Freescale Semiconductor
User's Guide

Document Number: WCT1000SYSUG
Rev. 3.3, 09/2014

WCT1000 A11 Reference Design System User's Guide

1 Introduction

This document describes how to use the 5 W low power wireless charger transmitter WCT_A11 reference board designed by Freescale. The A11 Reference solution is compliant with the WPC Qi V1.1 specification. It is a low cost reference solution which can be easily customized through the FreeMASTER GUI tool.

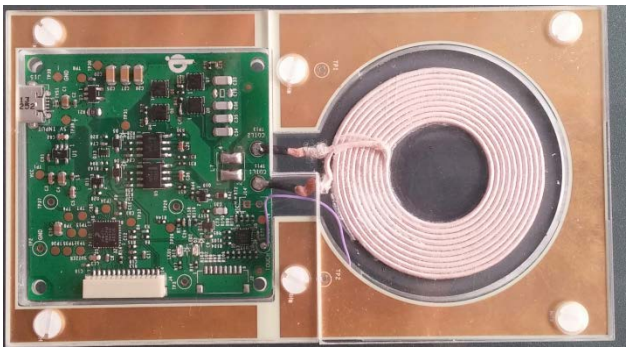

Figure 1 WCT_A11 reference board

Contents

1	Introduction	1
2	System Features	2
3	Package Checklist	2
4	System Block Diagram	2
5	Hardware Description	3
6	Getting Started	6
7	References	35

© Freescale Semiconductor, Inc., 2014. All rights reserved.

2 System Features

The WCT_A11 reference board has these features:

- Reference design that is compliant with low-power WPC Qi version 1.1 specification
- Integrated digital demodulation in chip
- Supports multiple types of Rx modulation signals (AC capacitor, AC resistor and DC resistor)
- Supports FOD and supports four types of Foreign Object protection
- Supports Resonance Shift FOD (RS FOD)
- Supports the Qi 1.1 receiver with 5 V DC@1A output power capability
- Super low standby power
- Full bridge topology with the freqency modulation power control strategy
- LED for Rx and Tx alignment indication
- Input voltage/current, coil current sensing for protection
- FreeMASTER GUI tool to enable customization and calibration

3 Package Checklist

Table 1 Package checklist

Name	Count
A11 Tx board with Tx coil	1
JTAG/UART debug board and cable	1
5V/2.4A DC power adapter	1

4 System Block Diagram

A11 Tx runs with Rx as shown in this figure to transfer power from the primary to the secondary side.

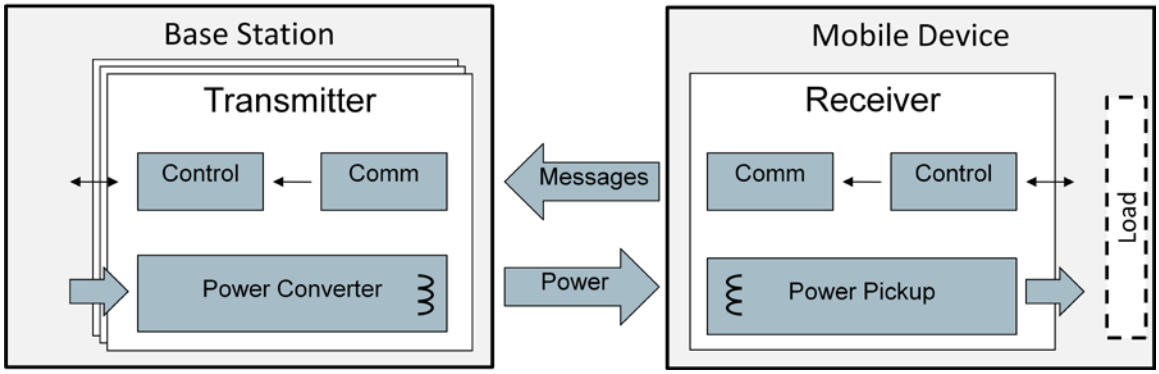

Figure 2 Wireless Charging system overview

Get WPC Qi information from: www.wirelesspowerconsortium.com/developers/.

5 Hardware Description

5.1 Reference board block diagram

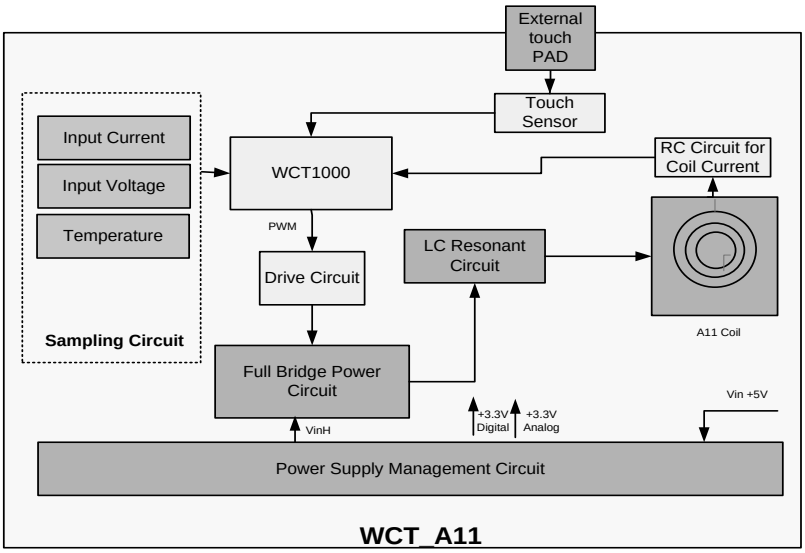

Figure 3 WCT_A11 board block diagram

5.2 Modules explanation

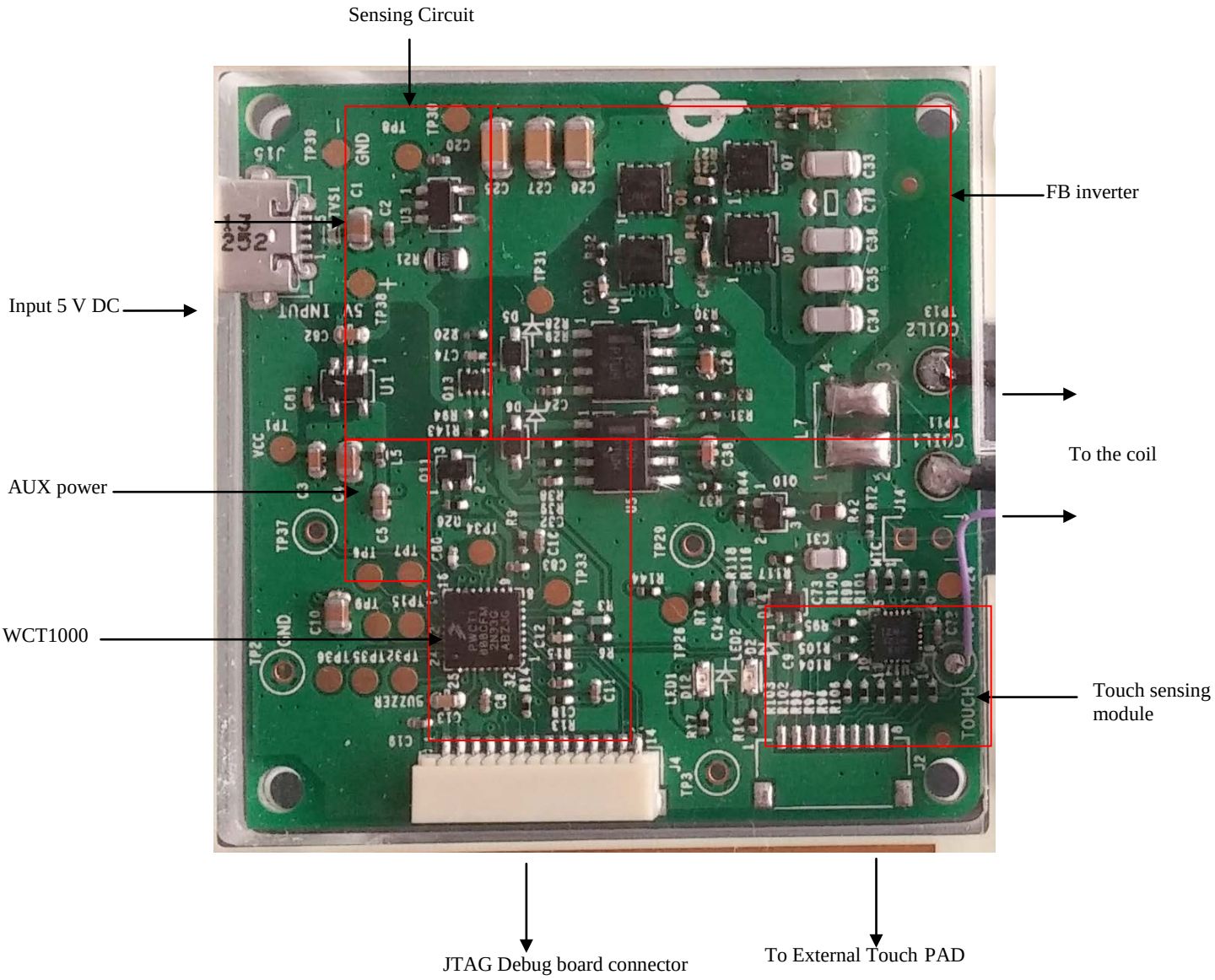

Figure 4 WCT_A11 board modules overview

- Controller
  Freescale WCT1000 chip is the central controller of the WCT_A11 board. It has rich I/O modules with low power consumption. It processes communication signals, controls power transfer start/stop, and controls a full bridge PWM inverter for output power control. These are the I/O modules used in this application:
  - Two PWM channels for full bridge DC/AC inverter control

- Three Timers for system timers and communication
- ADC for input voltage and current, coil current sampling
- GPIOs for pre-drivers control, low power, and LED control
- SCI for serial port debugging
- I2C for touch sensor MPR121 control

- Inverter
  The full bridge PWM control inverter converts 5 V DC input voltage to a higher AC voltage. The PWM frequency follows the WPC Qi specification, in the range of 110 KHz–205 KHz. The PWM duty is 50%, and starts duty control (50%–10%) when frequency is 205 KHz. Lower frequency gets a larger output power.
  Input voltage range: 4.5–5.5 V DC
  Output voltage range: 5–20 V AC
- Communication
  The communication of 2 kbps signal is demodulated from high frequency coil current AC signal (110 KHz–205 KHz). The RC sensing circuit gets resonant coil current and inputs to ADC for sampling. Digital demodulation module processes the input samples and extracts communication packets.
- Touch sensing for low power mode
  The board supports super low power mode with Freescale touch technology. When it is not charging, the controller shuts down the analog circuit power and activates the MPR121 touch sensor. WCT1000 runs in the LPSTOP mode to wait for the wakeup signal from touch. The user should connect an external electrode (placed around Tx coil) to TP28 or J2 on the board to enable touch. For details about the MPR121 sensor, navigate to freescale.com and search for “Touch Sensors”.

6 Getting Started

Freescale provides a SW package to modify WCT_A11 functions. The user can modify the system parameters or configurations to maintain system functionalities. For example, when either the Tx coil or main power components are changed, the user should calibrate to start the FOD. This document describes the basic debugging environment on WCT1000. For A11 software details, see the *WCT1000 TX Library User Guide* (WCT1000LIBUG).

6.1 System developing environment

Tx board debugging uses CodeWarrior and the FreeMASTER tool.

1. Set up the debugging connection as shown in Figure 5. The debugger and debugging board is between the PC and Tx board.
2. Connect a debugger (USBTAP or P&E-Multilink FX) to the JTAG port of a debugging board, and connect the debugging board to a Tx board through a 14-pins cable.

Figure 5 shows the connection and Figure 6 shows the real image.

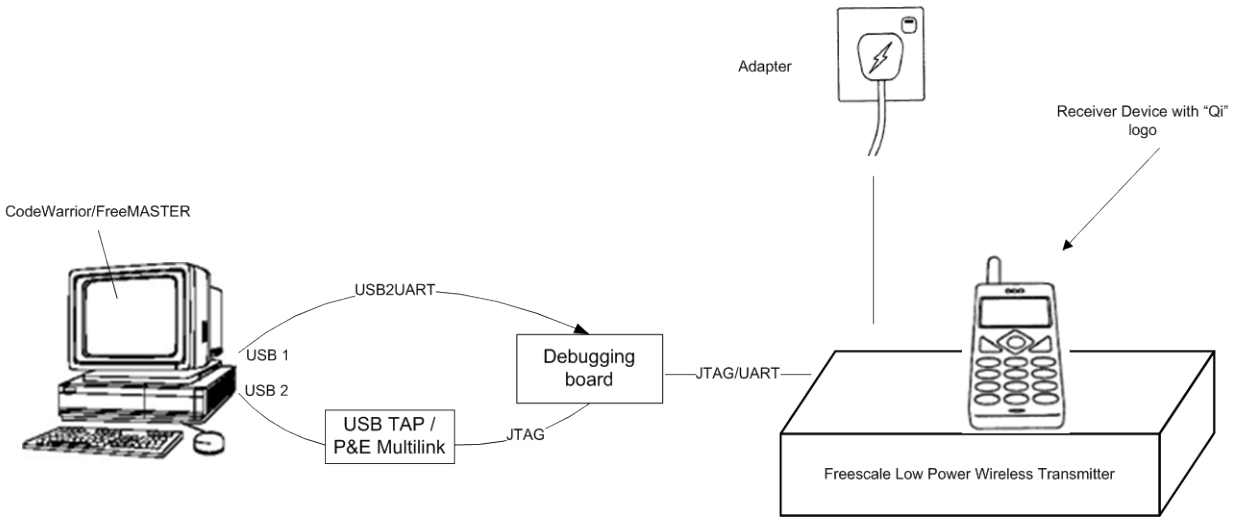

Figure 5 Debugging connections

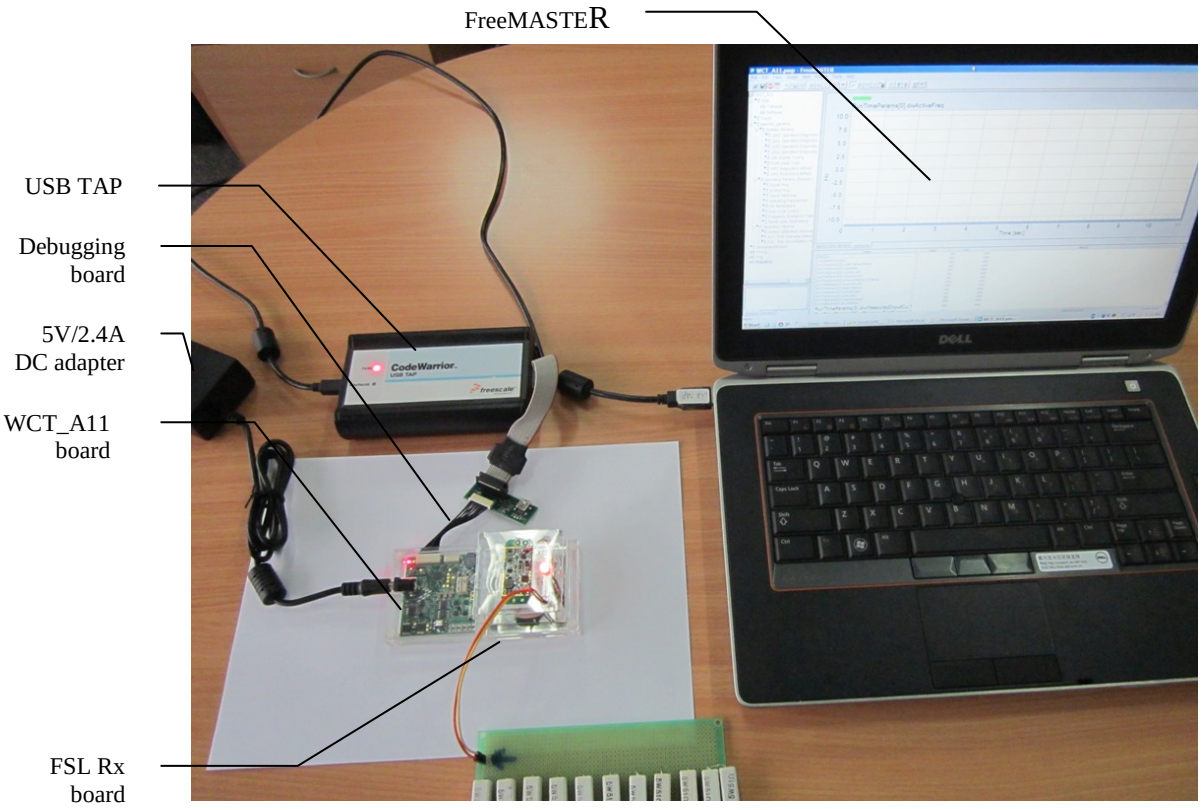

Figure 6 Developing environment

For details about the USB TAP debugger, see freescale.com and then search for “USB TAP for Once DSC”.

For details abut the P&E-Multilink FX debugger, see freescale.com and then search for “U-MULTILINK-FX”. It will take you to the “U-MULTILINK-FX: Universal Multilink FX High-Speed Development Interface” page.

6.2 Downloading and debugging firmware with CodeWarrior 10 IDE

6.2.1 Connecting the JTAG debugger

After CodeWarrior version 10 is installed, connect the Freescale MCU JTAG debugger, USB TAP, or P&E Multilink to the A11 board. The cable plug-in direction is shown in these figures.

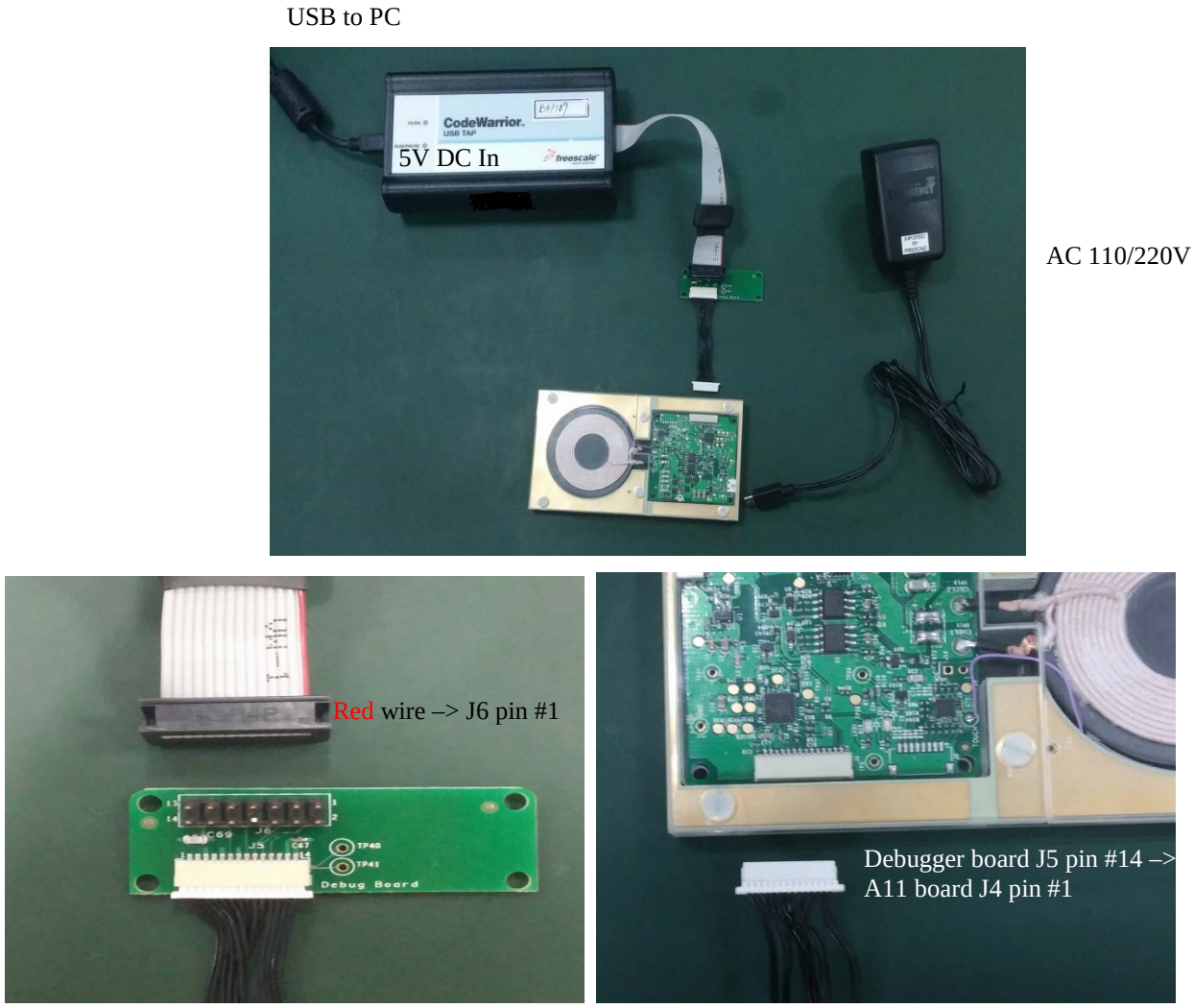

Figure 7 Debugger connections

When the debugger is plugged onto the PC, the device can be found in Windows “Device Manager”, as shown in these figures.

USB TAP

P&E Multilink

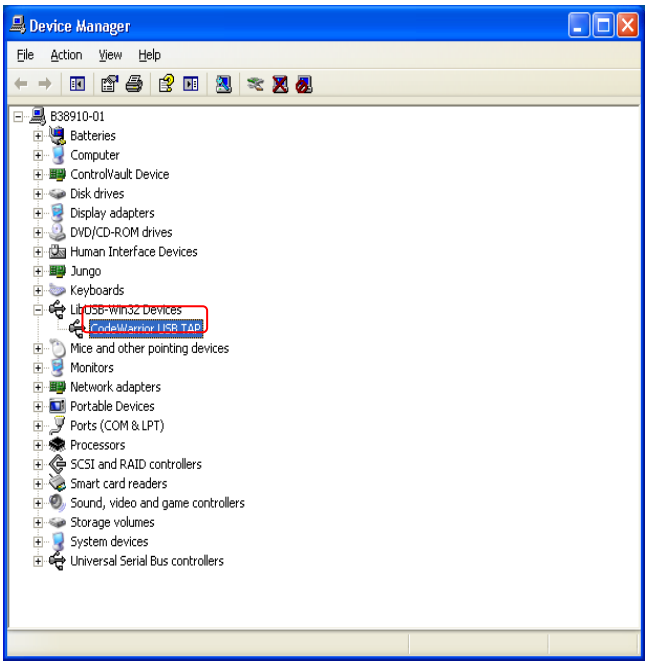

Figure 8 USB TAP debugger plugged in

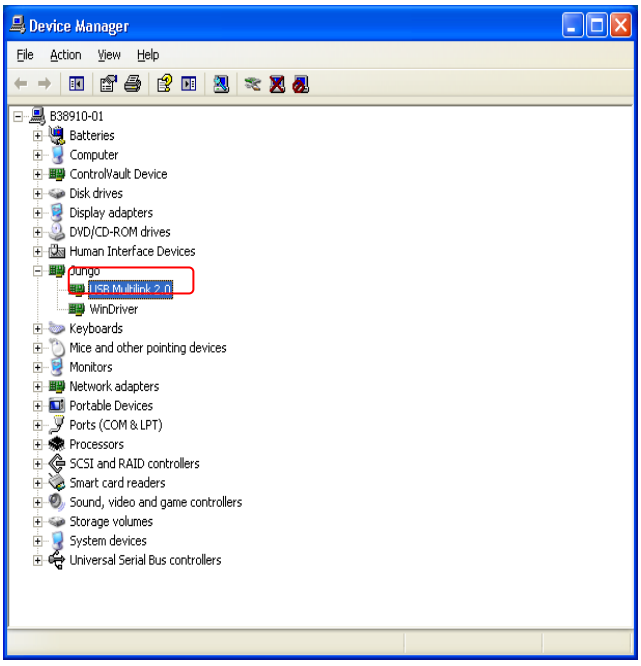

Figure 9 P&E multilink debugger plugged in

6.2.2 Downloading an existing WCT1000 project with CodeWarrior version 10

To download an existing WCT1000 project with CodeWarrior version 10, perform these steps:

1. Set the CodeWarrior version 10 Workspace.

   Open CodeWarrior version 10, and set the workspace to WCT1000 example project, wpt-tx.

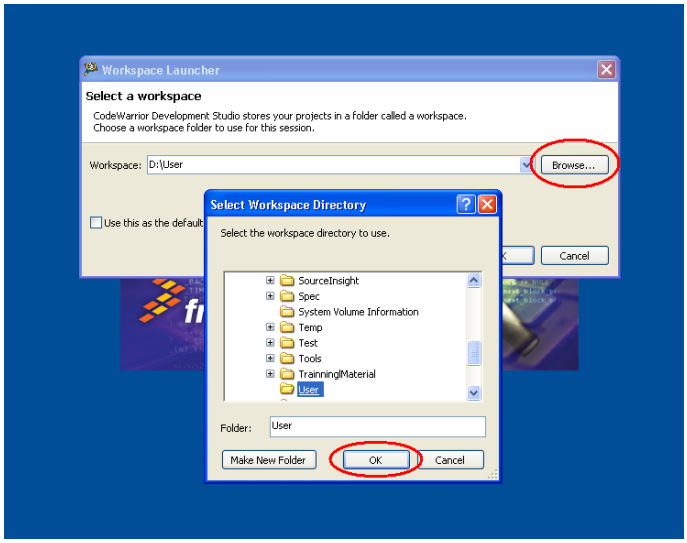

Figure 10 Setting the CodeWarrior version 10 workspace (1)

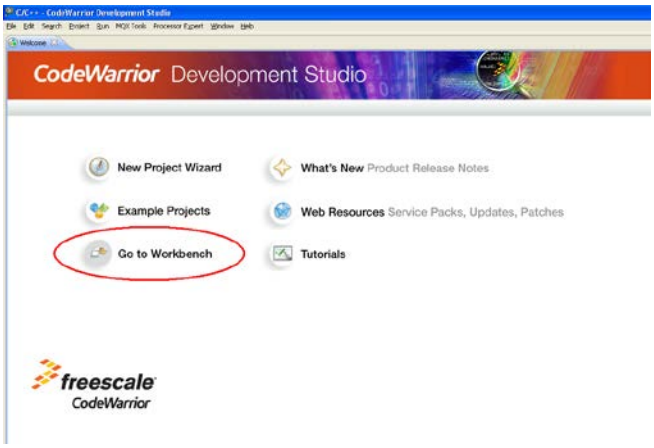

Figure 11 Setting the CodeWarrior version 10 workspace (2)

2. Update the MWCT1xxx service pack.
    a. Download the CW MCU v10.5 Wireless Charging MWCT1xxx Service Pack:
    b. Go to freescale.com and search for “CodeWarrior for MCU 10.5 Updates”.
    c. Install the service pack in CW10:
    d. Select “Help –> Install New Software” from the tool bar.

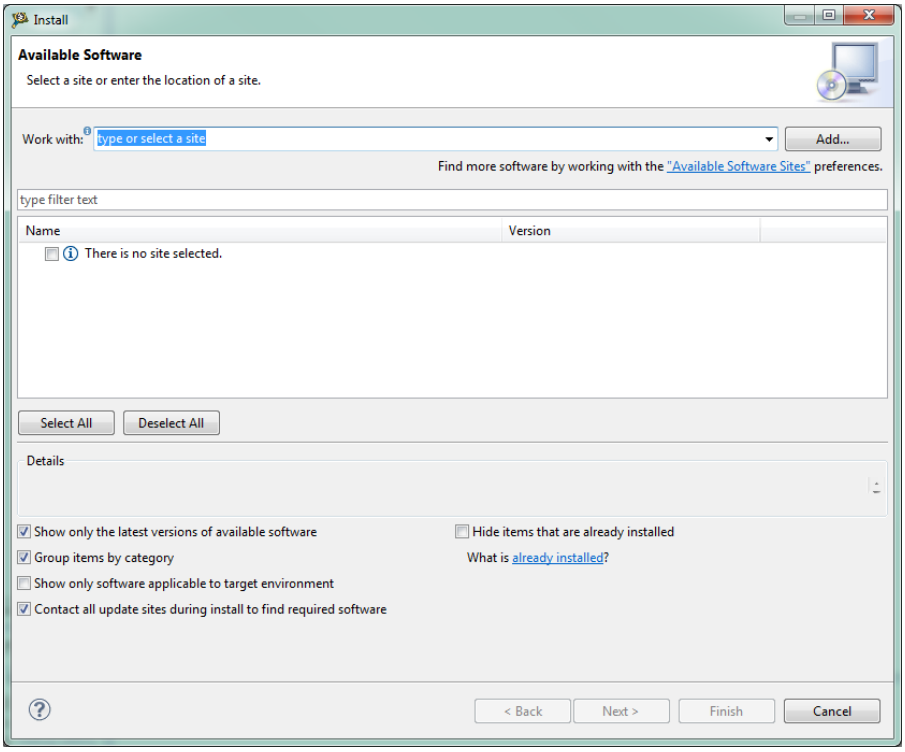

Figure 12 Updating the MWCT1xxx service pack (1)

    e. Select “Add –> Archive” to set the path of the upgrade patch.

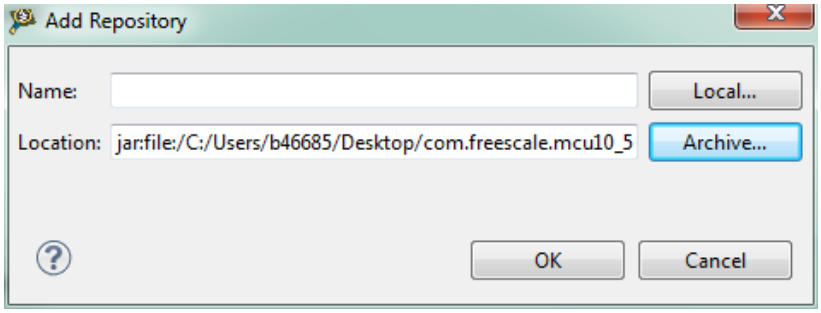

Figure 13 Updating the MWCT1xxx service pack (2)

f. Select “MCU v10.5 DSC Service Packs” and click “Next” to install the patch.

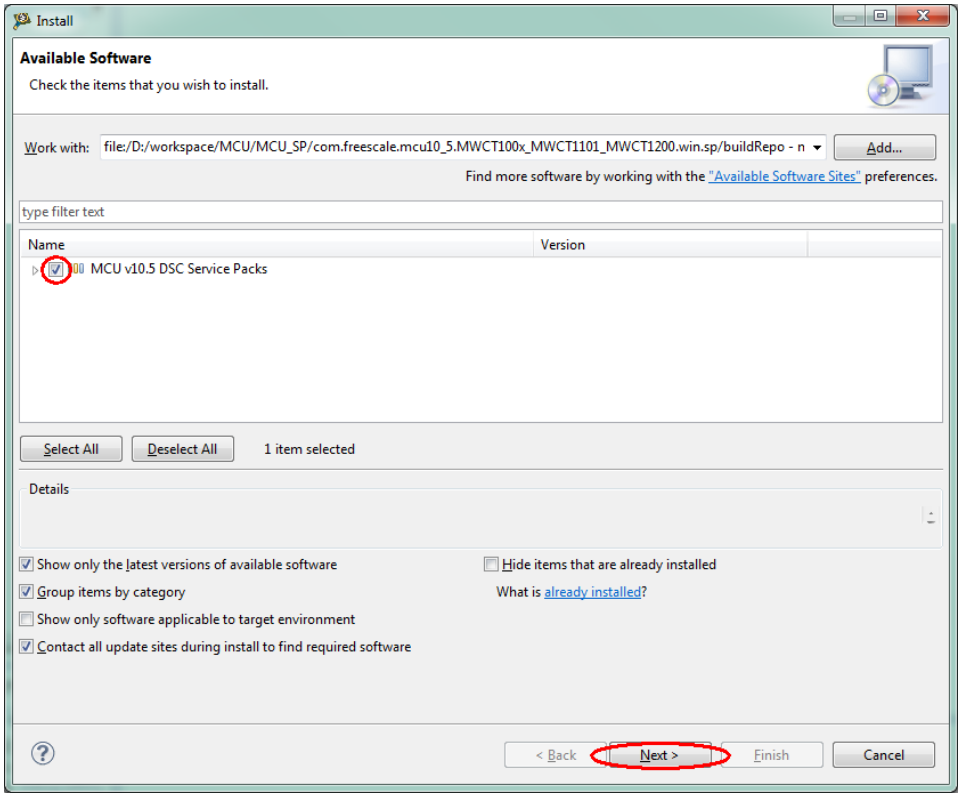

Figure 14 Updating the MWCT1xxx service pack (3)

g. Restart Code Warrior10 automatically after installation is completed, and then the MWCT1xxx Service Pack can be used.

3. Import the project.

    a. Right-click in the “CodeWarrior Projects” window and choose “Import” to import an existing project, as shown in these figures. If the “CodeWarrior Projects” window is not displayed, open it through Window –> Show View –> CodeWarrior Projects.

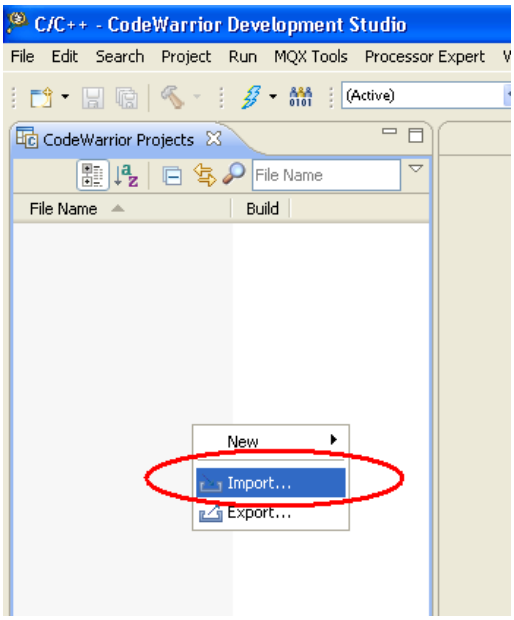

Figure 15 Importing a project (1)

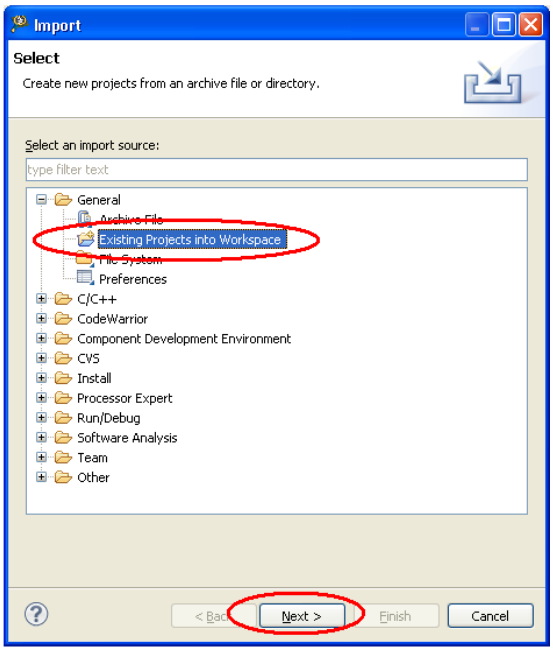

Figure 16 Importing a project (2)

b. Select the project directory, as shown in these figures.

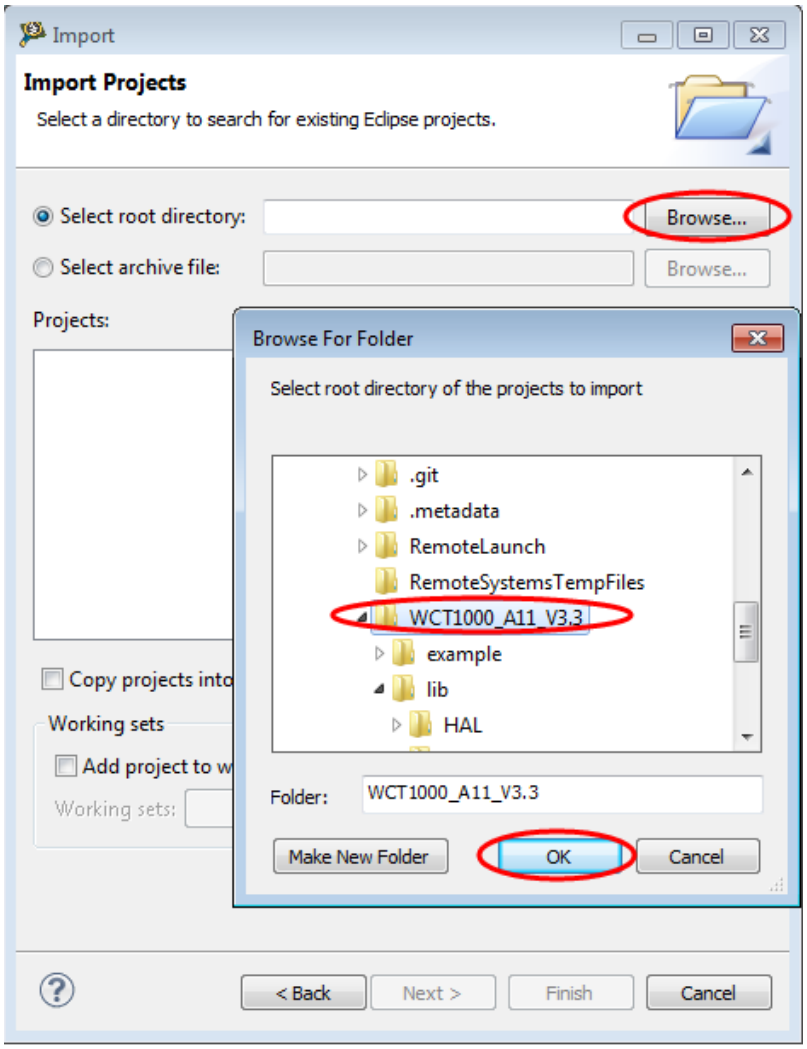

Figure 17 Importing a project (3)

c. Select the project found by CodeWarrior version 10.

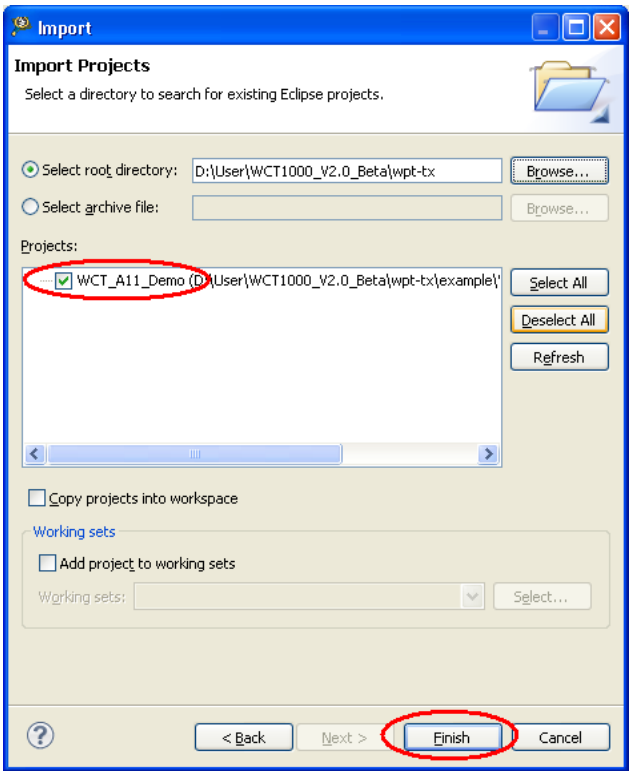

Figure 18 Importing a project (5)

4. Build a project.

a. Click the project name in the project window shown below, and select build configurations –> Debug or Release build. Debug build includes more debug information.

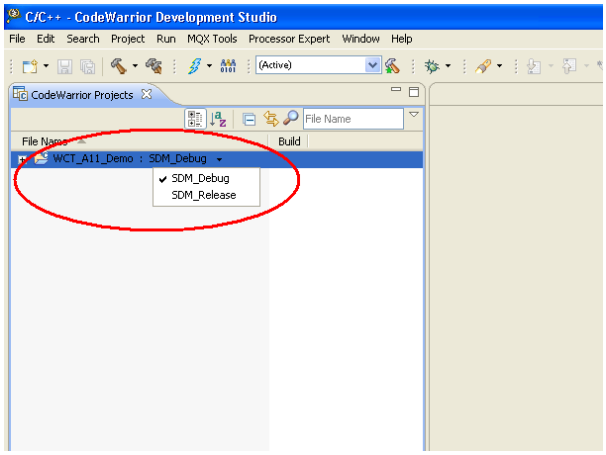

Figure 19 Building a project (1)

b. Right-click the project name, “WCT_A11_Demo : SDM_Debug”, and then select “Build Project”, “Clean Project”, or “Close Project”. You can also perform build from “Project”.

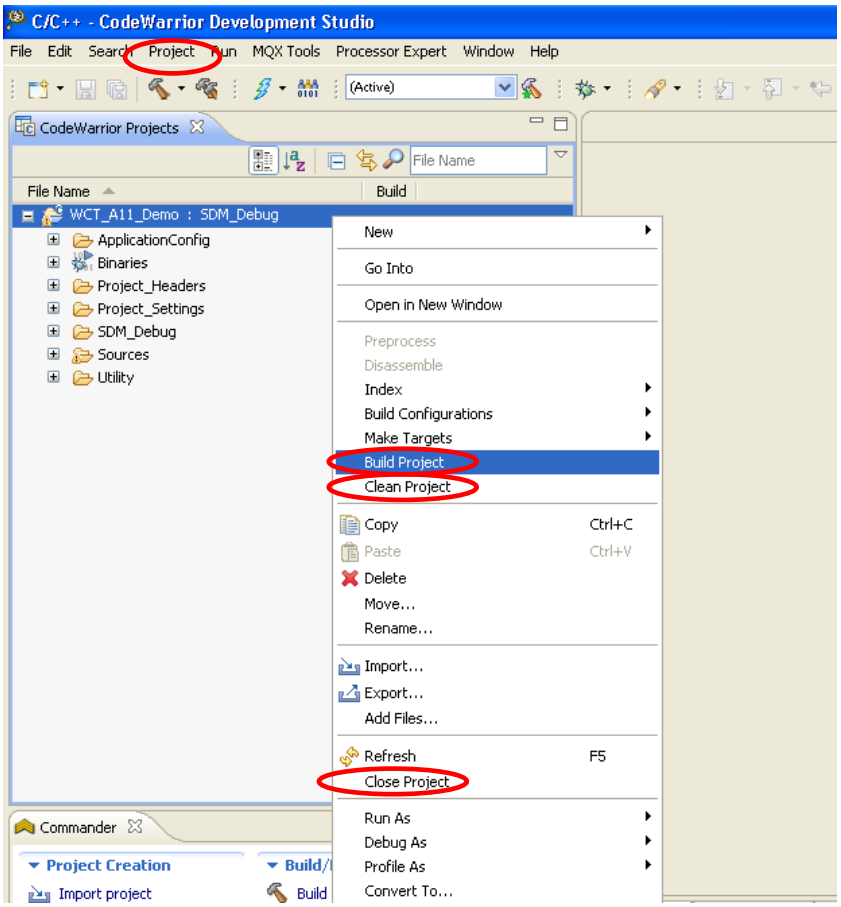

Figure 20 Building a project (2)

5. Download the project.

    a. After the project is built, the MCU binary files are generated to a folder, with the same name as the build configuration name, “SDM_Debug”.

    b. Download the project from the “Debug” drop-down list, or from “Run –> Debug”.

    c. In “Download Configurations”, select a download configuration according to your build configurations and debugger type, USB TAP, or PnE Multilink.

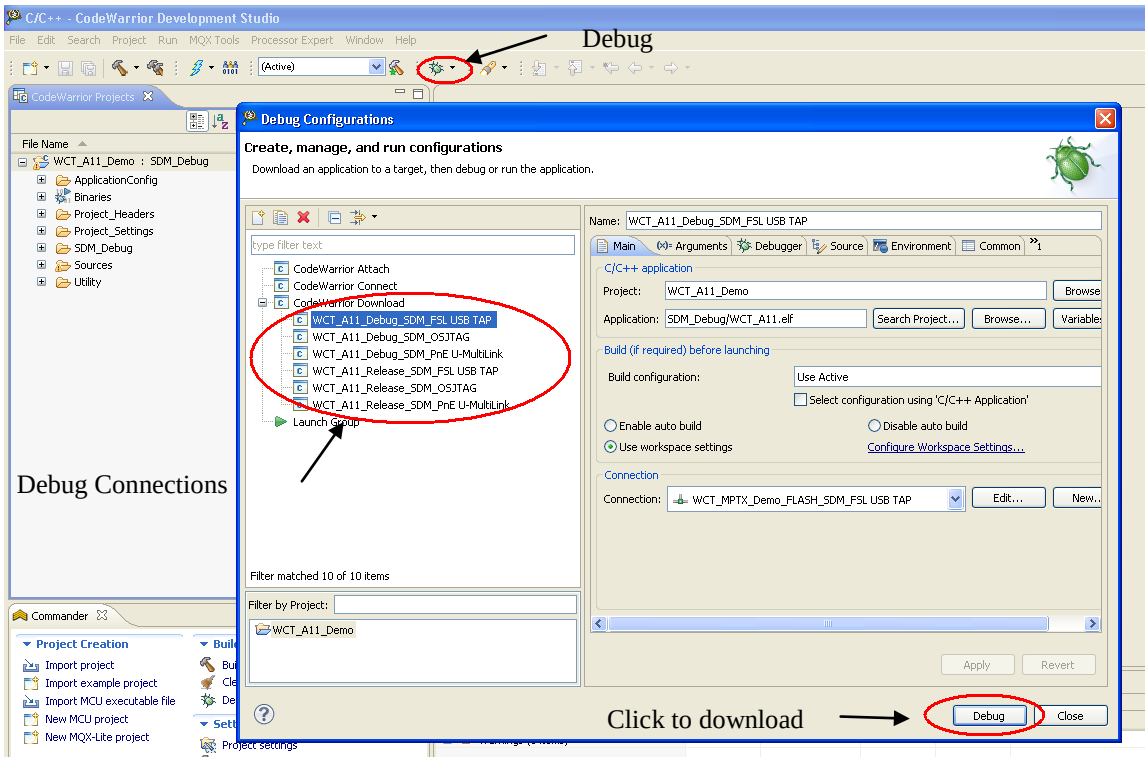

Figure 21 Downloading the project

d. After the project is downloaded, the MCU stops at startup code. Press F8 to release the MCU.

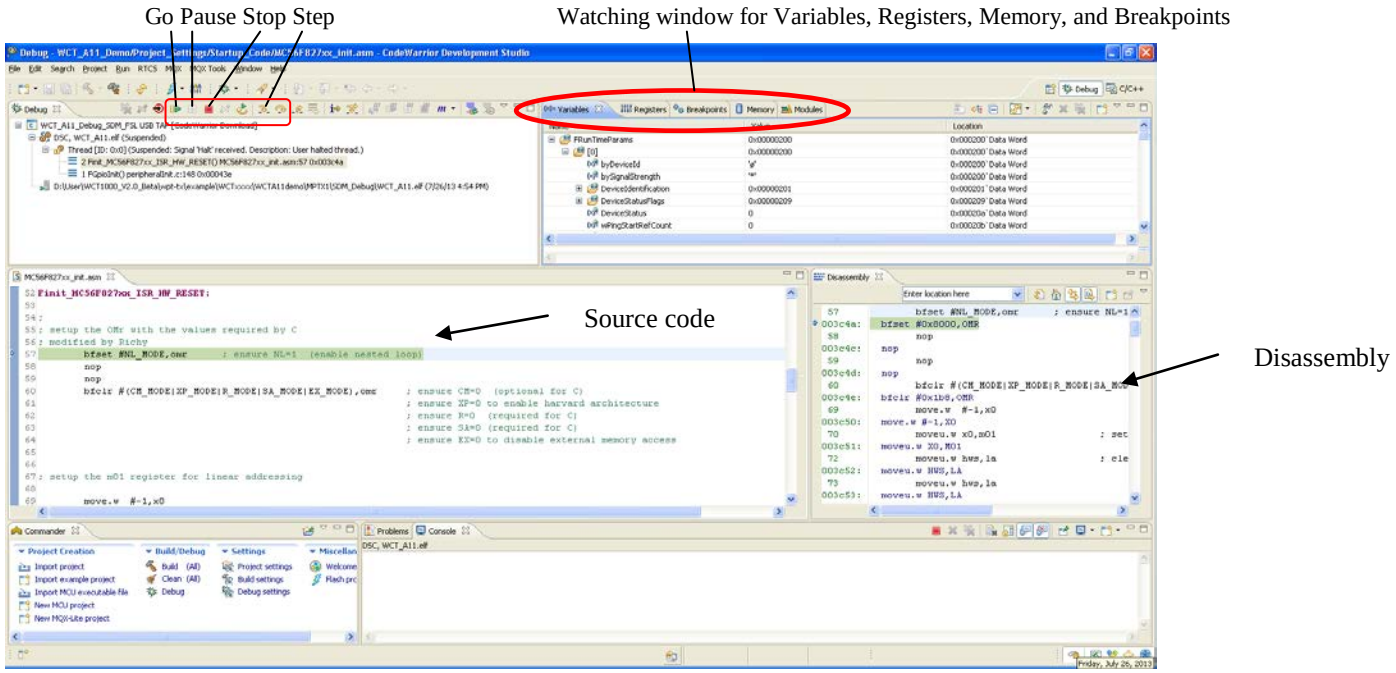

Figure 22 Project downloaded

6.2.3 Downloading an existing WCT1000 bin file (.s) with CodeWarrior version 10

To flash an .s file, perform these steps:

1. From the Flash Programmer drop-down list, select “Flash File to Target”.

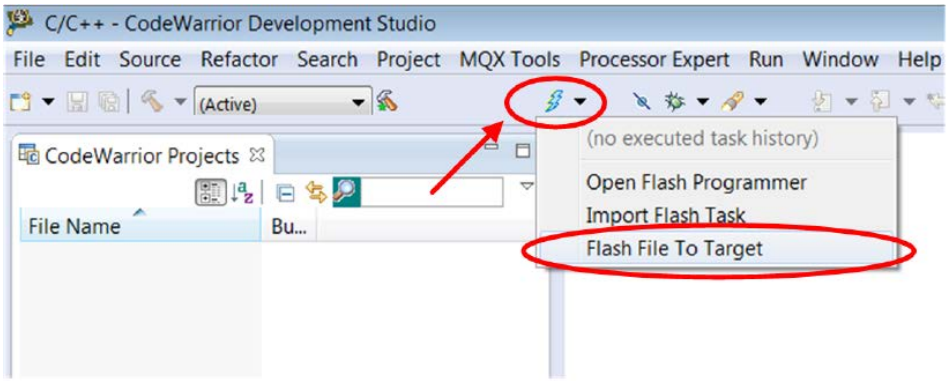

Figure 23 Selecting Flash File to Target

2. Click “New” to create a new connection.

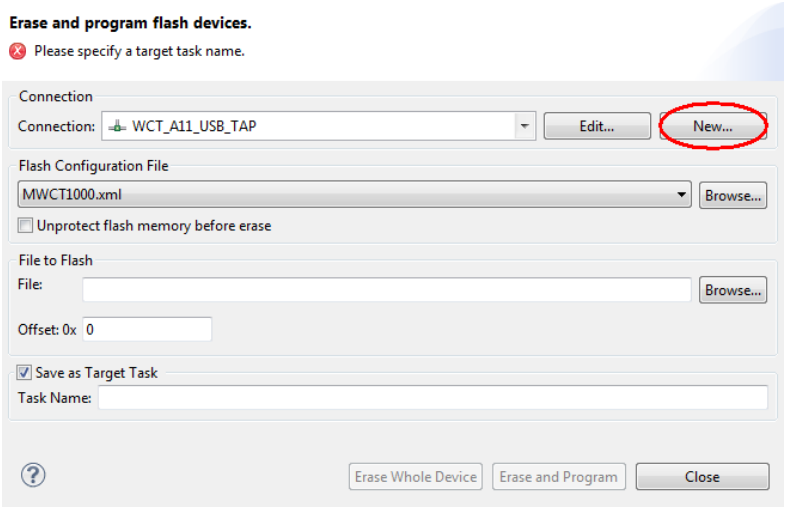

Figure 24 Creating a new connection

3. Enter a connection name and click “New” to create a target.

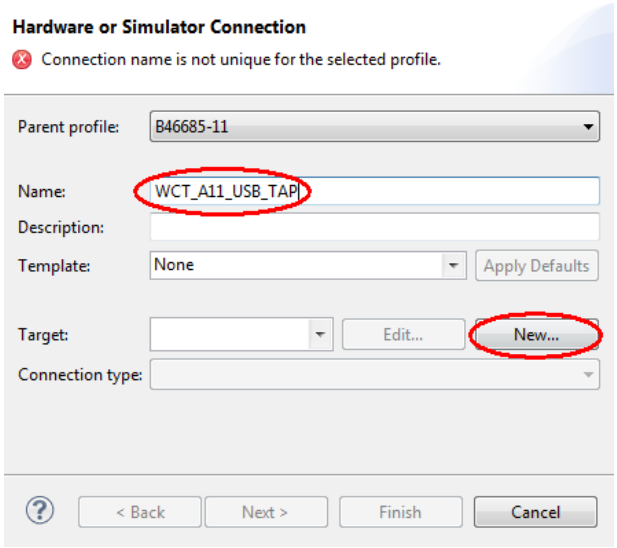

Figure 25 Creating a target

4. Enter a target name, and then select “MWCT1000” from the “Target type” drop-down list.

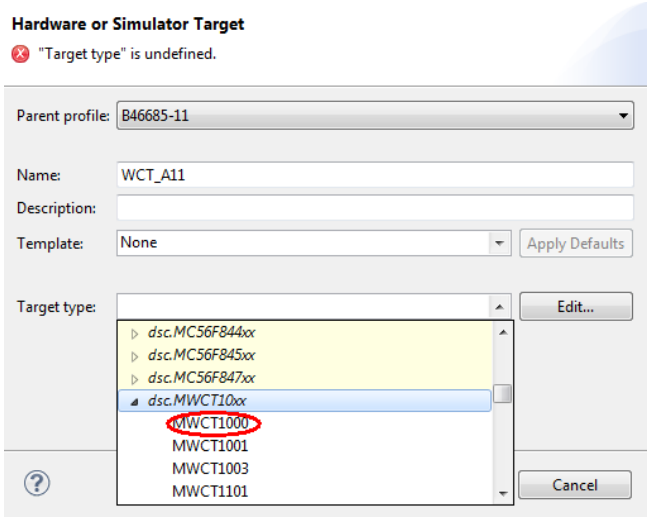

Figure 26 Selecting MWCT1000

5. Select “Execute reset” and “Initialize target”, set the initialization file path to the CodeWarrior version 10 installation folder, and select “MWCT1000.tcl” for the WCT1000 chip. The general path is:
C:/Program files/Freescale/CW10.5/CW MCU v10.5/MCU/lib/wizard_data/DSC/DataBase/init_files

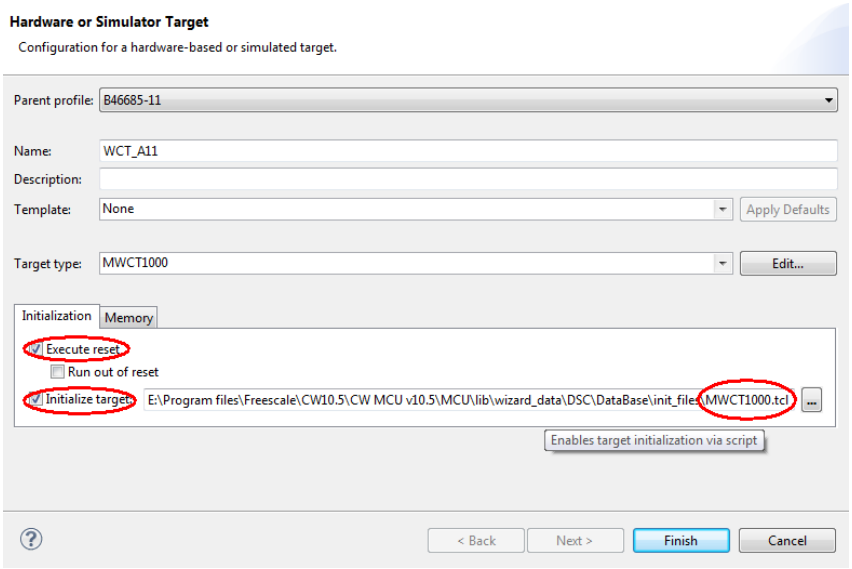

Figure 27 Setting the initialization file path

6. Set the memory configuration file path. For the WCT1000 chip, it is MWCT1000.mem, located under the CodeWarrior version 10 installation folder. Then, click “Finish”.

   The general path is:

   C:/Program files/Freescale/CW10.5/CW MCU v10.5/MCU/lib/wizard_data/DSC/DataBase/mem_files

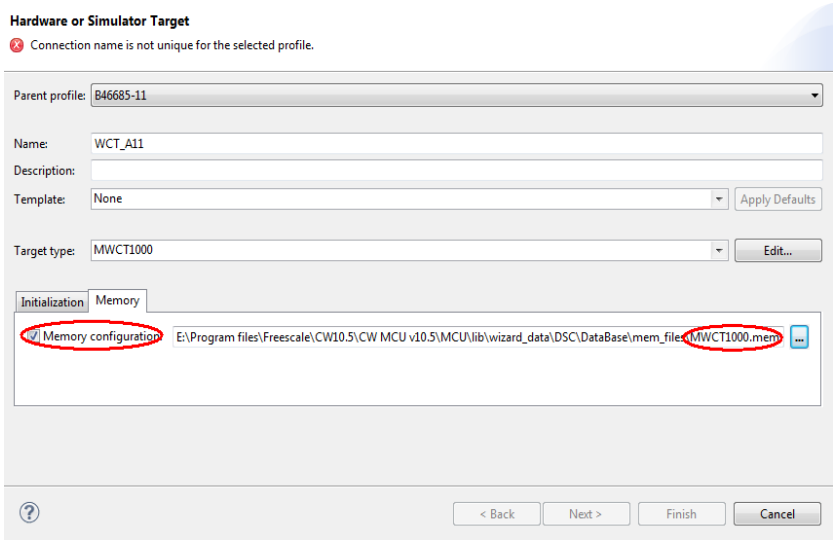

Figure 28 Setting the memory configuration file path

7. Select “USB TAP” or “P&E DSC Multilink/Multilink Universal/Cyclone Pro/OSJTAG” for the connection type. Then click “Finish”.

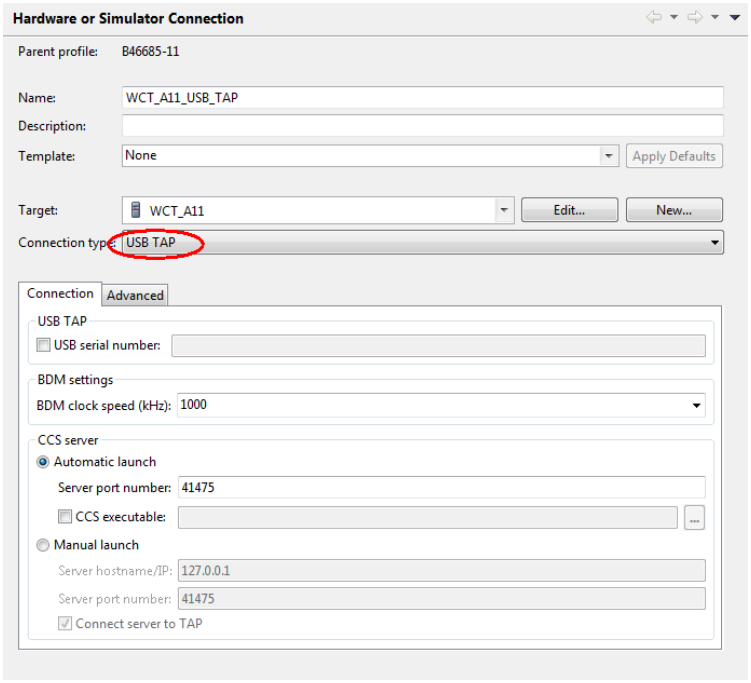

Figure 29 Setting the connection type

8. Set the Bin file path. Before downloading, you can save the configuration to the workspace for next-time downloading. Click “Erase and Program”.

   Note: The file path should contain only English letters. Otherwise, the flash cannot recognize it.

   For the new board, "Erase Whole Device" should be executed when you select "WCT_A11.elf" as the Bin file.

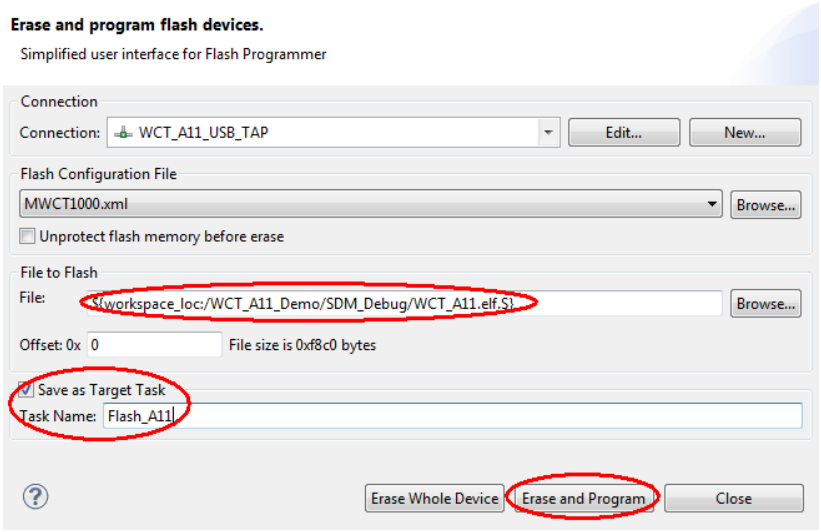

Figure 30 Setting the bin file path

9. The flashing progress is displayed in the CodeWarrior version 10 console window. After flashing is completed, reset the board to run the WCT1000.

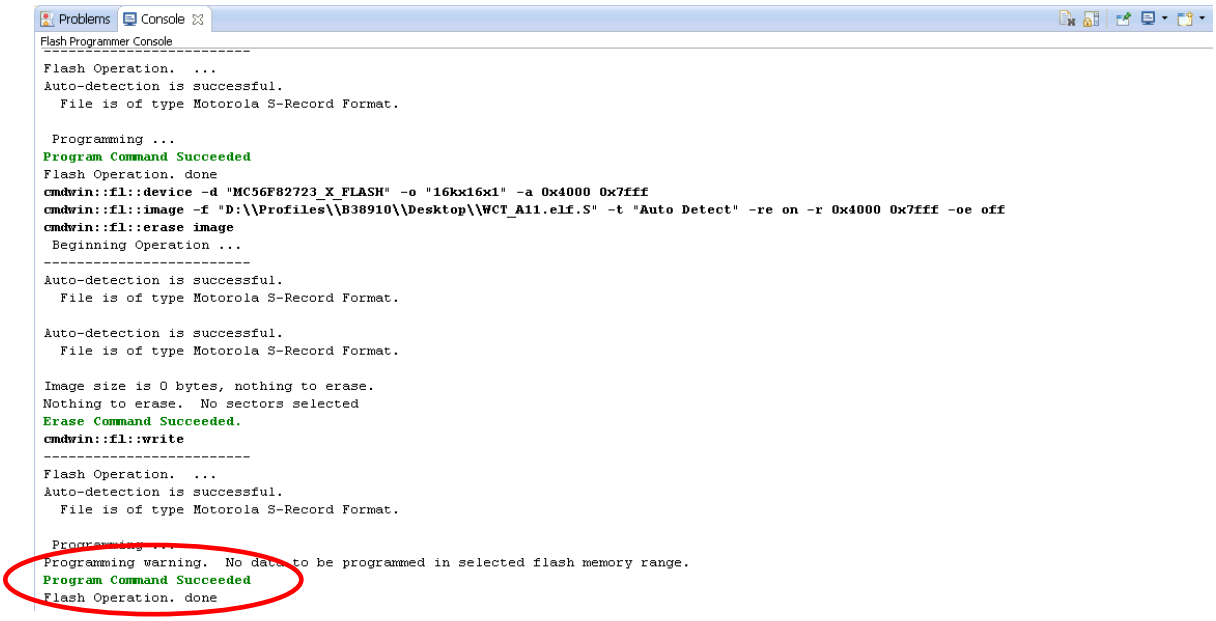

Figure 31 Flashing completed

6.2.4 Using the FreeMASTER GUI for calibration

Freescale provides the FreeMASTER GUI tool to calibrate and tune parameters. FreeMASTER configuration file WCT_A11.pmp is saved under /wpt-tx/example. See the *WCT1000 A11 Reference Design Calibration User's Guide* (WCT1000CALUG) for calibration. See the *WCT Runtime Debug User's Guide* (WCT1000RTDUG) for A11 paramters tuning. For the FreeMASTER tool, see freescale.com/Freemaster.

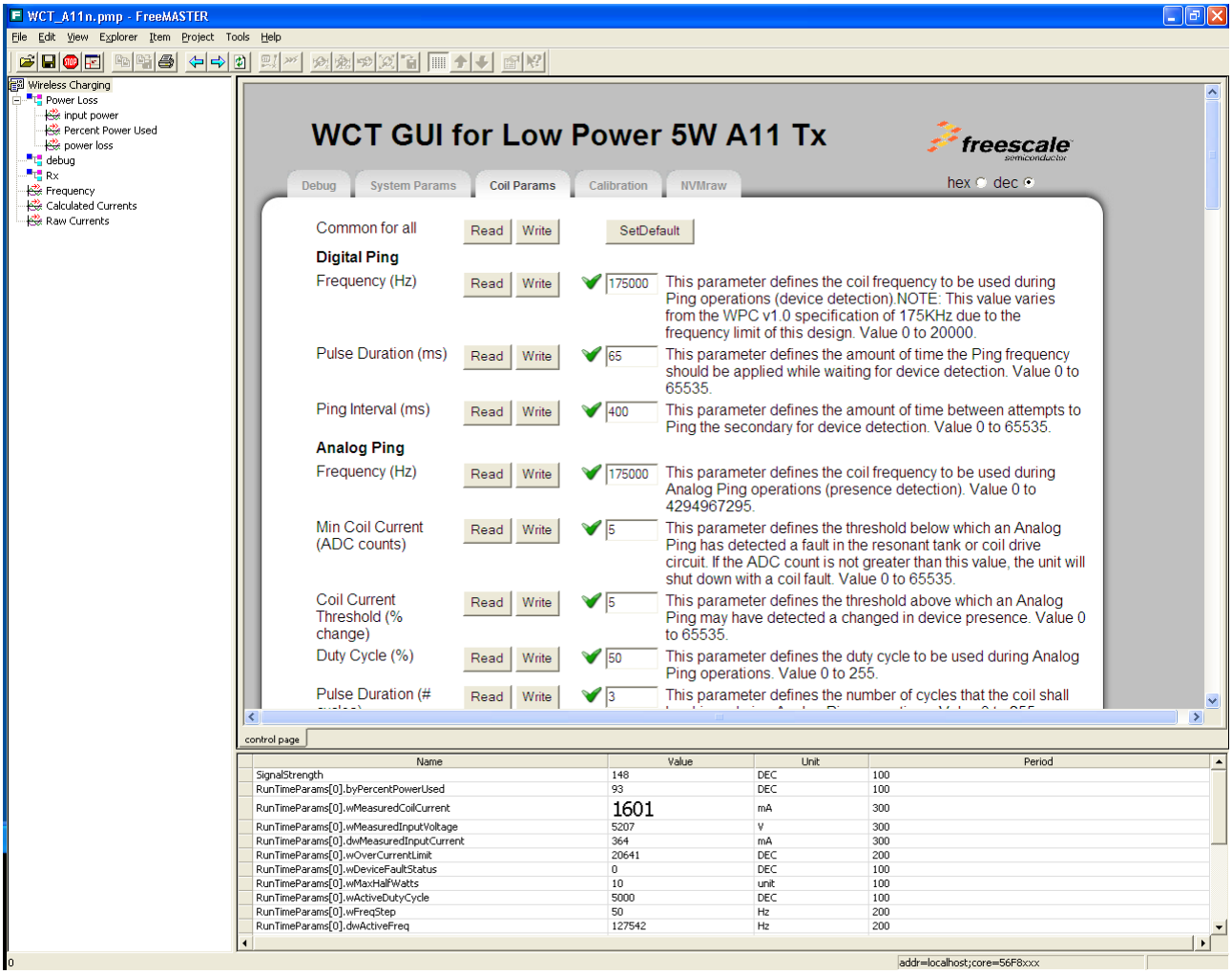

Figure 32 FreeMASTER GUI tool

To set up a FreeMASTER connection to the target board, perform these steps:

1. Set a symbol file for your project.
   Select a symbol file from FreeMASTER Project –> Options –> MAP Files, as shown in the following figure.

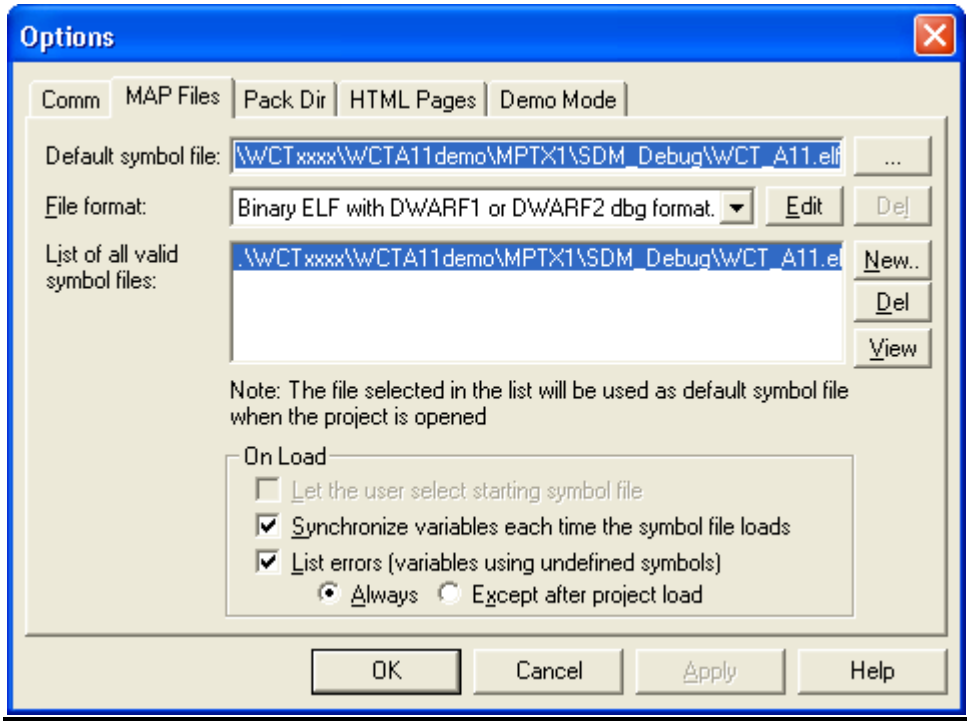

Figure 33 Selecting a symbol file

2. Configure the USB TAP debugger.
   Select “JTAG/OnCE” in Freemaster Project –> Options –> Comm as shown in the following figure.

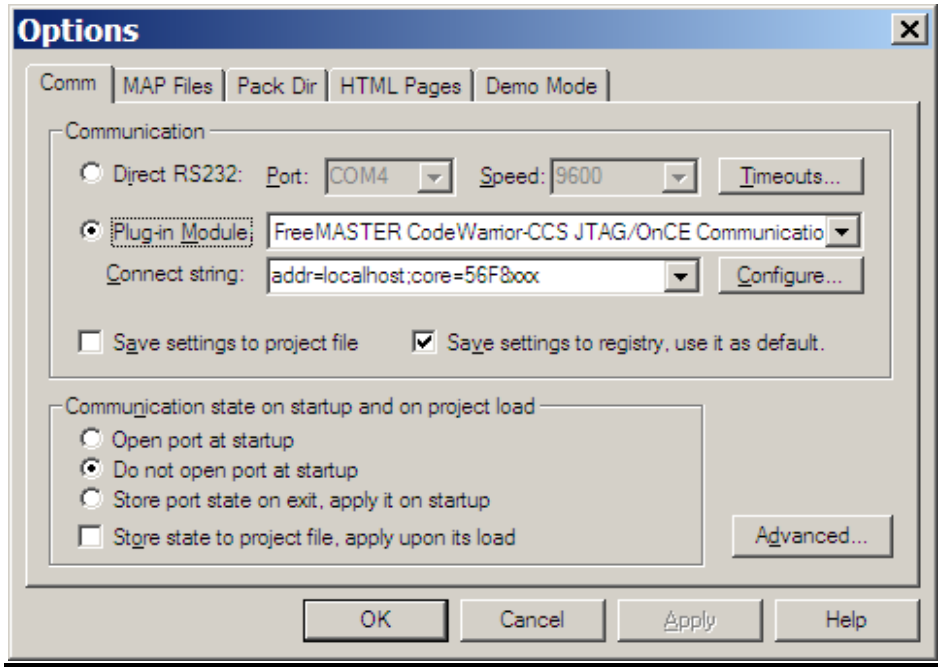

Figure 34 Options dialog box

3. Configure the P&E Multilink FX debugger.
   Select “FreeMASTER BDM JTAG/OnCE” in Project –> Options –> Comm as shown in the following figure.

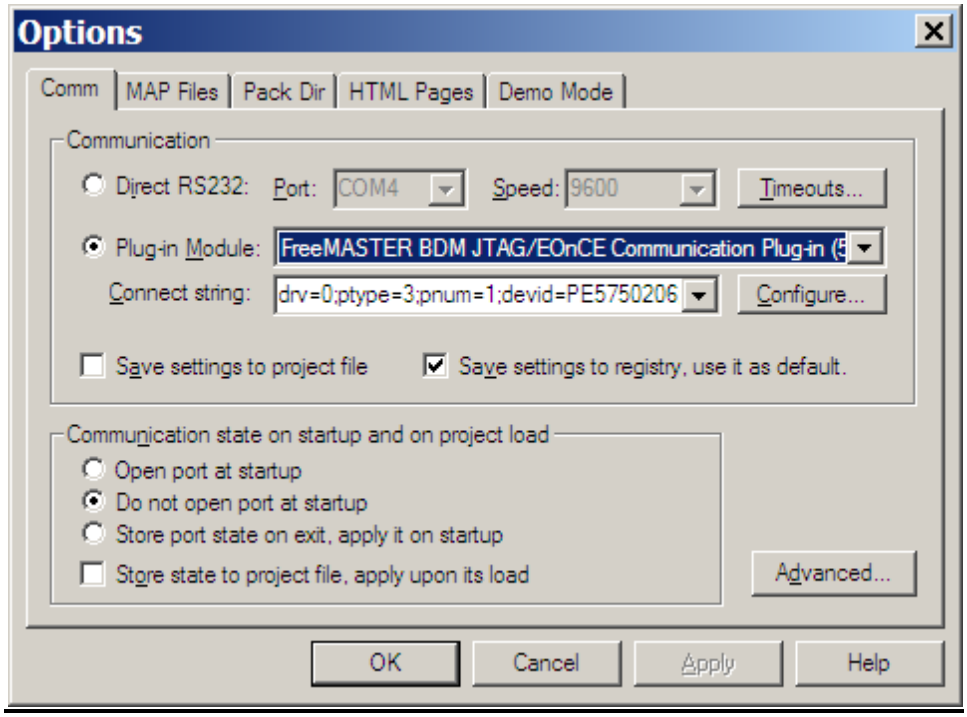

Figure 35 Options dialog box

6.2.5 Enabling or disabling board functions

Freescale provides full-featured wireless charging functions on the reference board. If you do not need a function, you can disable it with the definitions in the header file or with the parameters in the FreeMASTER GUI.

These header files are used to enable or disable functions, and configure a low-level driver.

/lib/MPTX_A1/wct_a11/wct_hal_cfg.h

/example/WCTxxxx/WCTA11demo/MPTX1/Sources/application_cfg.h, peripheral_cfg.h

In application_cfg.h, you can configure these functions:

- Low power mode enable/disable
  `#define LOW_POWER_MODE_SUPPORTED TRUE // FALSE for calibration or debug on FreeMASTER GUI.`
- Low power mode by Touch enable/disable
  `#define LOW_POWER_MODE_BY_TOUCH TRUE // FALSE if touch sensor is not needed.`
- Low power mode by Analog Ping enable/disable
  `#define LOW_POWER_MODE_BY_ANALOG_PING TRUE // FALSE if low power mode when analog ping is not needed.`
- Sound enable/disable
  `#define SOUND_SUPPORTED TRUE // FALSE if sound is not needed.`
- FOD enable/disable
  `#define FOD_ENABLE TRUE // FALSE if you don’t want FOD working.`
- Freemaster support enable/disable

```
#define FREEMASTER_SUPPORTED          TRUE    // FALSE if you don't want Freemaster working.
```

Note: Low power mode by Touch mode and low power mode by analog ping, only one can work at a time.

6.2.6 Enable RS FOD function

RS-FOD (Resonance Shift FOD) is one optional solution to detect FO in standby mode, when Rx is not on the surface of Tx. This can ensure that the charging never starts power transfer if FO is present. It does not belong to Qi 1.1, so the feature is disabled in the demo image. However, you can enable it by the following steps if the feature is required. Additionally, the feature cannot work with the touch sensor feature simultaneously.

The steps of enabling this function are as follows:

1. Right-click the project, and choose “Properties”.

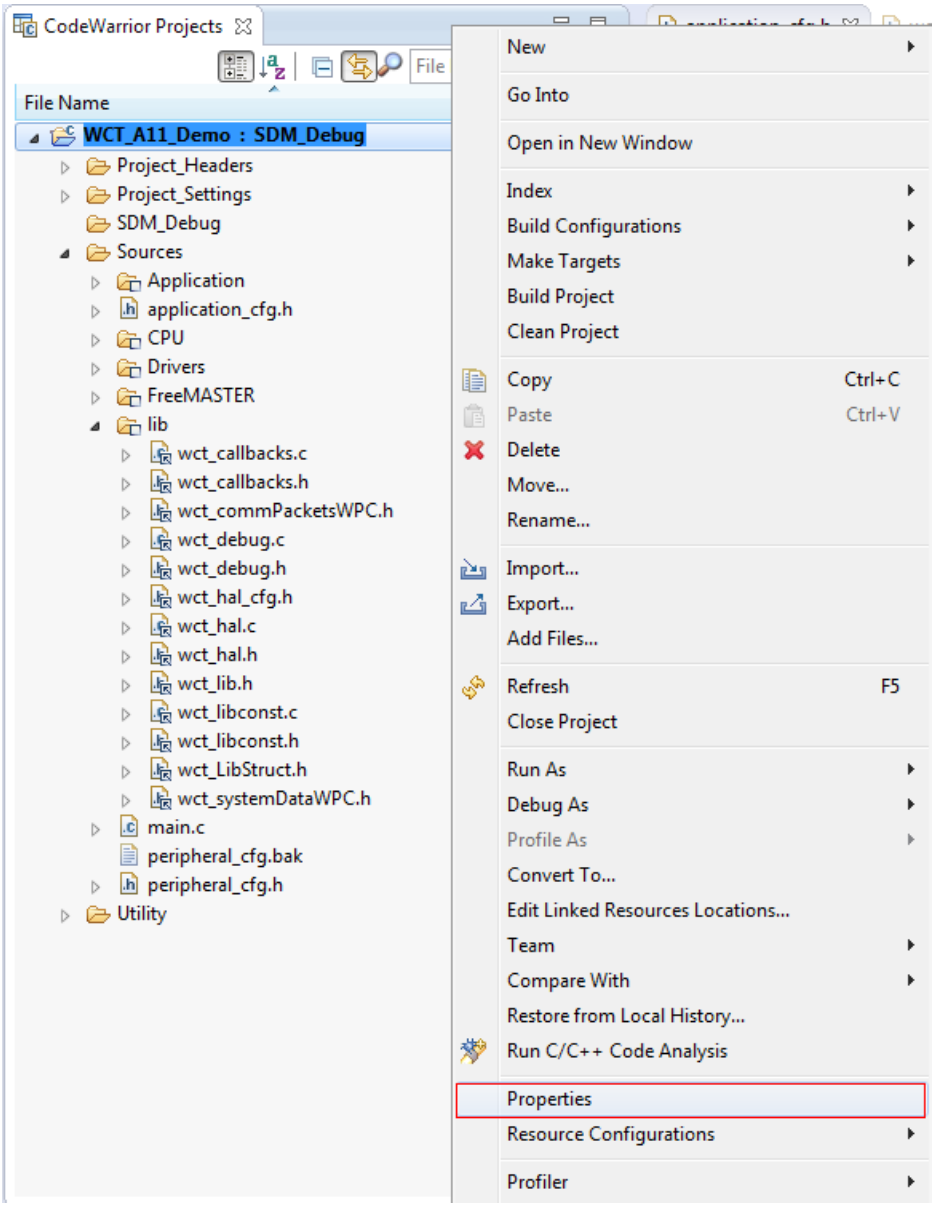

Figure 36 Choosing the properties of project

2. Choose C/C++ Build -> Setting -> DSC Linker -> Input, as shown in the following figure.

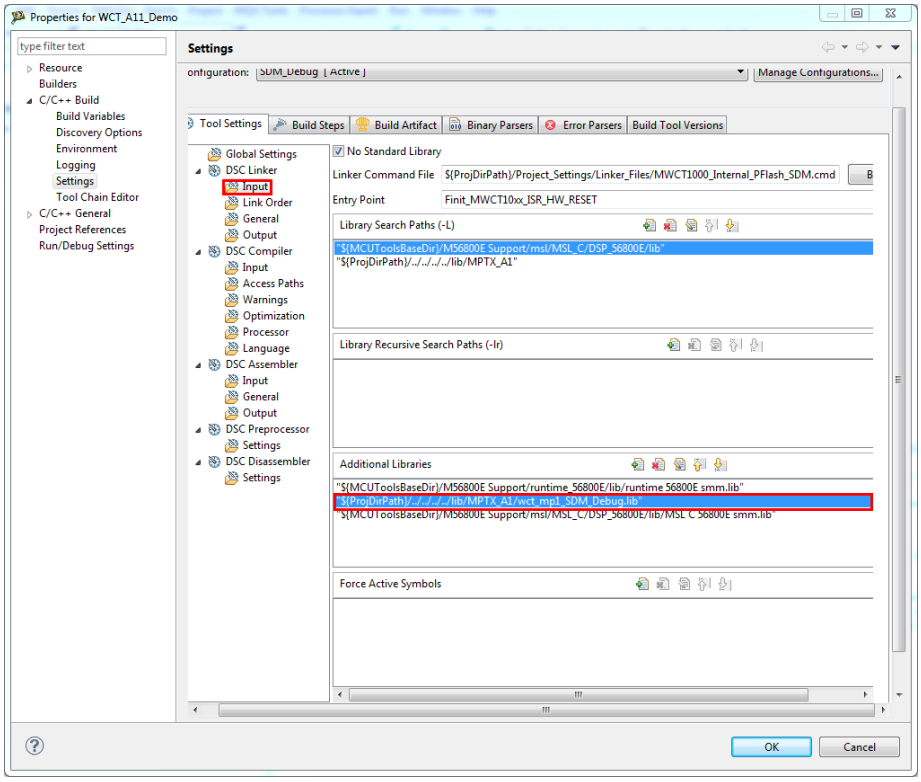

Figure 37 Finding the additional libraries option

3. Double click "…/…/wct_mp1_SDM_Debug.lib" in "Additional Libraries", and change to "wct_mp1_RSFOD_SDM_Debug.lib". Then click "Save" and exit.

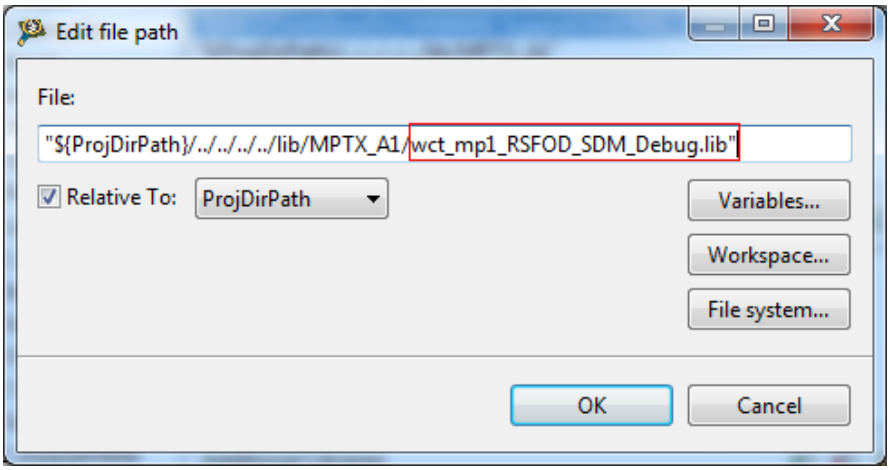

Figure 38 Modifying the relevant library

4. Set the macro RESONANCE_SHIFT_FOD to TRUE in wct_libconst.h, and confirm that macro LOW_POWER_MODE_BY_TOUCH in application_cfg.h is FALSE. Rebuild the project to make the RS FOD work.

6.3 Test

6.3.1 Signals on the board

Main signals on the A11 board are shown in this figure.

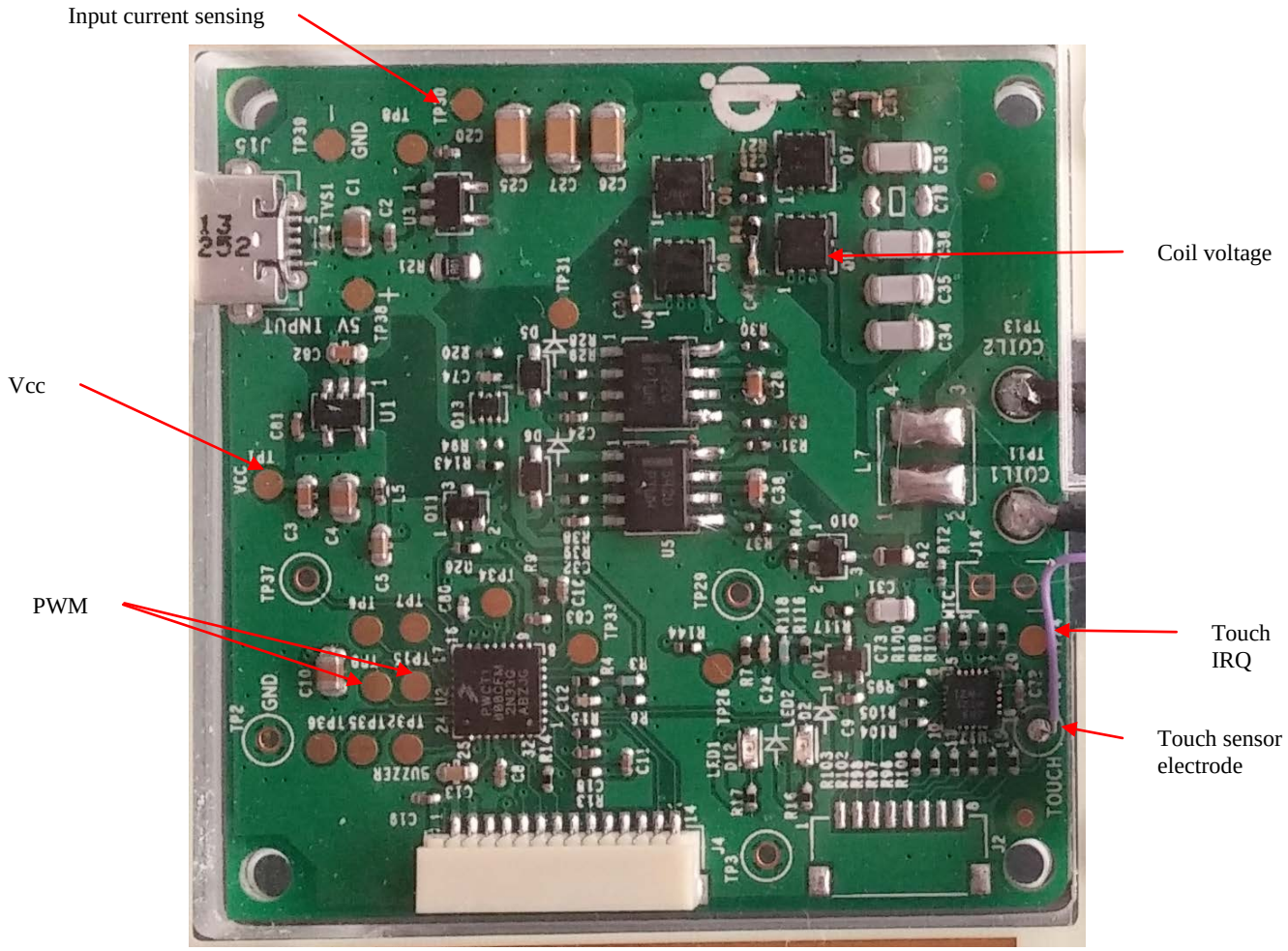

Figure 39 Test points on WCT_A11

- TP1: Vcc, controller input voltage 3.3V
- TP4: AUXP_Ctrl, power shut down signal when idle state
- TP8: Input current sensing
- TP26: Coil current sensing
- TP9&15: PWM1&2, PWM signals to pre-driver
- TP13: LC-Resonant voltage on the coil
- TP24: Touch_IRQ, wake up signal from touch sensor when receiver put on
- TP28: Touch sensing signal to an external electrode

6.3.2 Test environment

Set up the WCT_A11 test environment as shown in this figure by using the DC power supply and electronic load for input source and output load. Get the system efficiency by measuring the input and output power.

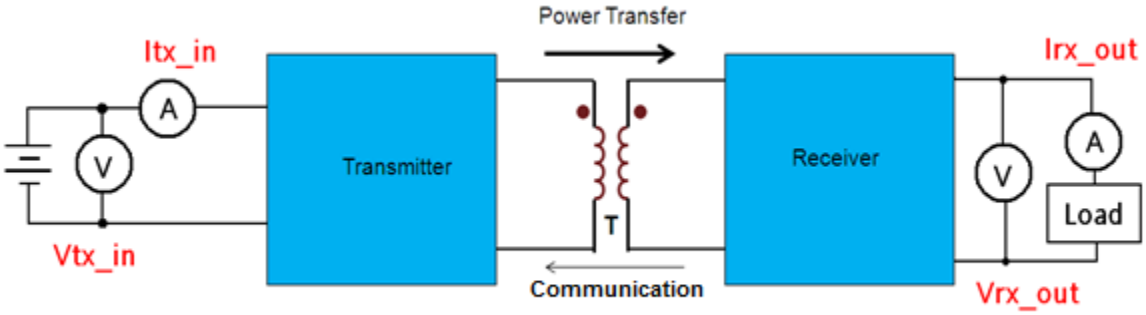

Figure 40 Test environment

6.3.3 Measurements

These are the examples to measure signals on the board.

1. Measure the signals when the Tx board works under ping and stand-by states, when Tx wakes up for charging, and from charging to power stop.

   Ch1: Rx communication signal Ch2: TP4 AUX power on/off

   Ch3: TP10 Coil voltage Ch4: Coil current

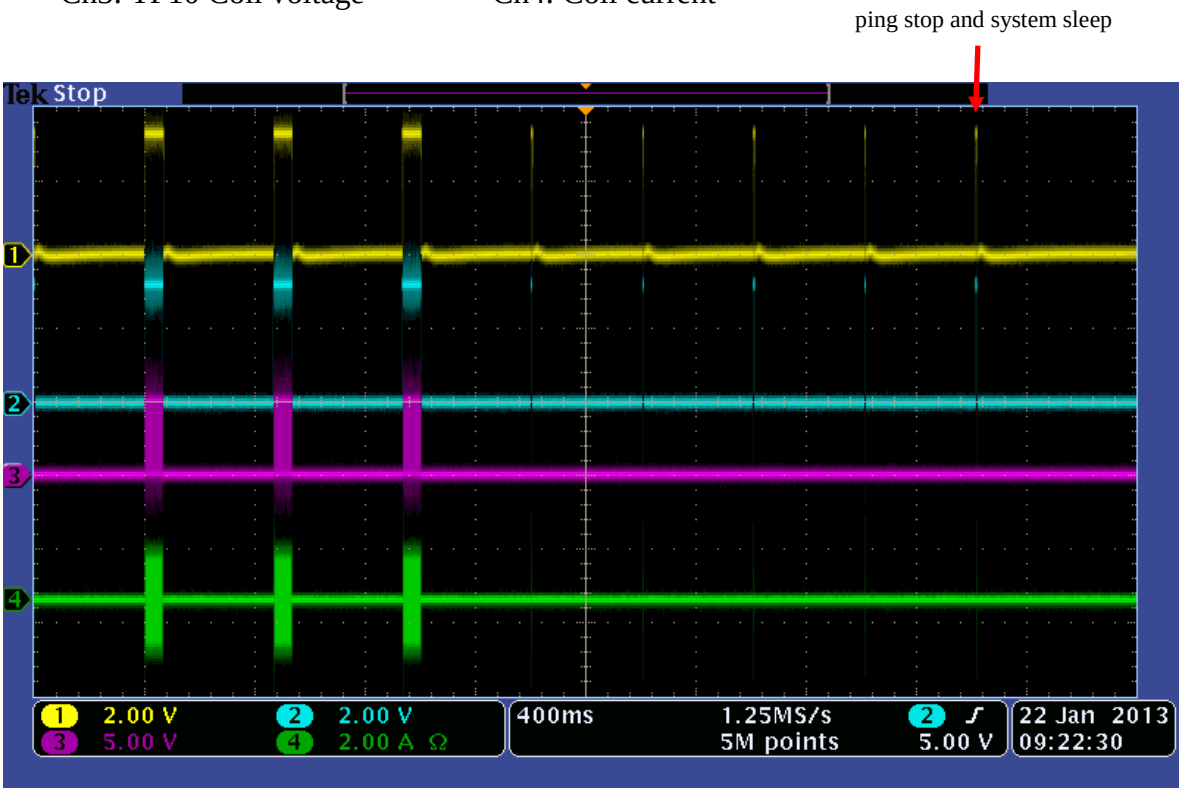

Figure 41 Signals from ping to sleep

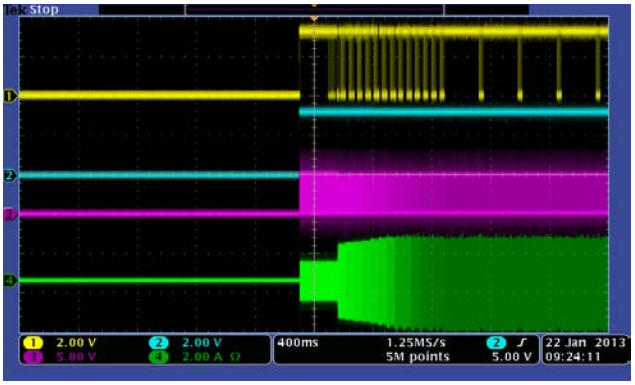

Figure 42 Signals from sleep to charging

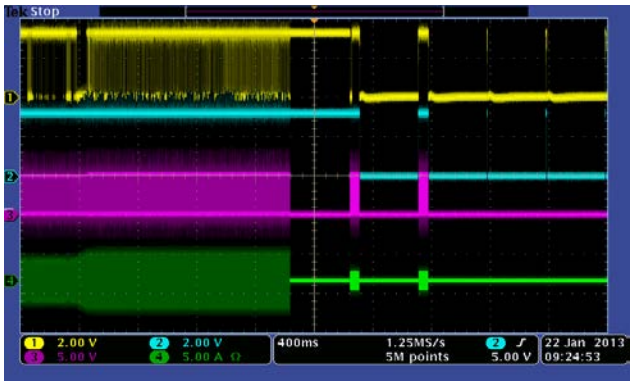

Figure 43 Signals when power stops

The AC signal frequency changes to adjust the output power when the load changes.

Ch1: Rx communication signal

Ch2: TP9 PWM1

Ch3: TP10 coil voltage

Ch4: Coil current

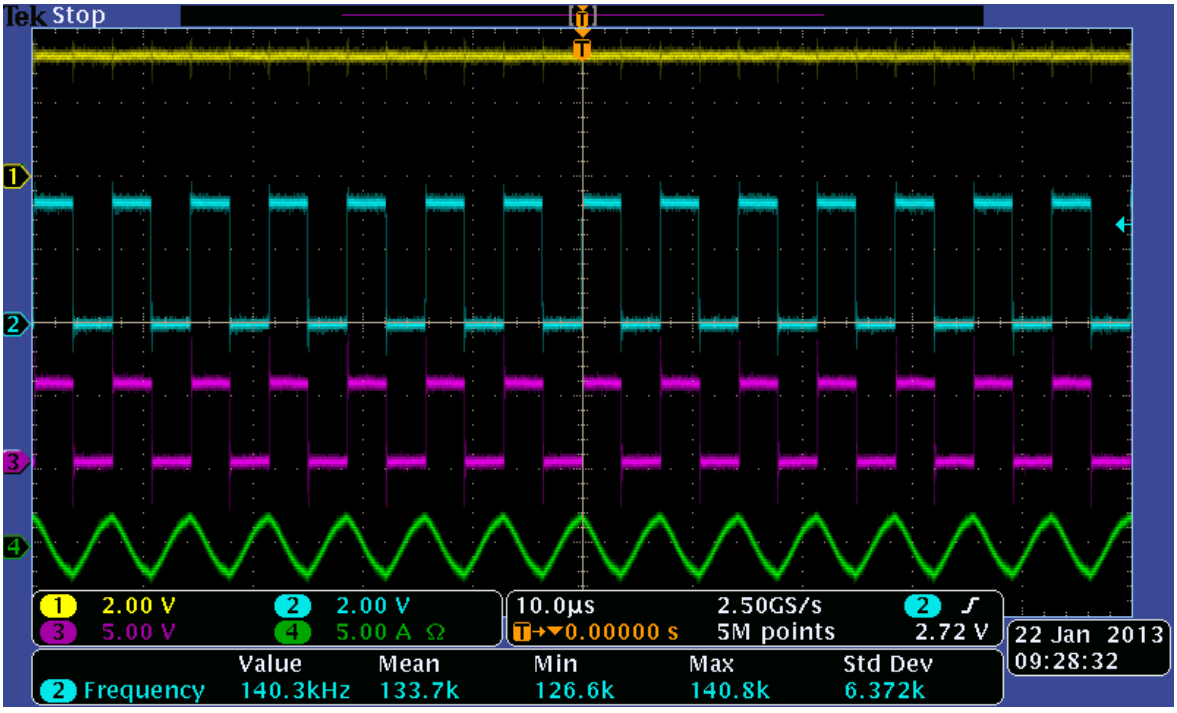

Figure 44 PWM frequency on 0.5 W power output

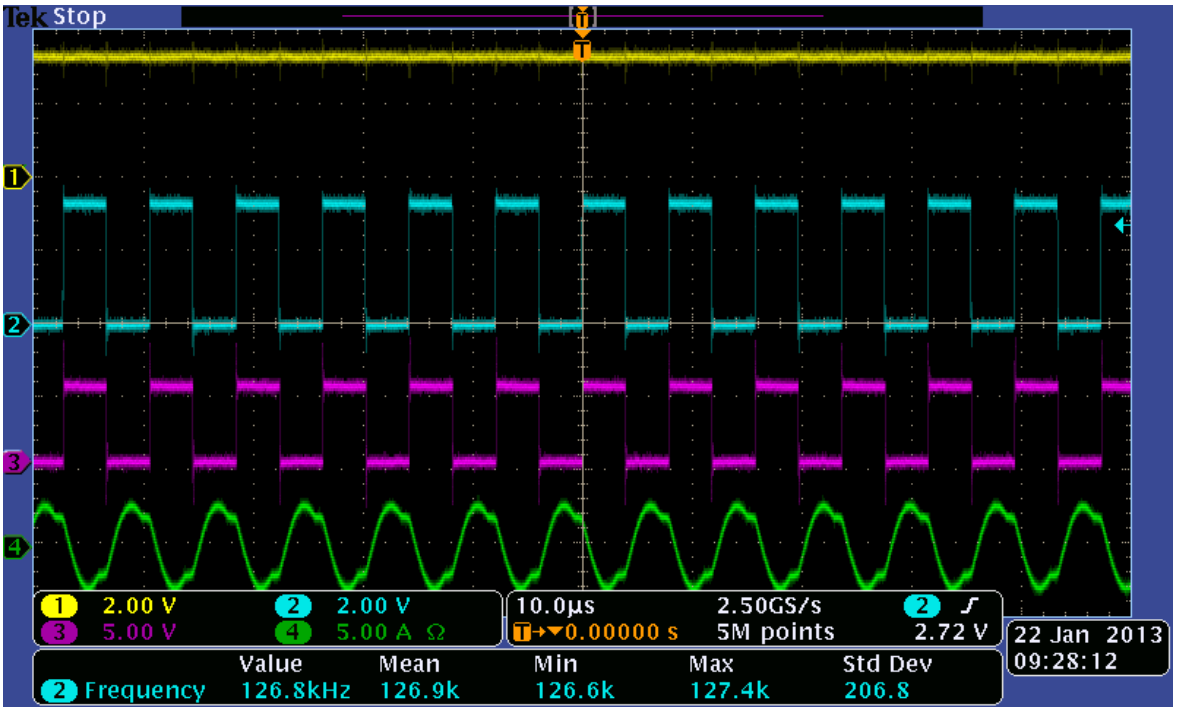

Figure 45 PWM frequency when 5W power output

2. System response measurement for load dump and load step test.

   Ch1: Rx communication signal Ch2: TP9 PWM1

   Ch3: TP10 coil voltage Ch4: Coil current

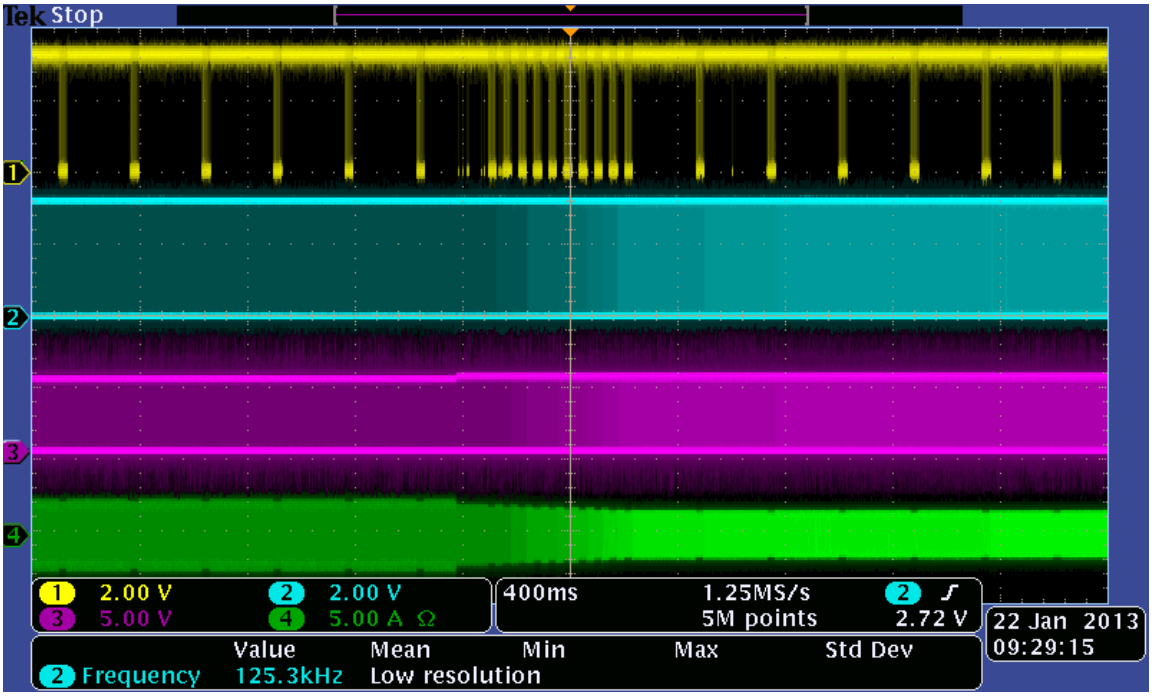

Figure 46 System response on load dump

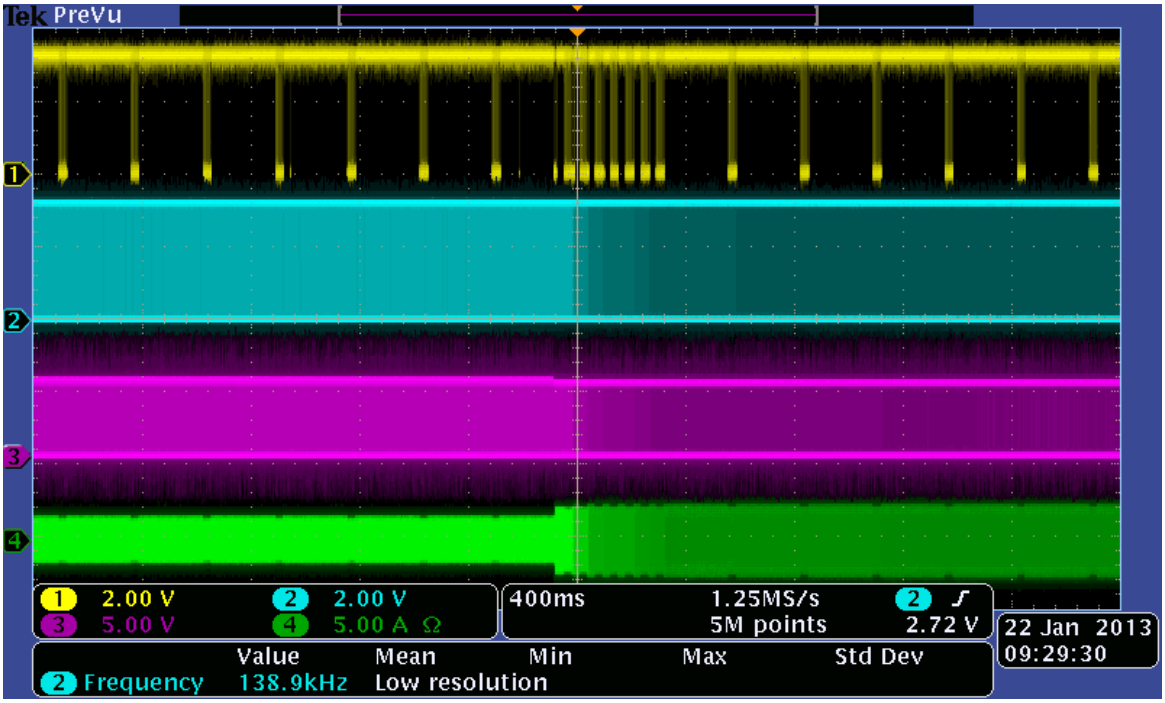

Figure 47 System response on load step

7 References

- Freescale wireless charging solution page:
  freescale.com/wirelesscharging
- Freescale MPR121 touch sensor page:
  freescale.com/webapp/sps/site/taxonomy.jsp?code=SNSPROXIMITY
- Freescale CodeWarrior 10 IDE page:
  freescale.com/codewarrior
- Freescale Freemaster tool page:
  freescale.com/Freemaster
- WPC page:
  www.wirelesspowerconsortium.com
- WCT1000 Documents:
  - *WCT1000 A11 Reference Design System User's Guide* (this document)
  - *WCT1000 TX Library User's Guide* (WCT1000LIBUG)
  - *WCT Runtime Debug User's Guide* (WCT1000RTDUG)
  - *WCT1000 A11 Reference Design Calibration User's Guide* (WCT1000CALUG)
  - *WCT1000 V3.3 Release Notes* (WCT1000RN)

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
www.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 CodeWarrior 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.

©2014 Freescale Semiconductor, Inc.

Document Number: WCT1000SYSUG
Rev. 3.3
09/2014