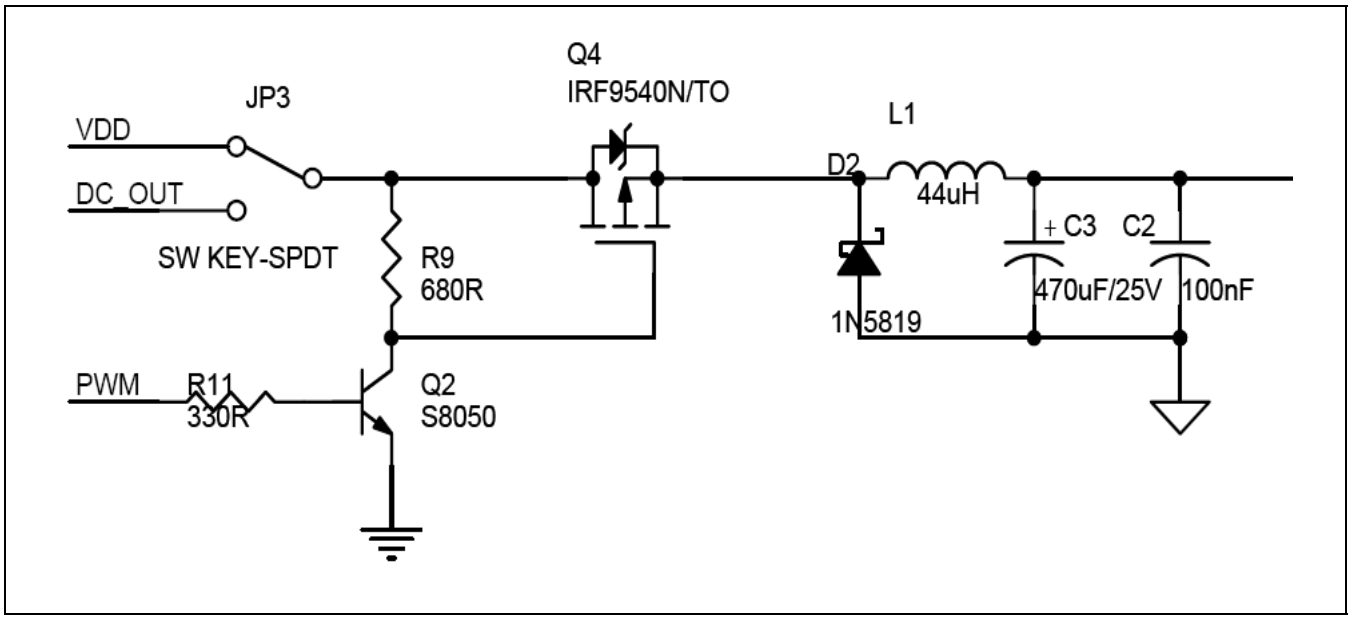

Figure 3. Buck Converter

2.4.1 Parameters for Layout

Given,

Oscillator frequency: f_{OSC} = 8MHz

Input voltage: V_I = 5.0v

Saturation voltage: V_{SAT} = 0.2v

Output voltage: V_O = 4.2v

Load current: I_{LOAD} = 0.3A

Ripple current: I_{ripple} = 0.3 I_{LOAD} = 0.09A (typically 30%)

So,

The switch frequency: f_{SW} = 8MHz / 2^6 = 125 KHz (12-bit PWM, only use 6-bit basic cycle)

The time in one cycle: T = 1/125 KHz = 8us

The duty cycle: D = V_O / V_I = 4.2v / 5.0v = 0.84

The time switch on: t_{ON} = D × T = 0.84×8us = 6.72us

For an inductor: V = L × ⊿I / ⊿t

So L = V × ⊿t / ⊿I

= (V_I - V_{SAT} - V_O) × t_{ON} / I_{ripple}

= (5v – 0.2v – 4.2v) × 6.72us / 0.09A

= 44.8uH

For L = 44uH

⊿t = L × ⊿I / V

= 44 × 0.09A / (5v – 0.2v – 4.2v)

= 6.6us

For a capacitor: ⊿V = ⊿I × (ESR + ⊿t/C + ESL/⊿t)

Assume: ESL = 0

ESR = 0.03Ω

⊿V = 0.05V

So

⊿t = 6.6us

C = ⊿I × ⊿t / (⊿V - ⊿I × ESR)

= 0.09A × 6.6us / (0.05V – 0.09A × 0.03Ω)

= 12.56uF (minimum value)

2.5 SIGNAL TRANSFORM CIRCUIT

Signal transform circuit include 3 analog inputs, the voltage, the current, and the temperature.

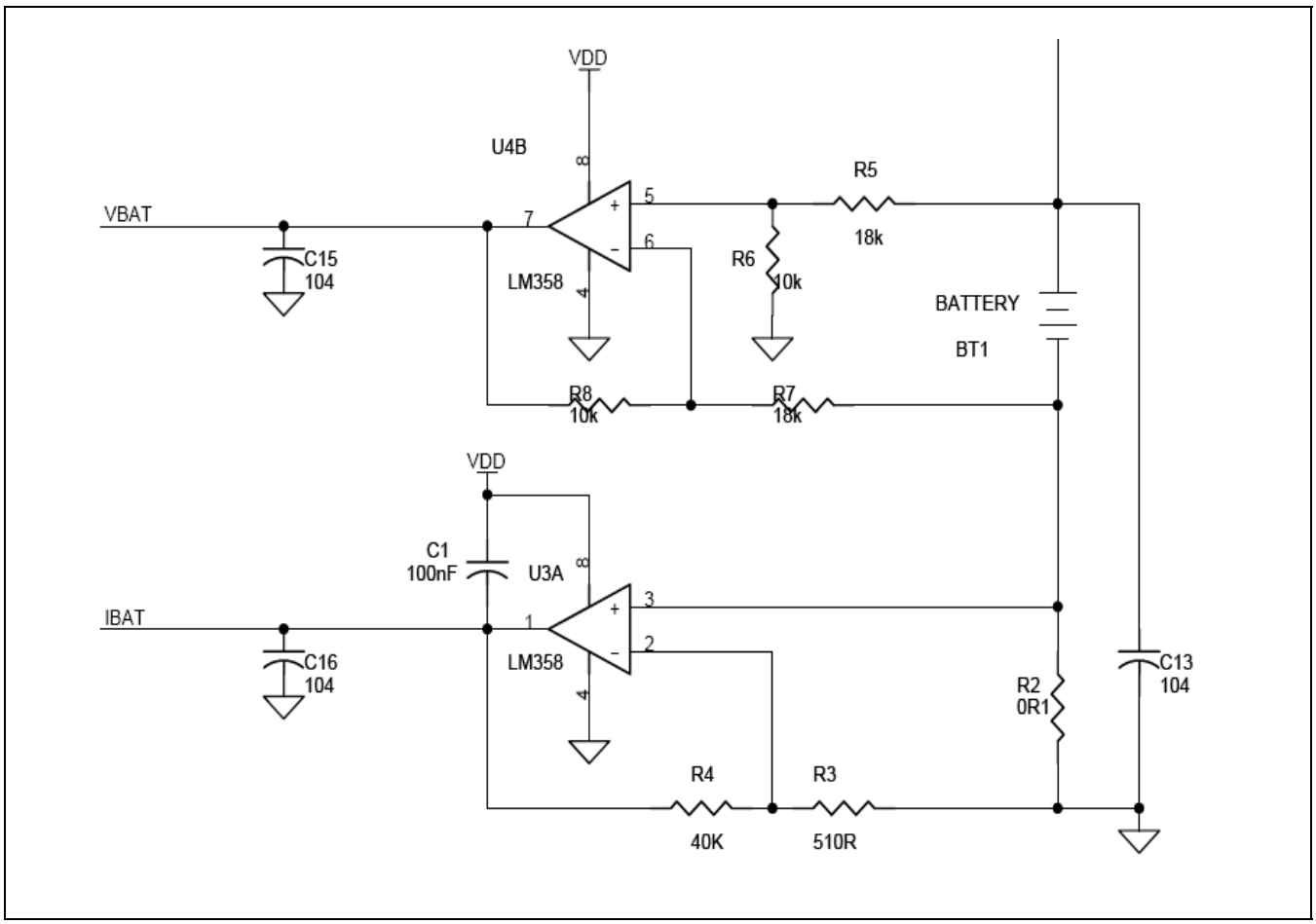

Figure 4. Signal Transform Circuit

2.5.1 Battery Voltage

As showed in the Fig above, one LM358 op-amp is used for the measurement of the voltage and the current. When using +5V to charge, R5 and R7 should change to 10kΩ.or the U4B could be unused, and the positive pole of battery could directly connect to the ADC input of VBAT. When using +9V to charge, or using a different voltage range battery, user should use the op-amp and carefully select the parameters of the resistors.

$$VBAT = (R5/R7) \times (V_+ - V_-)$$

$$R5 = R7,\ R6 = R8$$

Where:

VBAT is op-amp's output, connected to the analog input of microcontroller. This value should in the range of 0 to +5v.

V_+ and V_- is the positive and negative pole of the battery.

For

R5 = R7 = 18kΩ, R6 = R8 = 10kΩ

The maxim charge voltage is:

$(V_+ - V_-)_{max}$ = 1.8 VBAT = 9v (maxim ADC input for S3F84K4 is +5v)

The battery voltage measurement resolution:
(R5/R7) × V_{ref} / 2^{10} = 1.8 × 5v / 1024 = 8.79mV/step

2.5.2 Charge current

As showed in the Fig above, the charge current is measured by a resistor and amplified by the op-amp.

$$I_{charge} = R3/ (R3+R4)/ R2 \times V_0$$

Where

V_0 is the voltage of R2.

For

R3 = 510Ω, R4 = 40kΩ, R2 = 0.1Ω

$$I_{charge} = 510/ (40000+510)/0.1\ V_0 = 0.126\ V_0$$

$$V_0 = 7.943\ I_{charge}$$

The maxim charge current is:

$I_{charge\ max}$ = 0.126 $V_{0\ max}$ = 0.126×5v = 0.63A = 630mA

The charge current measurement resolution:
630mA / 2^{10} = 615 uA/step

2.5.3 Temperature

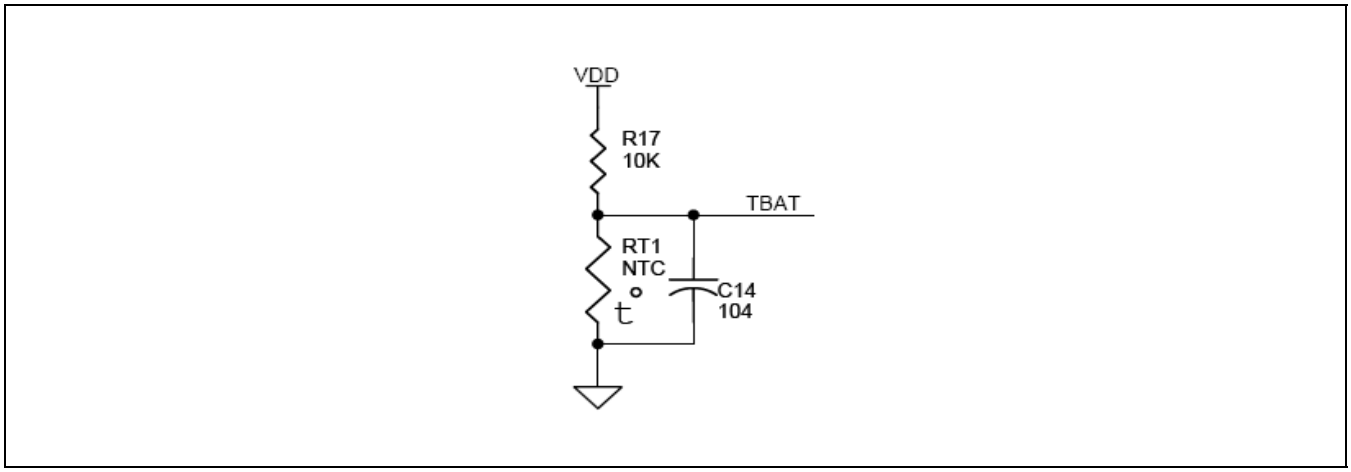

Figure 5. Temperature Detect Circuit

Temperature is measured by a negative temperature coefficient (NTC) resistor. It has an approximate resistance of 10k at 25°C.

The temperature measurement resolution:

$5V / 2^{10} = 4.88\ mV$

The temperature measured:

TBAT = RT1 / (RT1+R17)

The NTC resistance is not linear, which makes it difficult to calculate the temperature. But what we need to know is just the resistance under maxim and minim temperature for protecting the battery from over heating and over cooling.

3 SOFTWARE IMPLEMENTATION

This section describes the software used in the battery charger reference design. The software is written in assembly code.

3.1 SOURCE CODE FILES

The source code includes the following files:

Table 1. Source Code Files

File Name	Description	Note
Start.src	Program start, initiate the microcontroller, back ground job	
Exception.src	Exception handlers for Timer0 and PWM module	
Util.src	Collection of user functions	
Cofig.src	Config of IO, Timer, PWM and ADC	
DvDr.src	Collection of I/O device drivers	
Task.src	tasks for each Mode	
Const.src	Collection of Tables	
Var.src	To declare all variables	
Defines.src	Define of constants	

3.1.1 Start.src

This is the start part of the whole program. This part includes the smart option setting, the interrupt vector table setting, the internal register setting, RAM clear, and other peripheral initiation. After that, the system goes into an infinite loop, and waiting for interrupt.

3.1.2 Cofig.src

This part includes the initiate configs of ports, Timer0, PWM and ADC module.

3.1.3 Exception.src

This part is the exception handlers of Timer0 and PWM module. In PWM overflow interrupt, just clear the pending bit. In the Timer0 interrupt, the microcontroller should read ADC value, check temperature and charging time, and then call a task according to the mode right now.

3.1.4 Task.src

This part includes the tasks for each mode and the state machine of the modes. There are 5 modes during the charge process, open mode, constant current charge mode, constant voltage charge mode, full mode, and fault mode.

This file includes the following functions:

Table 2. Functions Included in Task.src

Function	Description	Note
tsk_OpenMode	The tasks of open mode	
tsk_Chrg_Cc_Mode	The tasks of constant current charge mode	
tsk_Chrg_Cv_Mode	The tasks of constant voltage charge mode	
tsk_FullMode	The tasks of full mode	
tsk_FaultMode	The tasks of fault mode	

3.1.5 Task Open Mode

After initiate, the charger goes into open mode, waiting for the insert of the battery.

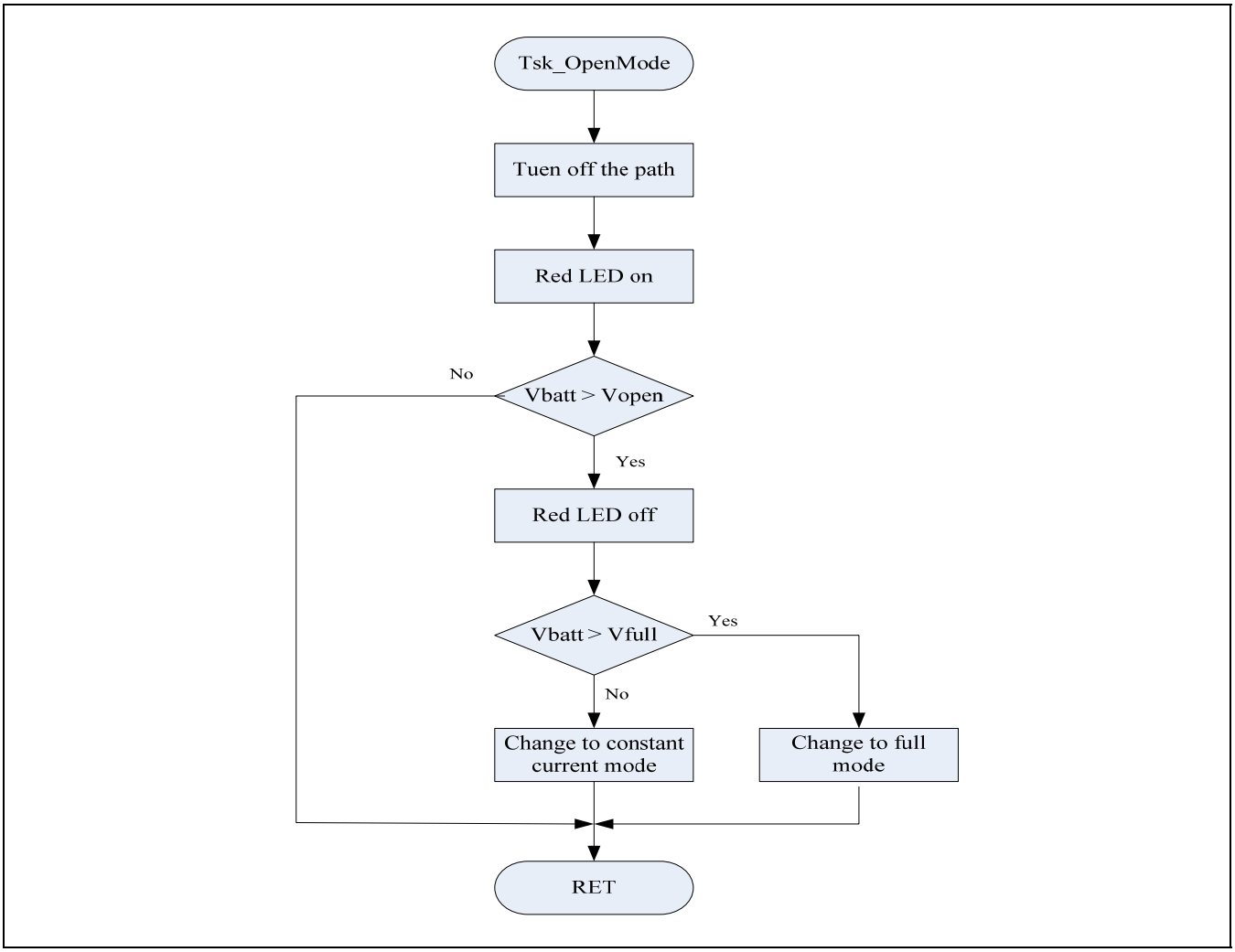

Figure 6. Task in Open Mode

3.1.6 Task Constant Current Charge Mode

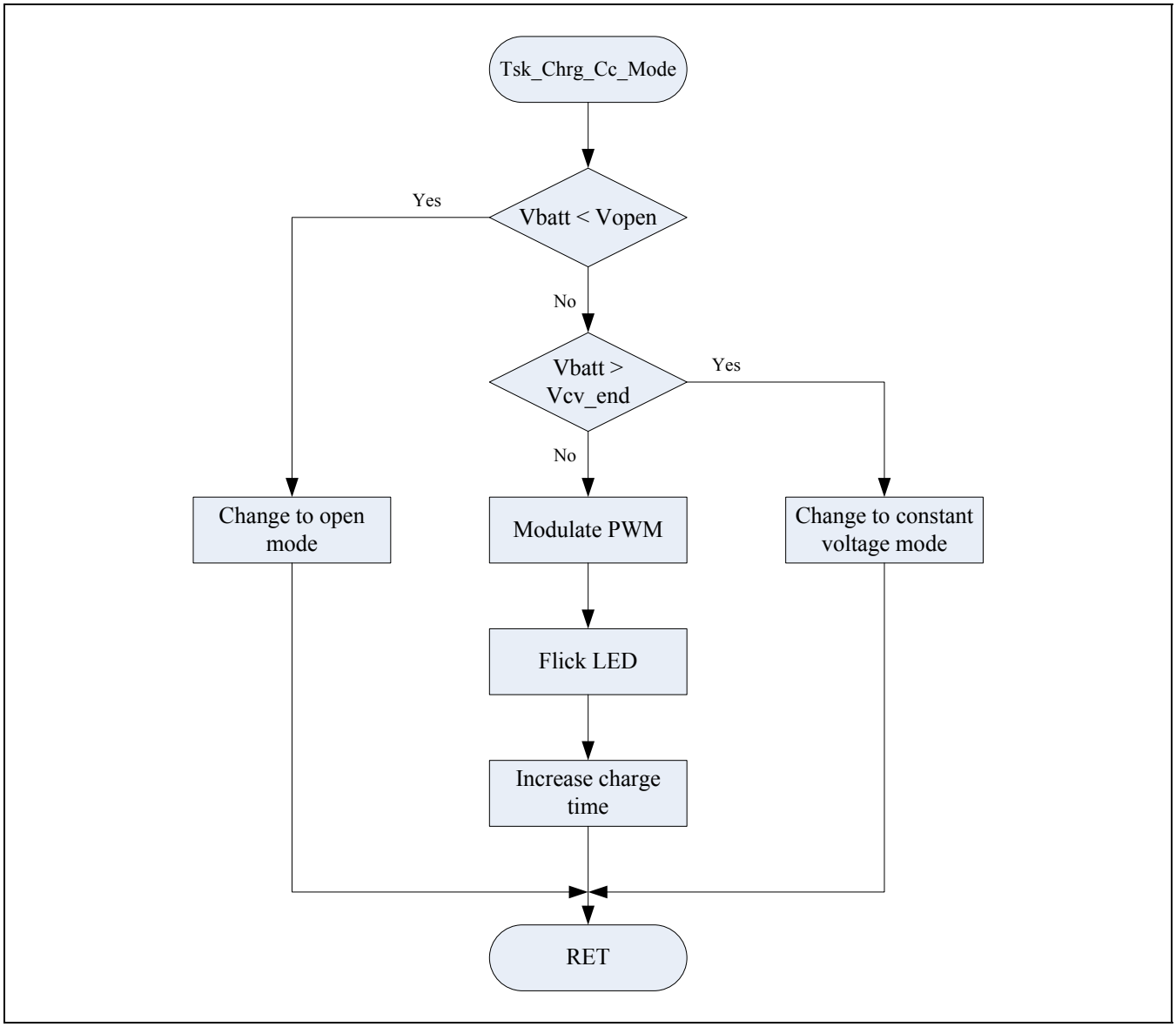

Figure 7. Task in Constant Current Mode

3.1.7 Task Constant Voltage Charge Mode

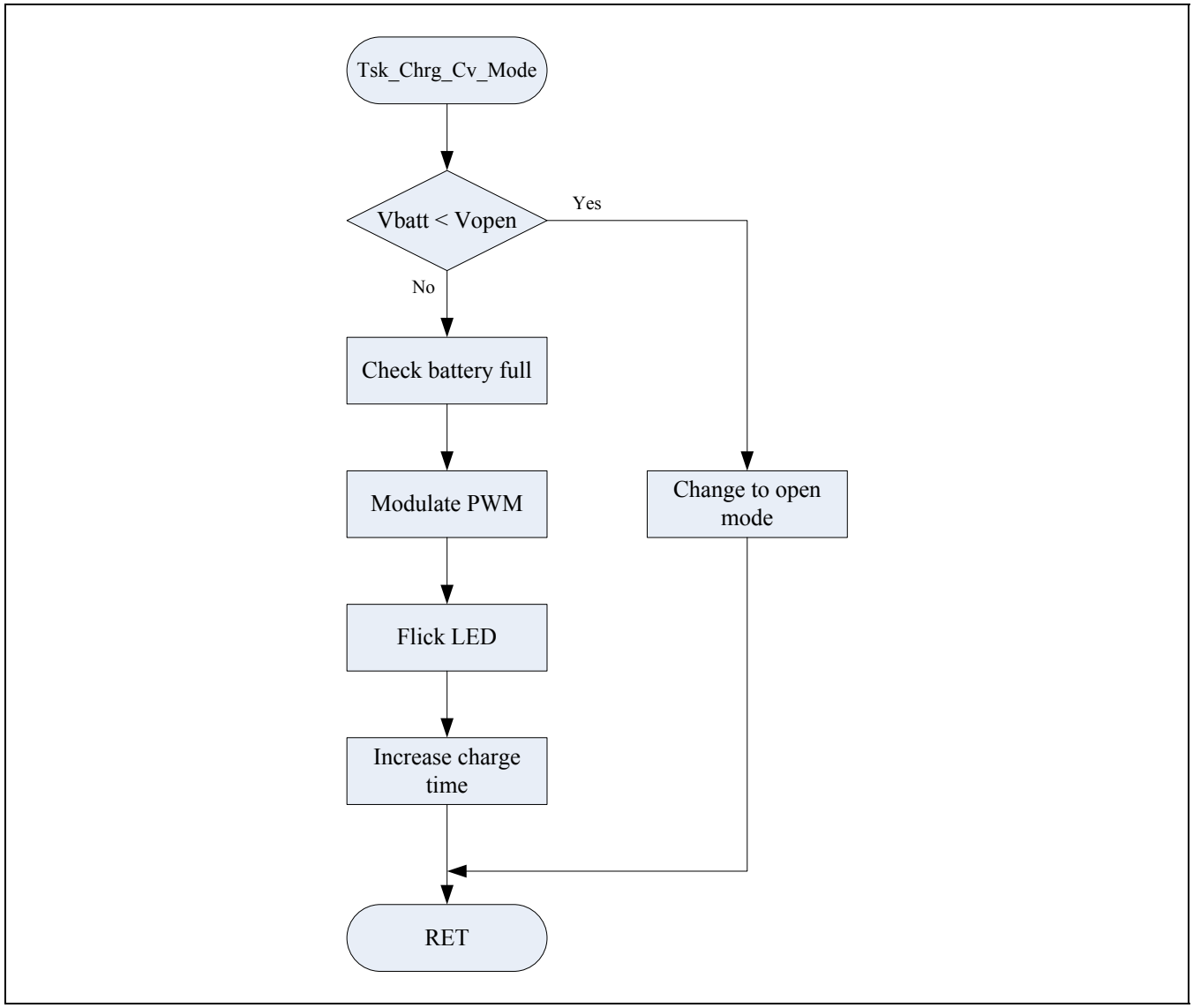

Figure 8. Task in Constant Voltage Mode

3.1.8 Task Full Mode

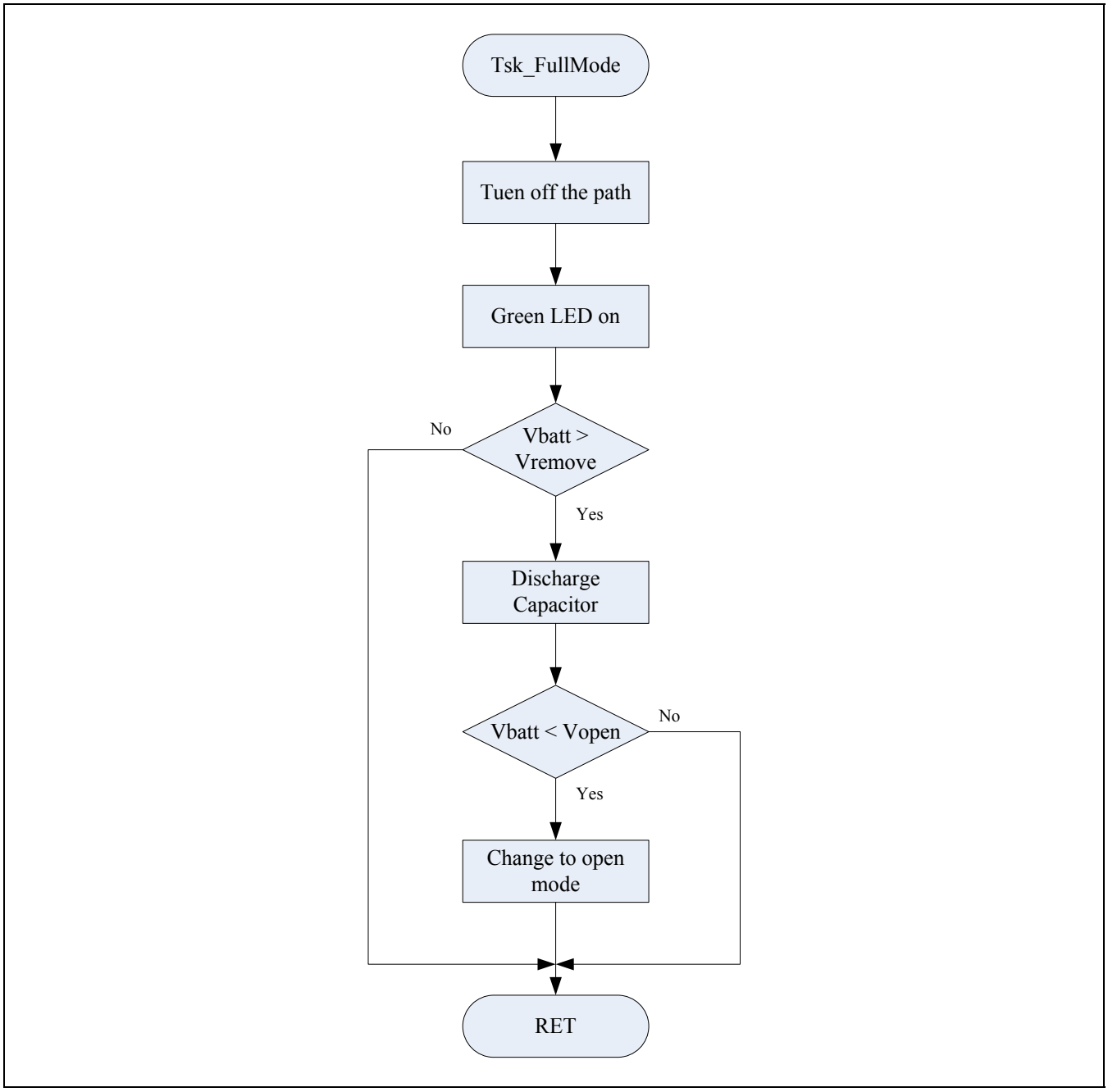

Figure 9. Task in Full Mode

3.1.9 Task fault mode

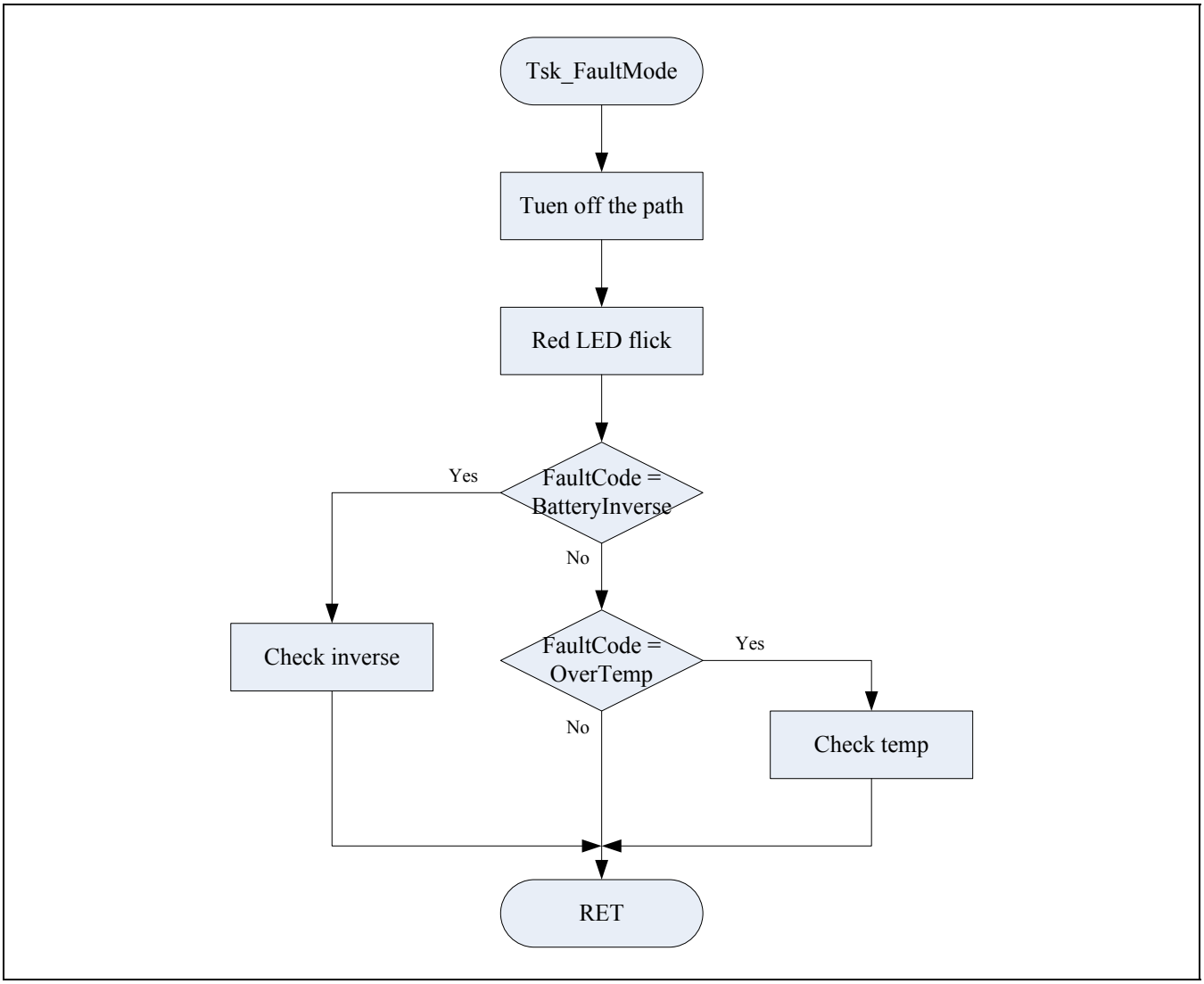

Figure 10. Task in Fault Mode

3.1.10 Util.src

This part is the collection of user functions.

Table 3. Functions Included in Util.src

Function	Description	Flags
ut_ReadAD	Read the AD convert result of current, temperature and voltage	
ut_ChkInverse	Check if the battery is inversed insert by check Vbatt 8 times	Mode, FaultCode
ut_ChkFull	Check if battery is full in CV mode by check Vbatt and Vcurrent 8 times	Mode
ut_ChkTempture	Check if the temperature is within the predefined range	Mode, FaultCode
ut_ChkChargingTime	Check if the charging time is within the predefined range	Mode
ut_IncChargeTime	Increase the charging time counter	
ut_ChkOverCurrent	Check if the charging current is within the predefined range	Mode, FaultCode
ut_CallTsk_MODE	Call task function referring to the current Mode	
ut_Delay_40us	Delay 40us @ 8MHz CPU clock	
ut_Delay_10ms	Delay 10ms @ 8MHz CPU clock	
ut_Delay_100ms	Delay 100ms @ 8MHz CPU clock	
ut_Delay_1s	Delay 1s @ 8MHz CPU clock	

3.1.11 DvDr.src

This part is the collection of I/O device drivers

Table 4. Functions Included in DvDr.src

Function	Description	Return
dr_ReadVbatt	Driver of read charge voltage and battery voltage. Read charge voltage first and then stop PWM and read battery voltage	Vcharge, Vbatt
dr_ReadVcurrent	Driver of read charge current, read 8 times and get average	Vcurrent
dr_ReadVtempture	Driver of read tempture	Vtempture
dr_DoutPathOff	Driver of turn off the charge path	
dr_DoutLEDRedOn	Driver of Bi-colour LED red on	
dr_DoutLEDGrnOn	Driver of Bi-colour LED green on	
dr_DoutLEDOff	Driver of Bi-colour LED off	
dr_DoutLEDFlick	Driver of Bi-colour LED flick	
dr_DoutLEDRedFlick	Driver of Bi-colour LED flick red	
dr_ModPWM	Driver to modulate PWM	

3.1.12 Var.src

This part declares all the variables.

Table 5. Functions Included in Var.src

Variable	Description	RAM Assignment (Hex)
Vbatt	Battery voltage	0
Vcharge	Charge voltage	1
Vcurrent	Charge current	2
Vtempture	Battery temperature	3
Mode	Charging mode	4
CurrentValue	PWM module function variable: the value right now	5
TargetValue	PWM module function variable: the target value	6
DurLEDFlick	LED flick counter	7
DurCvCurrentChk	Constant Voltage charging current check counter	8
FlagChkInverse	Flag of check inversed input of battery	9
FaultCode	Fault name	a
FlagChkFull	Flag of check battery full in CV mode	b
Dur_Charging_Hi	Charge time counter	c
Dur_Charging_Mid		d
Dur_Charging_Lo		e
DurADCRead	ADC read duration	10
DurADCDIV	ADC div duration	11
TempADCL	ADC read temporary	12
TempADCH		13

3.1.13 Defines.src

This part defines the constants.

Table 6. Functions Included in Defines.src

Constant	Description	Value
OPEN_MODE	Mode table	00H
CHRG_CC_MODE		01H
CHRG_CV_MODE		02H
FULL_MODE		03H
FAULT_MODE		04H
FAULT_BatteryInverse	Fault code table	00H
FAULT_OverCurrent		01H
FAULT_OverTempture		02H
LED_GRN_ON	LED display table	01H
LED_RED_ON		02H
LED_YEL_ON		03H
LED_GRN_OFF		FEH
LED_RED_OFF		FDH
LED_OFF		FCH
ADC_START_BATT	ADC control register settings	45H
ADC_START_CURR		35H
ADC_START_TEMP		25H
ADC_COMPLETION		08H
V_000 to V_500	ADC sample value table, stand for 0 to 5V	0 to FFH
V_CURRENT_MAX	Charge current limit	400mA
V_CURRENT_TARG		300mA
V_CURRENT_CV_END		20mA
V_OPEN	Charge voltage limit	2.5V
V_FULL		4.24V
V_REMOVE		4.5V
V_CC_END		4.2V
V_CV_END		4.22V
V_CV_TARG		4.2V
V_TEMPTURE_MAX	Battery temperature limit	50°C
V_TEMPTURE_MIN		30°C
DUR_LED_FLICK	LED flick time duration	1s
DUR_CV_CURRENT_CHK	Constant voltage charge current check time duration	1s
DUR_CHARGING_HI	Charge time limit, 4 hours	15H
DUR_CHARGING_MID		F9H
DUR_CHARGING_LO		00H

Constant	Description	Value
DURADCREADCURRENT	ADC read current time duration	08H
DURADCDIVCURRENT		03H

3.1.14 Const.src

This part is the table of task mode.

4 APPENDIX1: SCHEMATIC

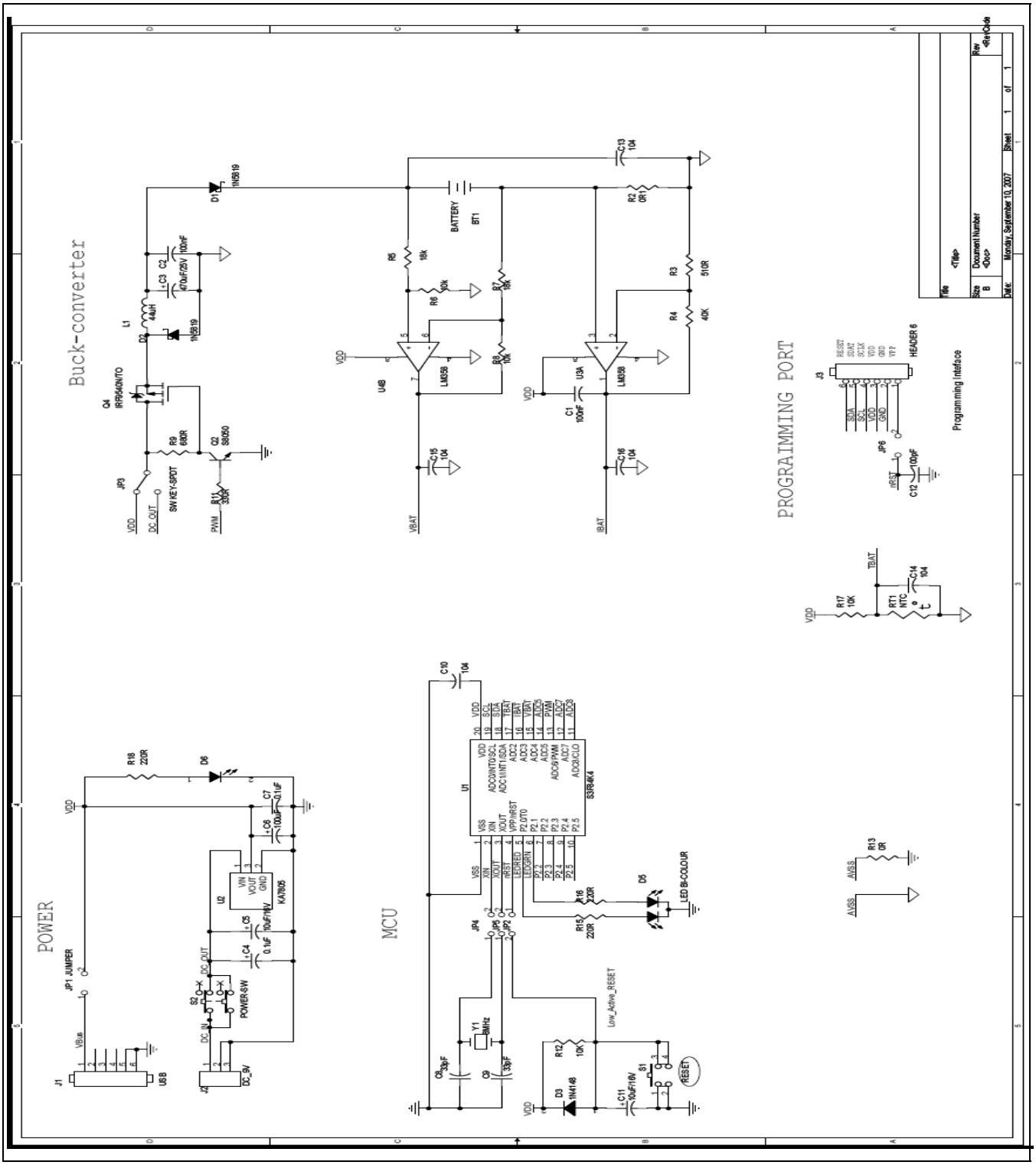

Figure 11. Schematic

5 APPENDIX2: BATTERY CHARGER TEST RESULT

5.1 TEST CONDITION

Table 7. Test Condition

		Test Condition	Note
Hardware		Battery charger ver.0.1	
Software		Battery charger basic version	
Equipment	Battery	500mAH Li-ion battery	
	Current measurement	34401A digital multimeter	Agilent
	Voltage measurement	TDS3034B digital phosphor oscillator	tektronix
	Power supply	LP-0520 AD/DC (100-240V AC input, 9V 1A DC output)	Powermax
Regulated parameters	Current	300mA	
	Voltage	4.2V	

5.2 TEST RESULT

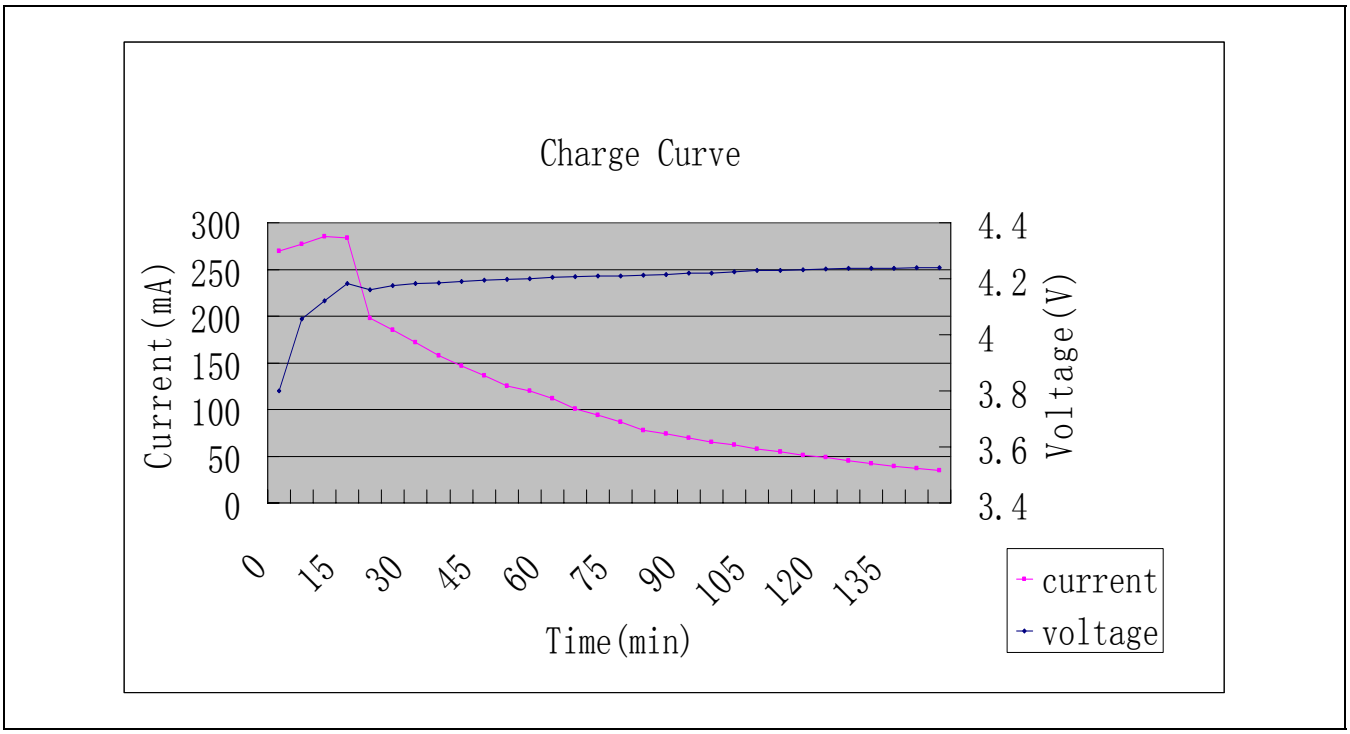

Figure 12. Charge Curve

Table 8. Test Result

No.	Time	Voltage(V)	Current(mA)
0	14：00	3.800	270
5	14：05	4.056	277
10	14：10	4.120	285
15	14：15	4.182	284
20	14：20	4.160	198
25	14：25	4.175	185
30	14：30	4.182	172
35	14：35	4.185	158
40	14：40	4.190	147
45	14：45	4.195	136
50	14：50	4.197	125
55	14：55	4.200	120
60	15：00	4.204	112
65	15：05	4.207	101
70	15：10	4.209	94
75	15：15	4.211	87
80	15：20	4.213	78
85	15：25	4.216	74
90	15：30	4.219	70
95	15：35	4.220	65
100	15：40	4.225	62
105	15：45	4.229	58
110	15：50	4.230	55
115	15：55	4.233	51
120	16：00	4.235	49
125	16：05	4.237	45
130	16：10	4.238	42
135	16：15	4.238	39
140	16：20	4.239	37
145	16：25	4.240	35

6 APPENDIX3: SOURCE CODE

```
;;*****************************************************************************
;;*                    Battery Charger Basic Version
;;*****************************************************************************
;;* File:              Start
;;* MCU:               S3F84K4
;;* Description:       Program start, initiate the MCU
;;* Date:              2007-7-4
;;* Modification:      2007-7-5
;;*****************************************************************************
;-----------------------------------------------------------------------------
; -- include files
;-----------------------------------------------------------------------------
.include "Var.src"
.include "Defines.src"
;-----------------------------------------------------------------------------
; -- inform that these modules are defined at other files in the same project
;-----------------------------------------------------------------------------
        .extern Int_PWM
        .extern Int_Tmr0_Match
        .extern cf_IoPort                ;config of each module
        .extern cf_Tmr0
        .extern cf_PWM
        .extern cf_ADC
        .extern ut_ReadAD
        .extern ut_Delay_1s
        .extern dr_DoutLEDRedFlick
;-----------------------------------------------------------------------------
; -- smart option
;-----------------------------------------------------------------------------
        .org    003CH
        .db     0FFH                    ; 003CH, must be initialized to 0
        .db     0FFH                    ; 003DH, must be initialized to 0
```

```
        .db     07FH                    ; 003EH, disable LVR (3.0v)
        ;.db    0FFH                    ; 003FH, External crystal
        .db     0FCH                    ; internal RC 8MHz
;-------------------------------------------------------------------------------
; -- Interrupt Vector Table
;-------------------------------------------------------------------------------
        .vector 0F2H,Int_PWM            ; IRQ2
        .vector 0F6H,Int_Tmr0_Match     ; IRQ0
;-------------------------------------------------------------------------------
; -- Initialize System and Peripherals
;-------------------------------------------------------------------------------
        .org 100H
RESET:
        DI                              ; disable interrupt
        LD      IMR,#11111111B
        LD      BTCON,#10100011B        ; Watch-dog disable
        LD      CLKCON,#00011000B       ; Select non-divided CPU clock
        LD      SPL,#0C0H               ; Stack pointer must be set
;-------------------------------------------------------------------------------
; -- To clear Internal Register(RAM range:00h ~ 0BFh)
;-------------------------------------------------------------------------------
        LD      r0,SPL
L_RamClr:
        DEC     r0
        CLR     @r0
        CP      r0,#00h
        JP      NE,L_RamClr
;-------------------------------------------------------------------------------
; -- To initiate I/O, Timer0, PWM, ADC
;-------------------------------------------------------------------------------
      CALL    cf_IoPort
    CALL    cf_Tmr0
    CALL    cf_PWM
```

```
  CALL    cf_ADC
;-------------------------------------------------------------------------------
; -- To initiate mode OPEN_MODE
;-------------------------------------------------------------------------------
    LD      Mode, #OPEN_MODE
    EI
;-------------------------------------------------------------------------------
; -- Main Loop, waiting for Timer interrupt
;-------------------------------------------------------------------------------
MAIN:
      NOP
      NOP
      NOP
      JP      T,MAIN
;-------------------------------------------------------------------------------
; -- End of Start.src
;-------------------------------------------------------------------------------
      .end
```

```
;;*****************************************************************************
;;*                     Battery Charger Basic Version
;;*****************************************************************************
;;* File:               Exception
;;* MCU:                S3F84K4
;;* Description: Exception handlers for Timer0 and PWM module
;;* Date:               2007-7-4
;;* Modification:
;;*****************************************************************************
;-----------------------------------------------------------------------------
; -- include files
;-----------------------------------------------------------------------------
.include "Var.src"
.include "Defines.src"
;-----------------------------------------------------------------------------
; -- inform that these modules are defined at other files in the same project
;-----------------------------------------------------------------------------
        .extern ut_ReadAD
        .extern ut_ChkInverse
        .extern ut_ChkOverCurrent
        .extern ut_ChkChargingTime
        .extern ut_ChkTempture
        .extern ut_CallTsk_MODE
;-----------------------------------------------------------------------------
; -- To allow global usage of following modules
;-----------------------------------------------------------------------------
        .global     Int_PWM
        .global Int_Tmr0_Match
;-----------------------------------------------------------------------------
; -- PWM Interrupt Service Routine
;       -- clear the interrupt pending bit
;-----------------------------------------------------------------------------
Int_PWM:
```

```
        AND     PWMCON,#11111110B   ;pending bit clear
        IRET
;-------------------------------------------------------------------------------
; -- Timer0 match interrupt service routine
;       -- clear the pending bit
;       -- read Vbatt, Vcurrent
;       -- call the task of the selected mode
;-------------------------------------------------------------------------------
Int_Tmr0_Match:
        CALL    ut_ReadAD               ; Read AD conversion result of Current,Voltage
        ;CALL    ut_ChkOverCurrent      ; Check the charging current
        CALL    ut_ChkTempture          ; check the tempture
        CALL    ut_ChkChargingTime      ; check the charging time
        CALL    ut_CallTsk_MODE         ; call task according to the mode selected
        AND     TACON,#11111110B        ; T0 pending bit clear
        IRET
;-------------------------------------------------------------------------------
; -- End of Exception.src
;-------------------------------------------------------------------------------
        .end
```

```
;;*****************************************************************************
;;*                    Battery Charger Basic Version
;;*****************************************************************************
;;* File:                 Util
;;* MCU:                  S3F84K4
;;* Description:          Collection of user functions
;;* Date:                 2007-7-4
;;* Modification:
;;*****************************************************************************
;-----------------------------------------------------------------------------
; -- include files
;-----------------------------------------------------------------------------
.include "Var.src"
.include "Defines.src"
;-----------------------------------------------------------------------------
; -- inform that these modules are defined at other files in the same project
;-----------------------------------------------------------------------------
        .extern dr_ReadVbatt
        .extern dr_ReadVcurrent
        .extern dr_ReadVtempture
        .extern dr_DoutPathOff
        .extern dr_DoutLEDOff
        .extern table_tsk_Mode
;-----------------------------------------------------------------------------
; -- To allow global usage of following modules
;-----------------------------------------------------------------------------
        .global ut_ReadAD
;       .global ut_SelMode
        .global ut_ChkInverse
        .global ut_ChkOverCurrent
        .global ut_ChkTempture
        .global ut_ChkChargingTime
        .global ut_ChkFull
```

```
        .global ut_IncChargeTime
        .global ut_CallTsk_MODE
        .global ut_Delay_40us
        .global ut_Delay_10ms
        .global ut_Delay_100ms
        .global ut_Delay_1s
;-------------------------------------------------------------------------
; -- Function:   read ADC
;              -- read Vbatt
;              -- read Vcurrent
;-------------------------------------------------------------------------
ut_ReadAD:
        CALL       dr_ReadVcurrent
        CALL    dr_ReadVtempture
        CALL       dr_ReadVbatt
        RET
;-------------------------------------------------------------------------
; -- Function:   check if the battery is inversed insert
;       -- check Vbatt 8 times
;       -- each time if Vbatt is '0', set the corresponding bit in the FlagChkInverse
;       -- otherwise clear the flag FlagChkInverse, turn Mode to Open_Mode
;       -- if Vbatt is '0' in 8 continuous check,the FlagChkInverse is '11111111'
;       -- then go to FAULT_MODE and set FaultCode
;-------------------------------------------------------------------------
ut_ChkInverse:
        CP      Vbatt, #0h
        JP      NE, L_ClrFlag
        RL      FlagChkInverse
        INC     FlagChkInverse
        CP      FlagChkInverse,#0ffh
        JP      EQ,L_ToFaultMode
        RET
```

```
L_ClrFlag:
        CLR     FlagChkInverse
        CLR     FaultCode
        LD      Mode, #OPEN_MODE
        RET
L_ToFaultMode:
        LD      Mode,#FAULT_MODE
        ;CLR     FlagChkInverse
        LD      FaultCode, #FAULT_BatteryInverse
        RET
;-------------------------------------------------------------------------------
; -- Function:  check battery is full or not in CV mode
;       -- check Vbatt and Vcurrent 8 times
;       -- each time if Vbatt > 4.24V && Vcurrent < 20mA,
;       -- set the corresponding bit in the FlagChkFull
;       -- otherwise clear the FlagChkFull, go on CV mode
;       -- if FlagChkFull is "11111111"
;       -- then battery is full , go to full mode
;-------------------------------------------------------------------------------
ut_ChkFull:
        CP      Vcurrent, #V_CURRENT_CV_END       ;20mA
        JP      UGE, L_ClrFullFlag
        CP      Vbatt, #V_CV_END          ; I<20mA && Vbatt > 4.24V
        JP      ULT, L_ClrFullFlag
        RL      FlagChkFull
        INC     FlagChkFull
        CP      FlagChkFull, #0ffh
        JP      EQ, L_ToFullMode
        RET
L_ClrFullFlag:
        CLR     FlagChkFull
        RET
```

```
L_ToFullMode:
        LD      Mode, #FULL_MODE
        CLR     FlagChkFull
        RET
;-------------------------------------------------------------------------------
; -- Function:  check the tempture
;       -- if tempture is above 50 C
;               -- turn off the path
;               -- go to fault mode
;       -- when in fault mode, check the tempture is below 30 C
;               -- if OK, turn to OPEN_MODE
;               -- if not, keep in FAULT_MODE
;-------------------------------------------------------------------------------
ut_ChkTempture:
        CP      Mode, #FAULT_MODE
        JP      NE, L_ChkMax
        CP      FaultCode, #FAULT_OverTempture
        JP      EQ, L_ChkMin
L_ChkMax:
        CP      Vtempture, #V_TEMPTURE_MAX
        JP      ULE, L_To_Fault
        RET
L_ChkMin:
        CP      Vtempture, #V_TEMPTURE_MIN
        JP      UGE, L_To_Open
L_To_Fault:
        CALL    dr_DoutPathOff
        LD      Mode, #FAULT_MODE
        LD      FaultCode, #FAULT_OverTempture
        RET
L_To_Open:
        CLR     FaultCode
        LD      Mode, #OPEN_MODE
```

```
        RET
;-------------------------------------------------------------------------------
; -- Function:  check the charging time
;       -- the whole charging time is set in 3 constants:
;               -- DUR_CHARGING_HI, DUR_CHARGING_MID, DUR_CHARGING_LO
;       -- if time is over then check the battery
;               -- battery voltage reached full mode => full_mode
;               -- battery voltage can't reach full mode => fault_mode
;-------------------------------------------------------------------------------
ut_ChkChargingTime:
        CP      Dur_Charging_Hi, #DUR_CHARGING_HI
        JP      NE, L_End_ChkChargingTime
        CP      Dur_Charging_Mid, #DUR_CHARGING_MID
        JP      NE, L_End_ChkChargingTime
        CP      Dur_Charging_Lo, #DUR_CHARGING_LO
        JP      NE, L_End_ChkChargingTime
 L_ChkFull:
        CP      Vbatt, #V_CV_END
        JP      ULT, L_ChkTime_To_Fault
        LD      Mode, #FULL_MODE
        RET
L_ChkTime_To_Fault:
        LD      Mode, #FAULT_MODE
L_End_ChkChargingTime:
        RET
;-------------------------------------------------------------------------------
; -- Function:  Increase the charging time, when in Cc and Cv charge mode
;-------------------------------------------------------------------------------
ut_IncChargeTime:
                INC  Dur_Charging_Lo
                CP   Dur_Charging_Lo,#00h
                JP   EQ,L_IncDurChrgMdbyte
                RET
```

```
L_IncDurChrgMdbyte:
                    INC  Dur_Charging_Mid
                    CP   Dur_Charging_Mid,#00h
                    JP   EQ,L_IncDurChrgHibyte
                    RET
L_IncDurChrgHibyte:
                    INC  Dur_Charging_Hi
                    RET
;-------------------------------------------------------------------------------
; -- Function:   check the charging current
;       -- if current is larger than 400mA
;             -- turn off the path
;             -- go to fault mode
;-------------------------------------------------------------------------------
ut_ChkOverCurrent:
      CP     Vcurrent, #V_CURRENT_MAX
      JP     ULT,   L_End_ChkOverCurrent
      CALL   dr_DoutPathOff
      LD     Mode, #FAULT_MODE
      LD     FaultCode, #FAULT_OverCurrent
L_End_ChkOverCurrent:
      RET
  ;-------------------------------------------------------------------------------
; -- Function:   select a mode from OPEN_MODE, CHRG_CC_MODE,
;             CHRG_CV_MODE or FULL_MODE
;             -- Vbatt < 3.0V              : OPEN_MODE (No battery insert)
;             -- 3.0V <= Vbatt < 4.2V      : CHRG_CC_MODE (constant current)
;             -- Vbatt >= 4.2V && I > 20mA  : CHRG_CV_MODE (constant voltage)
;             -- Vbatt >= 4.2V && I < 20mA  : FULL_MODE
;-------------------------------------------------------------------------------
; -- Function:   Call task function refering to the current Mode
;-------------------------------------------------------------------------------
ut_CallTsk_MODE:
```

```
                LD   r14,#0h                ;make the Mode index
                LD   r15,Mode
                RCF
                RL   r15
                LDC r0,#table_tsk_Mode[rr14]   ;take the address of task to be called from
                        ;table 'table_tsk_Mode'
                LDC r1,#table_tsk_Mode+1[rr14]
                CALL            @rr0
        RET
;-------------------------------------------------------------------------------
; -- Function:   Delay
;-------------------------------------------------------------------------------
ut_Delay_40us:
        LD      R12, #18
L_LOOP_D1:
        NOP
        NOP
        DJNZ    R12, L_LOOP_D1
        RET
;-------------------------------------------------------------------------------
ut_Delay_10ms:
        LD      R11, #250
L_LOOP_D2:
        CALL    ut_Delay_40us
        DJNZ    R11, L_LOOP_D2
        RET
;-------------------------------------------------------------------------------
ut_Delay_100ms:
        LD      R8, #10
L_LOOP_D4:
        CALL    ut_Delay_10ms
        DJNZ    R8, L_LOOP_D4
        RET
```

```
;-------------------------------------------------------------------------------
ut_Delay_1s:
        LD      R10, #100
L_LOOP_D3:
        CALL    ut_Delay_10ms
        DJNZ    R10, L_LOOP_D3
        RET
;-------------------------------------------------------------------------------
; -- end of Util.src
;-------------------------------------------------------------------------------
        .end
```

```
;;*****************************************************************************
;;*                    Battery Charger Basic Version
;;*****************************************************************************
;;* File:                 Config
;;* MCU:                  S3F84K4
;;* Description:          Config of IO, Timer, PWM and ADC
;;* Date:                 2007-7-4
;;* Modification:
;;*****************************************************************************
;-----------------------------------------------------------------------------
; -- include files
;-----------------------------------------------------------------------------
.include          "C:\OPENice\include\reg\S3F84K4.reg"
;-----------------------------------------------------------------------------
; -- To allow global usage of following modules
;-----------------------------------------------------------------------------
              .global cf_IoPort
              .global cf_Tmr0
              .global cf_PWM
              .global cf_ADC
              .extern ut_Delay_10ms
              .extern ut_Delay_100ms
              .extern ut_Delay_1s
;-----------------------------------------------------------------------------
; -- To config IO port
;-----------------------------------------------------------------------------
cf_IoPort:
;-----------------------------------------------------------------------------
;*            P0.7/ADC7(#12)  : push-pull - NOT USED(16-DIP has no this pin)
;*            P0.6/PWM(#13)   : PWM - Path On/Off
;*            P0.5/ADC5(#14)  : ADC - Vcf(C/F)
;*            P0.4/ADC4(#15)  : ADC - Vbatt(B+)
;*            P0.3/ADC3(#16)  : ADC - Vcurrent(current)
```

```
;*          P0.2/ADC2(#17)  : ADC - Vtemp(temperature)
;*          P0.1/SDA(#18)   : push-pull - Programming interface SDA
;*          P0.0/SCL(#19)   : push-pull - Programming interface SCL

      LD    P0CONH,#10101010b
      LD    P0CONL,#10101010b
      LD    P0, #0h
      CALL  ut_Delay_1s
      CALL  ut_Delay_1s
      CALL  ut_Delay_1s
     LD         P0CONH,#10011111b
     LD         P0CONL,#11111010b

;-------------------------------------------------------------------------
;*          P2.0(#05)     : push-pull - LED.Red On/Off
;*          P2.1(#06)     : push-pull - LED.Green On/Off
;*          P2.2(#07)     : push-pull - NOT USED
;*          P2.3(#08)     : push-pull - NOT USED
;*
;*          P2.4(#09)     : push-pull - NOT USED(16-DIP has no this pin)
;*          P2.5(#10)     : push-pull - NOT USED(16-DIP has no this pin)
;*          P2.6(#11)     : push-pull - NOT USED(16-DIP has no this pin)

       LD       P2CONH,#01001010b
      LD    P2CONL,#10101010b
      RET

;-------------------------------------------------------------------------------
; -- To config Timer0
;-------------------------------------------------------------------------------
cf_Tmr0:
       LD    TACON,#10100110b
```

```
;LD  TACON,#10000110b          ;Timer 0 Control register
                              ;.7 :Timer 0 operation mode selection bit:
                              ;          0 = Two 8-bit timers mode (Timer A/B)
                              ;          1 = One 16-bit timer mode (Timer 0)
                              ;.6 :Always "0"
                              ;.54:Timer 0 input clock selection bits
                              ;          00=fOSC/256
                              ;          01=fOSC/64
                              ;          10=fOSC/8
                              ;          11=fOSC/1
                              ;.3:Timer 0 counter clear bit
                              ;          0=No effect
                              ;          1=Clear the timer 0 counter(when write)
                              ;.2:Timer 0 counter run enable bit:
                              ;          0 = Disable counter running
                              ;          1 = Enable counter running
                              ;.1:Timer 0 Interrupt enable bit
                              ;          0=Disable T0 interrupt
                              ;          1=Enable T0 interrupt
                              ;.0:Timer 0 interrupt(T0INT,match/capture) pending bit
                              ;          0=No T0 interrupt pending(when reading)
                              ;          0=Clear T0 pending bit(when writing)
                              ;          1=T0 Interrupt is pending

    LD    TADATA,#10h
    LD    TBDATA,#27h    ;DATA/(fOSC/8) = 10ms -> DATA = 10000 (2710h)
                         ;TBDATA = 27h, TADATA=27h
                         ;fOSC = 8MHz
      RET

;---------------------------------------------------------------------------
; -- To config PWM
;---------------------------------------------------------------------------
```

```
cf_PWM:


      ;LD     PWMCON,#10000110b       ;64us, enable interrupt
                LD     PWMCON,#11000100b       ;PWM Control Register(130KHz PWM@8MHz CPU Clock)
                    ;.76 :PWM input clock selection bits
                    ;       00=fOSC/256
                    ;       01=fOSC/64
                    ;       10=fOSC/8
                    ;       11=fOSC/1
                    ;.54 :Not used for S3F84K4
                    ;.3  :PWM counter clear bit:
                    ;       0=No effect
                    ;       1=Clear the 12-bit up counter
                    ;.2  :PWM counter enable bit:
                    ;       0=Stop counter
                    ;       1=Start(Resume counting)
                    ;.1  :PWM counter interrupt enable bit:
                    ;       0=Disable PWM OVF interrupt
                    ;       1=Enable PWM OVF interrupt
                    ;.0  :PWM 12-bit OVF interrupt pending bit:
                    ;       0=No interrupt pending
                    ;       0=Clear pending condition(when write)
                    ;       1=Interrupt is pending


      LD      PWMEX,#00H
      LD      PWMDATA,#00H    ;PWM output always '0'
      RET


;-------------------------------------------------------------------------------
; -- To config ADC
;-------------------------------------------------------------------------------
cf_ADC:
                    ;No need
```

```
        RET
;-------------------------------------------------------------------------------
; -- end of Util.src
;-------------------------------------------------------------------------------
        .end
```

```
;;*****************************************************************************
;;*                    Battery Charger Basic Version
;;*****************************************************************************
;;* File:              DvDr
;;* MCU:               S3F84K4
;;* Description:       Collection of I/O device drivers
;;* Date:              2007-7-4
;;* Modification:
;;*****************************************************************************



;-------------------------------------------------------------------------------
; -- include files
;-------------------------------------------------------------------------------
.include "Var.src"
.include "Defines.src"


;-------------------------------------------------------------------------------
; -- To allow global usage of following modules
;-------------------------------------------------------------------------------
        .global     dr_ReadVbatt
        .global     dr_ReadVcurrent
        .global dr_ReadVtempture


        .global dr_DoutPathOff


        .global dr_DoutLEDRedOn
        .global dr_DoutLEDGrnOn
        .global dr_DoutLEDFlick
        .global dr_DoutLEDOff
        .global dr_DoutLEDRedFlick


        .global dr_ModPWM
```

```
        .extern ut_Delay_40us
        .extern ut_Delay_10ms
        .extern ut_Delay_1s

;-------------------------------------------------------------------------
; -- Driver of read Battery voltage
;       -- read charge voltage first and then stop PWM and read battery V
;       -- push P0CONH
;       -- change P0.6 to push-pull output to let PWM output '0'
;       -- read the AD convert result of Vbatt
;       -- pop the P0CONH
;-------------------------------------------------------------------------

dr_ReadVbatt:

        ;AND     PWMCON,#11111011B       ;stop PWM counter

                LD   ADCON,#ADC_START_BATT;start ADC of B+

L_VchargeEndOfConv:
                TM   ADCON,#ADC_COMPLETION              ;check bit of Eed of Conv.
                JR   Z,L_VchargeEndOfConv
                LD   Vcharge,ADDATAH


        PUSH    P0CONH
        AND     P0CONH,#11101111B       ;clr P0CONH.4
        OR      P0CONH,#00100000B       ;set P0CONH.5   to set P0.6 push-pull output

        AND     P0, #10111111B          ;Set P0.6 '0'
```

```
    NOP
    NOP
    NOP
    NOP
    NOP

    NOP
    NOP
    NOP
    NOP
    NOP

    ;AND    P0CONH,#11111110B      ;clr P0CONH.0
    ;OR     P0CONH,#00000010B      ;set P0CONH.1   to ser P0.4(Vbatt ADC in) to push-pull
                                   ;output to discharge the cap.
    ;CALL   ut_Delay_40us
    ;CALL   ut_Delay_40us
    ;CALL   ut_Delay_40us
    ;CALL   ut_Delay_40us
    ;CALL   ut_Delay_40us

    ;CALL   ut_Delay_40us
    ;CALL   ut_Delay_40us
    ;CALL   ut_Delay_40us
    ;CALL   ut_Delay_40us
    ;CALL   ut_Delay_40us

    ;CALL   ut_Delay_10ms
    ;CALL   ut_Delay_10ms          ;delay for cap diacharge

    ;LD     DurADCRead, #DURADCREADCURRENT
```

```
        LD          ADCON,#ADC_START_BATT       ;start ADC of B+

L_VbattEndOfConv:
                    TM   ADCON,#ADC_COMPLETION              ;check bit of Eed of Conv.
                    JR   Z,L_VbattEndOfConv

        ;ADD     TempADCL, ADDATAH
        ;ADC     TempADCH, #0h

        ;DEC     DurADCRead
        ;JP      NZ, L_VbattEndOfConv

        ;LD      DurADCDIV, #DURADCDIVCURRENT

L_VbattAvrg:
        ;clr     FLAGS
        ;RRC     TempADCH
        ;RRC     TempADCL

        ;DEC     DurADCDIV
        ;JP      NZ, L_VbattAvrg

        ;LD          Vbatt,TempADCL
        LD      Vbatt, ADDATAH
        POP     P0CONH




        ;CLR     TempADCH
        ;CLR     TempADCL
                    RET

;-------------------------------------------------------------------------------
```

```
; -- Driver of read charge current
;-------------------------------------------------------------------------------

dr_ReadVcurrent:

    LD      DurADCRead, #DURADCREADCURRENT

    LD          ADCON,#ADC_START_CURR      ;start ADC of Current

L_VcurrentEndOfConv:
                TM   ADCON,#ADC_COMPLETION          ;check bit of Eed of Conv.
                JR   Z,L_VcurrentEndOfConv

    ADD     TempADCL, ADDATAH
    ADC     TempADCH, #0h

    DEC     DurADCRead
    JP      NZ, L_VcurrentEndOfConv

    LD      DurADCDIV, #DURADCDIVCURRENT

L_VcurrentAvrg:

    ;LD      r0,TempADCL
    ;LD      r1,TempADCH

    ;DIV     rr0, #08h
    ;LD      Vcurrent, r0

    clr     FLAGS
    RRC     TempADCH
    RRC     TempADCL
```

```
        DEC     DurADCDIV
        JP      NZ, L_VcurrentAvrg

        LD          Vcurrent,TempADCL

        CLR     TempADCH
        CLR     TempADCL
                RET

;-------------------------------------------------------------------------
; -- Driver of read tempture
;-------------------------------------------------------------------------

dr_ReadVtempture:
        LD      ADCON,#ADC_START_TEMP   ; start ADC of tempture

L_VtemptureEndOfConv:
        TM          ADCON,#ADC_COMPLETION       ;check bit of Eed of Conv.
                    JR   Z,L_VtemptureEndOfConv
                    LD   Vtempture,ADDATAH
                    RET


;-------------------------------------------------------------------------
; -- Driver of turn off the charge path
;-------------------------------------------------------------------------

dr_DoutPathOff:

        ;LD      P0CONH,
        LD      PWMDATA, #0h
        RET
```

```
;-------------------------------------------------------------------------
; -- Driver of LED to turn on Red
;-------------------------------------------------------------------------

dr_DoutLEDRedOn:

                OR  P2,#LED_RED_ON          ;P2.1 is high
                ANDP2,#LED_GRN_OFF          ;P2.0 is low
                RET

;-------------------------------------------------------------------------
; -- Driver of LED to turn on Green
;-------------------------------------------------------------------------

dr_DoutLEDGrnOn:

    OR          P2,#LED_GRN_ON              ;P2.0 is high
                ANDP2,#LED_RED_OFF                  ;P2.1 is low
                RET

;-------------------------------------------------------------------------
; -- Driver of LED to turn off the LED
;-------------------------------------------------------------------------

dr_DoutLEDOff:
    AND     P2, #LED_OFF
    RET

;-------------------------------------------------------------------------
; -- Driver of LED to flick Green
;-------------------------------------------------------------------------

dr_DoutLEDFlick:
```

```
                INC  DurLEDFlick
                CP   DurLEDFlick,#DUR_LED_FLICK
                JP   UGE,L_LEDFlick
                RET

L_LEDFlick:

        TM      P2,#LED_GRN_ON
                JP   NZ,L_LEDGrnOn

        CALL    dr_DoutLEDGrnOn
                CLR DurLEDFlick
                RET
L_LEDGrnOn:
        CALL    dr_DoutLEDRedOn
                CLR DurLEDFlick
                RET

;-------------------------------------------------------------------------
; -- Driver of Red LED flick
;-------------------------------------------------------------------------

dr_DoutLEDRedFlick:
        XOR     P2,#LED_RED_ON              ;P2.1 is flick
                ANDP2,#LED_GRN_OFF          ;P2.0 is low
        RET

;-------------------------------------------------------------------------
; -- Driver to modulate PWM to make current value meet the tatget value
;-------------------------------------------------------------------------

dr_ModPWM:
```

```
        LD      r0,PWMDATA
        CP      CurrentValue,TargetValue

        JP      UGT, L_dec_PWM
L_inc_PWM:
        CP      r0, #3fh
        JP      UGE, L_max_PWM
        INC     r0
        JP      L_End_Mod

L_max_PWM:
        LD      r0,#3fh
        JP      L_End_Mod

L_dec_PWM:
        CP      r0, #01h
        JP      ULE, L_min_PWM
        DEC     r0
        JP      L_End_Mod

L_min_PWM:
        LD      r0,#00h

L_End_Mod:
        LD      PWMDATA,r0

        ;CALL    ut_Delay_1s
        ;CALL    ut_Delay_1s
       ;CALL    ut_Delay_1s

        RET
```

```
;-------------------------------------------------------------------------------
; -- end of DvDr.src
;-------------------------------------------------------------------------------
        .end
```

```
;;*****************************************************************************
;;*                    Battery Charger Basic Version
;;*****************************************************************************
;;* File:              Task
;;* MCU:               S3F84K4
;;* Description:       tasks for each Mode
;;* Date:              2007-7-4
;;* Modification:
;;*****************************************************************************



;-------------------------------------------------------------------------
; -- include files
;-------------------------------------------------------------------------
.include "Var.src"
.include "Defines.src"


;-------------------------------------------------------------------------
; -- To allow global usage of following modules
;-------------------------------------------------------------------------
        .global tsk_OpenMode
        .global tsk_Chrg_Cc_Mode
        .global tsk_Chrg_Cv_Mode
        .global tsk_FullMode
        .global tsk_FaultMode


;-------------------------------------------------------------------------
; -- inform that these modules are defined at other files in the same project
;-------------------------------------------------------------------------
        .extern ut_ChkInverse
        .extern ut_ChkTempture
        .extern ut_IncChargeTime
        .extern ut_ChkFull
```

```
        .extern dr_DoutPathOff
        .extern dr_DoutLEDRedOn
        .extern dr_DoutLEDGrnOn
        .extern dr_DoutLEDFlick
        .extern dr_DoutLEDOff
        .extern dr_DoutLEDRedFlick

        .extern dr_ModPWM
        .extern dr_ReadVbatt

        .extern ut_Delay_40us
        .extern ut_Delay_10ms
        .extern ut_Delay_100ms
        .extern ut_Delay_1s

;-------------------------------------------------------------------------------
;-------------------------------------------------------------------------------
; -- Tasks of each mode
; -- Open Mode:        Vbatt < 3.7V
; -- Chrg_Cc Mode:     3.7V <= Vbatt <= 4.16V
; -- Chrg_Cv Mode:     4.16V < Vbatt < 4.24V
; -- Full Mode:        Vbatt >= 4.24V
;-------------------------------------------------------------------------------
;-------------------------------------------------------------------------------



;-------------------------------------------------------------------------------
; -- Task open mode
;       -- turn off the path
;       -- LED RED on
;       -- Check battery inversed insert
;
```

```
; State machine:
;
;       -- If Vbatt < V_OPEN    => next mode is OPEN_MODE
;       -- If Vbatt > V_FULL    => next mode is FULL_MODE
;       -- Otherwise            => next mode is CHRG_CC_MODE
;-------------------------------------------------------------------------------

tsk_OpenMode:

        CALL    dr_DoutPathOff          ; Turn off the path
        CALL    dr_DoutLEDRedOn         ; LED red to indicate open mode

        CALL    ut_ChkInverse           ; Check if the battery is inversed insert

        CP      Vbatt, #V_OPEN
        JP      ULT, L_End_OpenMode

        CALL    dr_DoutLEDOff
        CP      Vbatt, #V_FULL
        JP      UGE, L_OpenToFull

        LD      Mode, #CHRG_CC_MODE
        JP      ULT, L_End_OpenMode


L_OpenToFull:
        LD      Mode, #FULL_MODE

L_End_OpenMode:
        RET


;-------------------------------------------------------------------------------
```

```
; -- Task charge constant current mode
;       -- modulate PWM to constant current (250mA)
;
; State machine:
;
;       -- If Vbatt < V_OPEN    => next mode is OPEN_MODE
;       -- If Vbatt > V_CC_END  => next mode is CHRG_CV_MODE
;       -- Otherwise            => next mode keeps unchange
;-------------------------------------------------------------------------------


tsk_Chrg_Cc_Mode:


        CP      Vbatt, #V_OPEN          ; Test if battery is removed
        JP      UGT, L_Cmp416           ; If removed change mode to open mode


L_Chrg_CcToOpen:
        LD      Mode, #OPEN_MODE
        JP      L_End_Chrg_Cc_Mode


L_Cmp416:
        CP      Vbatt, #V_CC_END        ;Test if battery Voltege is above 4.20V
        JP      ULE, L_ChargeCc



       ; PUSH    P0CONH
        ;AND     P0CONH,#11101111B       ;clr P0CONH.4
       ; OR      P0CONH,#00100000B       ;set P0CONH.5   to set P0.6 push-pull output
        ;CALL    ut_Delay_1s
        ;CALL    ut_Delay_1s
        ;CALL    ut_Delay_1s
       ;CALL    ut_Delay_1s
        ;CALL    ut_Delay_1s
        ;POP     P0CONH
```

```
        ;CALL    dr_ReadVbatt

        ;CP      Vbatt, #V_CC_END       ;Re test
        ;JP      ULT, L_End_Chrg_Cc_Mode

        LD      Mode, #CHRG_CV_MODE
        JP      L_End_Chrg_Cc_Mode

L_ChargeCc:
        LD      CurrentValue, Vcurrent  ; Set the modulate variable
        LD      TargetValue, #V_CURRENT_TARG     ; Target current value: 250mA
        CALL    dr_ModPWM

        CALL    dr_DoutLEDFlick

        CALL    ut_IncChargeTime

L_End_Chrg_Cc_Mode:

        RET

;-------------------------------------------------------------------------------
; -- Task charge constant voltage mode
;       -- modulate PWM to constant voltage (4.20V)
;
; State machine:
;
;       -- If Vbatt < V_OPEN                  => next mode is OPEN_MODE
;       -- If Current < 20mA & Vbatt > 4.24V    => next mode is full mode
;       -- Otherwise                          => next mode keeps unchange
;-------------------------------------------------------------------------------
```

```
tsk_Chrg_Cv_Mode:

        CP      Vbatt, #V_OPEN          ; Test if battery is removed
        JP      UGT, L_Chk_Current

L_Chrg_CvToOpen:
        LD      Mode, #OPEN_MODE
        JP      L_End_Chrg_Cv_Mode

L_Chk_Current:

        ;INC     DurCvCurrentChk
        ;CP      DurCvCurrentChk, #DUR_CV_CURRENT_CHK
        ;JP      ULE, L_ChargeCv

        CALL    ut_ChkFull

        ;CP      Vcurrent, #V_CURRENT_CV_END       ;20mA
        ;JP      UGE, L_ChargeCv

        ;CP      Vbatt, #V_CV_END          ; I<20mA && Vbatt > 4.24V
        ;JP      ULT, L_ChargeCv

        ;LD      Mode, #FULL_MODE
        ;CLR     DurCvCurrentChk
        ;JP      L_End_Chrg_Cv_Mode


        ;CP      Vbatt, #V_426
        ;JP      ULT, L_ChargeCv
        ;LD      Mode, #FULL_MODE
        ;JP      L_End_Chrg_Cv_Mode
```

```
L_ChargeCv:
        LD      CurrentValue, Vcharge    ; Set the modulate variable
        LD      TargetValue, #V_CV_TARG      ; Target voltage value: 4.2V

        CALL    dr_ModPWM

        CALL    dr_DoutLEDFlick
        CALL    ut_IncChargeTime

L_End_Chrg_Cv_Mode:
        RET

;-------------------------------------------------------------------------
; --  Task full mode
;       -- turn off the path
;       -- LED green on
;
; State machine:
;
;       -- if Vbatt < 3.70V            => next mode is open mode (battery is removed)
;       -- if Vbatt > 4.20V            => next mode keeps unchange
;       -- others                      => next mode is Cv mode
;-------------------------------------------------------------------------

tsk_FullMode:

        CALL    dr_DoutPathOff          ; Turn off the path
        CALL    dr_DoutLEDGrnOn         ; LED green to indicate full mode

        CP      Vbatt, #V_REMOVE
        JP      ULT, L_End_Full_Mode
```

```
L_Discharge:
        PUSH    P0CONH
        AND     P0CONH,#11101110B       ;clr P0CONH.4
        OR      P0CONH,#00100010B       ;set P0CONH.5   to set P0.6 push-pull output

        AND     P0, #10101111B          ;Set P0.6 '0', P0.4 '0'

        CALL    ut_Delay_10ms
        CALL    ut_Delay_10ms

        POP     P0CONH


        CP      Vbatt, #V_OPEN          ; Test if battery is removed
        JP      UGT, L_End_Full_Mode


L_FullToOpen:
        LD      Mode, #OPEN_MODE
        JP      L_End_Full_Mode

L_End_Full_Mode:
        RET


;-------------------------------------------------------------------------------
; -- Task Fault Mode
;       -- turn off the path
;       -- flick the red LED to indicate fault
;       -- check if battery is inversed insert
;       -- if invert the battery to the right connevtion, go to open mode
;-------------------------------------------------------------------------------
```

```
tsk_FaultMode:

        CALL    dr_DoutPathOff          ; Turn off the path
        CALL    dr_DoutLEDRedFlick      ; LED red flick to indicate fault mode

        CP      FaultCode, #FAULT_BatteryInverse
        JP      EQ, L_ChkInverse

        CP      FaultCode, #FAULT_OverTempture
        JP      EQ, L_ChkTempture

        RET

L_ChkInverse:
        CALL    ut_ChkInverse           ;
        RET

L_ChkTempture:
        CALL    ut_ChkTempture
        RET

;-------------------------------------------------------------------------------
; -- end of Task.src
;-------------------------------------------------------------------------------
        .end
```

```
;;*****************************************************************************
;;*                    Battery Charger Basic Version
;;*****************************************************************************
;;* File:              Var
;;* MCU:               S3F84K4
;;* Description:       To declear all variables
;;* Date:              2007-7-4
;;* Modification:
;;*****************************************************************************



;-----------------------------------------------------------------------------
; -- include files
;-----------------------------------------------------------------------------

.include "C:\OPENice\include\reg\S3F84K4.reg"



;-----------------------------------------------------------------------------
; -- To declear variables
;-----------------------------------------------------------------------------

        .ram_org            00h
        Vbatt:              .ram_ds 1       ; Battery voltage
        Vcharge:            .ram_ds 1       ; Charge voltage
        Vcurrent:           .ram_ds 1       ; Charge current
        Vtempture:          .ram_ds 1       ; Battery tempture


        .ram_org            04h
        Mode:               .ram_ds 1       ; Charging mode
        CurrentValue:       .ram_ds 1       ; PWM module function variable: the value right now
        TargetValue:        .ram_ds 1       ; PWM module function variable: the target value
```

```
DurLEDFlick:          .ram_ds 1      ; LED flick counter

.ram_org              08h
DurCvCurrentChk:      .ram_ds 1      ; Constant Voltage charging current check counter
FlagChkInverse:       .ram_ds 1      ; Flag of check inversed input of battery
FaultCode:            .ram_ds 1      ; Fault name
FlagChkFull:          .ram_ds 1      ; Flag of check battery full in CV mode

.ram_org              0ch
Dur_Charging_Hi:      .ram_ds 1      ; charge time counter
Dur_Charging_Mid:     .ram_ds 1
Dur_Charging_Lo:      .ram_ds 1

.ram_org              10h
DurADCRead:           .ram_ds 1      ; ADC read duration: 8 times
DurADCDIV:            .ram_ds 1      ; ADC div duration: 3 times
TempADCL:             .ram_ds 1      ; ADC read temp low byte
TempADCH:             .ram_ds 1      ; ADC read temp high byte

.ram_org              14h
```

```
;;*******************************************************************************
;;*                    Battery Charger Basic Version
;;*******************************************************************************
;;* File:              Defines
;;* MCU:               S3F84K4
;;* Description:       Define of constants
;;* Date:              2007-7-4
;;* Modification:
;;*******************************************************************************



;-------------------------------------------------------------------------------

                    .list off ; don't include in list file

;-------------------------------------------------------------------------------
; Mode table

OPEN_MODE:     .equ    0h
CHRG_CC_MODE:  .equ    1h
CHRG_CV_MODE:  .equ    2h
FULL_MODE:     .equ    3h
FAULT_MODE:    .equ    4h

;-------------------------------------------------------------------------------
; Fault Code Table

FAULT_BatteryInverse    .equ    1h
FAULT_OverCurrent       .equ    2h
FAULT_OverTempture      .equ    3h



;-------------------------------------------------------------------------------
```

```
;LED display

LED_GRN_ON:     .equ      00000001b          ;To Set P2.0
LED_RED_ON:     .equ      00000010b          ;To Set P2.1
LED_YEL_ON: .equ 00000011b          ;To Set P2.0-1

LED_GRN_OFF:    .equ      11111110b    ;To Clear P2.0
LED_RED_OFF:    .equ      11111101b          ;To Clear P2.1
LED_OFF:      .equ 11111100b          ;To Clear P2.0-1

;-------------------------------------------------------------------------
;used to start of ADC

ADC_START_BATT:          .equ      45h     ;ADC4 0100 0 10 1
ADC_START_CURR:          .equ      35h     ;ADC3 0011 0 10 1
ADC_START_TEMP:          .equ      25h     ;ADC2 0010 0 10 1

;-------------------------------------------------------------------------
;used to wait for completing of ADC

ADC_COMPLETION:          .equ      08h

;-------------------------------------------------------------------------
;                - ADC table
;- V_452 means 4520 mV

V_000          .equ 0          ;          0h
V_002          .equ 1          ;          1h
V_004          .equ 2          ;          2h
V_006          .equ 3          ;          3h
V_008          .equ 4          ;          4h
V_010          .equ 5          ;          5h
V_012          .equ 6          ;          6h
```

```
V_014   .equ 7    ;   7h
V_016   .equ 8    ;   8h
V_018   .equ 9    ;   9h
V_020   .equ 10   ;   Ah
V_022   .equ 11   ;   Bh
V_024   .equ 12   ;   Ch
V_026   .equ 13   ;   Dh
V_028   .equ 14   ;   Eh
V_030   .equ 15   ;   Fh
V_032   .equ 16   ;   10h
V_034   .equ 17   ;   11h
V_036   .equ 18   ;   12h
V_038   .equ 19   ;   13h
V_040   .equ 20   ;   14h
V_042   .equ 21   ;   15h
V_044   .equ 22   ;   16h
V_046   .equ 23   ;   17h
V_048   .equ 24   ;   18h
V_050   .equ 26   ;   19h
V_052   .equ 27   ;   1Ah
V_054   .equ 28   ;   1Bh
V_056   .equ 29   ;   1Ch
V_058   .equ 30   ;   1Dh
V_060   .equ 31   ;   1Eh
V_062   .equ 32   ;   1Fh
V_064   .equ 33   ;   20h
V_066   .equ 34   ;   21h
V_068   .equ 35   ;   22h
V_070   .equ 36   ;   23h
V_072   .equ 37   ;   24h
V_074   .equ 38   ;   25h
V_076   .equ 39   ;   26h
V_078   .equ 40   ;   27h
```

```
V_080     .equ 41     ;     28h
V_082     .equ 42     ;     29h
V_084     .equ 43     ;     2Ah
V_086     .equ 44     ;     2Bh
V_088     .equ 45     ;     2Ch
V_090     .equ 46     ;     2Dh
V_092     .equ 47     ;     2Eh
V_094     .equ 48     ;     2Fh
V_096     .equ 49     ;     30h
V_098     .equ 50     ;     31h
V_100     .equ 51     ;     33h
V_102     .equ 52     ;     34h
V_104     .equ 53     ;     35h
V_106     .equ 54     ;     36h
V_108     .equ 55     ;     37h
V_110     .equ 56     ;     38h
V_112     .equ 57     ;     39h
V_114     .equ 58     ;     3Ah
V_116     .equ 59     ;     3Bh
V_118     .equ 60     ;     3Ch
V_120     .equ 61     ;     3Dh
V_122     .equ 62     ;     3Eh
V_124     .equ 63     ;     3Fh
V_126     .equ 64     ;     40h
V_128     .equ 65     ;     41h
V_130     .equ 66     ;     42h
V_132     .equ 67     ;     43h
V_134     .equ 68     ;     44h
V_136     .equ 69     ;     45h
V_138     .equ 70     ;     46h
V_140     .equ 71     ;     47h
V_142     .equ 72     ;     48h
V_144     .equ 73     ;     49h
```

```
V_146   .equ 74   ;   4Ah
V_148   .equ 75   ;   4Bh
V_150   .equ 77   ;   4Ch
V_152   .equ 78   ;   4Dh
V_154   .equ 79   ;   4Eh
V_156   .equ 80   ;   4Fh
V_158   .equ 81   ;   50h
V_160   .equ 82   ;   51h
V_162   .equ 83   ;   52h
V_164   .equ 84   ;   53h
V_166   .equ 85   ;   54h
V_168   .equ 86   ;   55h
V_170   .equ 87   ;   56h
V_172   .equ 88   ;   57h
V_174   .equ 89   ;   58h
V_176   .equ 90   ;   59h
V_178   .equ 91   ;   5Ah
V_180   .equ 92   ;   5Bh
V_182   .equ 93   ;   5Ch
V_184   .equ 94   ;   5Dh
V_186   .equ 95   ;   5Eh
V_188   .equ 96   ;   5Fh
V_190   .equ 97   ;   60h
V_192   .equ 98   ;   61h
V_194   .equ 99   ;   62h
V_196   .equ 100  ;   63h
V_198   .equ 101  ;   64h
V_200   .equ 102  ;   66h
V_202   .equ 103  ;   67h
V_204   .equ 104  ;   68h
V_206   .equ 105  ;   69h
V_208   .equ 106  ;   6Ah
V_210   .equ 107  ;   6Bh
```

```
V_212       .equ 108    ;       6Ch
V_214       .equ 109    ;       6Dh
V_216       .equ 110    ;       6Eh
V_218       .equ 111    ;       6Fh
V_220       .equ 112    ;       70h
V_222       .equ 113    ;       71h
V_224       .equ 114    ;       72h
V_226       .equ 115    ;       73h
V_228       .equ 116    ;       74h
V_230       .equ 117    ;       75h
V_232       .equ 118    ;       76h
V_234       .equ 119    ;       77h
V_236       .equ 120    ;       78h
V_238       .equ 121    ;       79h
V_240       .equ 122    ;       7Ah
V_242       .equ 123    ;       7Bh
V_244       .equ 124    ;       7Ch
V_246       .equ 125    ;       7Dh
V_248       .equ 126    ;       7Eh
V_250       .equ 128    ;       7Fh
V_252       .equ 129    ;       80h
V_254       .equ 130    ;       81h
V_256       .equ 131    ;       82h
V_258       .equ 132    ;       83h
V_260       .equ 133    ;       84h
V_262       .equ 134    ;       85h
V_264       .equ 135    ;       86h
V_266       .equ 136    ;       87h
V_268       .equ 137    ;       88h
V_270       .equ 138    ;       89h
V_272       .equ 139    ;       8Ah
V_274       .equ 140    ;       8Bh
V_276       .equ 141    ;       8Ch
```

```
V_278       .equ 142    ;       8Dh
V_280       .equ 143    ;       8Eh
V_282       .equ 144    ;       8Fh
V_284       .equ 145    ;       90h
V_286       .equ 146    ;       91h
V_288       .equ 147    ;       92h
V_290       .equ 148    ;       93h
V_292       .equ 149    ;       94h
V_294       .equ 150    ;       95h
V_296       .equ 151    ;       96h
V_298       .equ 152    ;       97h
V_300       .equ 153    ;       99h
V_302       .equ 154    ;       9Ah
V_304       .equ 155    ;       9Bh
V_306       .equ 156    ;       9Ch
V_308       .equ 157    ;       9Dh
V_310       .equ 158    ;       9Eh
V_312       .equ 159    ;       9Fh
V_314       .equ 160    ;       A0h
V_316       .equ 161    ;       A1h
V_318       .equ 162    ;       A2h
V_320       .equ 163    ;       A3h
V_322       .equ 164    ;       A4h
V_324       .equ 165    ;       A5h
V_326       .equ 166    ;       A6h
V_328       .equ 167    ;       A7h
V_330       .equ 168    ;       A8h
V_332       .equ 169    ;       A9h
V_334       .equ 170    ;       AAh
V_336       .equ 171    ;       ABh
V_338       .equ 172    ;       ACh
V_340       .equ 173    ;       ADh
V_342       .equ 174    ;       AEh
```

```
V_344   .equ 175   ;   AFh
V_346   .equ 176   ;   B0h
V_348   .equ 177   ;   B1h
V_350   .equ 179   ;   B2h
V_352   .equ 180   ;   B3h
V_354   .equ 181   ;   B4h
V_356   .equ 182   ;   B5h
V_358   .equ 183   ;   B6h
V_360   .equ 184   ;   B7h
V_362   .equ 185   ;   B8h
V_364   .equ 186   ;   B9h
V_366   .equ 187   ;   BAh
V_368   .equ 188   ;   BBh
V_370   .equ 189   ;   BCh
V_372   .equ 190   ;   BDh
V_374   .equ 191   ;   BEh
V_376   .equ 192   ;   BFh
V_378   .equ 193   ;   C0h
V_380   .equ 194   ;   C1h
V_382   .equ 195   ;   C2h
V_384   .equ 196   ;   C3h
V_386   .equ 197   ;   C4h
V_388   .equ 198   ;   C5h
V_390   .equ 199   ;   C6h
V_392   .equ 200   ;   C7h
V_394   .equ 201   ;   C8h
V_396   .equ 202   ;   C9h
V_398   .equ 203   ;   CAh
V_400   .equ 204   ;   CCh
V_402   .equ 205   ;   CDh
V_404   .equ 206   ;   CEh
V_406   .equ 207   ;   CFh
V_408   .equ 208   ;   D0h
```

```
V_410   .equ 209  ;  D1h
V_412   .equ 210  ;  D2h
V_414   .equ 211  ;  D3h
V_416   .equ 212  ;  D4h
V_418   .equ 213  ;  D5h
V_420   .equ 214  ;  D6h
V_422   .equ 215  ;  D7h
V_424   .equ 216  ;  D8h
V_426   .equ 217  ;  D9h
V_428   .equ 218  ;  DAh
V_430   .equ 219  ;  DBh
V_432   .equ 220  ;  DCh
V_434   .equ 221  ;  DDh
V_436   .equ 222  ;  DEh
V_438   .equ 223  ;  DFh
V_440   .equ 224  ;  E0h
V_442   .equ 225  ;  E1h
V_444   .equ 226  ;  E2h
V_446   .equ 227  ;  E3h
V_448   .equ 228  ;  E4h
V_450   .equ 230  ;  E5h
V_452   .equ 231  ;  E6h
V_454   .equ 232  ;  E7h
V_456   .equ 233  ;  E8h
V_458   .equ 234  ;  E9h
V_460   .equ 235  ;  EAh
V_462   .equ 236  ;  EBh
V_464   .equ 237  ;  ECh
V_466   .equ 238  ;  EDh
V_468   .equ 239  ;  EEh
V_470   .equ 240  ;  EFh
V_472   .equ 241  ;  F0h
V_474   .equ 242  ;  F1h
```

```
V_476           .equ 243        ;           F2h
V_478           .equ 244        ;           F3h
V_480           .equ 245        ;           F4h
V_482           .equ 246        ;           F5h
V_484           .equ 247        ;           F6h
V_486           .equ 248        ;           F7h
V_488           .equ 249        ;           F8h
V_490           .equ 250        ;           F9h
V_492           .equ 251        ;           FAh
V_494           .equ 252        ;           FBh
V_496           .equ 253        ;           FCh
V_498           .equ 254        ;           FDh
V_500           .equ 255        ;           FFh


;-------------------------------------------------------------------------
; Current limit,

;V_CURRENT_LIMIT .equ    0a1h      ;398mA (close to 400mA), V_316, 3.16V

V_CURRENT_MAX         .equ    V_318         ; charge max current: 400mA

;V_CURRENT_TARG        .equ    V_198         ; charge current: 250mA
V_CURRENT_TARG        .equ    V_238         ; charge current: 300mA
;V_CURRENT_TARG        .equ    V_318         ; charge current: 400mA
;V_CURRENT_TARG        .equ    V_396         ; charge current: 500mA

;V_CURRENT_MAX         .equ    V_400         ; charge max current: 504mA


V_CURRENT_CV_END      .equ    V_158         ; cv mode end current: 20mA
;V_CURRENT_CV_END      .equ    V_080         ; cv mode end current: 10mA
```

```
;-------------------------------------------------------------------------------
; Voltage Limit
V_OPEN          .equ    V_250           ; Vbatt < V_OPEN  => open mode
V_FULL          .equ    V_424           ; Vbatt > V_FULL  => full mode
V_REMOVE        .equ    V_450           ; Vbatt > V_REMOVE => battery is remove
V_CC_END        .equ    V_420           ; Vbatt > V_CC_END => CV  mode
V_CV_END        .equ    V_422           ; Vbatt > V_CV_END => full mode
V_CV_TARG       .equ    V_420           ; Vcharge target value in cv mode

;-------------------------------------------------------------------------------
; Temptrue Limit
V_TEMPTURE_MAX  .equ    V_144           ; tempture > 50C: stop charging, to fault mode
V_TEMPTURE_MIN  .equ    V_206           ; tempture < 30C: back to open mode



;-------------------------------------------------------------------------------

DUR_LED_FLICK:          .equ    100             ;10ms * 100 = 1s
DUR_CV_CURRENT_CHK:     .equ    100             ;10ms * 100 = 1s

;DUR_LED_FLICK:         .equ    1               ;1s
;DUR_CV_CURRENT_CHK:    .equ    1               ;1s

;-------------------------------------------------------------------------------
; Charging Time Limit
; Set to 4h
; 4*3600 = 14400s
; 14400/0.01 = 1440000 cycle (10ms/cycle)
; 1440000 = 23280h
DUR_CHARGING_HI:        .equ    15h
DUR_CHARGING_MID:       .equ    F9h
```

```
DUR_CHARGING_LO:        .equ    00h



;-------------------------------------------------------------------------------
; ADC Read Duration

DURADCREADCURRENT:      .equ    8               ;8 times ADC sample
DURADCDIVCURRENT:       .equ    3


        .list on; include in list file
```

```
;;*****************************************************************************
;;*                    Battery Charger Basic Version
;;*****************************************************************************
;;* File:              Const
;;* MCU:               S3F84K4
;;* Description:       Collection of Tables
;;* Date:              2007-7-4
;;* Modification:
;;*****************************************************************************


;-----------------------------------------------------------------------------
; -- include files
;-----------------------------------------------------------------------------
.include           "C:\OPENice\include\reg\S3F84K4.reg"



;-----------------------------------------------------------------------------
; -- inform that these modules are defined at other files in the same project
;-----------------------------------------------------------------------------

                   .extern         tsk_OpenMode
                   .extern         tsk_Chrg_Cc_Mode
        .extern tsk_Chrg_Cv_Mode
                   .extern         tsk_FullMode
        .extern tsk_FaultMode



;-----------------------------------------------------------------------------
; -- To allow global usage of following modules
;-----------------------------------------------------------------------------

                   .global         table_tsk_Mode
```

```
;-------------------------------------------------------------------------
;                   - We call one of the following task functions according to 'Mode'
;
;ut_CallTsk_MODE:
;
;;                  - To make the Mode index
;                   LD   r14,#0h
;                   LD   r15,Mode
;                   RCF
;                   RL   r15
;
;                   - To take the address of task to be called from table 'table_tsk_Mode'
;                   LDC r0,#table_tsk_Mode[rr14]
;                   LDC r1,#table_tsk_Mode+1[rr14]
;
;                   CALL            @rr0
;
;                   RET
;
table_tsk_Mode:



                    .dw  tsk_OpenMode
                    .dw  tsk_Chrg_Cc_Mode
        .dw     tsk_Chrg_Cv_Mode
                    .dw  tsk_FullMode
        .dw     tsk_FaultMode



;-------------------------------------------------------------------------
; -- End of Const.src
;-------------------------------------------------------------------------
                    .end ;End of ConstA.asm
```