Document Number:WSSCUG
Rev 1, 10/2013

Watt Saver Software Component (WSC)

Contents

1 Introduction

This document describes the basic steps for getting started with the Watt Saver Component (WSC) using Freescale's Processor Expert Software.

Freescale Semiconductor is finding innovative ways to stop "vampire" energy loss. With Watt Saver, an AC powered device or equipment can achieve zero standby power consumption. The Watt Saver solution can be used in the industrial and consumer electronics field to save standby power consumption in devices such as AC adapters for mobile phone battery chargers, AC adapters for laptop computer/ tablet battery chargers, and so on.

Freescale's Watt Saver technology consists of patent-pending hardware and software. The Watt Saver software component is a Processor Expert software package. You modify settings in Processor Expert and it generates the configuration files required. This enables easy and fast project development by the customer. After the component is installed, the designer can quickly complete the project development by configuring some parameters.

You can download WSC from the link below:

www.freescale.com/WattSaver

2 How to use the WSC component

The Watt Saver component is designed for a 5 W mobile phone charger application based on the following hardware platform and software (SW) tools. You can modify the related hardware circuit for different power requirements.

Hardware (HW) environment
MCU — MC9RS08
Demo board — WATTSAVER-EVB
Debug tool — PE USB Multilink

Software (SW) environment
CodeWarrior 10.2 (CW)
CodeWarrior 6.3

To use the software component, follow the steps:

1. Install the Watt Saver component.
2. Set the WSC properties according to the customer design.
3. Generate Processor Expert code using CodeWarrior tools.
4. Implement user application specific code, build, and then burn the code to the target board.

2.1 Creating a new project named *Demo*

1. In the CodeWarrior, click on the command New / Bareboard Project... in the menu File in the CodeWarrior window to create a new project.
2. The Project Wizard dialog window appears. Enter the name of the project —- Demo. Click on the Next button.
3. Select RS08 / RS08KA Family / MC9RS08KA4. Click on the Next button.
4. P&E Universal / USB Multilink is set as the default connection. Click on the Next button.
5. Skip addition of existing files by clicking on the Next button.
6. C is set in Languages dialog. Continue by clicking on the Next button.
7. Keep default settings. Continue by clicking on the Next button.
8. Select the Processor Expert from Rapid Application Development and click on the Next button.
9. Select the MC9RS08KA4CWG in MCU Pin variants and the Release for MC9RS08KA4 in Configurations. Click on the Finish button.

The new empty project is created and ready for adding new components.

2.2 Install WSC in CodeWarrior 10.2

The Watt Saver component is released as the file WSC_PE.PEupd. To install this file, follow the directions below.

1. First, open the PE components Library view by selecting CW menu: Processor Expert ◊ Show Views.
2. Use the Processor Expert ◊ Import Package command, refer to Figure 1

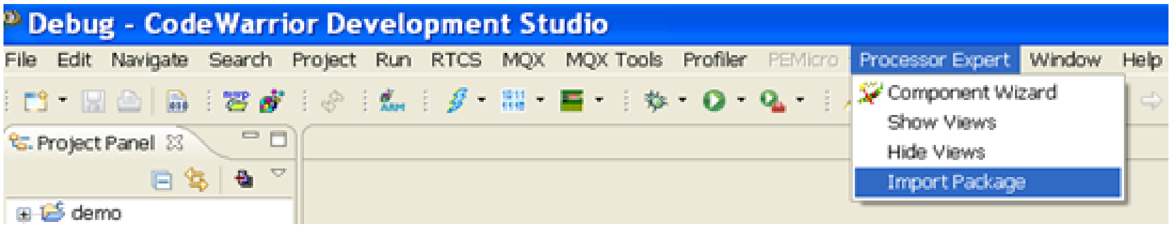
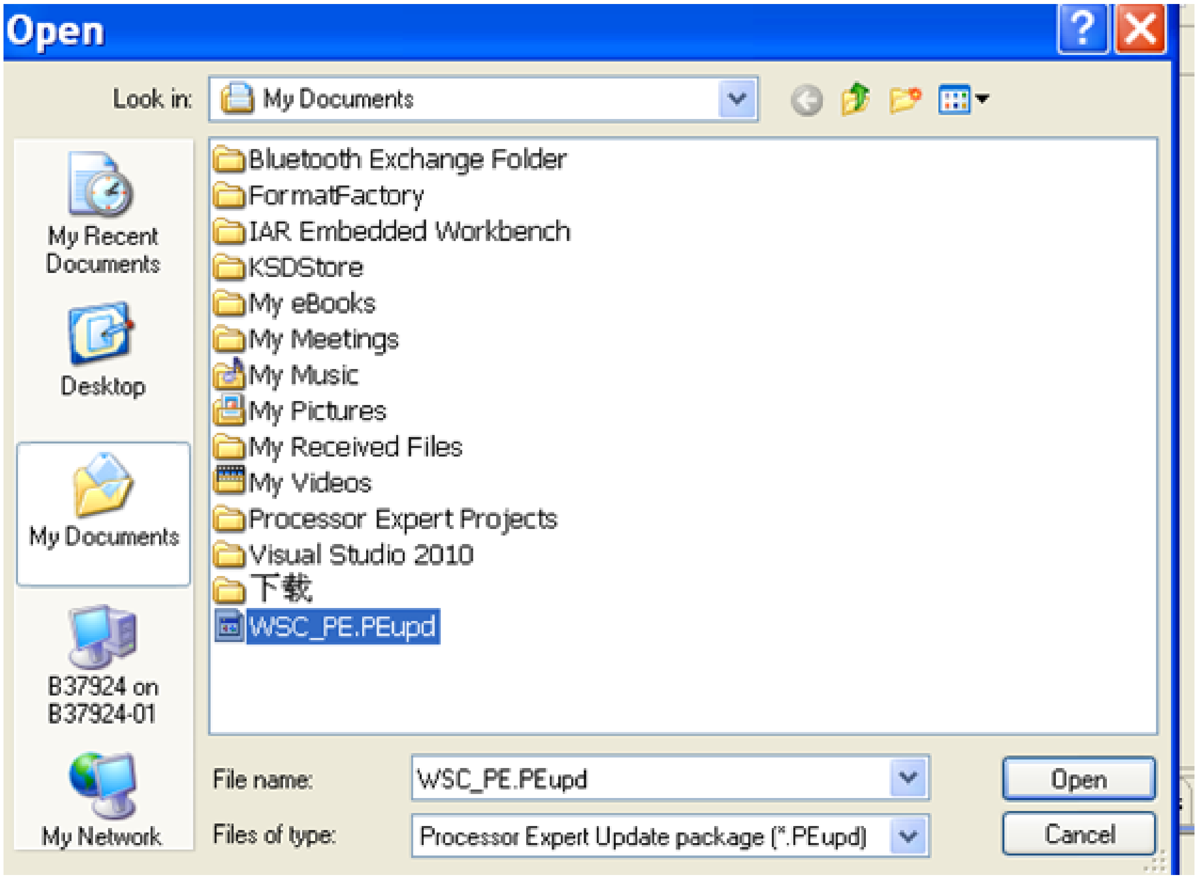

Debug - CodeWarrior Development Studio
File Edit Navigate Search Project Run RTCS MQX MQX Tools Profiler PEMicro Processor Expert Window Help
Component Wizard
Show Views
Hide Views
Import Package
Project Panel
demo

Figure 1. Import package

3. Select the file WSC_PE.PEupd and click the Open button, refer to Figure 2



File name: WSC_PE.PEupd

Files of type: Processor Expert Update package (*.PEupd)

Figure 2. Select WSC component

4. After installing the WSC, it can be found in the User Components of the SW group, in the Processor Expert Components Library. If it does not appear, right click in the library area and choose “Refresh.” , refer to Figure 3

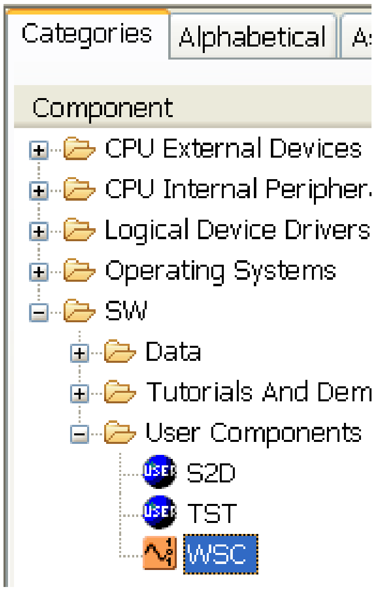


Figure 3. WSC component in PE Components Library

5. You can get component help by right clicking the WSC component , refer to Figure 4

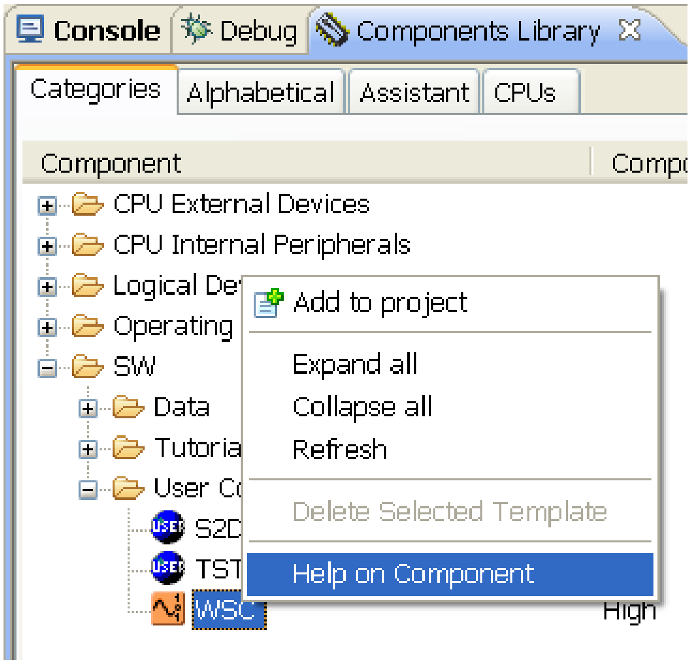


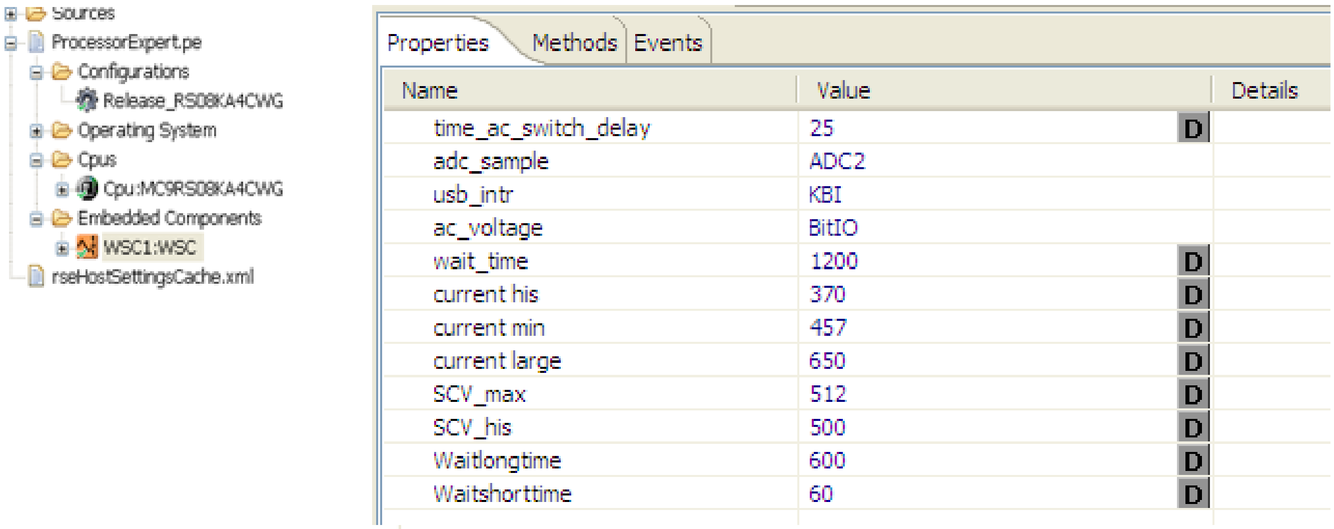
Figure 4. Show help doc

6. Add WSC to project Refer to Figure 4, select Add to project

The components are now visible in the Component folder in the Project Panel

2.3 Configure properties

After the component is in the project, you can select the component and modify the properties of the component, refer to Figure 5



- Sources
- ProcessorExpert.pe
  - Configurations
    - Release_RS08KA4CWG
  - Operating System
  - Cpus
    - Cpu:MC9RS08KA4CWG
  - Embedded Components
    - WSC1:WSC
- rseHostSettingsCache.xml

Name	Value		Details
time_ac_switch_delay	25	D	
adc_sample	ADC2		
usb_intr	KBI		
ac_voltage	BitIO		
wait_time	1200	D	
current his	370	D	
current min	457	D	
current large	650	D	
SCV_max	512	D	
SCV_his	500	D	
Waitlongtime	600	D	
Waitshorttime	60	D	

Figure 5. WSC properties

The default value in the following Table 1 and Table 2 is typical for 5 W AC adapter of mobile phone battery charger

Table 1. Software related properties class1

Properties	Properties	Description
wait_time	1200 s	Maximum (Max) time before timer expires during wait mode. .
current his	370	If sampling current less than this value, battery is fully charged.
current min	457	If sampling current more than this value, the battery is charging.
current large	650	If sampling current more than this value, battery is almost fully discharged. Charging is ongoing

NOTE

current_large > current_min > current_his

Table 2. Software related properties class2

Properties	Default	Description
time_ac_switch_delay	25 s	Delay for a while to avoid fast switching between AC mode and Super Cap mode for relay stability.
SCV_max	512	ADC value — Super Cap voltage max value. If sampling super cap voltage more than this value, super cap is fully charged.
SCV_his	500	ADC value — If sampling super cap voltage less than this value, super cap needs to be charged

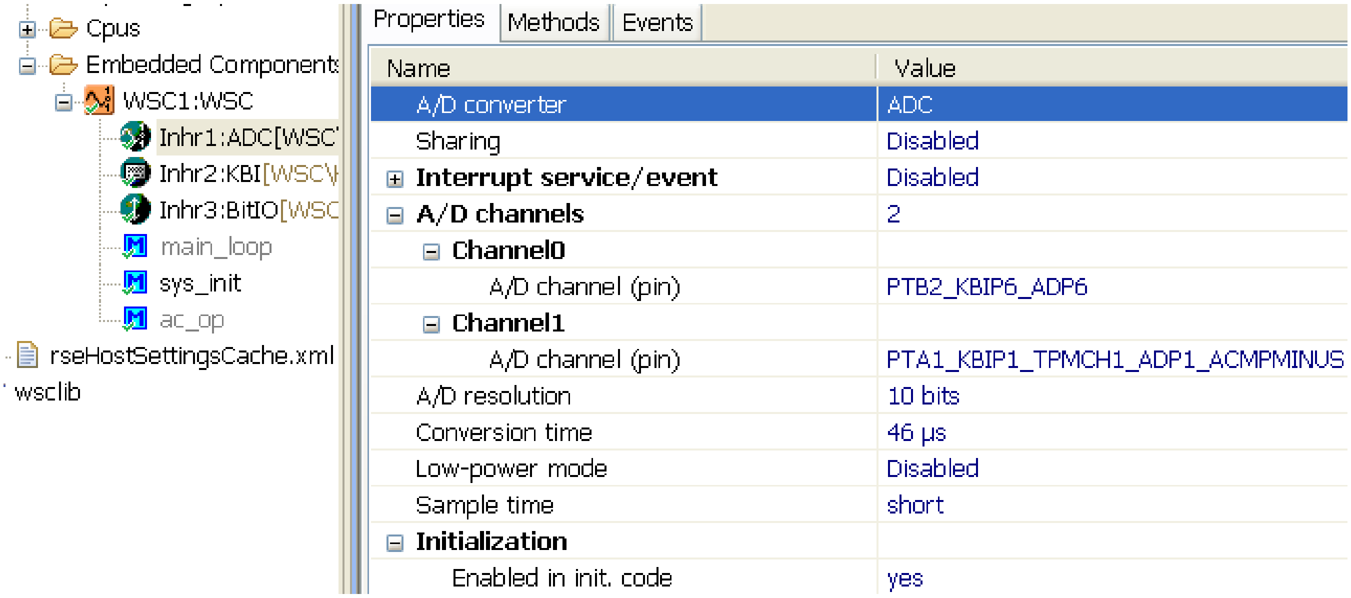
Table continues on the next page...

Table 2. Software related properties class2 (continued)

Properties	Default	Description
Waitlongtime	600s	If sampling current is greater than or equal to the value of current large property, the MCU will enter stop mode until Waitlongtime second expires.
Waitshorttime	60	If sampling current is greater than value of current min property and less than the value of Current large property the MCU will enter stop mode until Waitshorttime second expires.

Example— Please set related properties as shown in the following figures.

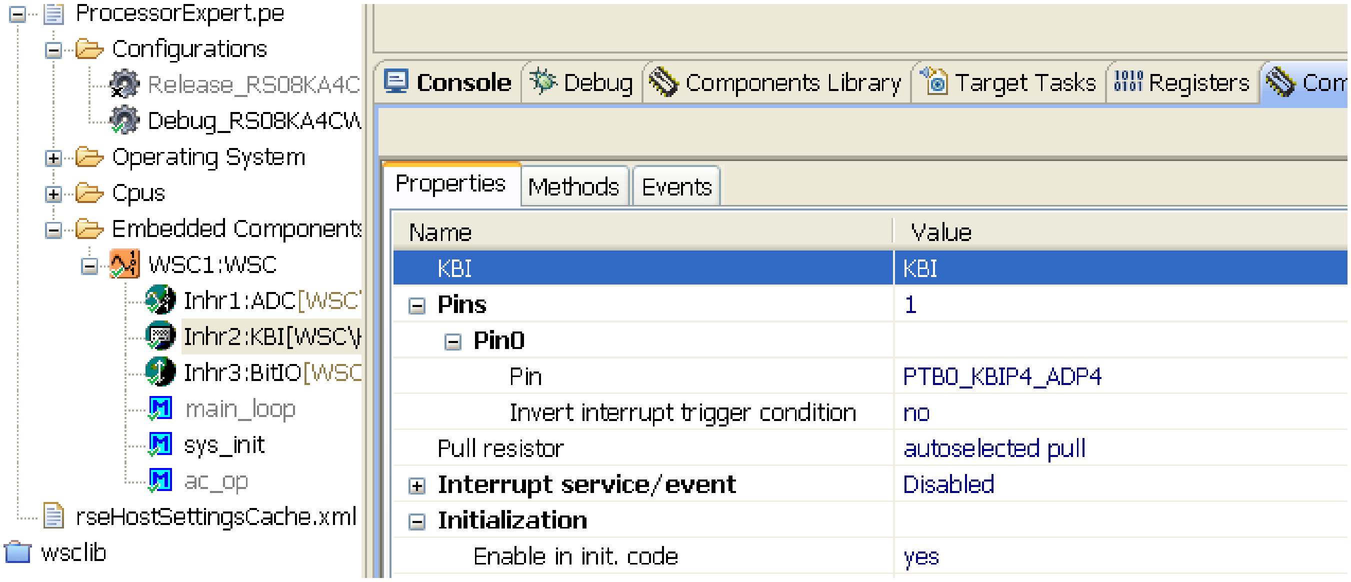
Adc_sample configure, refer to Figure 6.



Name	Value
A/D converter	ADC
Sharing	Disabled
Interrupt service/event	Disabled
A/D channels	2
Channel0	
A/D channel (pin)	PTB2_KBIP6_ADP6
Channel1	
A/D channel (pin)	PTA1_KBIP1_TPMCH1_ADP1_ACMPMINUS
A/D resolution	10 bits
Conversion time	46 µs
Low-power mode	Disabled
Sample time	short
Initialization	
Enabled in init. code	yes

Figure 6. Set adc_sample properties

Usb_intr configure— Refer to Figure 7

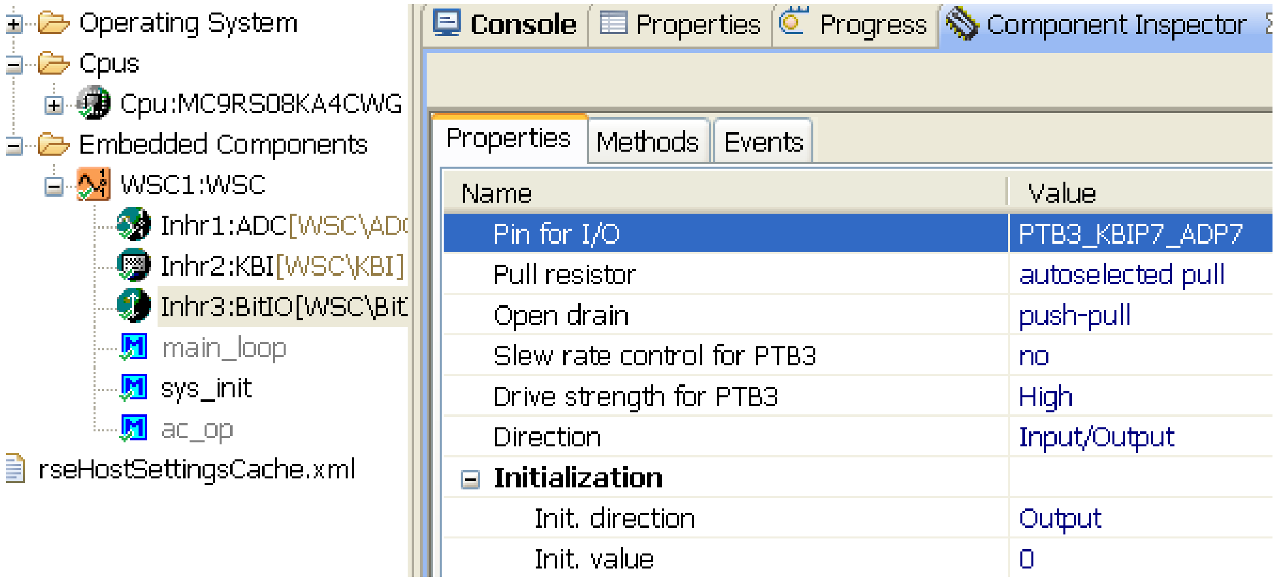


ProcessorExpert.pe
Configurations
Release_RS08KA4C
Debug_RS08KA4CW
Operating System
Cpus
Embedded Components
WSC1:WSC
Inhr1:ADC[WSC
Inhr2:KBI[WSC\
Inhr3:BitIO[WSC
main_loop
sys_init
ac_op
rseHostSettingsCache.xml
wsclib

Name	Value
KBI	KBI
Pins	1
Pin0	
Pin	PTB0_KBIP4_ADP4
Invert interrupt trigger condition	no
Pull resistor	autoselected pull
Interrupt service/event	Disabled
Initialization	
Enable in init. code	yes

Figure 7. Set usb_intr properties

Ac_voltage configure, refer to Figure 8



Operating System
Cpus
Cpu:MC9RS08KA4CWG
Embedded Components
WSC1:WSC
Inhr1:ADC[WSC\ADC
Inhr2:KBI[WSC\KBI]
Inhr3:BitIO[WSC\Bit
main_loop
sys_init
ac_op
rseHostSettingsCache.xml

Name	Value
Pin for I/O	PTB3_KBIP7_ADP7
Pull resistor	autoselected pull
Open drain	push-pull
Slew rate control for PTB3	no
Drive strength for PTB3	High
Direction	Input/Output
Initialization	
Init. direction	Output
Init. value	0

Figure 8. Set ac_voltage properties

2.4 MCU and project configuration

The CPU correct configuration and library path settings are explained in this section.

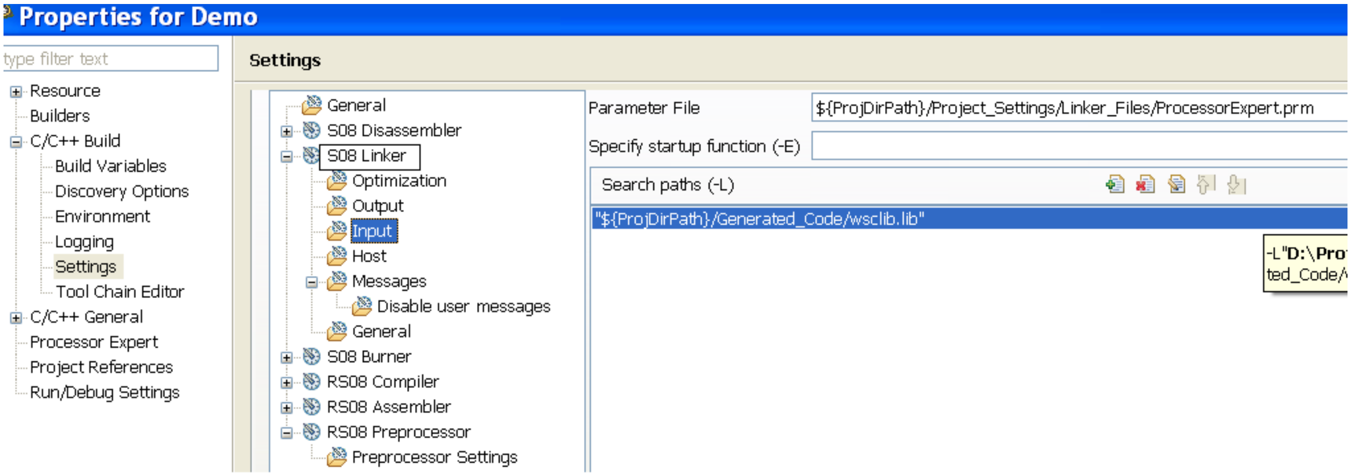
2.4.1 WSC component lib path setting

Before settings, put CodeWarrior IDE window focus on the project (select project name like demo in the Project Panel group).

Refer to Figure 9 for WSC component lib path setting

Project → Properties → C/C++ Build → Settings → Tool Setting → S08 Linker → input → click “+” button on ‘Libraries’ box.

Add lib path— "${ProjDirPath}/Generated_Code/wsclib.lib"



Parameter File: ${ProjDirPath}/Project_Settings/Linker_Files/ProcessorExpert.prm

Specify startup function (-E):

Search paths (-L)

"${ProjDirPath}/Generated_Code/wsclib.lib"

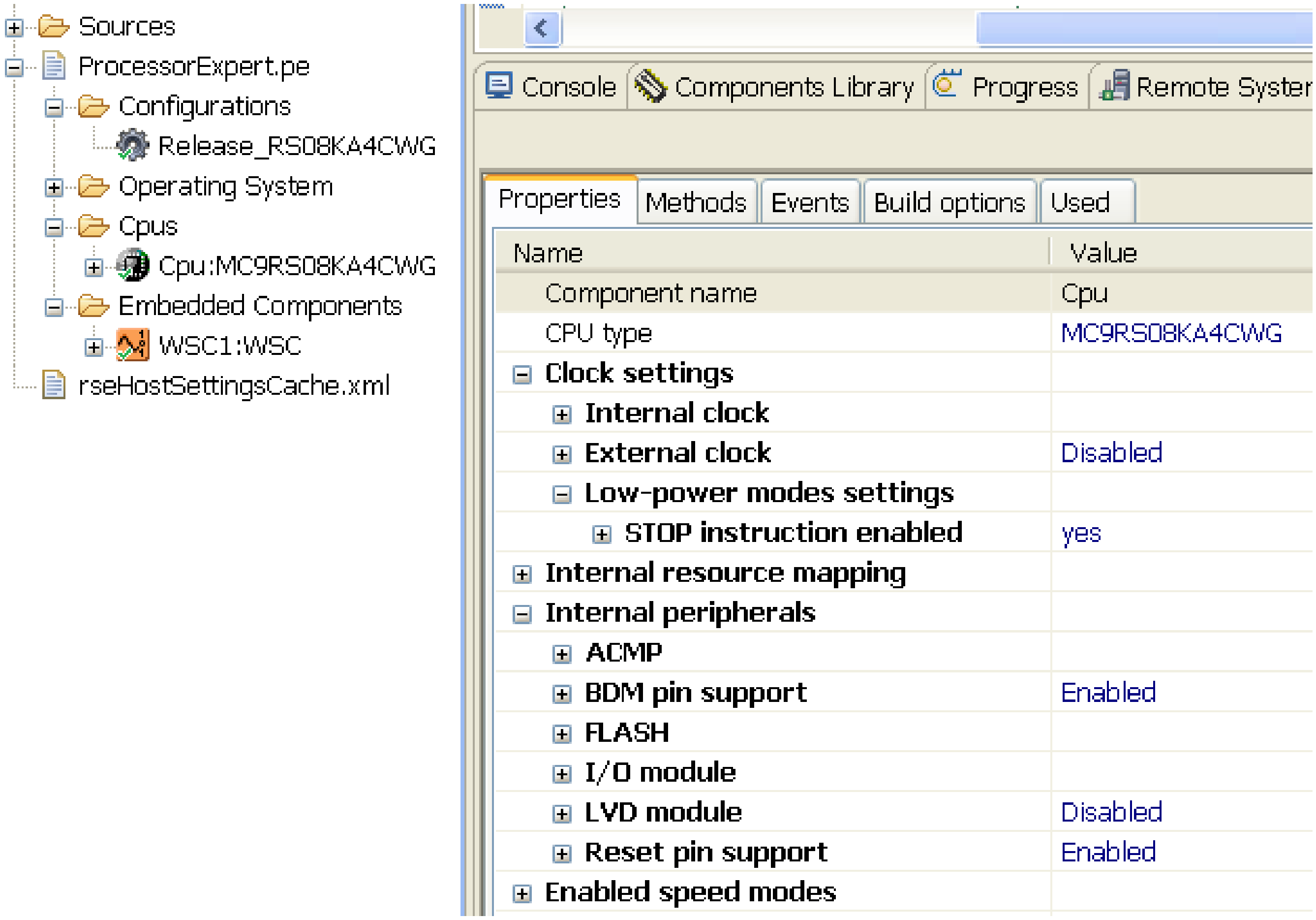
-L"D:\Pro
ted_Code/\

Figure 9. Add wsclib.lib to lib path

2.4.2 CPU property settings

Refer to Figure 10 for CPU property settings

- CW10.2 → Expert view → cpu properties → clock settings → low power mode settings → stop instruction enabled → yes
- CW10.2→ Expert view → cpu properties → Internal peripherals → Reset Pin support → Enabled.
- CW10.2→ Expert view → cpu properties → Internal peripherals → LVD module → Disabled.
- CW10.2→ Expert view → cpu properties → Internal peripherals → BDM pin support → enabled.



- Sources
- ProcessorExpert.pe
  - Configurations
    - Release_RS08KA4CWG
  - Operating System
  - Cpus
    - Cpu:MC9RS08KA4CWG
  - Embedded Components
    - WSC1:WSC
- rseHostSettingsCache.xml

Name	Value
Component name	Cpu
CPU type	MC9RS08KA4CWG
Clock settings	
Internal clock	
External clock	Disabled
Low-power modes settings	
STOP instruction enabled	yes
Internal resource mapping	
Internal peripherals	
ACMP	
BDM pin support	Enabled
FLASH	
I/O module	
LVD module	Disabled
Reset pin support	Enabled
Enabled speed modes	

Figure 10. Set CPU related properties, expert view

2.5 Generate processor expert code via CodeWarrior

After all properties and MCU settings are configured you can generate code by implementing the following steps.

Project -> Generate Processor Expert Code, refer to Figure 11

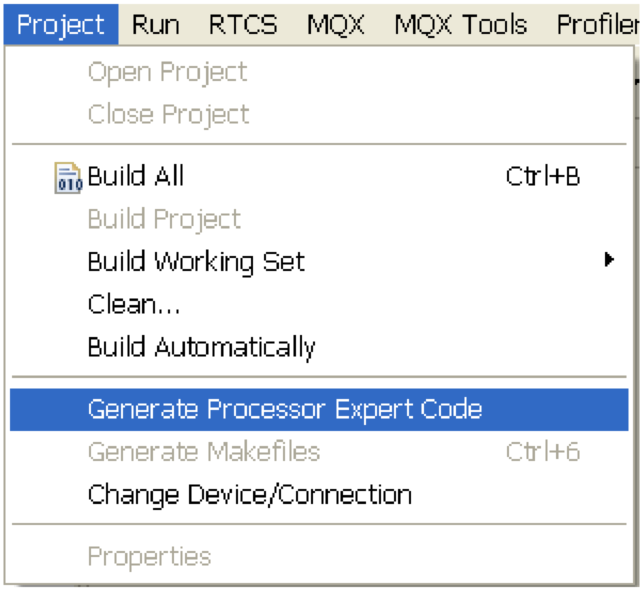
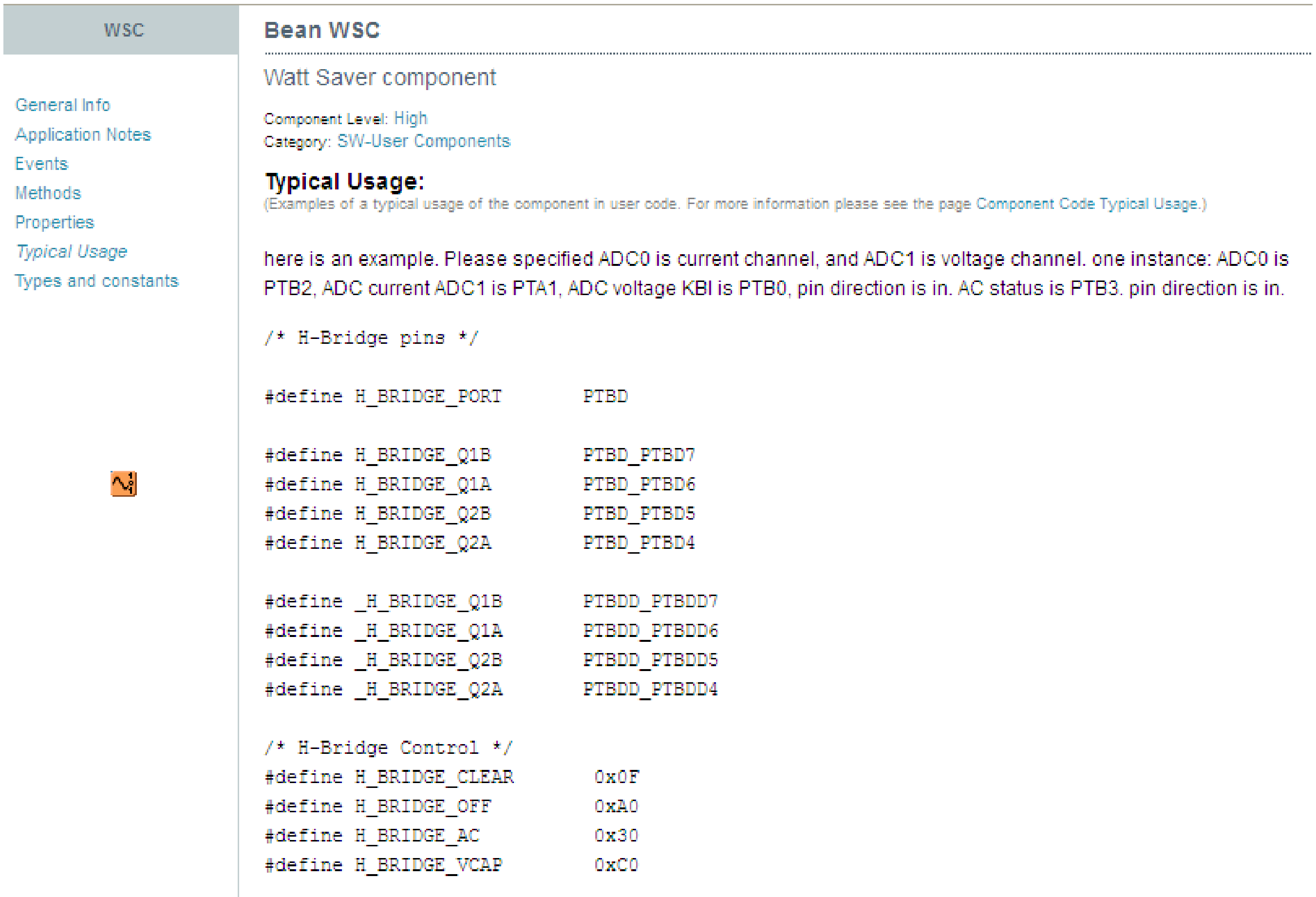


Figure 11. Generate processor code

2.6 Implement the H-bridge Code

H-bridge is used to drive relay to switch ON/OFF AC source in the demo. You can refer to the example code from component online help document.

Users can copy the example source code to their project, refer to Figure 12



WSC

- General Info
- Application Notes
- Events
- Methods
- Properties
- *Typical Usage*
- Types and constants

Bean WSC

Watt Saver component

Component Level: High
Category: SW-User Components

Typical Usage:
(Examples of a typical usage of the component in user code. For more information please see the page Component Code Typical Usage.)

here is an example. Please specified ADC0 is current channel, and ADC1 is voltage channel. one instance: ADC0 is PTB2, ADC current ADC1 is PTA1, ADC voltage KBI is PTB0, pin direction is in. AC status is PTB3. pin direction is in.

```
/* H-Bridge pins */

#define H_BRIDGE_PORT        PTBD

#define H_BRIDGE_Q1B         PTBD_PTBD7
#define H_BRIDGE_Q1A         PTBD_PTBD6
#define H_BRIDGE_Q2B         PTBD_PTBD5
#define H_BRIDGE_Q2A         PTBD_PTBD4

#define _H_BRIDGE_Q1B        PTBDD_PTBDD7
#define _H_BRIDGE_Q1A        PTBDD_PTBDD6
#define _H_BRIDGE_Q2B        PTBDD_PTBDD5
#define _H_BRIDGE_Q2A        PTBDD_PTBDD4

/* H-Bridge Control */
#define H_BRIDGE_CLEAR        0x0F
#define H_BRIDGE_OFF          0xA0
#define H_BRIDGE_AC           0x30
#define H_BRIDGE_VCAP         0xC0
```

Figure 12. Typical use in online help doc

2.7 Build image

Please build image by:

Project → Build Project

The built binary image is demo.abs.s19.

2.8 Flash binary image to the demo board

There are two options to download the software into the WATTSAVER – EVB, both options use the common PE USB multilink 6 pin interface.

Option 1 — CW 10.2

1. Use the debug icon located in the tool bar as shown in Figure 13

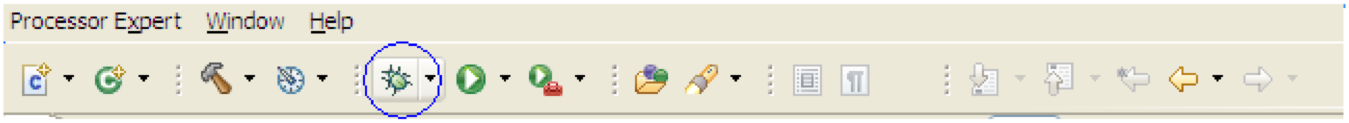


Figure 13. Flashing image

2. After download, click on the red button in the debug box. Refer to Figure 14

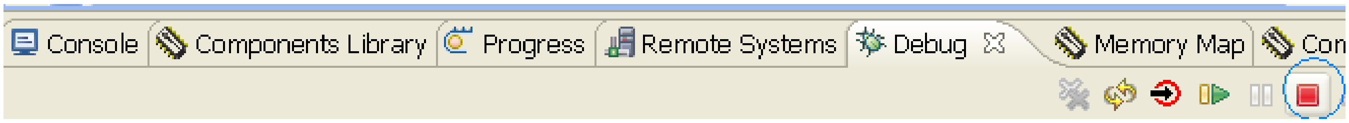


Figure 14. Stop debugger

Option 2 — Install CodeWarrior 6.3, and used hiwave.exe tool to flash. Steps:

1. Install Codewarrior 6.3
2. Run application hiwave.exe in CodeWarrior install path (program sub directory). You can get help from the online help document. (cw_install_path\Help\PDF\ Debugger_HC08.pdf)
3. Connect it to the board through a PE USB multilink interface and set a connection as mentioned below.

   Please refer to page 48 of the Debugger_HC08.pdf, When you set the connection, select RS08 as Processor and 'P&E Multilink/Cyclone Pro' as Connection

4. Burn code

   Select the binary file to burn by selecting file -> Load Application. Page 34 of the Debugger_HC08.pdf.

   For details, refer to document Debugger_HC08.pdf.

   After the code is flashed
   a. Cut off the board power
   b. Remove the debugger cable from the board.
   c. Press the reset button on the board (SW2);
   d. Power ON the board.

The board is ready to be tested.

3 WSC theory

Watt Saver software control includes several operation modes, operation parameters, and features that will be explained in this chapter.

3.1 Application Block Diagram

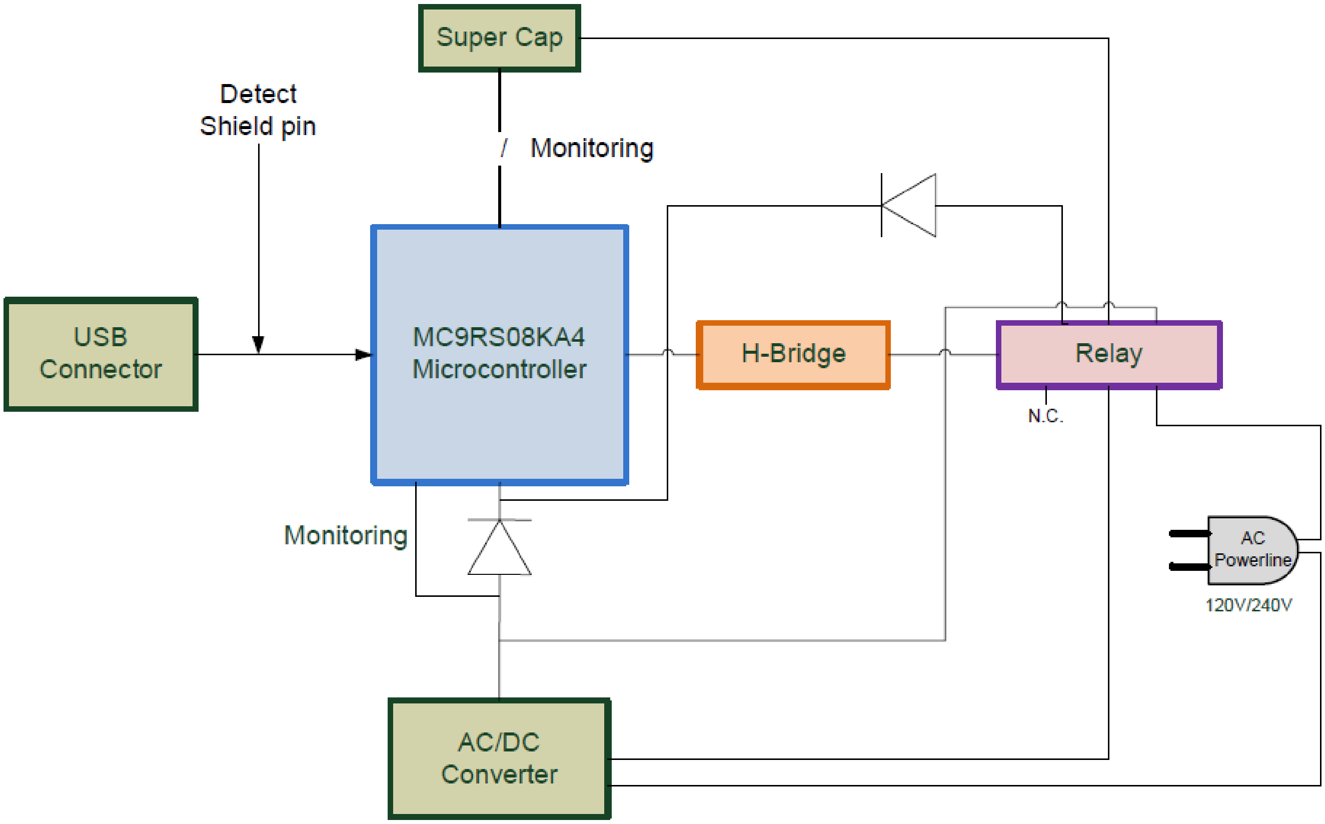


Figure 15. Application block diagram

3.2 State diagram

The state diagram is depicted in Figure 16.

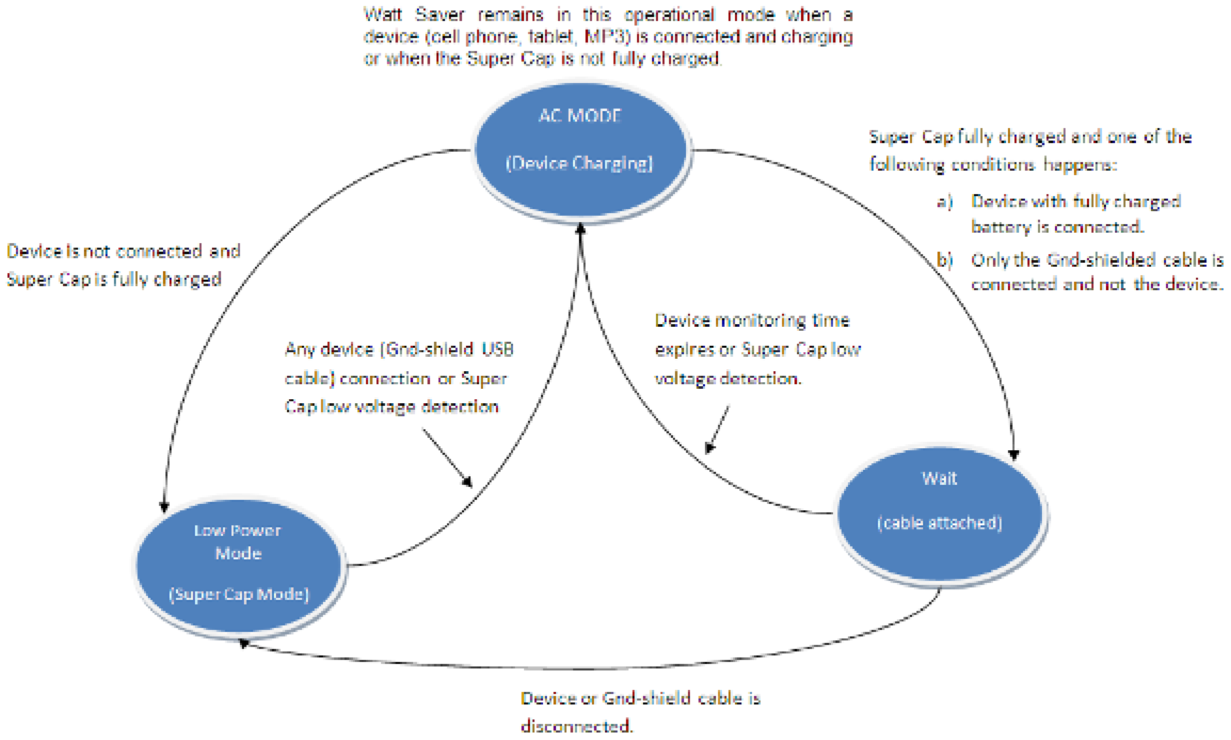

Watt Saver remains in this operational mode when a device (cell phone, tablet, MP3) is connected and charging or when the Super Cap is not fully charged.
AC MODE
(Device Charging)
Super Cap fully charged and one of the following conditions happens:
a) Device with fully charged battery is connected.
b) Only the Gnd-shielded cable is connected and not the device.
Device is not connected and Super Cap is fully charged
Any device (Gnd-shield USB cable) connection or Super Cap low voltage detection
Device monitoring time expires or Super Cap low voltage detection.
Wait
(cable attached)
Low Power Mode
(Super Cap Mode)
Device or Gnd-shield cable is disconnected.

Figure 16. State diagram

NOTE

1. The difference between Wait Mode and Low Power mode:

   Load connected—Enter wait mode Load not connected—Enter low power mode

   .
2. wait_time property is used in wait mode for periodically monitoring load current.

   The MCU will be wakes after wait_time expires or super cap low voltage detection.

3.2.1 Description—configuring current properties

Three current threshold values are:

- current_his
- current_min
- current_large

The value order is shown in the Figure 17.

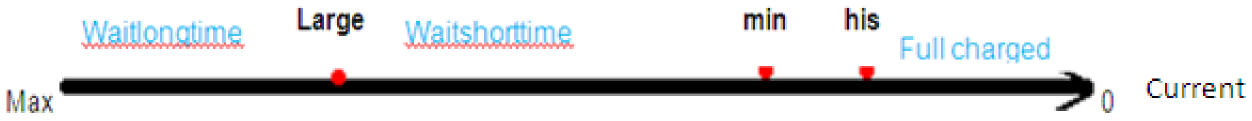


Figure 17. Current threshold

- current_min > current_his — The hysteris between current_min and current_his is to avoid operation vibration or oscillation
- current > current_large — MCU samples current every waitlongtime seconds;

- current > current_min and current < current_large — MCU samples current every waitshorttime seconds
- current < current_his — Device is not charging (charging full).

3.2.2 How to set current properties value

Below is the example used for charging a mobile phone on WATTSAVER-EVB.

1. Plug in a mobile phone into the demo board
2. Measure the JB5 voltage (U2, the Vout pin of MAX4372H) and calculate the current by using formula: I = V * 100 The voltage unit is volt, and the current unit is mA.
3. Monitor the charging period. The charging curve is similar to the figure below:

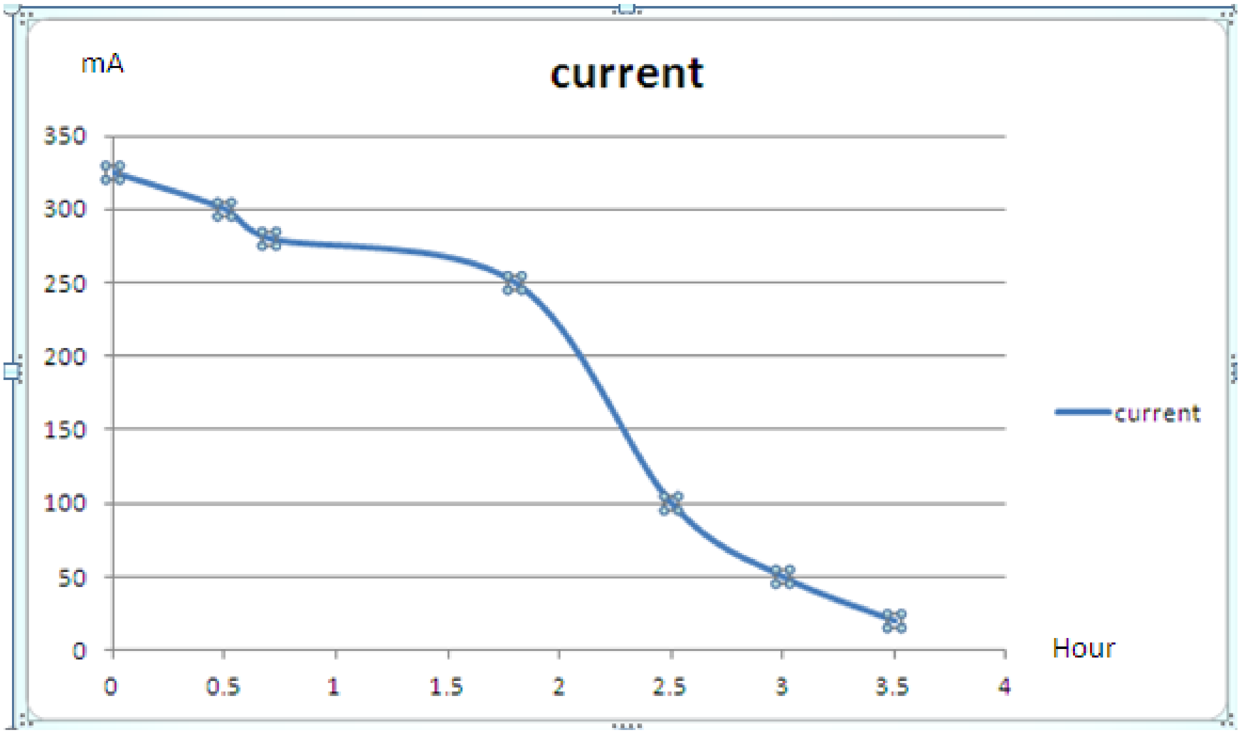


Figure 18. Charging curve

4. Observe the current curve

   Current > 50 mA, device is charging (325 mA to 50 mA, takes 3 hours)

   Current < 30 mA, is fully charged.

   50 mA and 30 mA are current threshold values.

5. Set the current properties value

   R9 = 0.1, 10-bit ADC = 1024, magnification scale factor = 100,

   Maximum voltage of JB5 is (5 V – voltage of diode) 4.72 V.

   1 mA representative unit is :

   0.001 A(1 mA) * 0.1 Ω * 100 * 1024 / 4.72 V = 2.169
   current_large = 2.169 * 50 = 108
   current_his = 2.169 * 30 = 65

   35 mA (a little greater than current) is set as current minimum threshold,

   Therefore:

current_min = 2.169 * 35= 76

3.2.3 2 APIs from Wsclib.lib

The 2 API, listed below, can be called by the user.

void MCU_Init(void) : Initiates the MC9RS08 public peripherals

void main_fsm(void): It implements the watt saver power consumption algorithm.

4 Limitations

- User implements H bridge code to drive AC ON/OFF
- Only used for RS08 MCU family
- Tested with CodeWarrior 10.2
- Only tested with 5 W AC source application

5 Conclusions

This document described the basic steps for getting started with the Watt Saver software component and the steps to configure it. For technical details about the board design and solution implemented, please check the documentation section in the following website:

www.freescale.com/WattSaver

6 References

- RS08KA: 8-bit General Purpose Ultra-Low-End Market KA MCUs, available at www.freescale.com/RS08KA
- Watt Saver Solution for Chargers, at www.freescale.com/WattSaver

7 Revision history

Revision number	Date	Changes
1	10/2013	Initial public release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. Freescale reserves the right to make changes without further notice to any products herein.

Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 that may be provided in Freescale 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 does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.