Watt Saver for a Cell Phone AC Adapter

Reference Design

Document Number: DRM130
Rev 1, 10/2013

Contents

Section number **Title** **Page**

Chapter 1
Introduction

Chapter 2
Hardware Description

Chapter 3
Firmware Description

Chapter 4
Performance Results

Chapter 1
Introduction

1.1 Overview

Innovative Watt Saver no-load technology for AC adapters prevents wasting power when no device is connected.

Freescale Semiconductor is finding innovative ways to stop “vampire” energy loss, that is, is the loss of power that occurs when an AC adapter is plugged into an electrical outlet but isn’t charging a device.

Freescale’s new Watt Saver technology automatically eliminates no-load power consumption for AC adapters, potentially providing substantial energy savings over existing manual versions. Freescale’s Watt Saver technology consists of patent-pending hardware and software implementations enabling the main power source to be disconnected when no power is required by the connected device.

1.2 Freescale solution

The Watt Saver reference design is an example of how a “Smart Switch” can be built taking advantage of the MC9RS08KA4 features. The main features of this microcontroller that make it ideal for Watt Saver solution are:

- Low cost implementation
- Power saving modes
- 12-channel 10-bit resolution analog-to-digital converter (ADC)
- Analog Comparator (ACMP)
- Real-Time Interrupt (RTI)
- Up to 18 GPIOs including one output-only and one input-only pin
- Background debugging system on-chip
- Wake from stop mode based on analog comparator module, real time interrupt and key board interrupt events

The Watt Saver charger block diagram is shown in Figure 1-1.

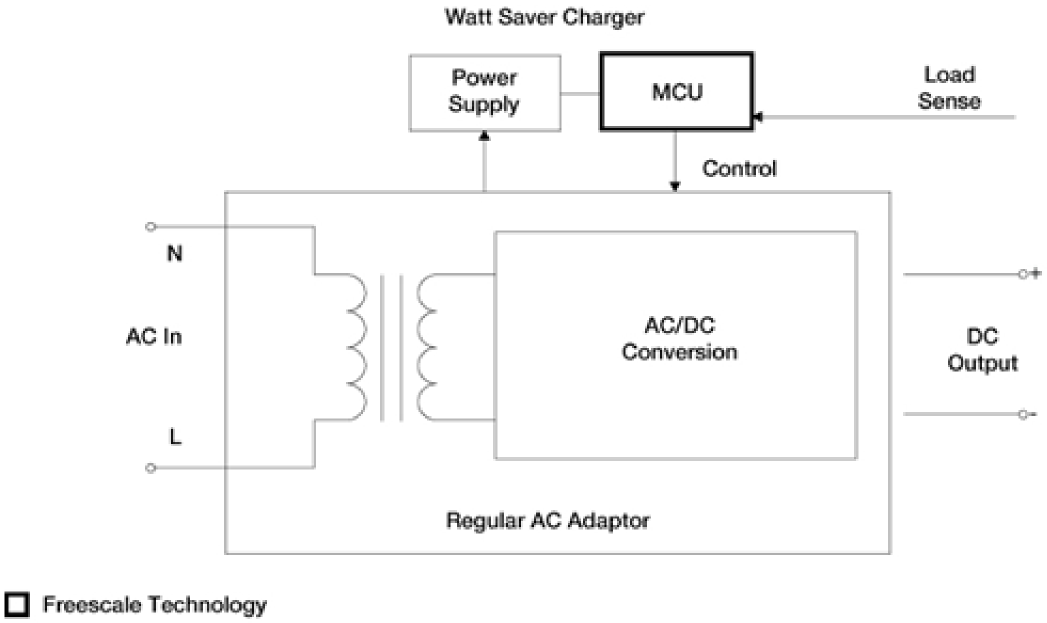


Figure 1-1. Watt Saver block diagram

The last hardware version of watt saver is a 5 W zero no-load power AC/DC converter with current measurement and low power saving improvements, this Watt Saver demonstration board is shown in Figure 1-2.

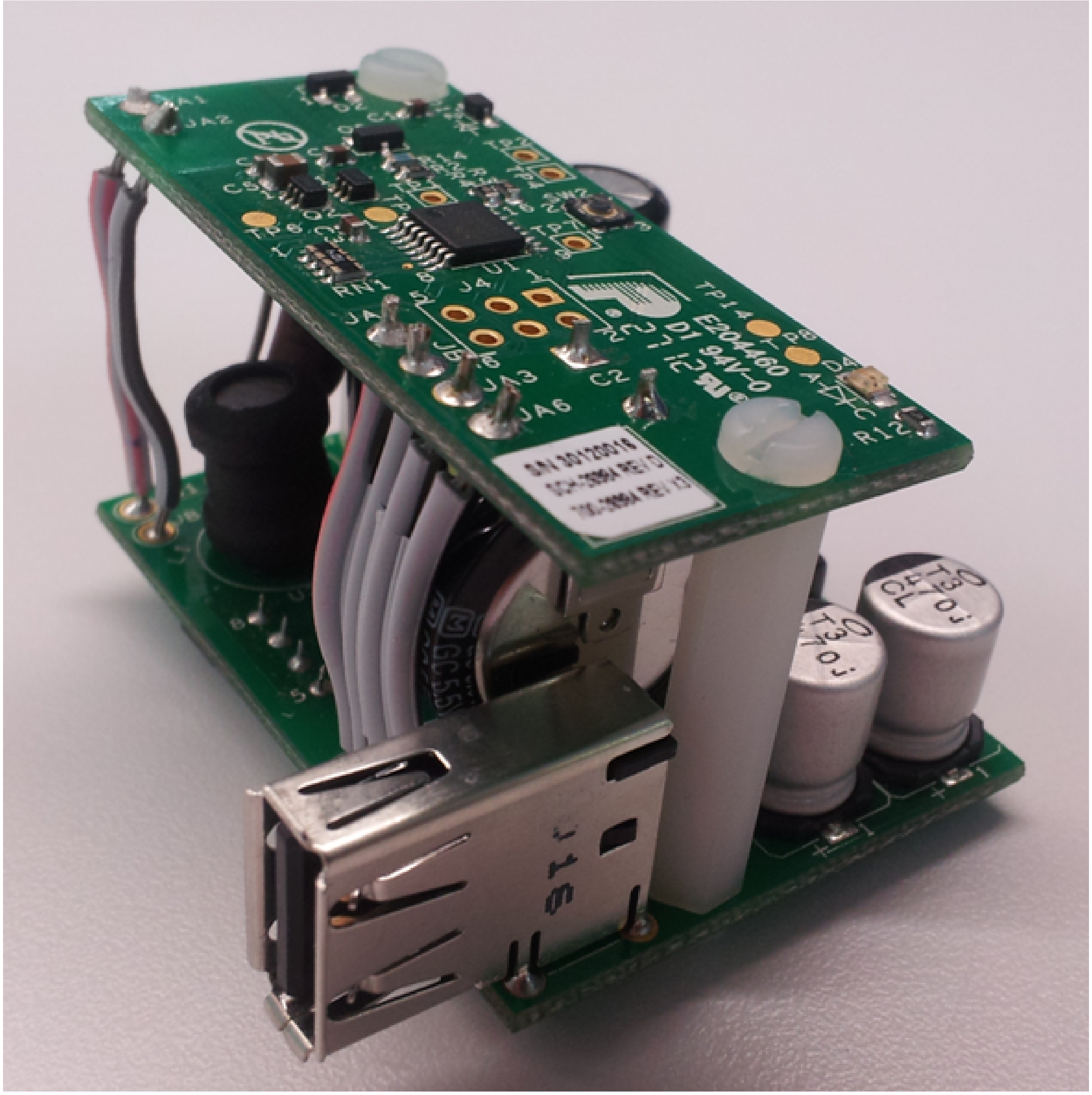


Figure 1-2. Watt Saver demonstration board

1.3 Demo board features

The Watt Saver demonstration board includes the following features:

- Support for 110 V and 220 V
- Discrete H-Bridge
- 1 F – 5.5 V super cap
- Support common and ground (GND)-shield USB cables
- Device end-of-charge detection (cell phone, mp3 player)

NOTE

A GND-shield cable is a cable with GND and shield tied together on the cable.

Chapter 2
Hardware Description

2.1 Demo board application block diagram

The design block diagram is found on Figure 2-1.

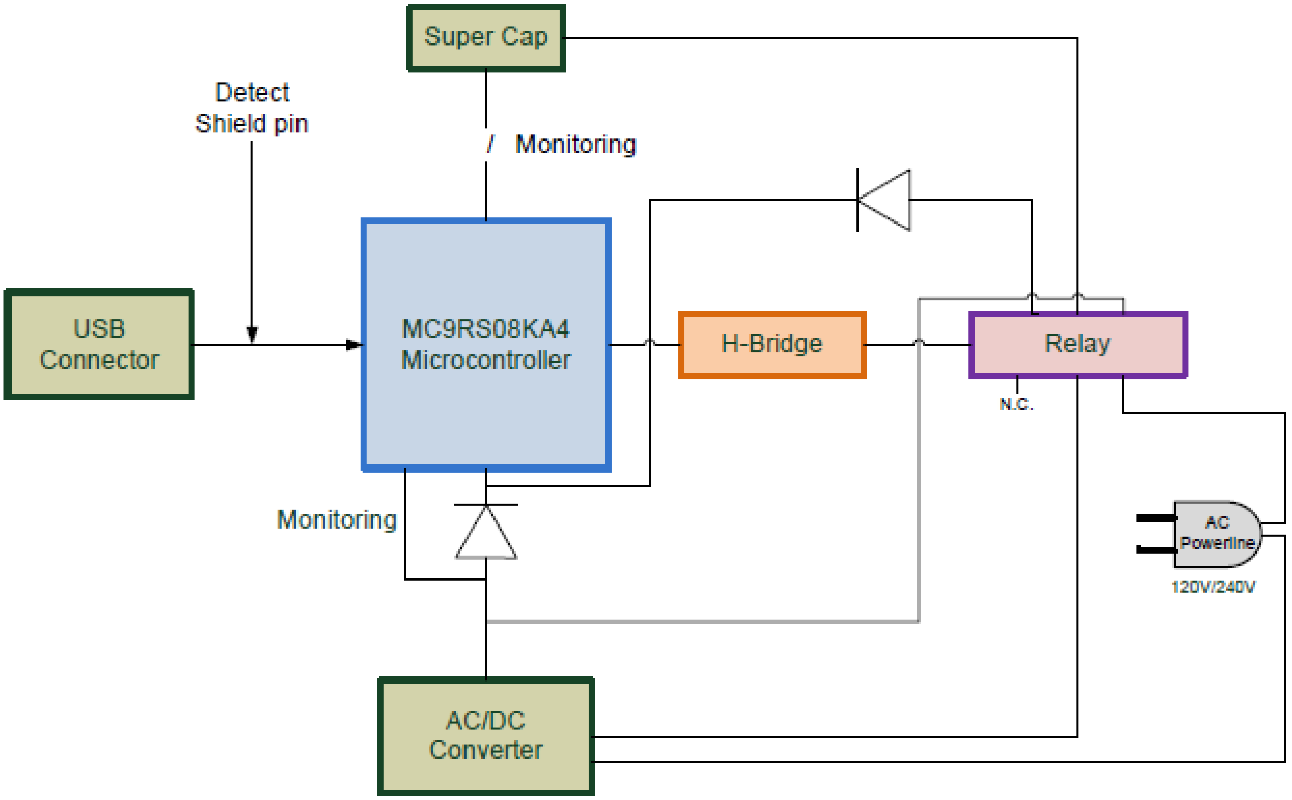


Figure 2-1. Design block diagram

2.2 Modules explanation

Microcontroller (MCU)

The RS08KA4 microcontroller is used to control all operational modes, current, voltage measurements, and GND-shield detection.

The following are the MCU modules used in this application:

- Analog comparator (ACMP)
- Keyboard Interrupt (KBI)
- Real-Time Interrupt (RTI)
- 10-bit Analog to Digital Converter (ADC)

Discrete AC/DC Converter

This module has the following characteristics.

- Input — 110–220 V
- Output — 5 V / 1 A

Super Capacitor (super cap.)

Super cap is the power supply for Watt Saver when it is in low-power mode. It is used instead of a battery. This is a 1 Farad, 5.5 Volts super cap.

H–Bridge

Discrete H-bridge is created with two complementary pair enhancement mode field-effect transistors, H-bridge is used to generate the control signal needed to switch relay between AC mode and low power mode.

Relay

A bistable coil type relay is used to disconnect the AC power line from the rest of the circuit when the Watt Saver is in low-power mode. This component has the following characteristics:

- Bistable coil type
- 110/220 VAC compatible
- 2.4 VDC rated voltage

2.3 Demo board content

The Watt Saver demo board contains several blocks needed for the implementation of a no-load AC charger. The main blocks and components of this demo board are highlighted In Figure 2-2 and Figure 2-3.

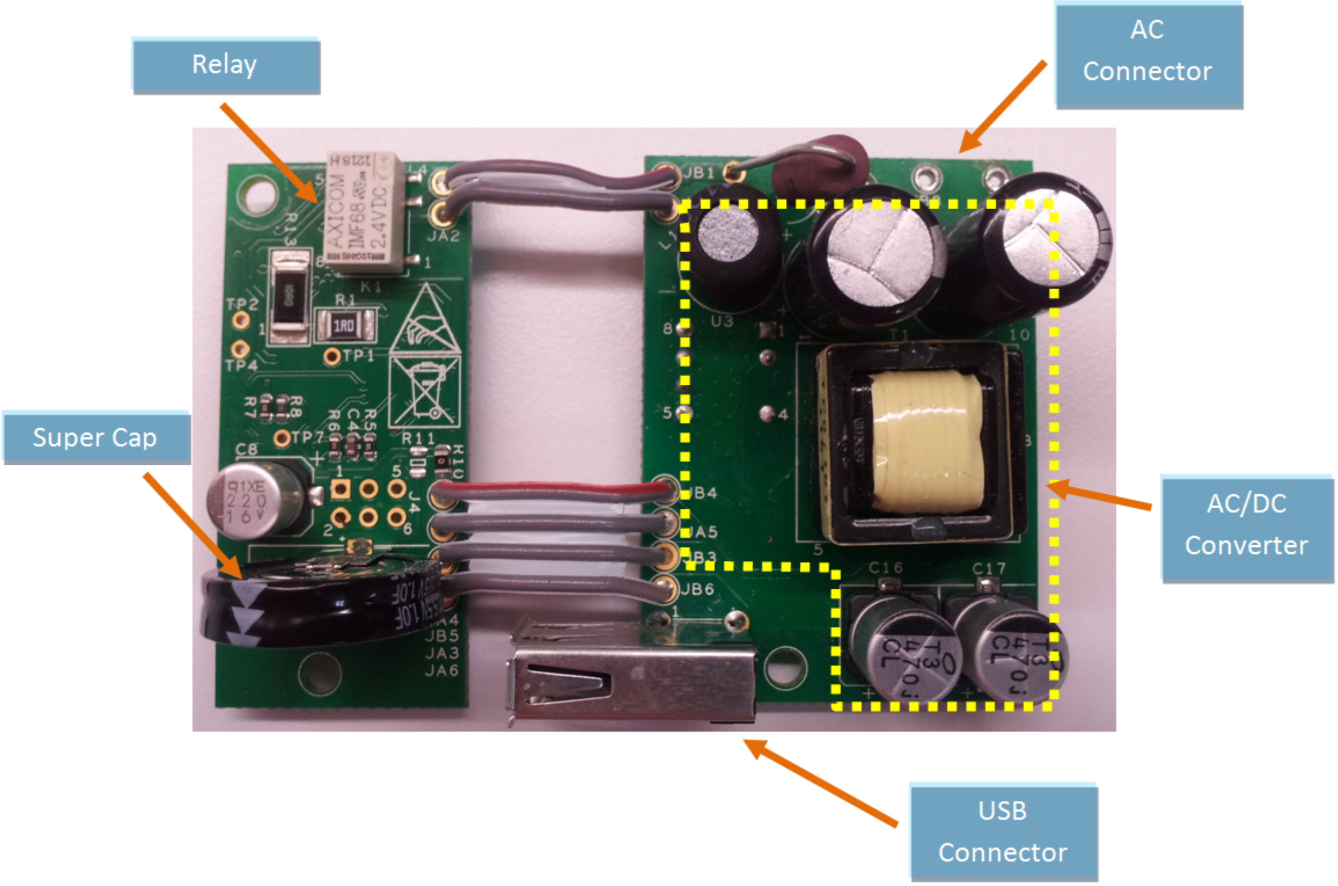


Figure 2-2. Watt Saver demo board content (Inside)

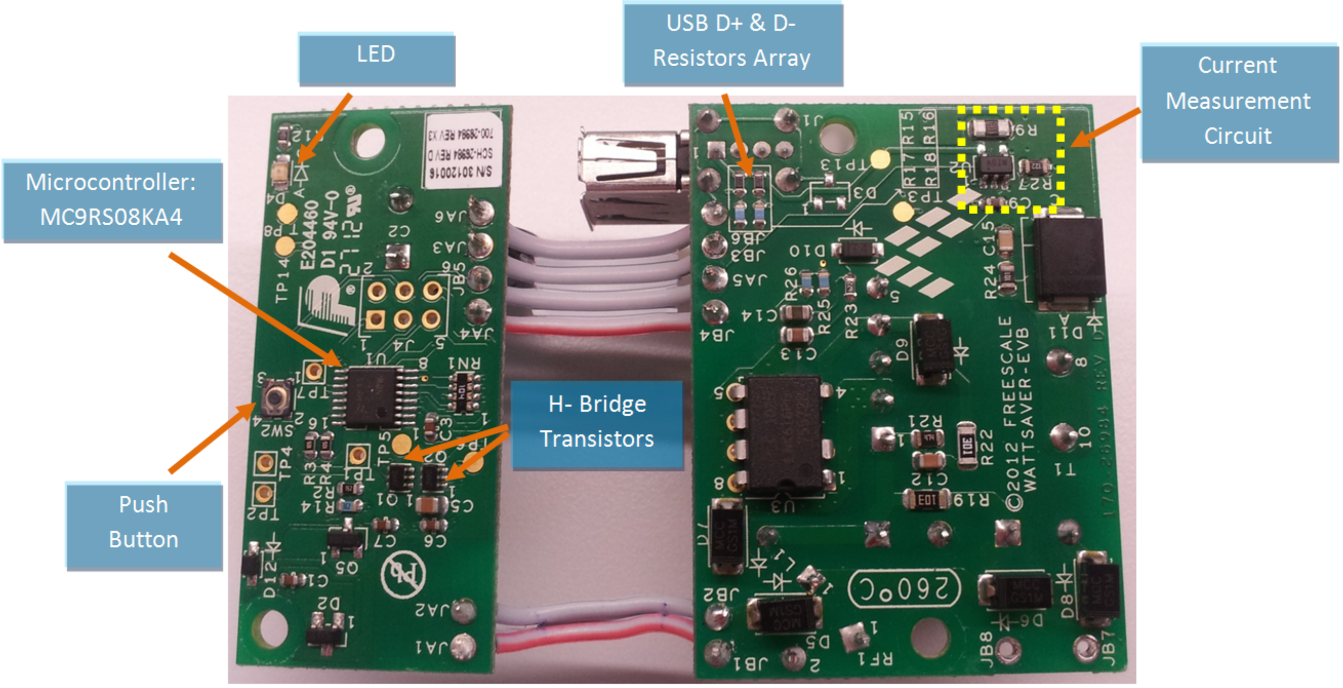


Figure 2-3. Watt Saver demo board content (outside)

2.4 Relay type and connections

To obtain a zero current draw charger the Watt Saver solution uses a bi-stable coil type relay, a discrete H- bridge and software control. The complete switching solution is shown in Figure 2-4.

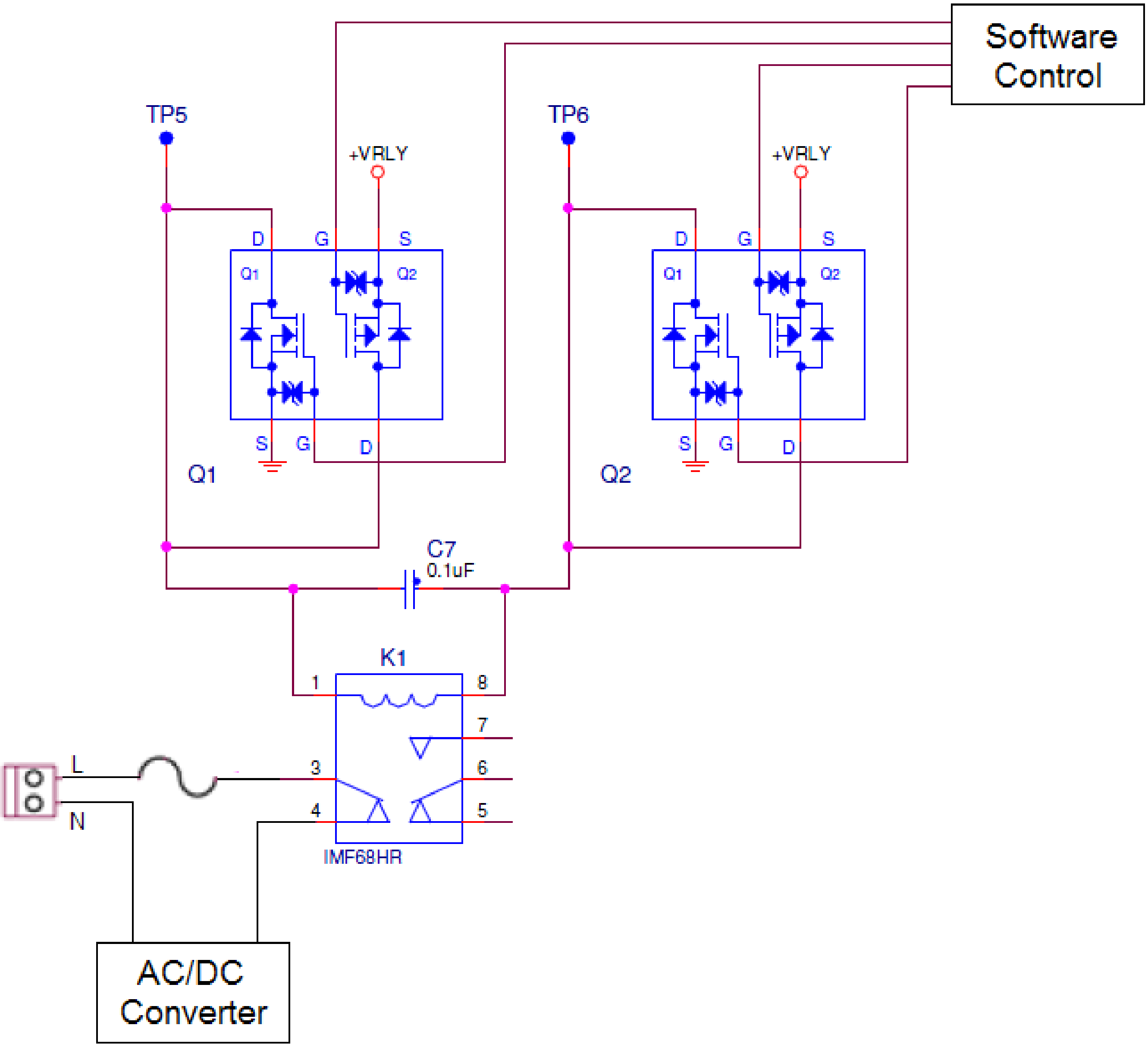


Figure 2-4. AC switching solution

A bi-stable relay is used because the application needs to improve the consumption in low power mode. The main feature of this relay is that it requires a current pulse with polarity to switch between both states. While the coil is relaxed the contacts remain in its previous position and no energy is being consumed. When the current pulse is provided the coil consumes power only for an instant.

2.5 Super capacitor low voltage detection circuit

When the watt saver operates in AC mode, the software control uses a 10-bit ADC module to measure the voltage in the super cap. and determines if it is fully charged.

In low power mode, the software control uses ACMP (bandgap enabled) to detect low voltage in the super cap.. The MCU remains in stop mode until the ACMP flag is set. This occurs when Vcap is less than 2.4 V. This is because a voltage divider is created with R3 and R4 (see Figure 2-5). In addition, the PTA0 is used as a virtual GND, created by the MCU (pin as output and 0 logic) to avoid energy waste while the voltage divider is not in use.

Super cap. charging and low voltage detection circuits are shown in Figure 2-5.

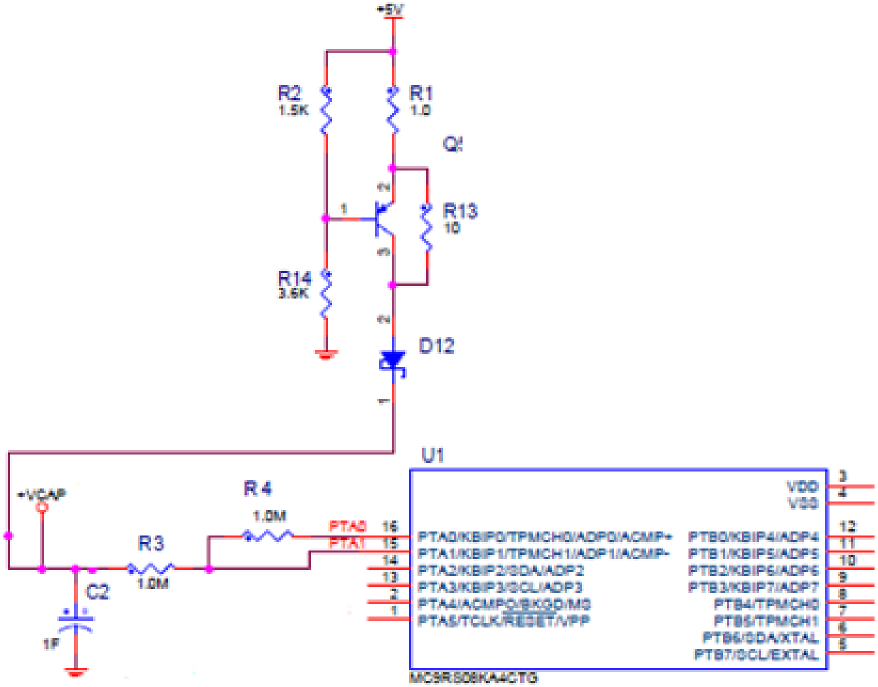


Figure 2-5. Low voltage detection and charging circuit

2.6 Device detection circuit

Most cell phones have the ability to tie together GND and Shield signals when a USB cable is connected. The Watt Saver takes advantage of this behavior using a pull-up resistor circuit and a GPIO to detect when a device is connected. This circuit is shown in Figure 2-6.

When no device is connected the MCU reads a logic “1” in the GPIO pin. If a cell phone is connected, the shield signal is tied to GND. When the MCU detects a falling edge it wakes the Watt Saver and enters in AC mode.

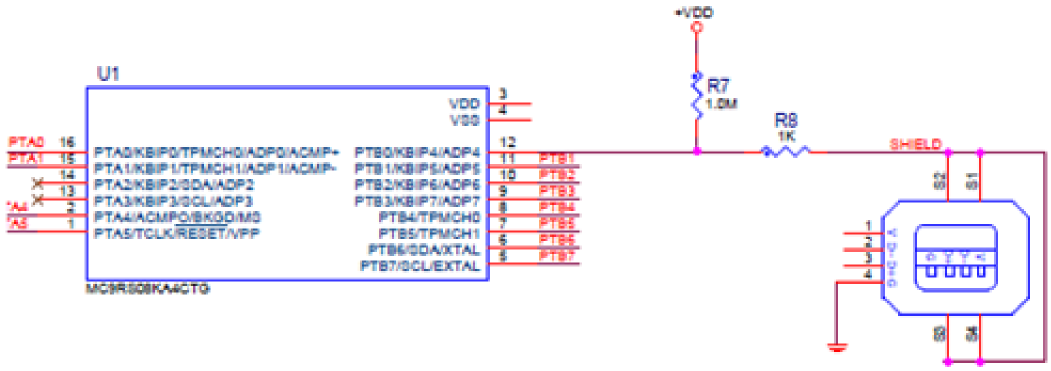


Figure 2-6. Device detection circuit

2.7 Current measurement circuit

One of the most relevant features is the device end-of-charge detection and the ability to support ground-shield cables. The Watt Saver detects a device based on the union of GND and shield, this occurs when the device is connected to the cable, although currently some cables have this union without any device connected.

This feature is implemented using a 10-bit ADC and the current measurement circuit. This circuit includes the amplifier MAX4372 (x100 gain), which increases the measurement resolution. The Watt Saver detects when a cable is connected to the USB connector. The current is measured to determine if there is any connected device. The highest and the lowest current values can be selected via software, this indicates that the Watt Saver can be customized for specific or generic devices. The current measurement circuit is shown in Figure 2-7.

The operational ranges are:

Device charging — 210 mA DC or higher

Device fully charged — 170 mA DC or lower

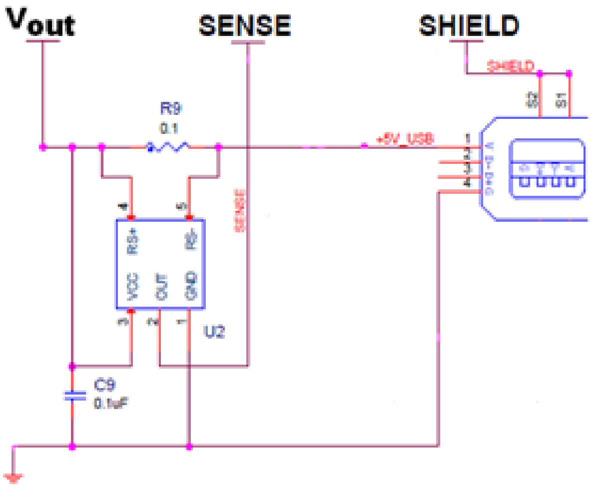


Figure 2-7. Current measurement circuit

Chapter 3
Firmware Description

3.1 Demo board operation modes

The Watt Saver has three operational modes:

- AC mode—The Watt Saver remains in this operational mode when a device is connected and charging or when the super cap is not fully charged. The LED remains turned on.
- Low Power mode (super cap. mode)—The Watt Saver switches to this operational mode when a device is not connected and super cap. is fully charged. Any device or GND-shield cable connection wakes the watt saver and switches to AC mode. The LED remains turned off.
- Wait Cable mode—The Watt Saver switches to this operational mode when the super cap. is fully charged and one of the following conditions happens:
  - The device is connected and the device battery is fully charged.
  - Only the GND–shield cable is connected and the device is not.

If device or GND-shield cable is disconnected, then the Watt Saver switches to super cap. mode.

If the Watt Saver detects a low voltage in the super cap., it then switches to AC mode.

The Watt Saver returns to AC mode periodically to detect the device current measurement and to charge the device again, if needed. The LED turns on periodically every 20 minutes. This time can be configured with the watt saver component property WAIT_TIME.

3.2 State diagram

The state diagram is found on Figure 3-1.

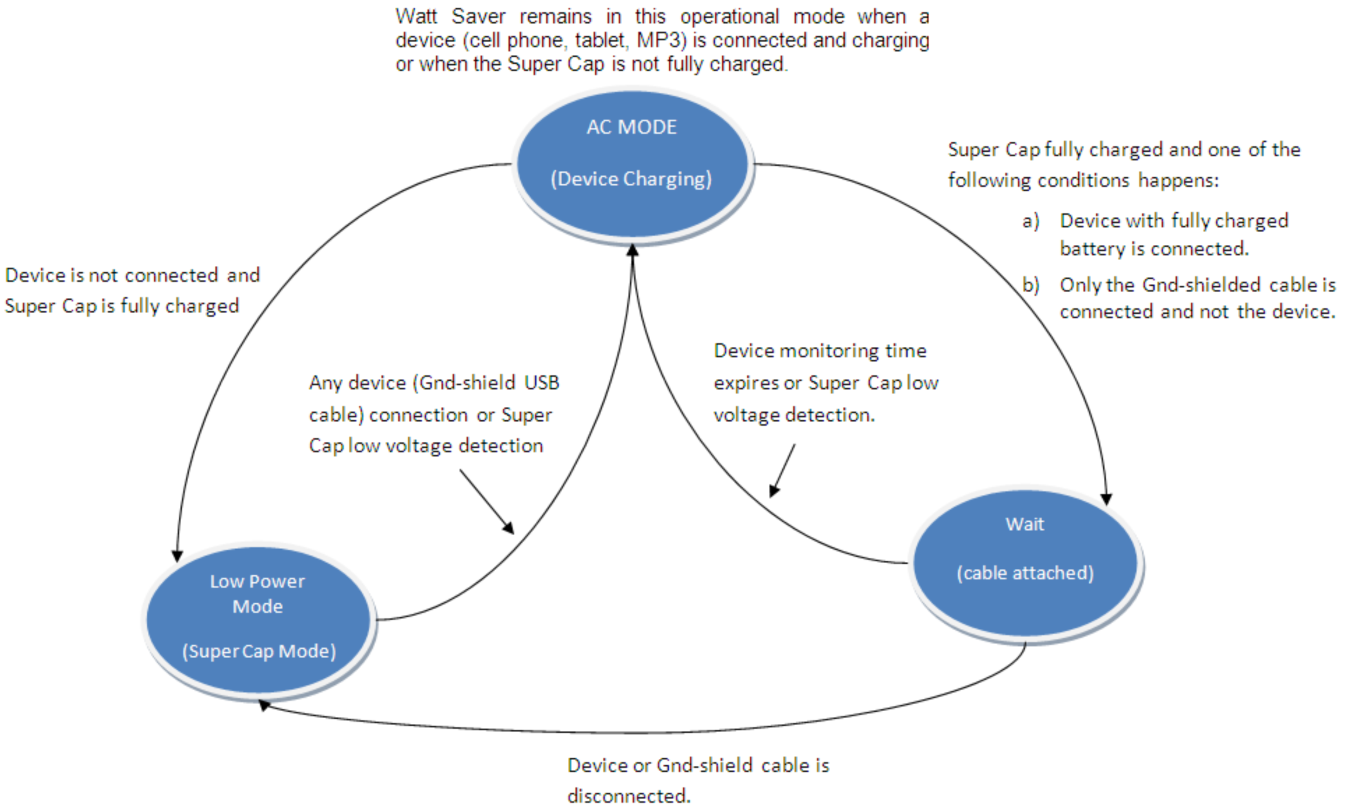


Figure 3-1. State diagram

3.3 Relay software control

The software control implemented to switch the relay using the discrete H-bridge is based on four GPIOs, one for each transistor, these GPIOs are initialized using macro definitions trying to make the code easy to modify for customized hardware platforms. In the current application port B pins 4 to 7 are used. GPIOs are initialized as follow:

```
/* 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
```

After the GPIO´s are initialized control macro definitions need to be created with the purpose to generate a clear and easy to understand code, these control macro definitions will be created to manage all the logic used to clear, switch, and turn off the H-bridge. The following are the macros created for this specific hardware:

```
#define H_BRIDGE_CLEAR           0x0F
#define H_BRIDGE_OFF             0xA0
#define H_BRIDGE_AC              0x30
#define H_BRIDGE_VCAP            0xC0
```

With the macro definitions created the switching of the relay can be implemented in three lines, in the following piece of code the first group of instructions are used to switch the relay to AC mode, and the second group is used to switch to low power mode.

```
gu8PTB_temp = H_BRIDGE_PORT;                    // Store PTBD value in temp variable
gu8PTB_temp = (gu8PTB_temp & H_BRIDGE_CLEAR);   // Clear H-Bridge Bits (7-4)
H_BRIDGE_PORT = (gu8PTB_temp | H_BRIDGE_AC);    // Change to AC Mode

gu8PTB_temp = H_BRIDGE_PORT;                    // Store PTBD value in temp variable
gu8PTB_temp = (gu8PTB_temp & H_BRIDGE_CLEAR);   // Clear H-Bridge Bits (7-4)
H_BRIDGE_PORT = (gu8PTB_temp | H_BRIDGE_VCAP);  // Change to Vcap Mode
```

3.4 Super capacitor voltage measurement algorithm

The Watt Saver hardware used to detect when the super cap. is fully charged is an ADC pin in combination with a voltage divider circuit. A GPIO is used to enable or disable the voltage divider and avoid energy waste. Following are the macro definitions created for this purpose.

```
#define    SUPCAP_MEASUREMENT_CTRL                PTAD_PTAD0
#define   _SUPCAP_MEASUREMENT_CTRL                PTADD_PTADD0
#define   CAP_ADC                                 1              /*Super Cap ADC #1*/
```

To enable the voltage divider and perform a voltage measurement the GPIO needs to be set as an Output with a logic value of 0, this will create a virtual ground to close the circuit. The instructions are shown below.

```
SUPCAP_MEASUREMENT_CTRL     =  0;             // Set por as 0
 _SUPCAP_MEASUREMENT_CTRL  = _OUT;            // Set port as ouput for measure Vcap
```

The output of the voltage divider circuit is Vin/2. In the Watt Saver application, a macro definition with the desired value at which the application detects that the super cap. is fully charged needs to be created. In the watt saver application 4.7 volts is the desired limit for the super cap. because it uses a 1 F, 5.5 V super cap. The charging circuit is designed to charge up to 5 volts.

```
#define     V_MAX                       512             /*4.7 Volts*/
#define     V_HIS                       500             /*Histeresis(10 Units)*/
```

The ADC counts assigned to the macro V_MAX are obtained as follows:

ADC resolution is 10-bit, so the highest value is 1024. The voltage that is powering the MCU is 4.7 volts so, (1024 * ((4.7)V / (2))) / 4.7 V = 512 ADC counts.

After the macro definitions are created, the ADC module needs to be initialized as follow:

```
ADCCFG= ADCCFG_MODE1_MASK;      // 10 bit conversion
APCTL1_ADPC1=1;                 // Enable ADC1(PTA1) for super cap. voltage measurement
```

To improve the accuracy of the voltage measurement, an algorithm to obtain the median value was developed, the first eight samples are stored in an array then the following segment code is implemented.

```
do
  {
     u8CompletFlag=0;
     for (gu8Counter=0;gu8Counter<7;gu8Counter++)
    {
        if (gu16TempArray[gu8Counter] > gu16TempArray[gu8Counter+1])
        {
           gu16TempValue = gu16TempArray[gu8Counter];
          gu16TempArray[gu8Counter] = gu16TempArray[gu8Counter+1];
         gu16TempArray[gu8Counter+1] = gu16TempValue;
         u8CompletFlag=1;
       }
    }
 }while (u8CompletFlag)

gu16ADCVoltage_avg = ((gu16TempArray[3] + gu16TempArray[4])/2);       // ADC median equation
```

After this, compare the V_MAX macro definition value with this ADC median result and decide if the super cap. is fully charged or not.

3.5 Super capacitor low voltage detection method

The ACMP module in combination with a voltage divider circuit used to detect when the super cap. is working at a low voltage range. As mentioned in the previous section the voltage divider needs to be enabled to obtain an accurate comparison.

The ACMP compares the voltage divider output with the Bandgap reference (1.2 V), therefore the bandgap reference must be enabled and the ACMP must be initialized.

```
SPMSC1=SPMSC1_BGBE_MASK;
ACMPSC= ACMPSC_ACME_MASK  |ACMPSC_ACBGS_MASK | ACMPSC_ACMOD0_MASK;
```

In the previous configuration, select the bandgap reference as one of the signals to compare and set the ACMP output as rising edge. This generates the ACMPSC_ACO flag set when the super cap. voltage is lower than the bandgap reference.

The ACMP module is enabled in the super cap. mode function, each desired time to perform a comparison and decide if the super cap. is working in low voltage range. The application needs to operate with low power consumption. A device connection or an accidental switching of the relay events are implemented in the algorithm for super cap. mode, if one of these conditions occurs then the Watt Saver switches to AC mode.

3.6 Device detection and current measurement algorithm

As mentioned in the hardware section, the Watt Saver uses a pull-up resistor and a GPIO to detect when a device is connected in the charger, macro definitions created for this purpose are shown below.

```
#define DEVICE                PTBD_PTBD0
#define _DEVICE               PTBDD_PTBDD0
```

When no device is connected the DEVICE macro definition is read as logic “1”, If a device is connected, the shield signal is tied with GND and the DEVICE macro definition is read as logic “0”.

For the current measurement algorithm it is necessary to create the following macro definitions.

```
#define CURRENT_ADC               6             /*Super Cap ADC #6*/
#define CURRENT_MIN               457           /*Minium current     [210 mA DC]*/
#define CURRENT_HIS               370           /*Current Histeresis [170 mA DC] */
```

As already mentioned, the ADC module and the desired channel needs to be enabled as shown below.

```
ADCCFG= ADCCFG_MODE1_MASK;        // 10 bit conversion
APCTL1_ADPC6=1;                   // Enable ADC6(PTB2) for device current measurement
```

Because of hardware improvements a current amplifier circuit was added, it increases the current measurement resolution. To set the values for CURRENT_HIS and CURRENT_MIN the operational current value range for your specific device needs to be obtained. This application sets the current parameters for generic devices, when the device is charging the current value is higher than 210 mA @DC and when the device is fully charged the current value is lower than 170 mA @DC. The ADC counts are obtained as below.

Current Amplifier Vout = I * R * gain = (0.170 A) * (0.1 Ohm) * 100 = 1.7 V.

ADC resolution is 10 bit, so the highest value is 1024. The voltage that is powering the MCU is 4.7 volts, therefore, (1024 * 1.7 V) / 4.7 V = 370 ADC counts. The same calculation needs to be implemented for the highest current value.

After the limits are defined, the median algorithm explained in section 3.4 needs to be applied, with this algorithm the measurement accuracy increases.

The ADC median result value is compared with the upper and lower limits of current as follows:

```
if (gu16ADCCurrent_avg > CURRENT_MIN)
    {
        gu8I_charging=0;                              // Device is
charging
    }
```

```
    else if(gu16ADCCurrent_avg < CURRENT_HIS)
    {
        gu8I_charging++;                              // Device is not charging
and                                                         // counter increase
        if (gu8I_charging==224) gu8I_charging=6;      // Reset counter to avoid overflow but
maintain not charging count.
    }
```

When the algorithm detects that the device is not charging, it increases a counter variable, and is related to the number of seconds that the Watt Saver remains in AC mode, that is before it switches to super cap. mode when the device is fully charged or disconnected. In the Watt Saver application this time is called Switch off delay and can be configured with the macro definition TIME_AC_SWITCH_DELAY, this was implemented due to particular device charge cycles, these devices start charging as usual at high current values, but in a few minutes stops charging for a specific time (few seconds) and then is followed with the charging cycle.

Chapter 4
Performance Results

4.1 Operating parameters

Table 4-1. Operating parameters

Parameter	Measurement	Notes
Super cap. charge time	~ 1 minute	Watt Saver takes only 1 minute to charge the 1 Farad super cap. up to 4.7 Volts
Super cap. discharge time	~ 15 days	Watt Saver takes 15 days to discharge super cap. up to 2.3 Volts. This voltage is needed to switch the relay to AC mode.
Watt Saver current consumption in AC mode without device connected	~ 6.5 mA	This is the current consumption while super cap. is charging and the device is not connected.
Watt Saver current consumption in low power mode (AC side)	0 A	Watt Saver is disconnected when no power is required by the connected device.
MCU current consumption in low power mode	~ 1 μA	This is the average current consumption in low-power mode. The MCU is working with the voltage stored in the super cap., this way it does not waste energy.

4.2 Average power consumption

This is the average power consumption for the Watt Saver using a discrete AC/DC converter.

Table 4-2. Average power consumption

AC/DC Converter	AC/DC Converter Average power Consumption - No Load			AC/DC Converter plus the Watt Saver Average power Consumption - No Load			Power Saving Percent
	Current Consumption (mA)	Active Time Percent	Power Consumption (mW) at 110 VAC	Current Consumption (mA)	Active Time Percent	Power Consumption (mW) at 110 VAC	
Watt Saver (10 W)	15	100 %	1650	15	0.024 %	0.38	99.97
Watt Saver zero no-load power AC/DC converter (5 W)	1.2	100 %	132	6.5	0.005 %	0.03	99.97

These numbers were obtained as follow:

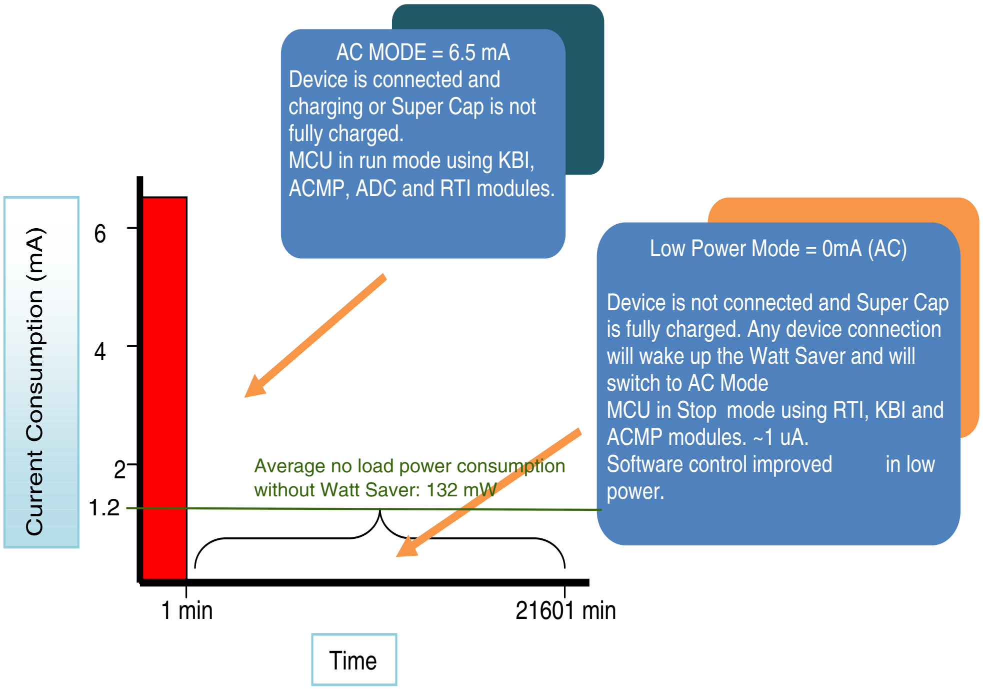
- Power consumption for AC/DC converter with 100% active time
  P = IV, so P = (1.2 mA) * (110 V) = 132 mW

- Active time percent AC/DC converter plus the Watt Saver
  Watt saver cycle is 21601 min (~15 days) = 100%.The active time using the Watt Saver is 1 min
  21601 min = 100%, 1 min = X%,
  X= (1)*(100) / 21601= 0.004629= ~0.005 %

- AC/DC converter plus the watt saver average power consumption no load.
  The watt saver cycle is 21601 min (~15 days), Watt Saver wastes 6.5 mA for 1 minute, while super cap. is charging..
  AC mode current consumption—0.0065 A * 1 min = 0.0065 A*min
  Low Power Mode current consumption—0 A* 21601 min = 0 A*min
  Total current consumption—0.0065 A*min
  Average current consumption = 0.0065 A*min / 21601 min = 0.000000301 A
  Average power consumption
  P = IV, so P = (0.000000301 A) * (110 V) = 0.000033 W = 0.033 mW

- Power saving percent
  The discrete AC/DC converter consumes 132 mW while in a no load condition, implementing Watt Saver to this AC/DC converter only consumes 0.033 mW
  132 mW = 100 %
  0.033 mW = X %
  X= (0.033)*(100) / 132= 0.025 %

This means that it consumes only 0.025 % or a power saving percent of 99.97 %.



AC MODE = 6.5 mA
Device is connected and charging or Super Cap is not fully charged.
MCU in run mode using KBI, ACMP, ADC and RTI modules.

Low Power Mode = 0mA (AC)

Device is not connected and Super Cap is fully charged. Any device connection will wake up the Watt Saver and will switch to AC Mode
MCU in Stop mode using RTI, KBI and ACMP modules. ~1 uA.
Software control improved in low power.

Average no load power consumption without Watt Saver: 132 mW

Figure 4-1. Current consumption

Appendix A
FAQs

1. **Is the 5 V output from the AC/DC connected directly to charge the super cap.?**

   No. There is a circuit used to control the charging current in the super cap. Refer to section “2.5 Super capacitor low voltage detection circuit” to see the charging circuit schematic..
2. **When a mobile phone is charging and the AC power is suddenly disconnected because of some power line issue. Does a mobile phone draw power directly from the super cap.?**

   No. It does not draw power directly from the super cap. The Watt Saver application contains hardware and software restrictions to avoid this particular issue. The Watt Saver only switches the relay to super cap. mode when an AC power line is active.
3. **If super cap. mode is active and the cable is unplugged, and then plugged back; How can the Watt Saver go back to the AC mode?**

   You need to push the wakeup button or connect a discharged device to go back to the AC mode.
4. **In the case where a mobile phone is fully charged and is still connected to the Watt Saver (wait cable mode); Can the WAIT_TIME parameter be modified according to customer´s needs?**

   Yes, this parameter can be modified according to customer´s needs. Refer to section “3.1 Demo board operation modes” for related information
5. **The charging current can be as high as 1 A, therefore the Vsense maximum can be as high as 10 V. Is it necessary to add a divider before the input to the ADC sampling if my circuit does not handle 10 V?**

   No. A divider is not necessary, because the maximum output voltage of the current measurement circuit is the same as the output of the AC/DC converter, which in this case is 5 V.

Appendix B
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.

Freescale, the Freescale logo, and Processor Expert are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© 2010-2013 Freescale Semiconductor, Inc.

Document Number DRM130
Revision 1, 10/2013