NXP Semiconductors
User's Guide

Document Number: WCT1012V31SYSUG

Rev. 0 02/2017

WCT1012 15W Single Coil TX V3.1 Reference Design System User's Guide

1 Introduction

This document describes how to use the 15W medium power wireless charger transmitter WCT-15W1COILTX (MPA4) reference board designed by NXP. The reference solution is compliant with Qi Medium Power V1.2.2 specification. It is a low-cost reference solution, which can be easily customized through the FreeMASTER GUI tool.

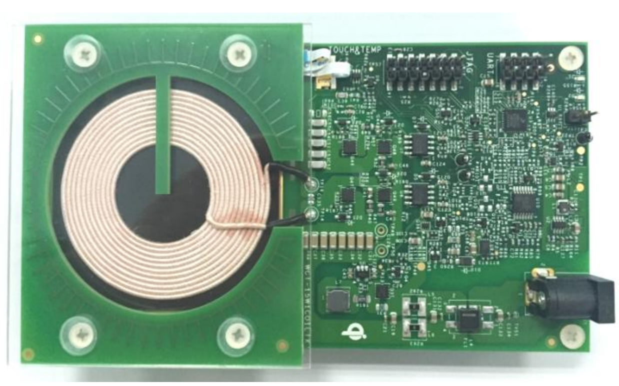


Figure 1. WCT-15W1COILTX reference board

Contents

2 System Features

The WCT-15W1COILTX reference board has these features:

- Compliant with the Qi Medium Power v1.2.2 specification
- Integrated digital demodulation in chip
- Supports multiple types of RX modulation signals (AC capacitor, AC resistor, and DC resistor)
- Supports two-way communication
- Supports FOD based on quality factor (Q factor) change
- Supports FOD based on calibrated power loss accounting
- Supports Qi MP receiver with 15W output power capability
- Supports Qi LP receiver with 5W output power capability
- Super low standby power
- Supports switch between full bridge topology and half-bridge topology
- Supports frequency control, phase shift control, and duty control algorithm
- Supports WPID
- LED for system status indication
- Input voltage, input current, and coil current sensing
- FreeMASTER GUI tool to enable customization and calibration

3 Package Checklist

Table 1. Package checklist

Name	Count
WCT-15W1COILTX board	1

4 System Block Diagram

MP TX runs with RX to transfer power from TX to RX as well as send messages to RX by frequency shift keyed modulation (FSK), as shown in this figure.

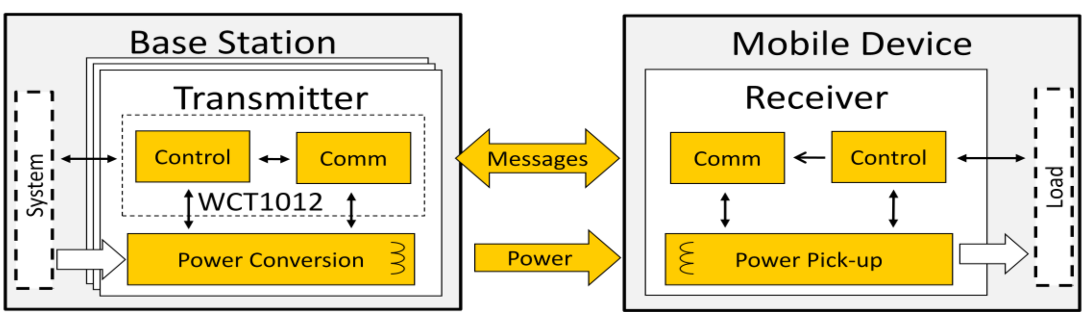


Figure 2. Wireless Charging System overview

To find WPC Qi information, visit www.wirelesspowerconsortium.com/developers/.

5 Hardware Description

5.1 Reference board block diagram

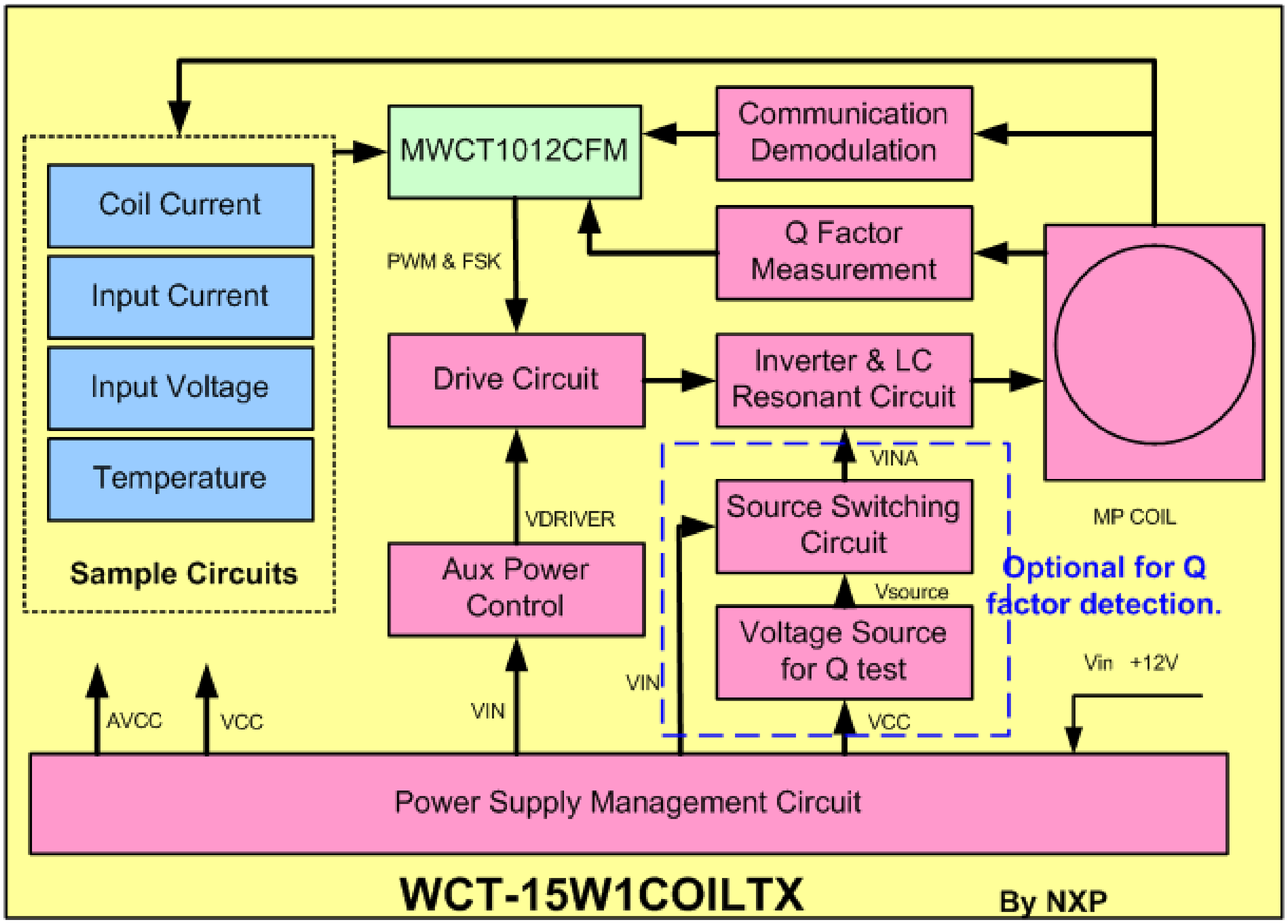


Figure 3. Block diagram

5.2 Modules explanation

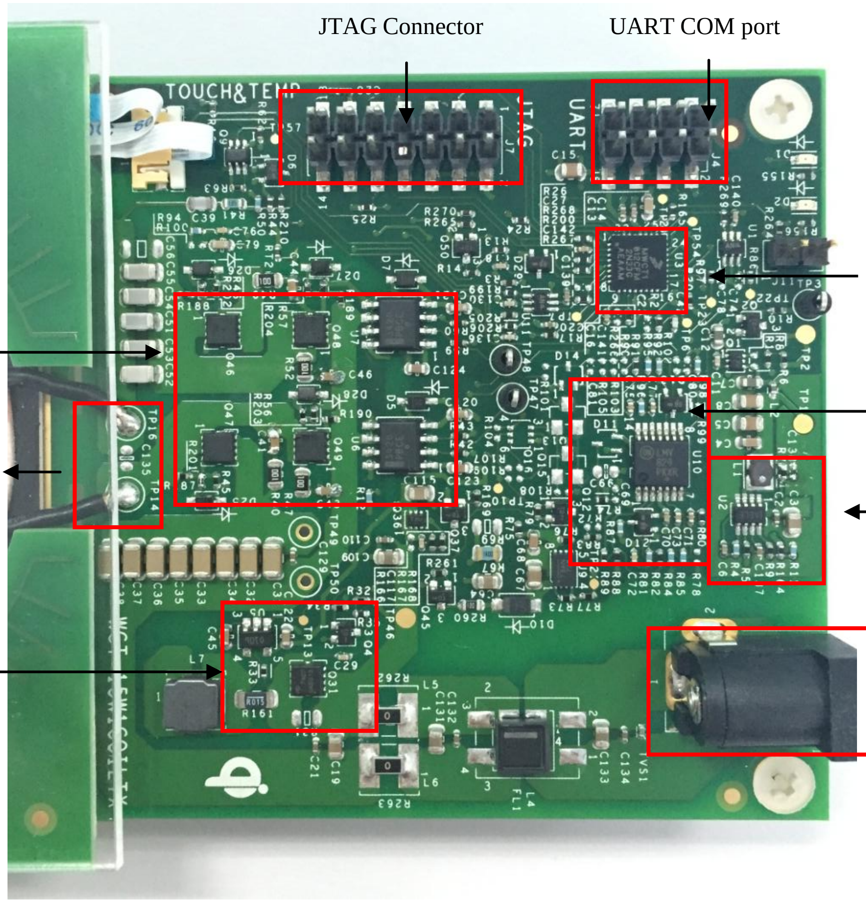


Figure 4. WCT_MP board modules overview

- Controller
  The NXP WCT1012 chip is the central controller of WCT-15W1COILTX board. It has rich peripherals 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 peripherals used in this application:
  - Two PWM channels for full bridge DC/AC inverter control
  - Three Timers for system timing and communication
  - ADCs for input voltage and current, coil current sampling, temperature sensing, and quality factor detecting
  - DAC for generate reference voltage for LDO to produce driver signal for the quality factor detecting circuit
  - GPIOs for pre-drivers control, LED control, and multiplexer switch control

- SCI for serial port debugging or debug console

- Inverter
  The full bridge PWM control inverter converts 12 V DC input voltage to a higher AC voltage.
  - For charging LP RX, the PWM frequency follows the WPC Qi specification. In the 110 KHz–205 KHz range with half bridge, lower frequency outputs larger power with 50% PWM duty and starts duty control (50%–10%) when the frequency reaches 205 KHz.
  - When charging MP RX, TX works with half bridge in the 172 KHz–205 KHz frequency range and starts phase control (16%–100%) at 172 KHz. When PWM phase shift reaches 100%, TX works in full bridge frequency control mode in the range of 110 KHz–172 KHz with 50% PWM duty. Lower frequency outputs larger power and starts duty control (50%–10%) when the frequency reaches 205 KHz.

    Input voltage range: 11 V DC–13 V DC
- Communication
  - Communication from RX to TX: 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 input to ADC for sampling. The digital demodulation module processes the input samples and extracts communication packets.
  - Communication from TX to RX: TX shall negotiate with RX in the negotiation phase if requested by RX. TX uses FSK modulation to communicate to RX, and the communication frequency is about 512 times power signal frequency.
- Q factor detecting
  - When the medium RX is put on the interface of TX, TX starts to detect Q factor of the coil. If Q factor is lower than the threshold, which is determined by RX reported Q factor, FO is detected. This method can detect FO before power transfer.
  - There are two methods of Q factor detection methods provided in this release: free resonance and external driver, which need different hardware modules as shown in Figure 5-a and Figure 5-b. The Q Factor measurement circuit is also different. See Section 6.1 for detailed information.

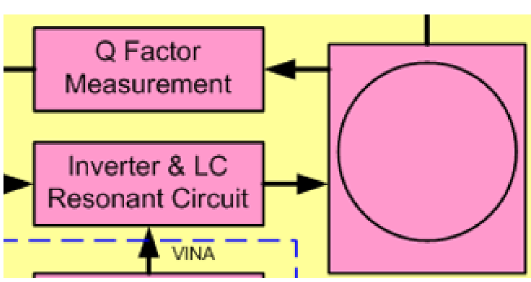


Figure 5-a. Free resonance Q measurement

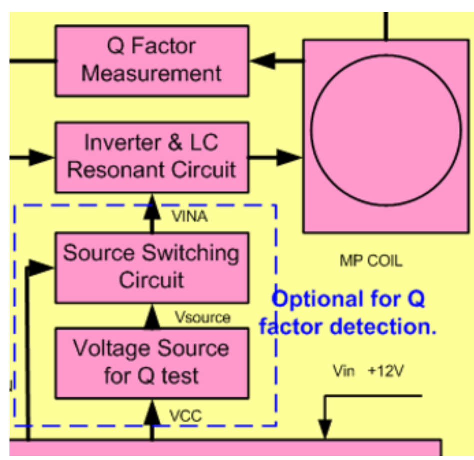


Figure 5-b. External driver Q measurement

6 Foreign Object Detection Architecture

The NXP WCT-15W 1COIL TX solution provides two methods of Foreign Object Detection: FOD based on quality factor (Q factor) change and FOD based on power loss accounting. The former can detect FO before power transfer and the later can work during power transfer phase.

6.1 FOD based on Q factor change

A change in the environment of the TX coil typically causes its inductance to decrease and its equivalent series resistance to increase. Both effects lead to a decrease of the TX coil's Q factor. The RX would send a packet including the reference Q factor for TX to compare and determine if FO exists, as shown in Figure 6.

The reference Q factor is defined as the Q factor of Test Power Transmitter #MP1's Primary Coil at an operating frequency of 100 kHz with RX positioned on the interface surface and no FO nearby. Due to the differences between its design and that of Test Power Transmitter #MP1, the difference between the frequency it uses to determine its Q factor and 100 kHz, TX needs to convert the Q factor it measured to that of Test Power Transmitter #MP1. NXP provides the conversion method and needs to get the parameters on board at first. The TX would do auto-calibration and get parameters the first time power up after flashing a new image, and then these parameters are written to flash. Therefore, it is necessary to make sure there is no object on the TX surface the first time powering up after flashing a new image.

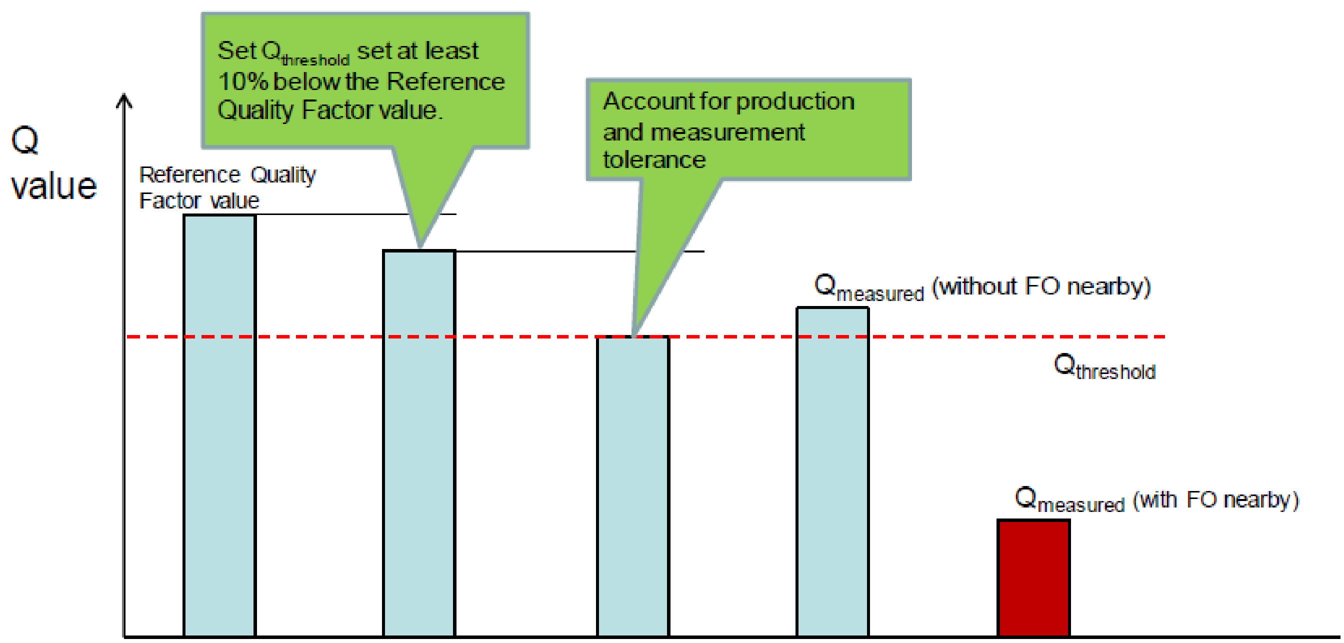


Figure 6. Quality factor threshold example

The NXP WCT-15W1COILTX provides two optional methods to detect Q factor: free resonance and external driver. The free resonance method is enabled by default in the released V3.1 f/w. To apply free resonance method, the WCT-15W1COILTX Rev.3 board (SCH-28122 REVC3, 700-28122 REVA) needs rework. The external driver method could be applied on the WCT-15W1COILTX Rev.3 board directly. Customers can apply external driver Q factor detection by changing the related macro to enable it as follows:

```
#define QF_EXTERNAL_DRIVER                        TRUE
#define QF_FREE_RESONANCETRUE                     FALSE
```

6.1.1 Free Resonance Q factor

The free resonance Q factor detection aims to detect the decay rate of the resonance signal, as shown in Figure 7. With the system's high Q, driving just a few pulses near resonant frequency is sufficient to serve as an impulse and start the system ringing. The decay rate of the signal can be found by collecting the ADC data of the tank voltage (or coil current).

$$Q=\pi/(-\ln(Rate))$$

Rate is the decay rate value of the resonance signal.

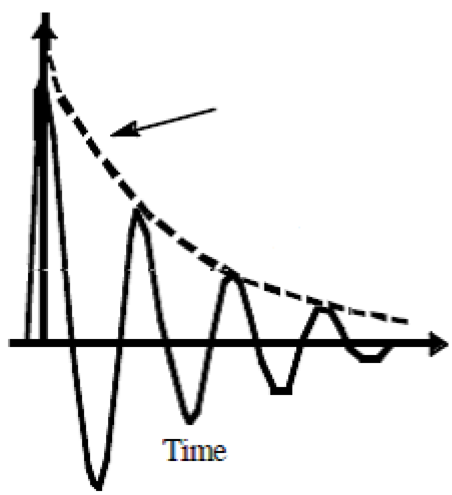


Figure 7. Resonance signal

The circuit for free resonance Q measurement is as shown in Figure 8, which is sampling circuit of signal on resonance capacitors.

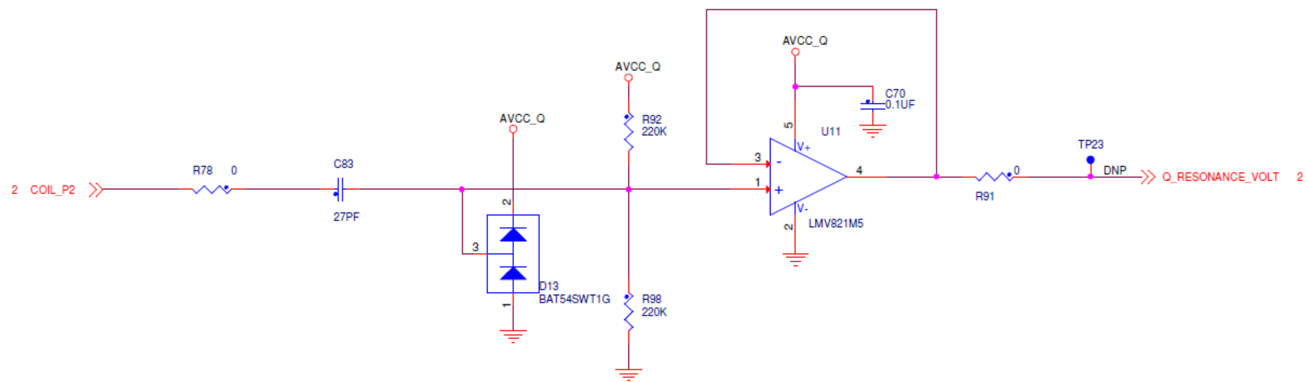


Figure 8. Free resonance Q measurement circuit

6.1.2 External Driver Q factor

The external driver Q factor detection method is described as shown in Figure 9. It consists of a series connection of the coil and a resonance capacitor that is driven by a sinusoidal voltage. The capacitance value of the resonance capacitor is chosen so that the resonance frequency of the system is in a suitable range. In this example, the resonance frequency is 100 kHz. The Q factor of the coil follows from this system as the ratio of the RMS voltage across the coil and the RMS voltage that is driving the system at the resonance frequency.

$$Q = V_2/V_1 \text{ (at resonant frequency)}$$

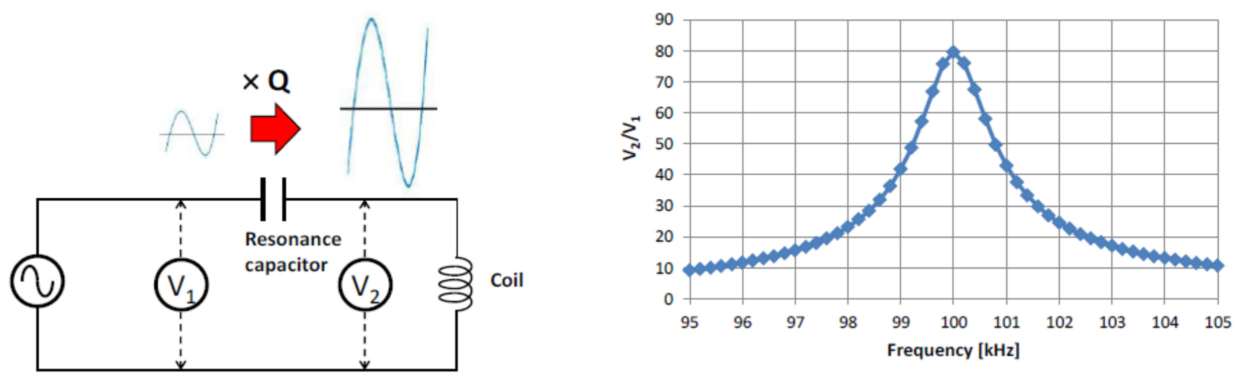


Figure 9. External Driver Q factor measurement method

The sample circuits include driver signal sampling and resonance signal sampling, as shown in Figure 10:

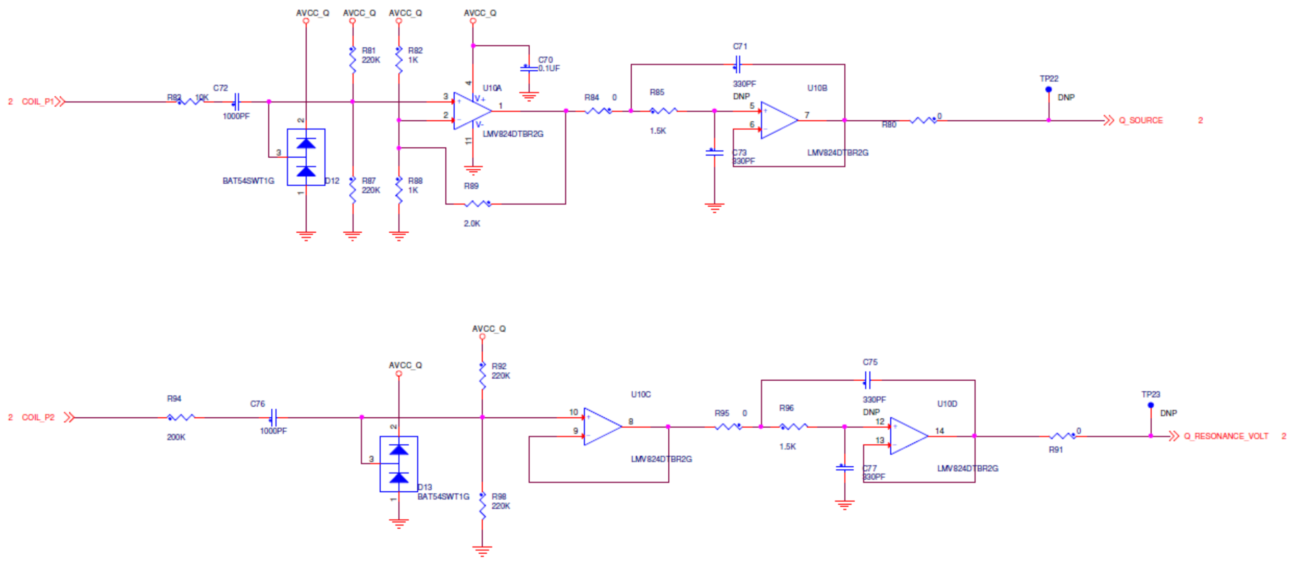


Figure 10. External driver Q measurement circuit

The driver signal should be small to avoid waking up the RX during Q factor measurement. Therefore, it is necessary to use an auxiliary power to generate a low voltage for the driver signal.

6.2 FOD based on power loss accounting

The power loss P_{LOSS}, which is defined as the difference between the Transmitted Power P_{PT} and the Received PowerP_{PR}, i.e. $P_{LOSS} = P_{PT} - P_{PR}$, provides the power absorption in Foreign Objects, as shown in the following figure.

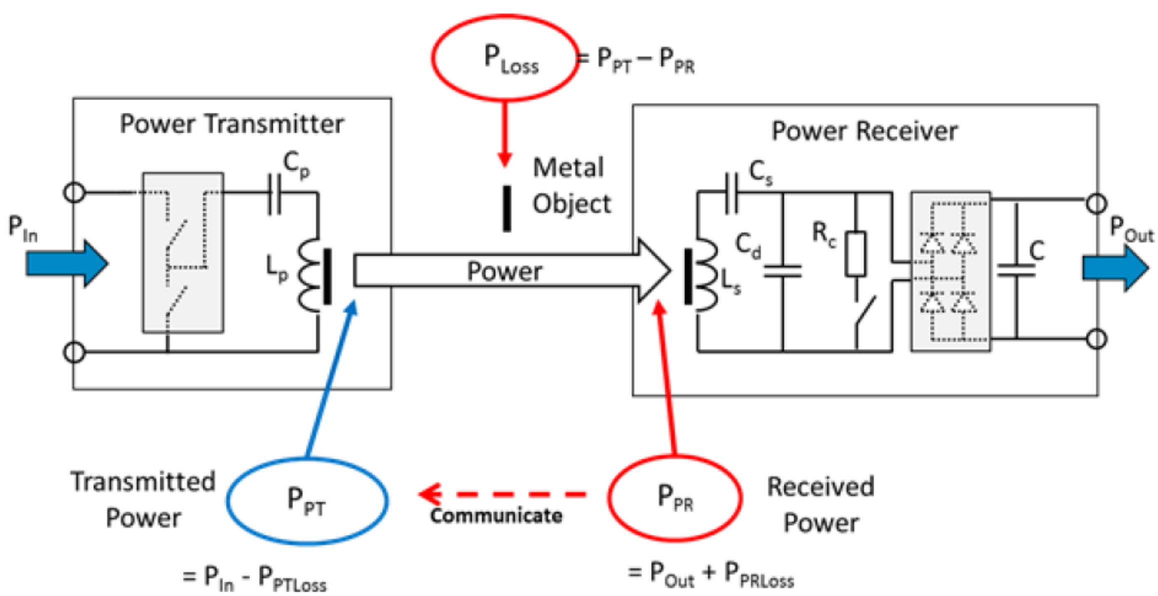


Figure 11. Power loss illustrated

When FO is implemented in the power transfer, the power loss increases accordingly. Then the FO can be detected based on the power loss method.

Power loss FOD method is divided into two types:

- FOD for baseline power profile (TX and RX can transfer no more than 5W of power)
- Extensions power profile (TX and RX can transfer power above 5W)

6.2.1 Power loss FOD baseline

The equation for power loss FOD baseline is $P_{LOSS} = P_{PT} - P_{PR}$.

The Transmitted Power P_{PT} represents the amount of power that leaves TX due to the magnetic field of the TX, and $P_{PT} = P_{in} - P_{PTloss}$, where the P_{in} represents the input power of TX and P_{PTloss} is the power dissipated inside the TX. P_{in} can be calculated by sampling input power and input current, and P_{PTloss} can be estimated through coil current.

The Received Power P_{PR} represents the amount of power that is dissipated within the RX due to the magnetic field of the TX, and $P_{PR} = P_{Out} + P_{PRloss}$. The power P_{Out} is provided at the RX's output and P_{PRloss} is the power lost inside the RX.

When the NXP 15W 1COIL TX charges the RX baseline, the power loss baseline is applied. The TX would continuously monitor the P_{LOSS}, and if it exceeds the threshold several times, the TX would terminate power transfer.

6.2.2 Power loss FOD extensions

Typically, the RX estimates the power loss inside itself to determine its Received Power. Similarly, the TX estimates the power loss inside itself to determine its Transmitted Power. A systematic bias in these estimated results in a difference between the Transmitted power and the Received Power, even if there is no Foreign Object present on the Interface Surface. To increase the effectiveness of the power loss method, the TX can remove the bias in the calculated power loss by calibration. For this purpose, the TX and RX execute the *calibration* phase before the *power transfer* phase starts. The TX needs to verify that there is no FO present on its interface surface before calibration phase and pre-power FOD based on Q factor can work.

Because the bias in the estimates can be dependent on the power level, the TX and RX determine their Transmitted Power and Received Power at two load conditions—a "light" load and a "connected" load. The "light" load is close to the minimum expected output power, and the "connected" load is close to the maximum expected output power. Based on these two load conditions, the Power Transmitter can calibrate its Transmitted Power using linear interpolation. Alternatively, the Power Transmitter can calibrate the reported Received Power.

Take the calibrated Transmitted Power as an example：

$$P_{PT}^{cal} = a * P_{PT} + b$$

$$a = \frac{P_{PR}^{(connected)} - P_{PR}^{(light)}}{P_{PT}^{(connected)} - P_{PT}^{(light)}}$$

$$b = \frac{P_{PT}^{(connected)} * P_{PR}^{(light)} - P_{PR}^{(connected)} * P_{PT}^{(light)}}{P_{PT}^{(connected)} - P_{PT}^{(light)}}$$

Subsequently, the TX should use the calibrated Transmitted Power to determine the power loss as follows:

$$P_{LOSS} = P_{PT}^{cal} - P_{PR}$$

When an RX baseline is charged by the WCT-15W1COILTX, only the power loss FOD baseline works. If a RX extension is placed on the WCT-15W1COILTX, Q factor would be measured at first to detect if there is a FO present. If there is, the TX would stop charging; otherwise, the TX can proceed to calibration phase and power transfer phase, and power loss FOD extension works to detect if an FO is inserted during power transfer phase.

7 Getting Started

NXP provides a software package to modify WCT_MP functions. The user can modify system parameters or configurations to maintain system functionalities. For example, when TX coil or main power components are changed, it is better to calibrate to get the Foreign Object Detection (FOD) working. This document describes the basic debugging environment on WCT1012. For MP software details, see the *WCT1012 TX V3.1 Library User's Guide* (WCT1012V31LIBUG).

7.1 System developing environment

TX board debugging uses CodeWarrior and the FreeMASTER tool. Set up the debugging connection as shown in Figure 12. The debugger is between the PC and TX board. Connect a debugger (USBTAP, P&E-Multilink FX or OSJTAG) to the JTAG port of MP TX board through a 14-pin cable.

Figure 12 shows the connection and Figure 6. shows a real image.

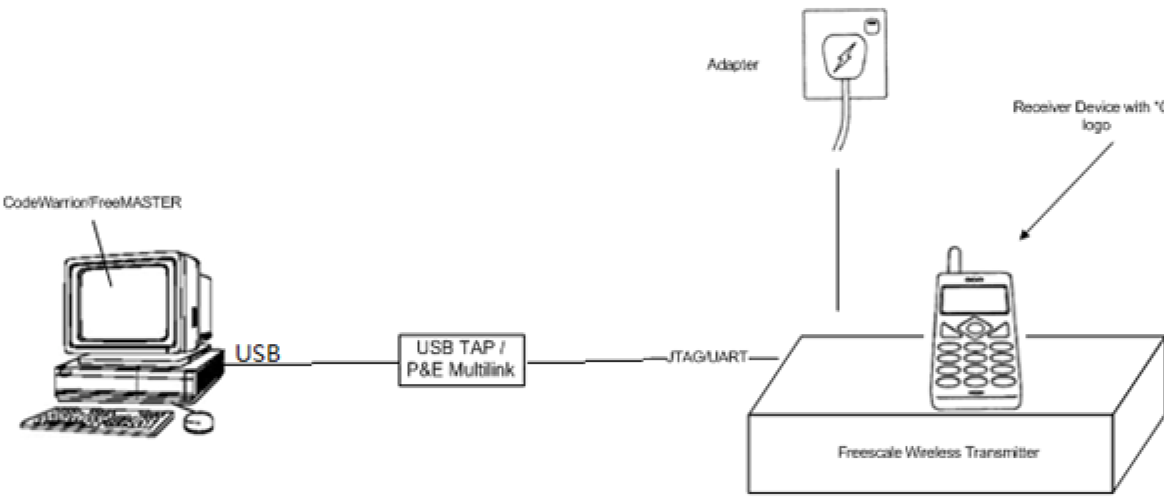


Figure 12. Debugging connections

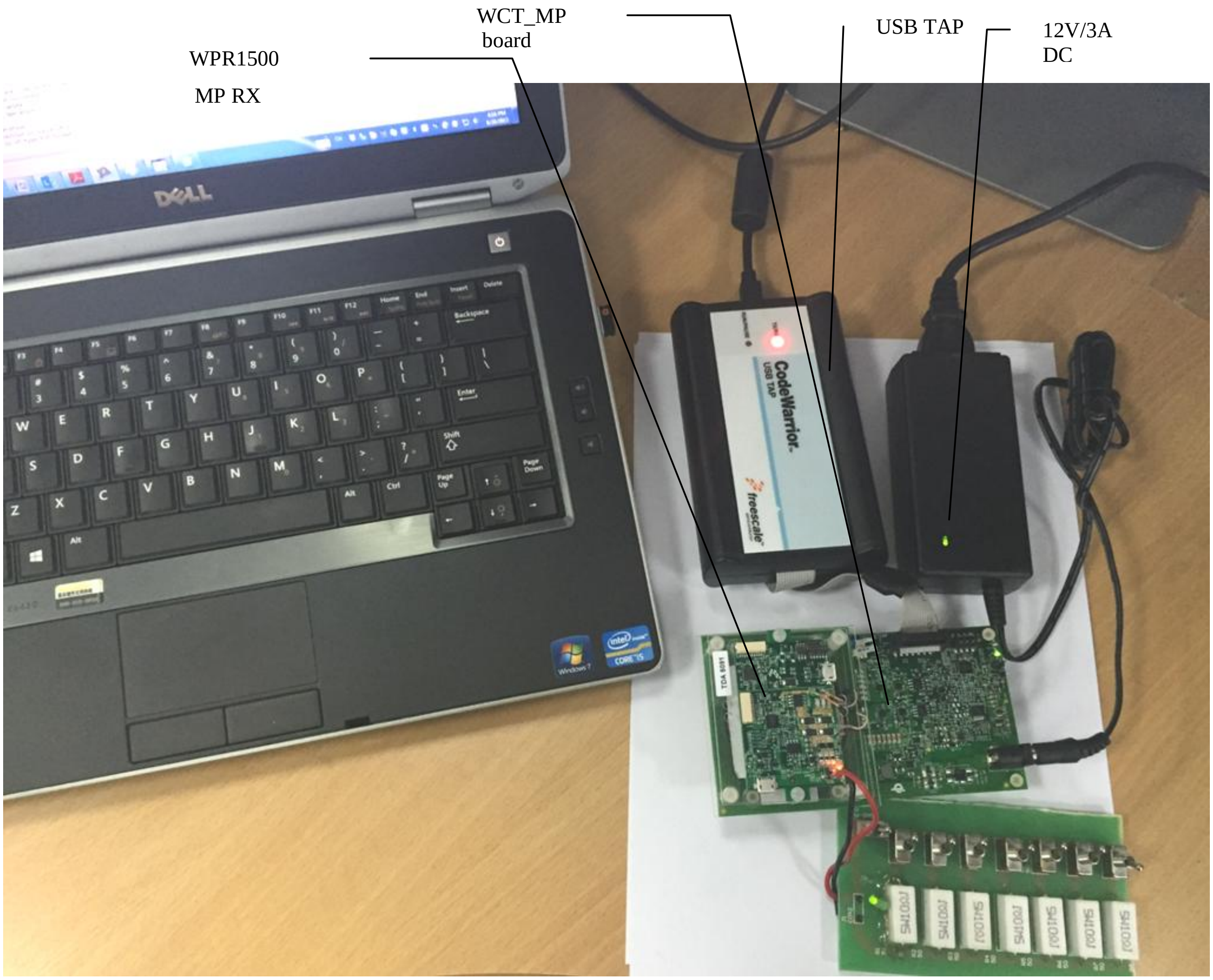


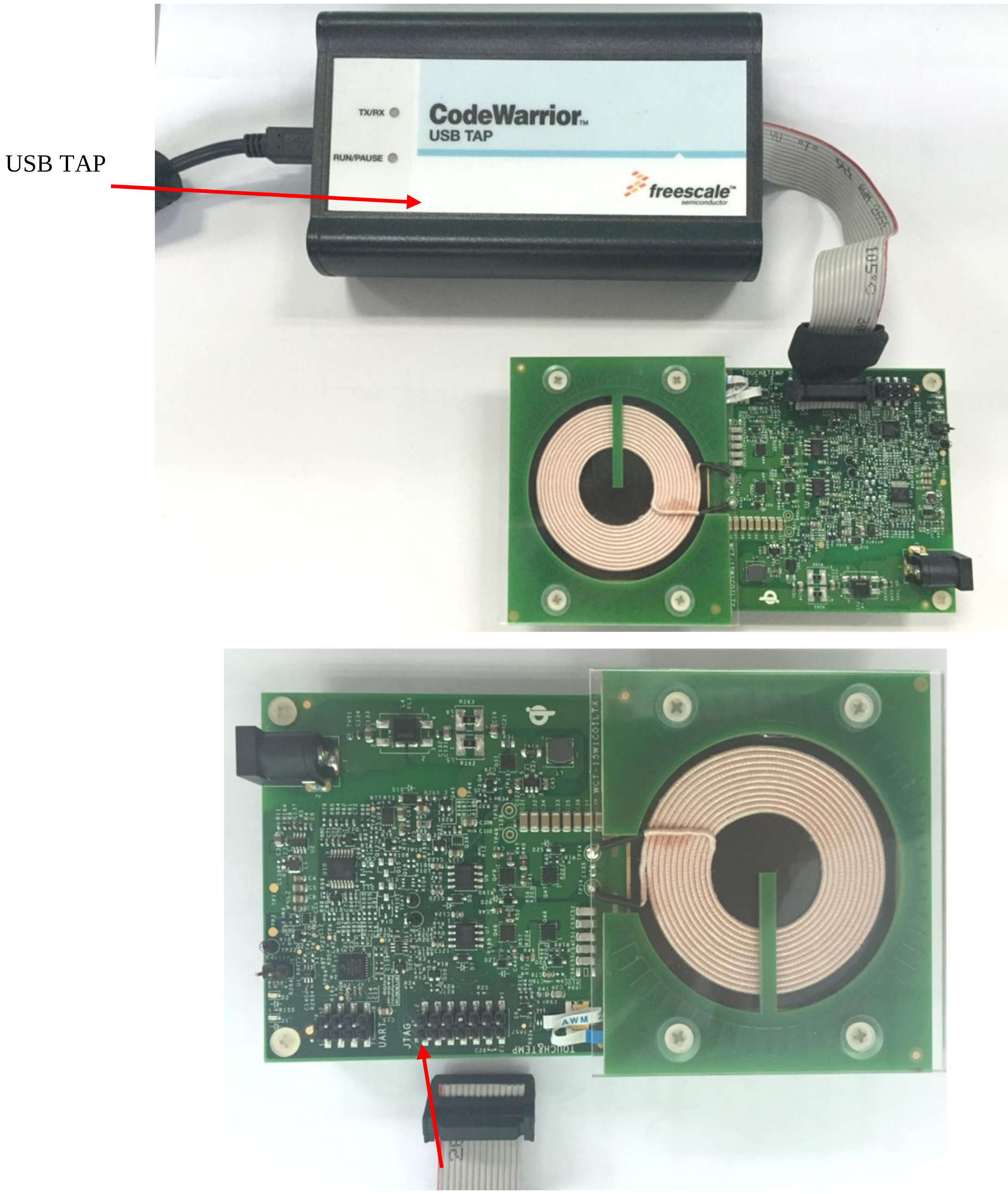
Figure 13. Development environment

For details about the P&E-Multilink FX debugger, visit [nxp.com]() and search for “U-MULTILINK-FX”. Then the **U-MULTILINK-FX: Universal Multilink FX High-Speed Development Interface** page is displayed.

7.2 Downloading and debugging firmware with CodeWarrior 10 IDE

7.2.1 Connecting the JTAG debugger

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



Red line linked to Pin 1

Figure 14. Debugger connecting

When the debugger is plugged onto the PC, the device can be found in Windows® operating system **Device Manager**, as shown in these figures.

USB TAP

P&E Multilink

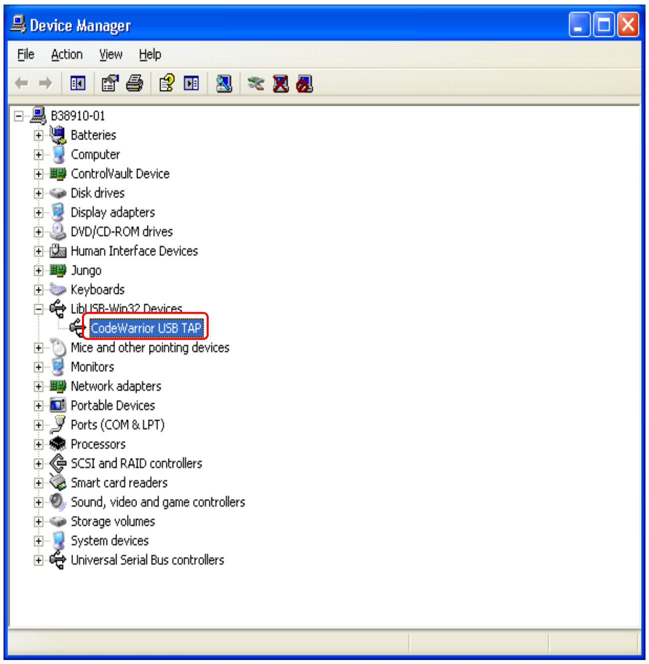


Figure 15. USB TAP debugger plugged in

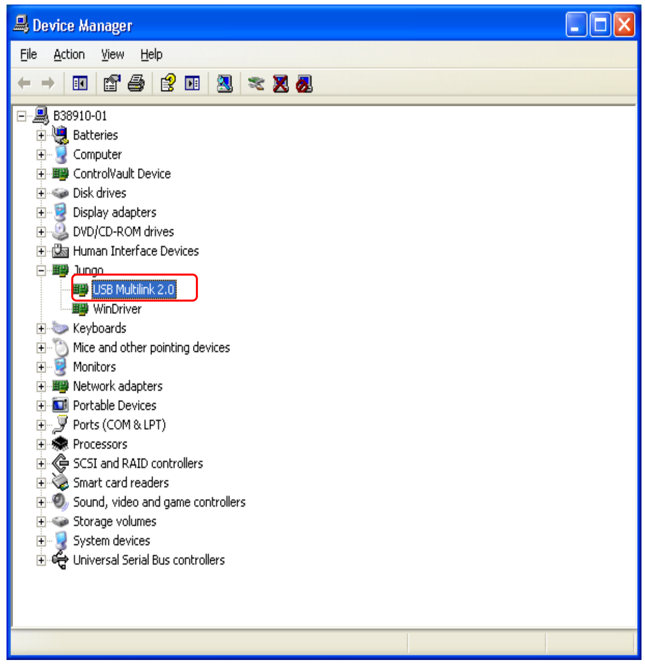
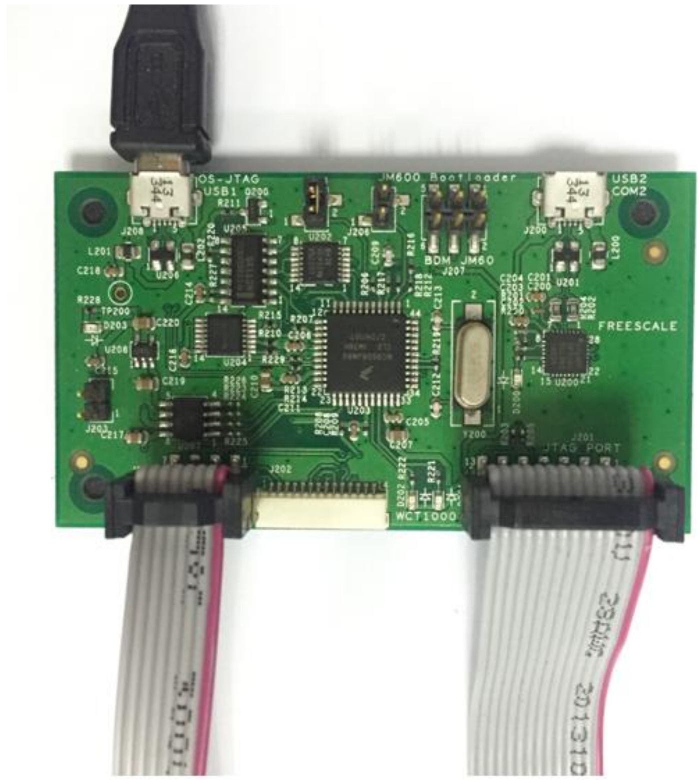


Figure 16. P&E multilink debugger plugged in



OSJTAG

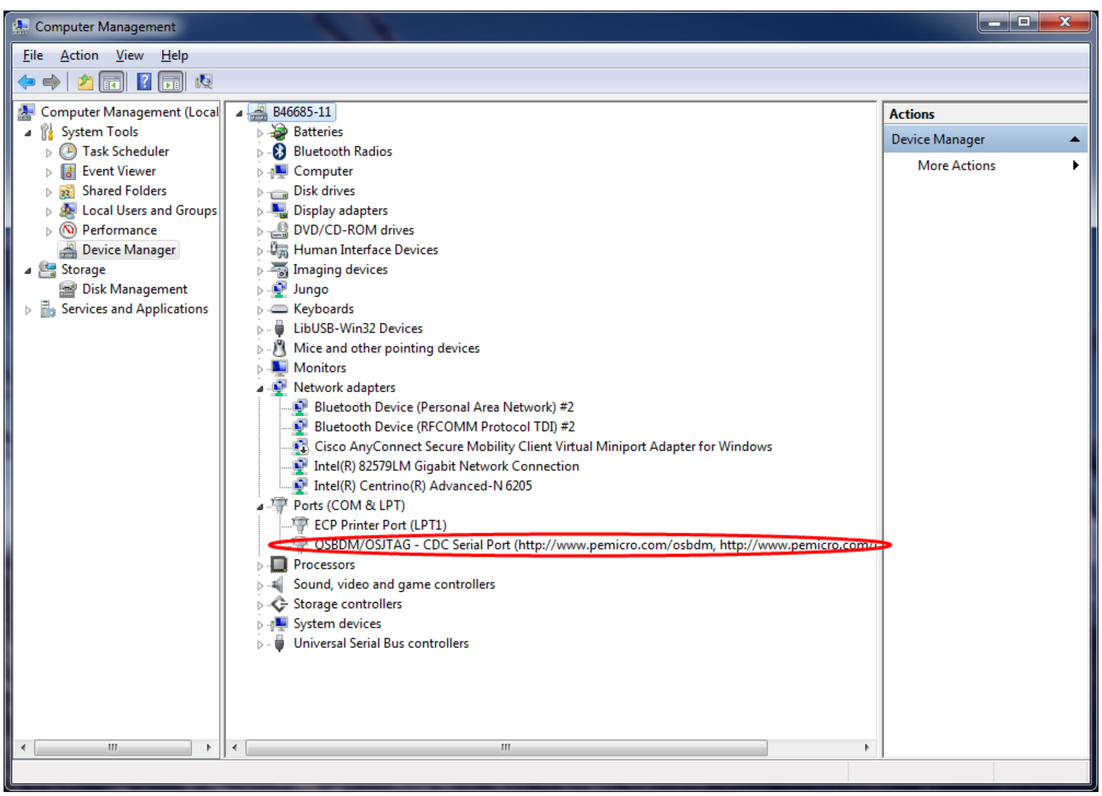


Figure 17. OSJTAG debugger plugged in

7.2.2 Downloading an Existing WCT1012 Project with CodeWarrior Version 10.7 or later

To download an existing WCT1012 project with CodeWarrior version 10.7, perform these steps:

Make sure there is no object on TX surface the first time TX runs after flashing a new image.

1. Set the CodeWarrior version 10.7 Workspace.

   Open CodeWarrior version 10.7, and set the workspace to WCT1012 example project.

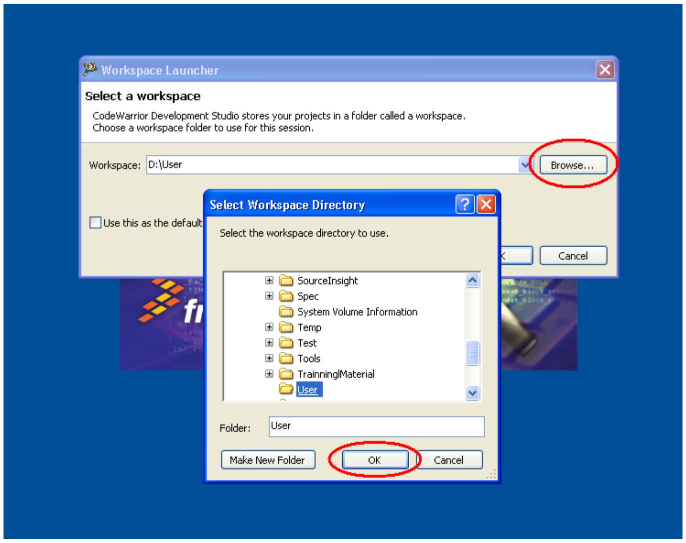


Figure 18. Setting the CodeWarrior version 10.7 workspace (1)

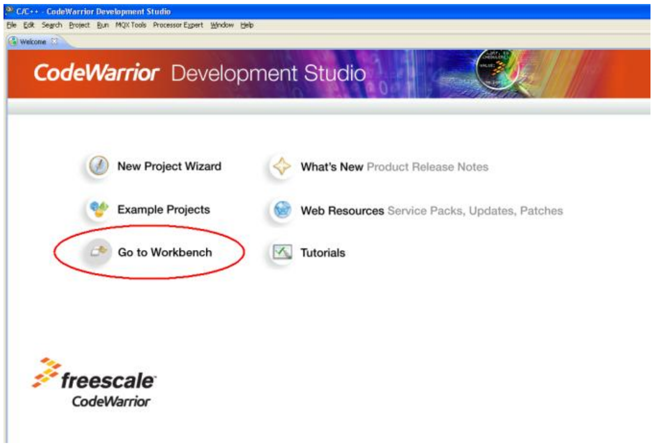


Figure 19. Setting the CodeWarrior version 10.7 workspace (2)

2. Install the MCU v10.7 service package.

   a. Ensure that the CW is updated to the latest version. Choose **Help** -> **Check for updates**.

      If there are updates, install them.

   b. Download MCU v10.7 service package.

      Access the following webpage and sign in:

      www.nxp.com/products/power-management/wireless-charging-ics/15-watt-wireless-charging-transmitter-ics-for-automotive-applications:MWCT1x1xA

      Click the **Software & Tools** tab.

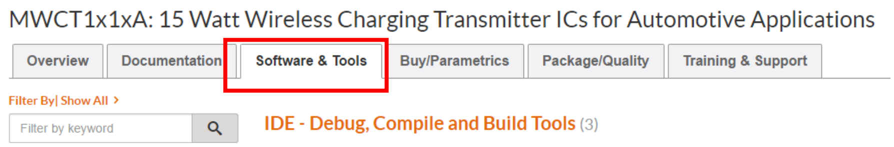


Figure 20. Downloading MCU v10.7 service package (1)

      Click **Download** to get the service package.

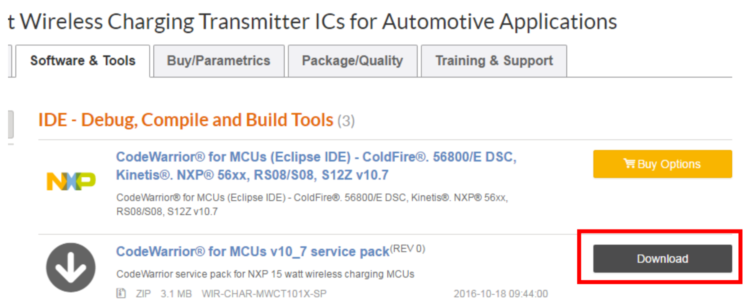


Figure 21. MCU v10.7 service package (2)

   c. On the CW tool bar, choose **Help** –> **Install New Software**.

      Use the downloaded.zip file as the local update archive: Click the **Add...** button, and then click **Archive...** to point to the **com.freescale.mcu10_7.Wireless_Charging_MWCT101x.win.sp.v1.0.1.zip** file to be downloaded.

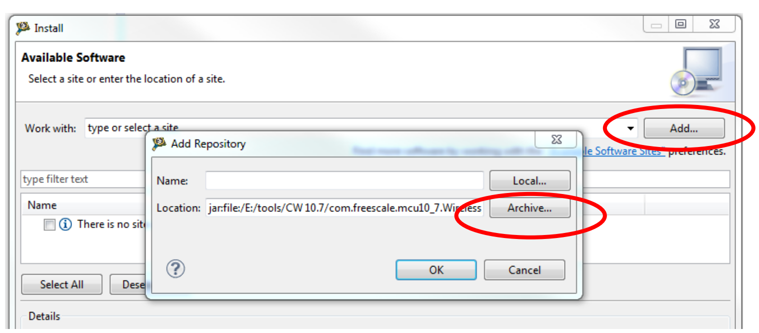


Figure 22. Updating the MCU v10.7 service package (1)

Select the update(s) and go through the installation process with **Next**.

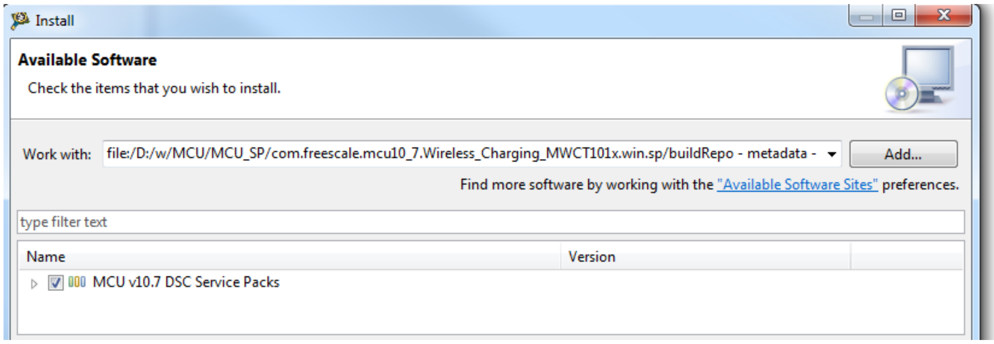


Figure 23. Updating the MCU v10.7 service package (2)

Code Warrior10 is restarted automatically after installation is completed.

3. Import a project.

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

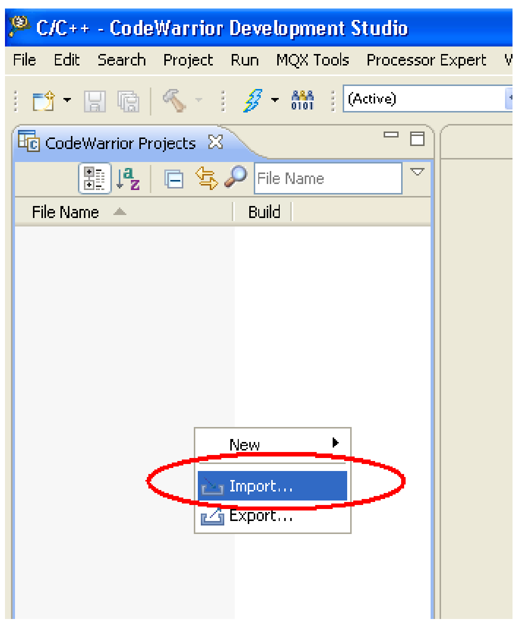


Figure 24. Importing a project (1)

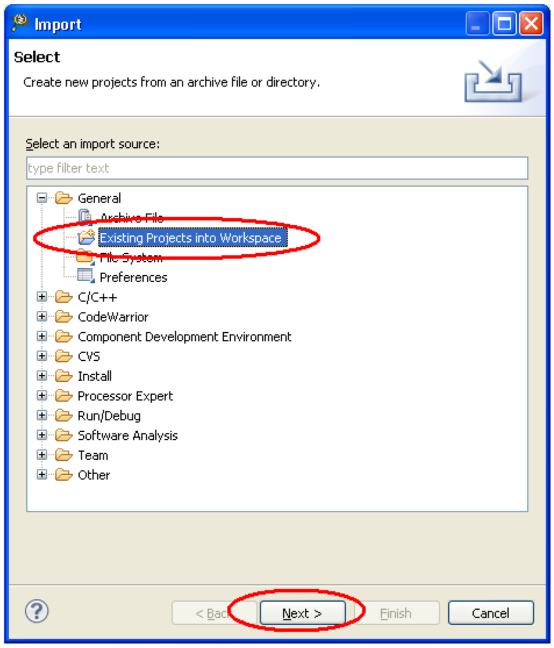


Figure 25. Importing a project (2)

Select the project directory, as shown in this figure.

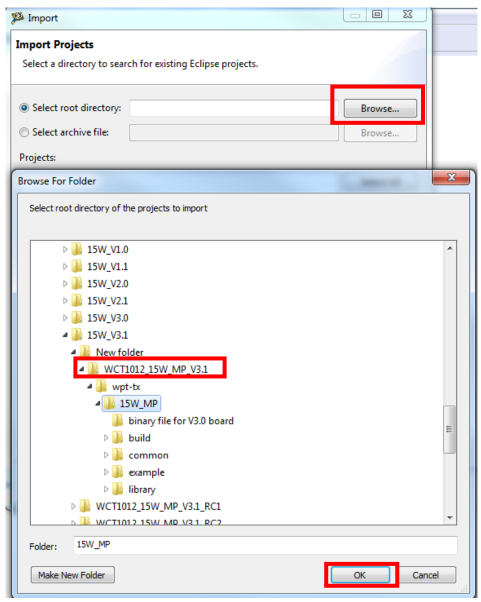


Figure 26. Importing a Project (3)

Select the project found by CodeWarrior version 10.

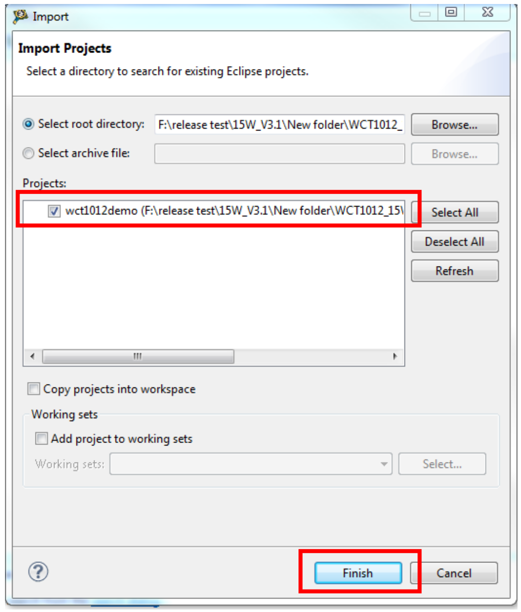


Figure 27. Importing a project (4)

4. Build a project.

You can select build configurations –> Debug or Release build, by clicking the project name in the project window shown in the following figure. Debug build includes more debug information. In this release v3.1, only Release mode is enabled due to code size.

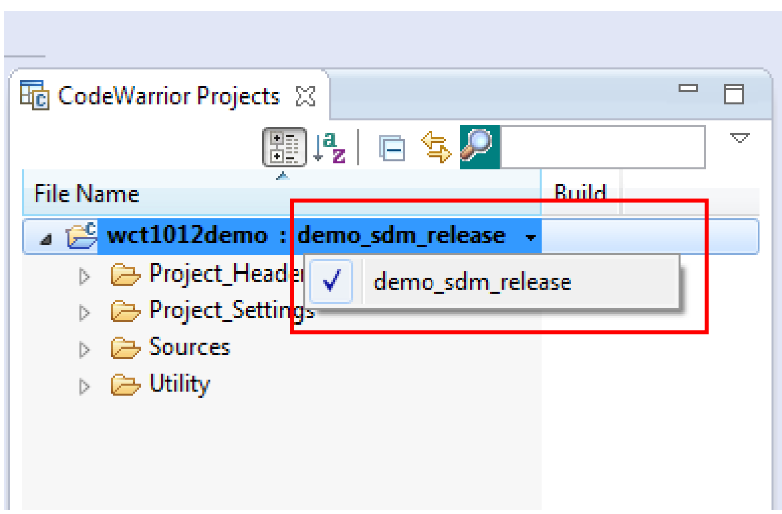


Figure 28. Building a project (1)

Right-click the project name **wct1012demo: demo_sdm_release** and then you can select **Build Project**, **Clean Project**, or **Close Project**. You can also perform building from **Project**.

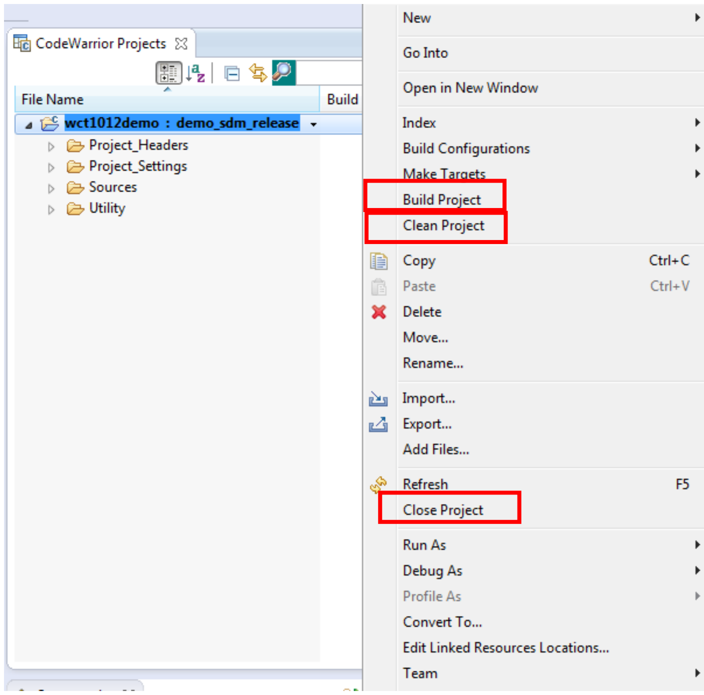


Figure 29. Building a project (2)

5. Download the project.

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

   Download the project from the **Debug** drop-down list or from **Run** –> **Debug**.

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

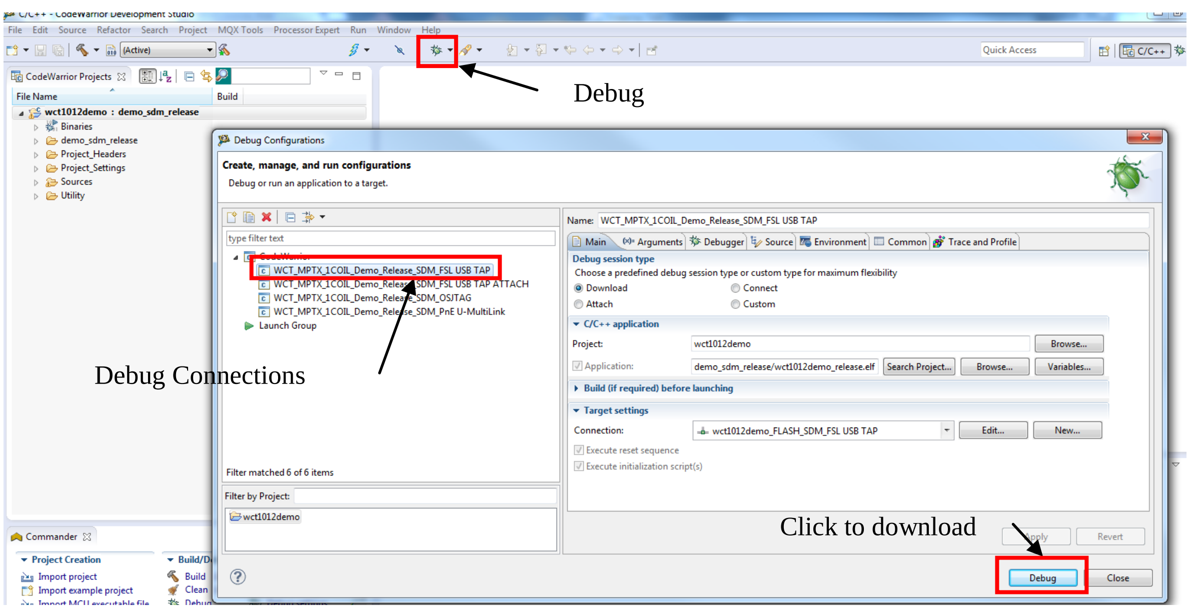


Figure 30. Downloading the project

After the project is downloaded, the MCU stops at the startup code. Press F8 to let MCU continue.

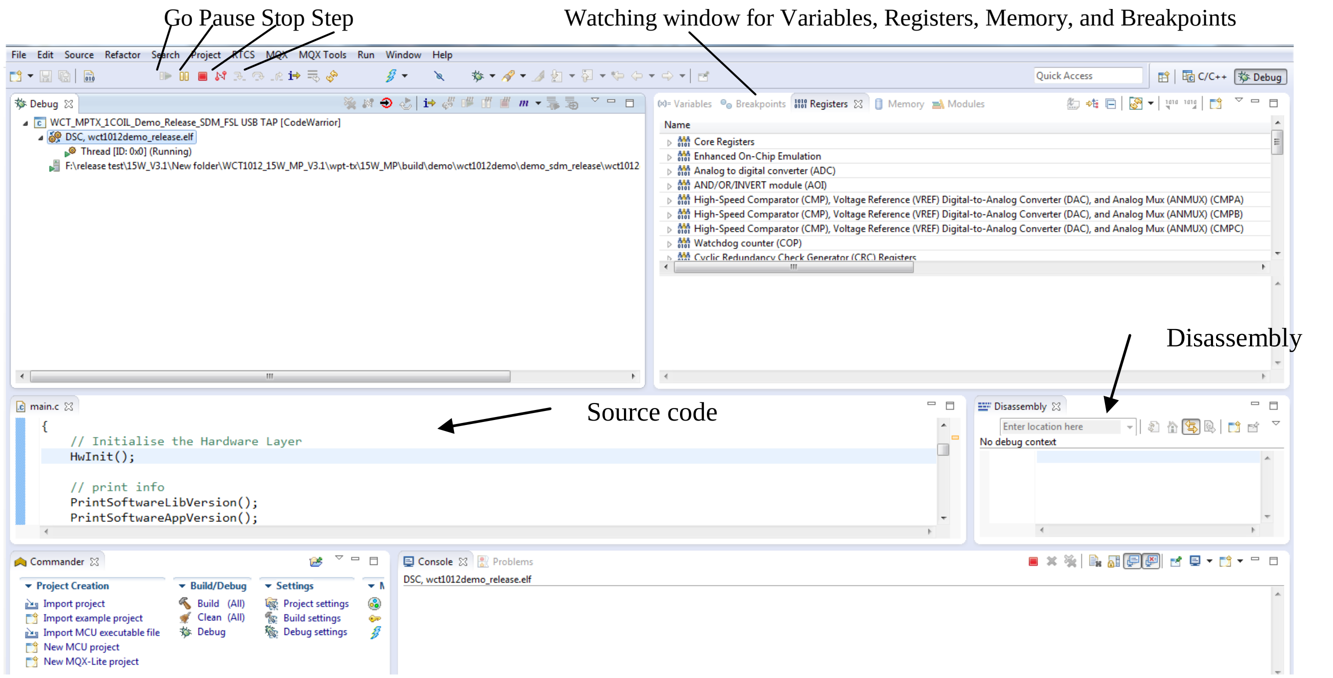


Figure 31. Project Downloaded

7.2.3 Downloading an existing WCT1012 bin file (.elf) with CodeWarrior version 10

To flash an .elf file, perform these steps:

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

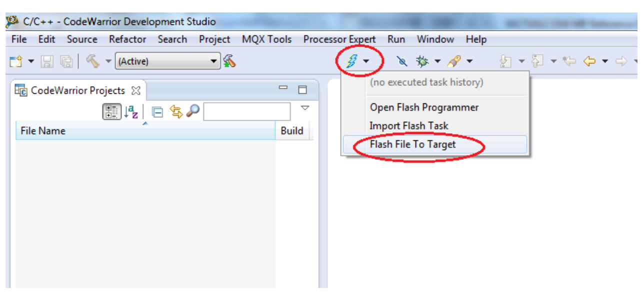


Figure 32. Bin file download (1)

2. Click **New** to create a new connection.

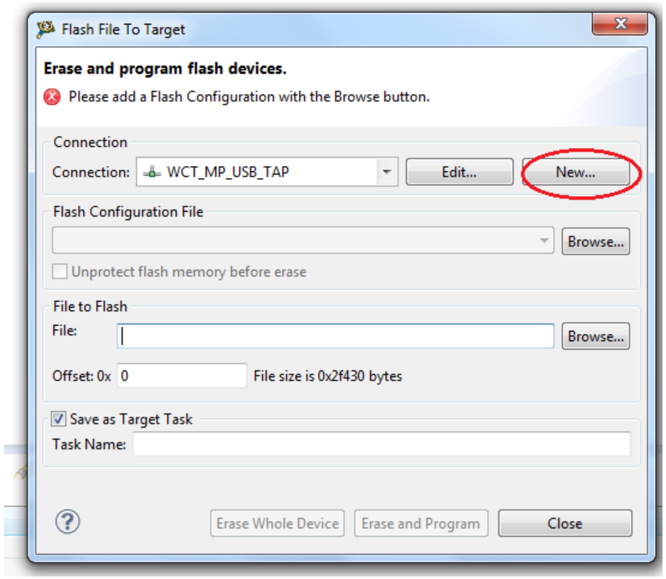


Figure 33. Bin file download (2)

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

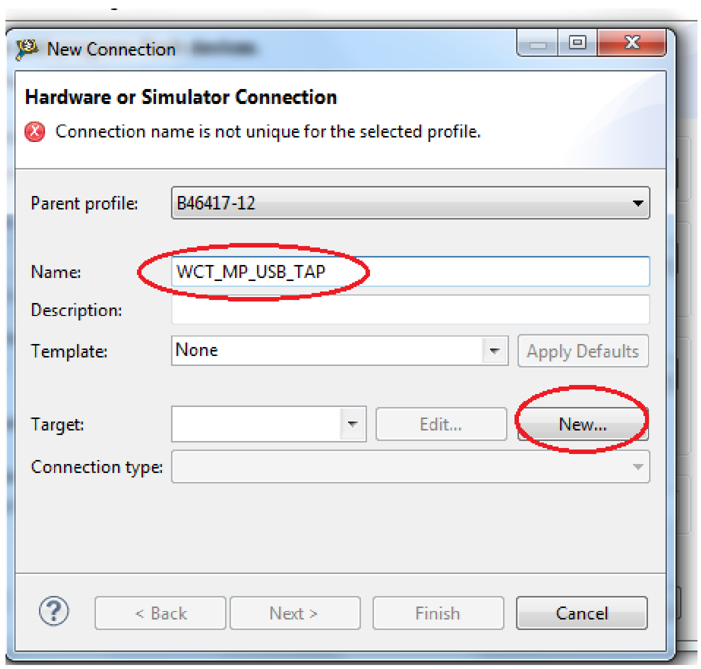


Figure 34. Bin file download (3)

4. Enter a target name, and then select **MWCT1012** from the **Target type** drop-down list.

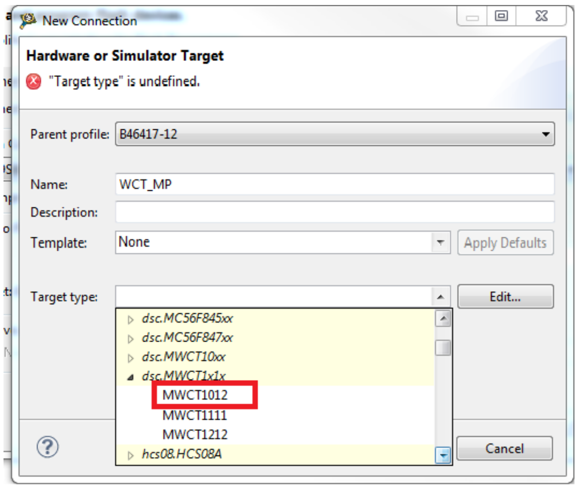


Figure 35. Bin file download (4)

5. Select **Execute reset** and **Initialize target**, set the initialization file path to the CodeWarrior version 10 installation folder, and select **MWCT1012.tcl** for the MWCT1012 chip. The general path is:

   C:\Freescale\CW MCU v10.7\MCU\lib\wizard_data\DSC\DataBase\init_files.

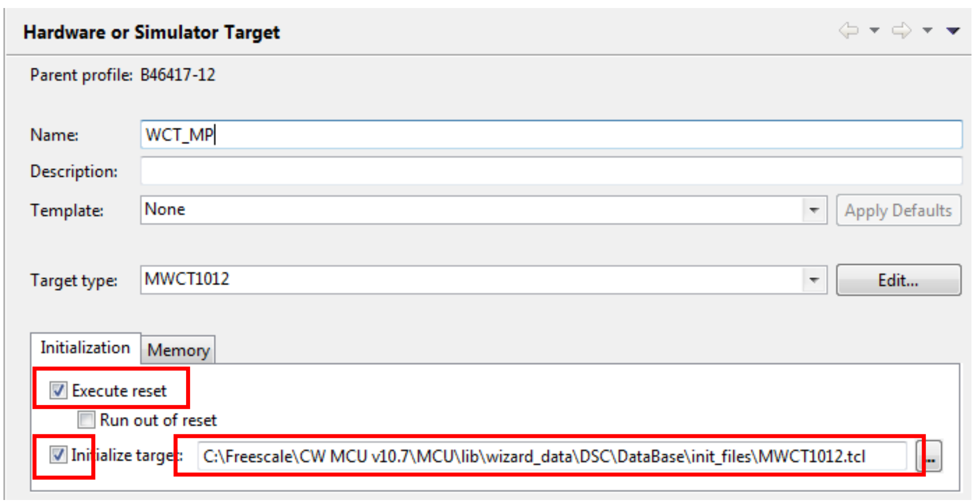


Figure 36. Bin file download (5)

6. Set the memory configuration file path. For the MWCT1012 chip, it is MWCT1012.mem, located under the CodeWarrior version 10 installation folder. Then, click **Finish**.

   The general path is:

   C:\Freescale\CW MCU v10.7\MCU\lib\wizard_data\DSC\DataBase\mem_files

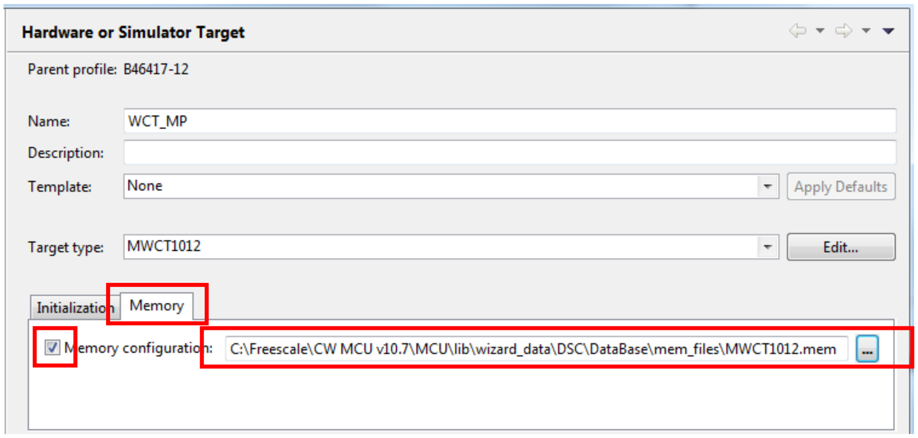


Figure 37. Bin file download (6)

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

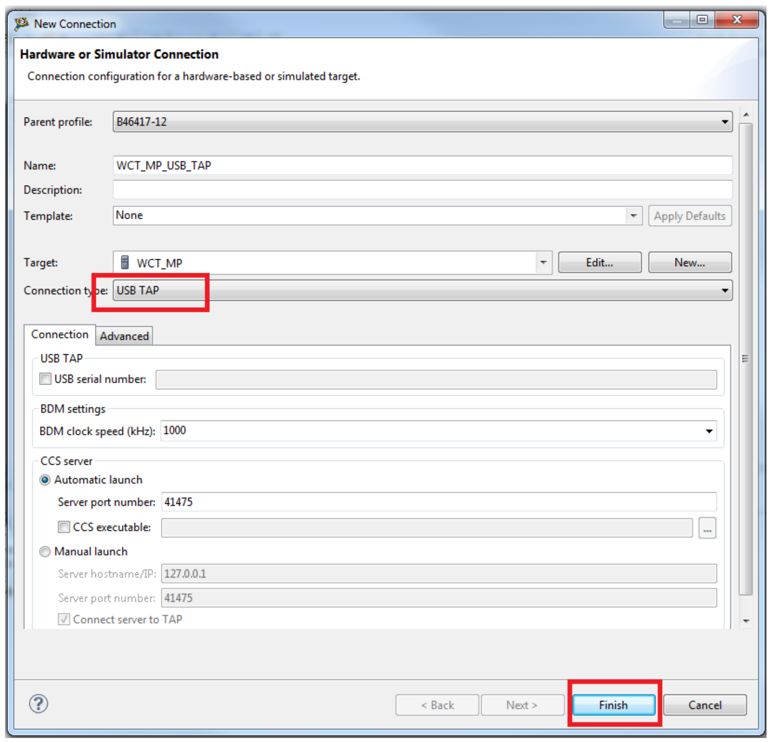


Figure 38. Bin file download (7)

8. Set the Bin file path. Before downloading, save the configuration to the workspace for future downloading. Click **Erase and Program**.

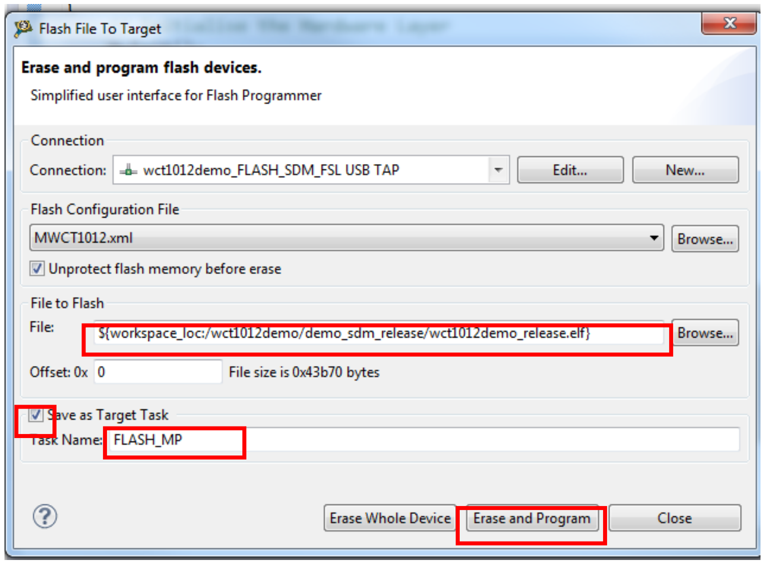


Figure 39. Bin file download (8)

NOTE

The file path should contain only English letters. Otherwise, the flash cannot recognize it. For a new board, execute **Erase Whole Device** when you select .elf as the Bin file.

9. The flashing progress is displayed in the CodeWarrior version 10 console window. After flashing is completed, reset the board to make MWCT1012 run.

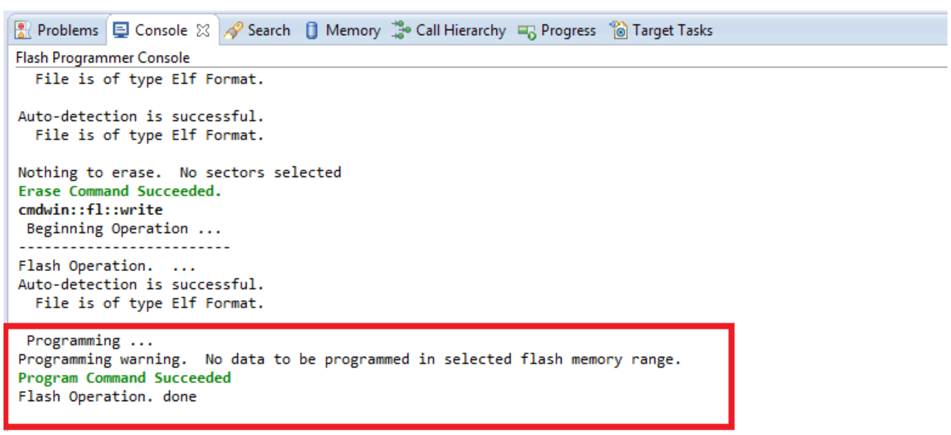


Figure 40. Bin file download (9)

7.2.4 Using the FreeMASTER GUI for calibration

NXP provides the FreeMASTER GUI tool for calibration and parameters tuning. FreeMASTER configuration file **wct1012.pmp** is saved under <Software_Release>/example/wct1012. See the *WCT1012 15W Single Coil TX V3.1 Runtime Debug User's Guide* (WCT1012V31RTDUG) for calibration and MP parameters tuning. For the FreeMASTER tool, see www.nxp.com/products/power-management/wireless-charging-ics/freemaster-run-time-debugging-tool:FREEMASTER.

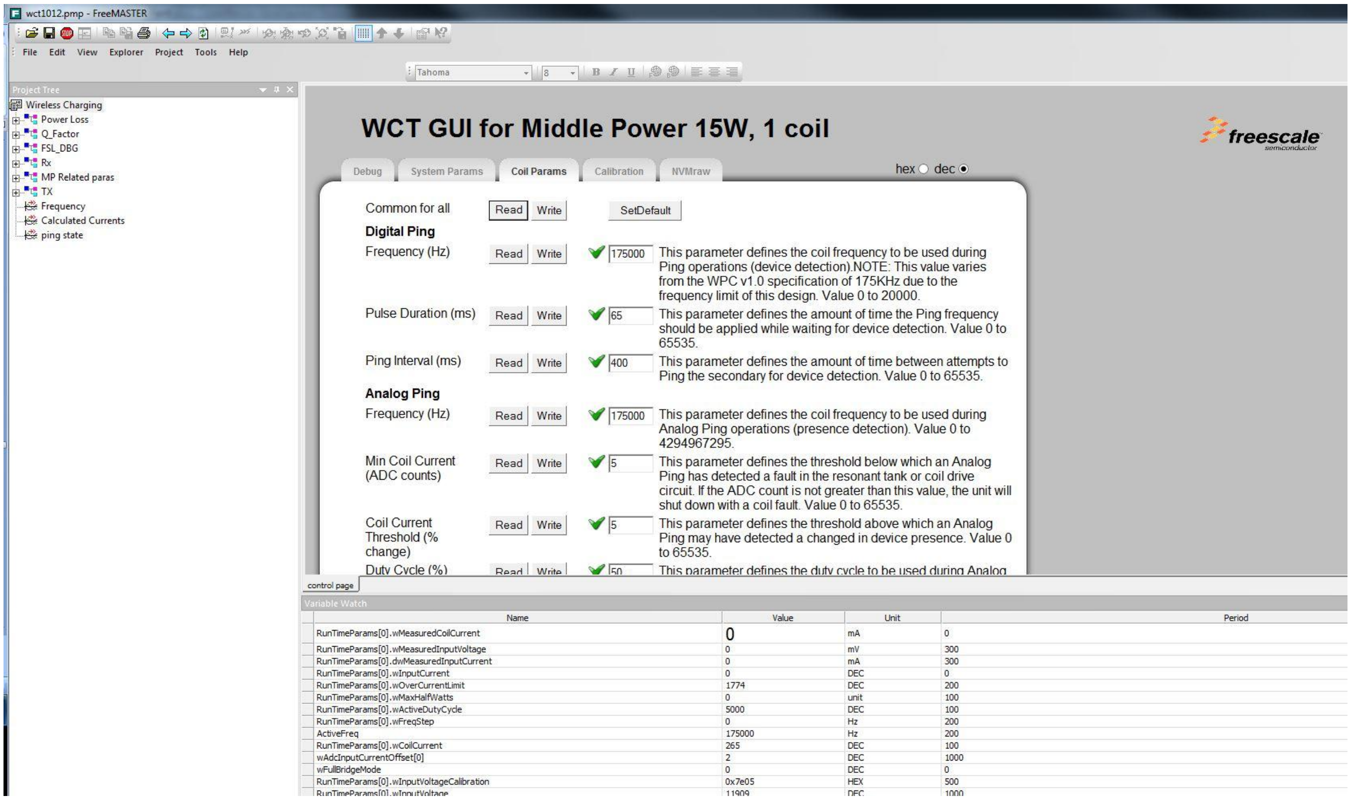


Figure 41. FreeMASTER GUI tool

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

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

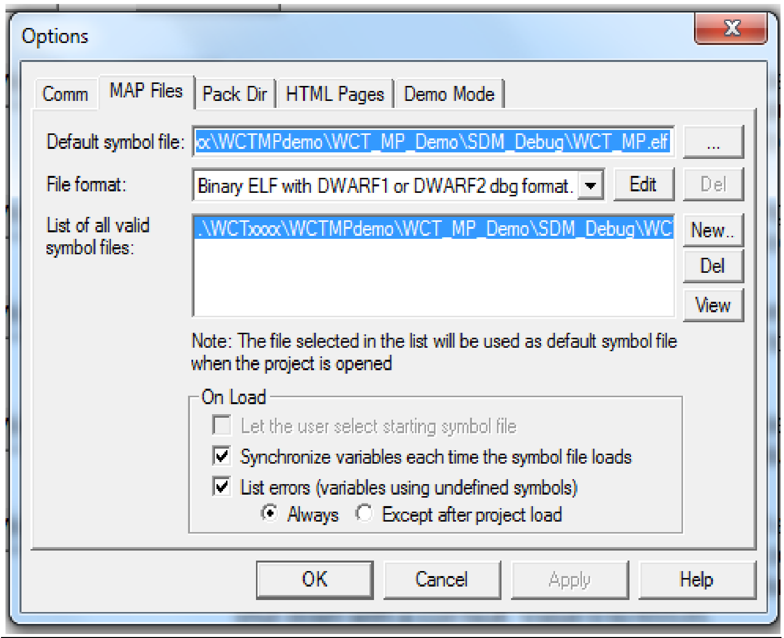


Figure 42. Selecting symbol file

2. Perform settings for using the USB TAP debugger.
   Select **FreeMASTER Code Warrior-CCS JTAG/OnCE** in **Freemaster Project** –> **Options** –> **Comm**, as shown in the following figure.

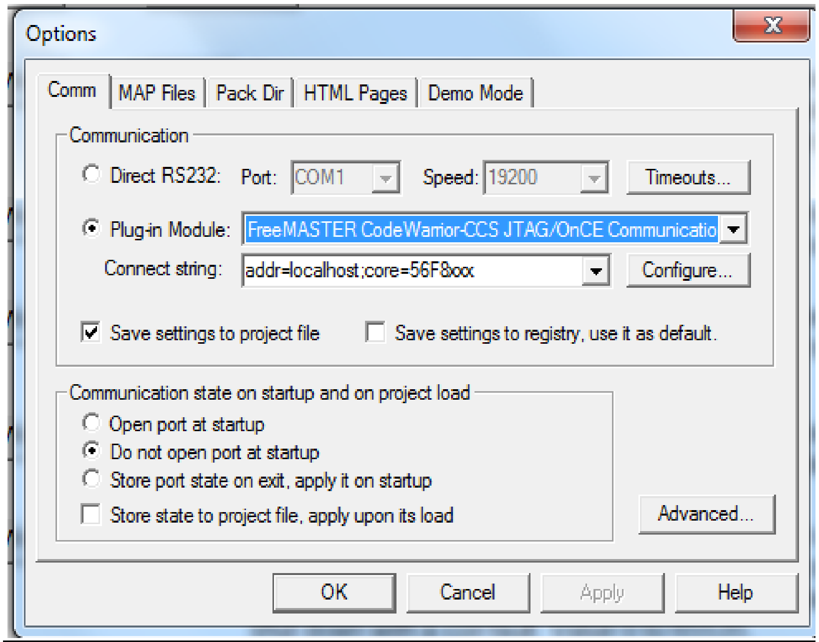


Figure 43. Options dialog box

3. Perform settings for using P&E Multilink FX debugger.
   Select **FreeMASTER BDM JTAG/OnCE** in **Project** –> **Options** –> **Comm**, as shown in the following figure.

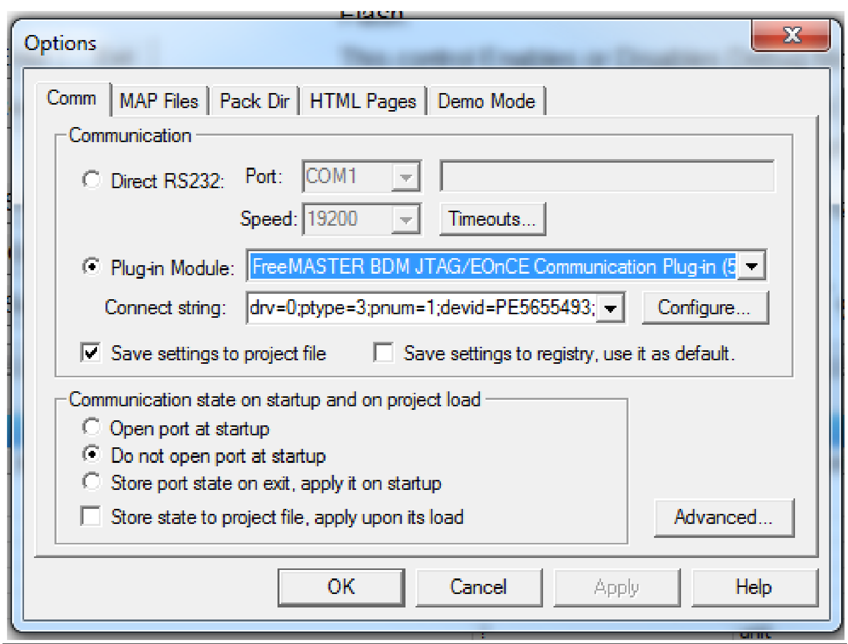


Figure 44. Options dialog box

4. Perform settings for using the SCI/OSJTAG debugger.

   SCI can be used for FreeMASTER connection in MP demo, and the baud rate is 19200. If OSJTAG is used to link FreeMASTER through SCI, some changes are needed. Perform the following steps to enable the OSJTAG debugger connection:

   a. Import wct1012demo in CW10. Enable macro "FMSTR_USE_SCI" and disable "FMSTR_USE_JTAG". They are defined in 15W_MP –> example –> wct1012 –>hal -> freemaster_cfg.h as follows:

   ```
   #define FMSTR_USE_SCI 1 /* To select SCI communication interface */

   #define FMSTR_USE_JTAG 0 /* 56F8xxx: use JTAG interface */
   ```

   Enable SCI0 by setting macro "QSCI0_ENABLED" TRUE, which is defined in example -> wct1012 –> wct_hal_cfg.h as follows:

   ```
   #define QSCI0_ENABLED      TRUE   // Enable the SCI0 driver
   ```

   If USB TAP or P&E Multilink FX is used, these macros need to be set as follows:

   ```
   #define FMSTR_USE_SCI 0 /* To select SCI communication interface */

   #define FMSTR_USE_JTAG 1 /* 56F8xxx: use JTAG interface */
   ```

   b. Rebuild the demo, and download it according to the used debugger.

   c. Link the SCI port on OSJTAG to MP board as shown in the following figure.

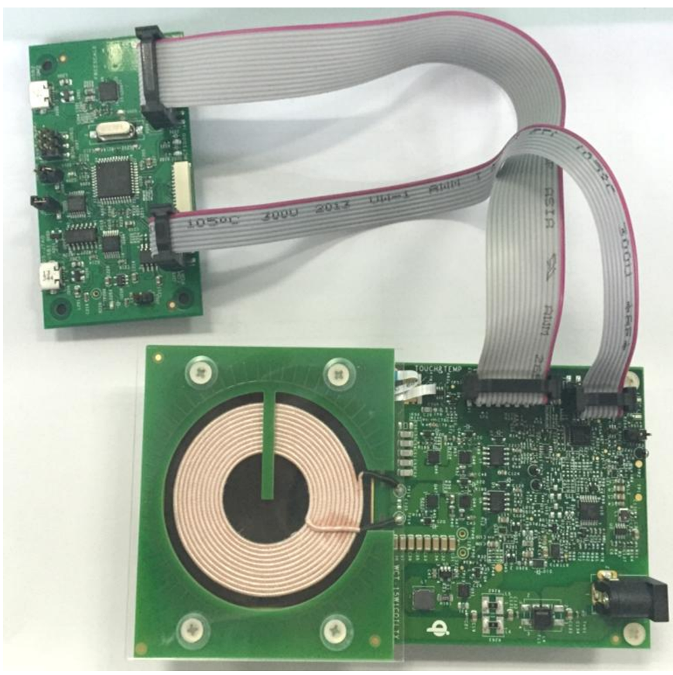


Figure 45. Using OSJTAG for FreeMASTER connection

NOTE

Insert one row of the pins into the SCI port, as shown in the following figure.

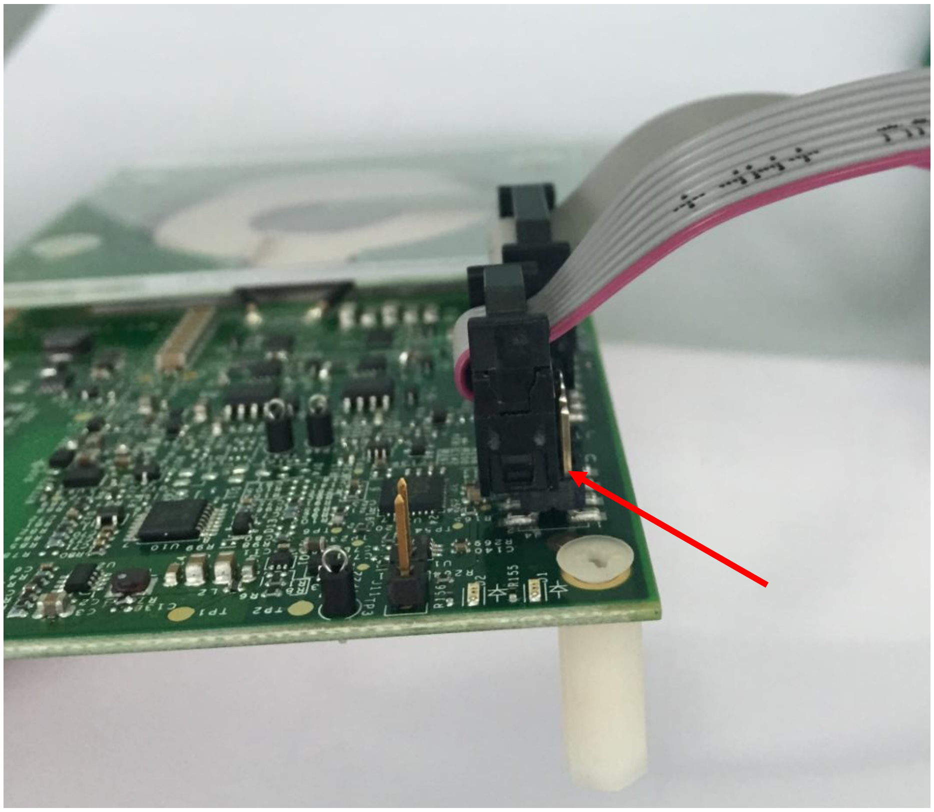


Figure 46. SCI port on OSJTAG board

d. Before connecting FreeMASTER, confirm that the baud rate of the computer com port is 19200. It can be found in Computer –> Manage –> System Tools –> Device Manager –> Ports. Right-click **OSBDM/OSJTAG** and choose **Properties**. Then the baud rate can be changed as shown in the following figure.

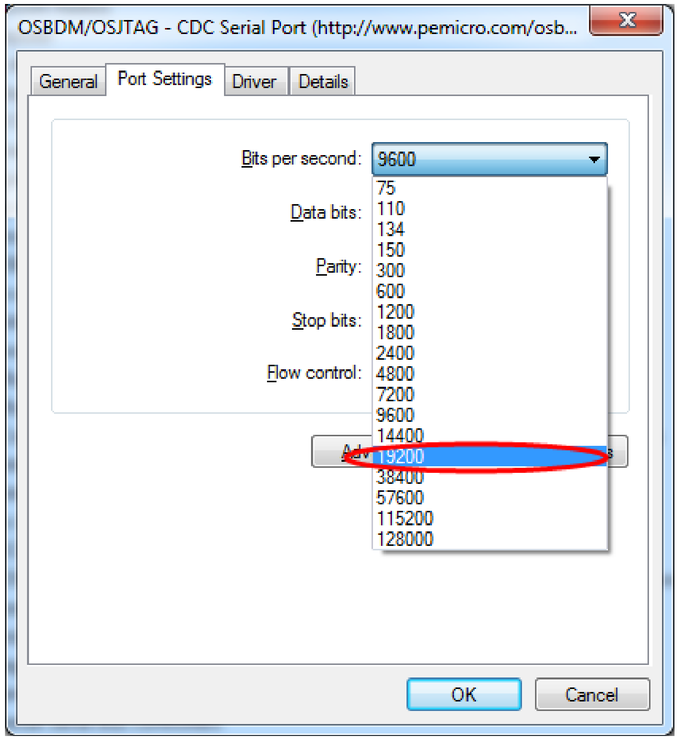


Figure 47. Computer Band Rate Setting

e. Select **FreeMASTER BDM JTAG/OnCE** in **Project** –> **Options** –> **Comm** in FreeMASTER tool, as shown in the following figure.

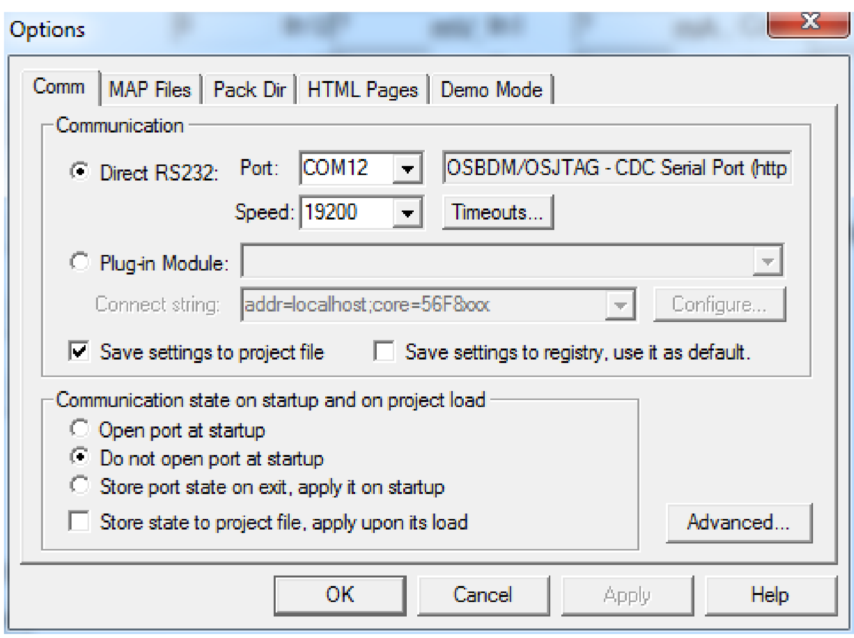


Figure 48. Option dialog box

f. Click the **Start/Stop** button to make the FreeMASTER connection work.

7.2.5 Enabling or disabling board functions

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

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

```
/example/wct1012/wct_hal_cfg.h, peripheral_cfg.h

/example/wct1012/application/application_cfg.h

/example/wct1012/application/main.c
```

In application_cfg.h, you can configure these functions:

- `FOD enable/disable`
  `#define FOD_ENABLE         TRUE   // FALSE to ensure that FOD is not working.`

In wct_hal_cfg.h, you can configure these functions:

- `Debug console enable/disable`
  `#define DEBUG_CONSOLE_QSCI0     TRUE    // We are using Peripheral QSCI0 for diagnostics`

In main.c, you can configure these functions:

- `Q measurement enable/disable`
  `gStaticConf.wctlib_cfg_switch.qfactordetection = 1;  // 0 to disable Q measurement`
- `enable/disable protection in library`
  `gStaticConf.wctlib_cfg_switch.libprotect_enable = 1; // 0 to disable protection in library`

NOTE

Only one macro of DEBUG_CONSOLE_QSCI0 and FMSTR_USE_SCI can be TRUE at one time. The debug console function is disabled in the release v3.1 due to code size limitation.

7.3 Test

7.3.1 Basic charging test

When the software work is complete, power on the MP demo with a standard 12 V adapter to make it work.

Put the MP Qi receiver on the charging pad, and make sure the coil is aligned and the load is in the allowed range (up to 15 W). The TX charges the RX properly.

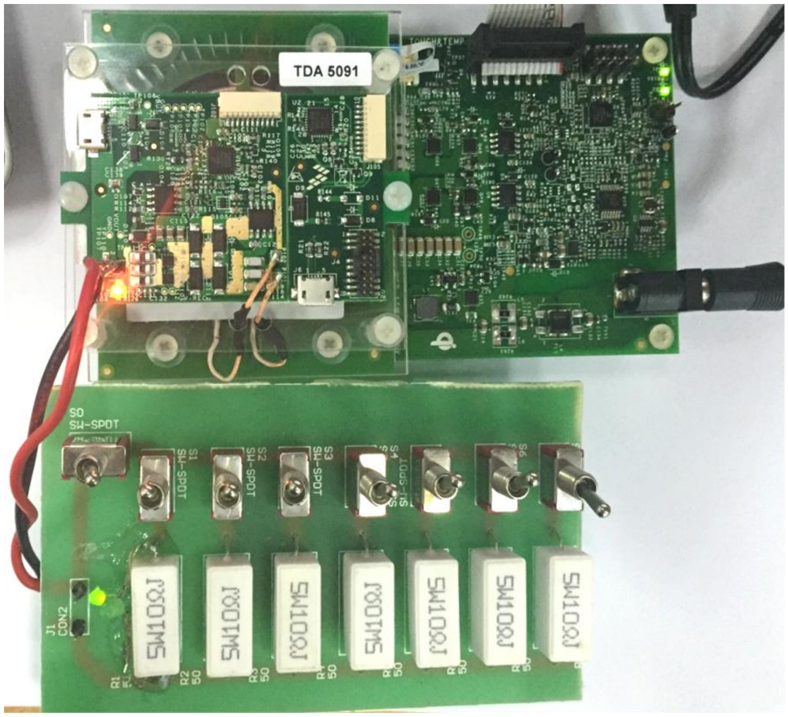


Figure 49. Working system

The defined LED display modes for different TX working states are shown in the following table:

Table 2. LED display modes

LED configuration option	Description	LED No.	LED operational status					
			Standby	**Charging**	**Charging complete**	**FOD fault**	**TX fault**	**RX fault**
Default	Default choice	LED1	Off	Blink slow	Off	On	On	On
		LED2	Blink slow	On	On	Off	Off	Off
Option-1	Choice-1	LED1	Off	Blink slow	On	Off	Off	Off
		LED2	Off	Off	Off	Blink fast	Blink fast	Blink fast
Option-2	Choice-2	LED1	Off	On	Off	Off	Off	Off
		LED2	Off	Off	Off	On	Blink slow	Blink slow
Option-3	Choice-3	LED1	Off	Blink slow	On	Blink fast	Blink fast	Blink fast
		LED2	–	–	–	–	–	–

7.3.2 Signals on the board

The main signals on the MP TX board are shown in the following figure.

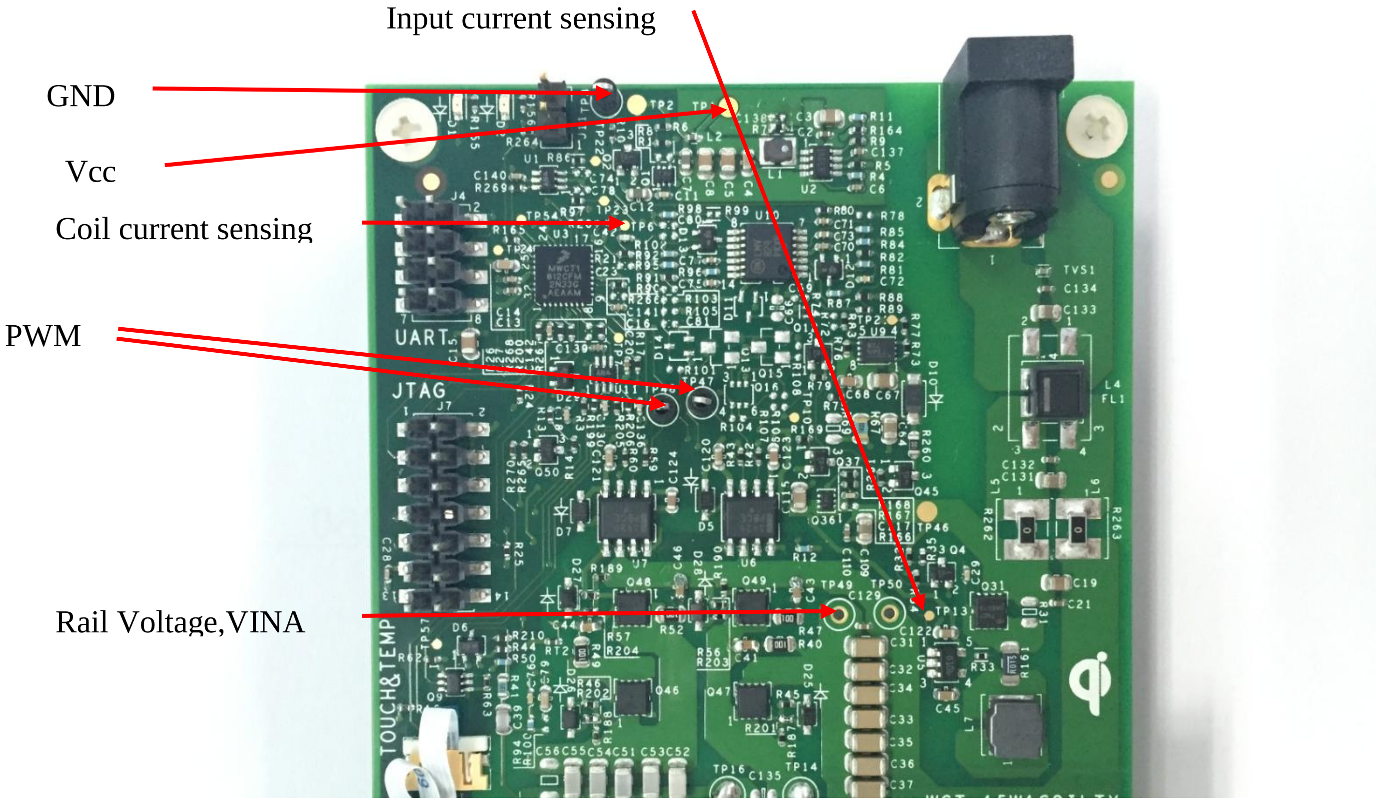


Figure 50. Test Points on WCT_MP

- TP1: Vcc, controller input voltage 3.3V
- TP3: GND
- TP13: Input current sensing
- TP6: Coil current sensing
- TP47&48: PWM1&2, PWM signals to pre-driver
- TP49: Rail voltage, VINA, 12V except during Q factor measurement

7.3.3 Test environment

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

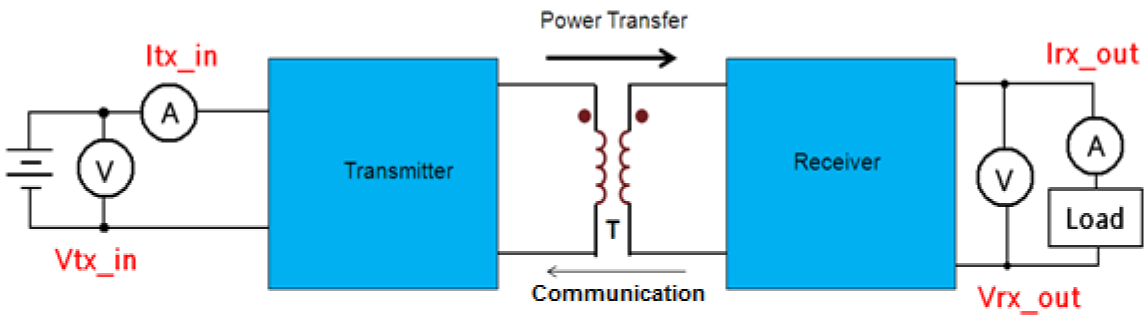


Figure 51. Test environment

7.3.4 Measurements

WCT_MP is compatible with low-power RX and medium-power RX. With low-power RX, WCT_MP runs normally under half bridge. When the medium power RX is put on WCT_MP, full bridge mode of TX is enabled. The operational mode of TX can be controlled by controlling PWM1 and PWM2. The following provides the examples for measuring signals on the board.

1. Measure the signals when the TX board works under analog ping and digital ping.

   Ch1: PWM1 Ch2: PWM2

   Ch4: Coil current

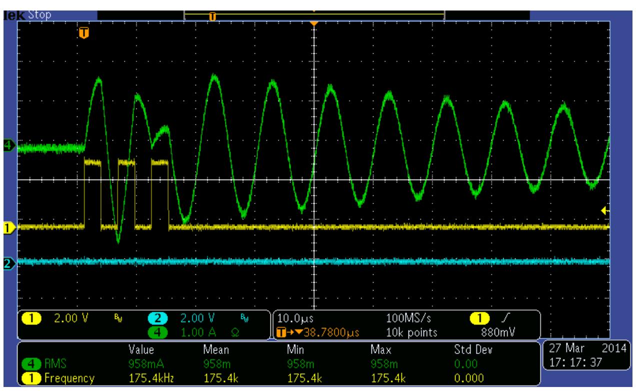


Figure 52. Analog ping

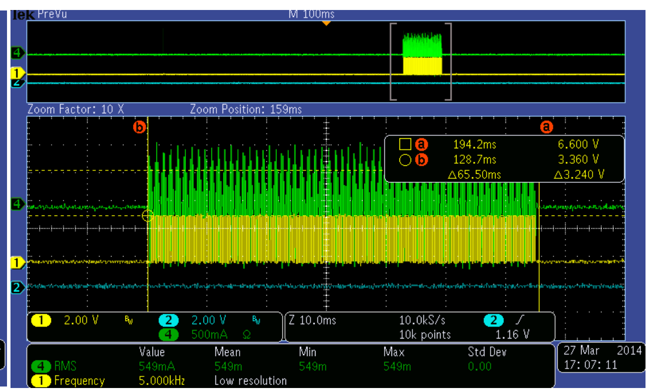


Figure 53. Digital ping

2. MP TX can work under different control mode with different load and RX.

   Ch1: PWM1 Ch2: PWM2

   Ch3: RX bus voltage Ch4: Coil current

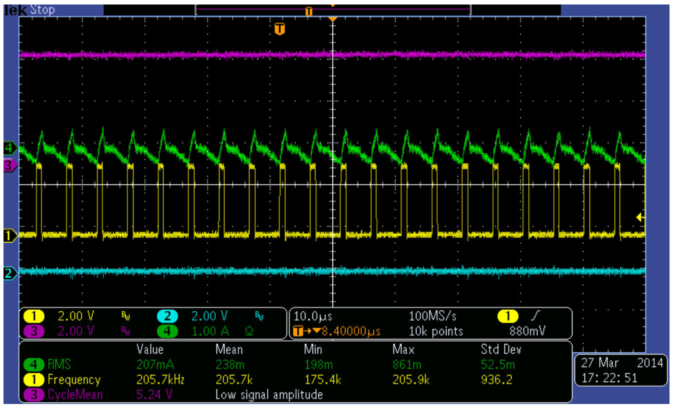


Figure 54. Half bridge, duty cycle control

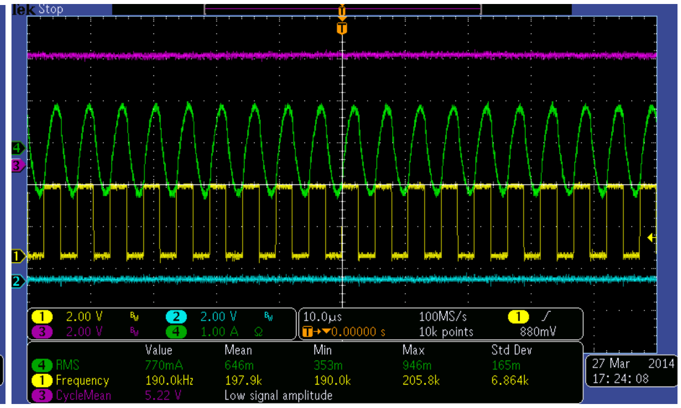


Figure 55. Half bridge, frequency control

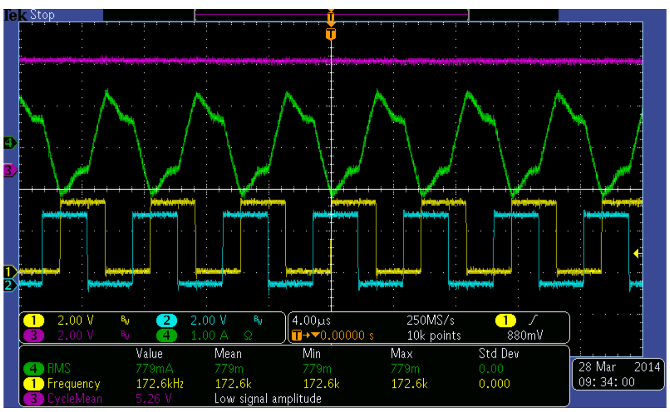


Figure 56. Full bridge, phase shift control

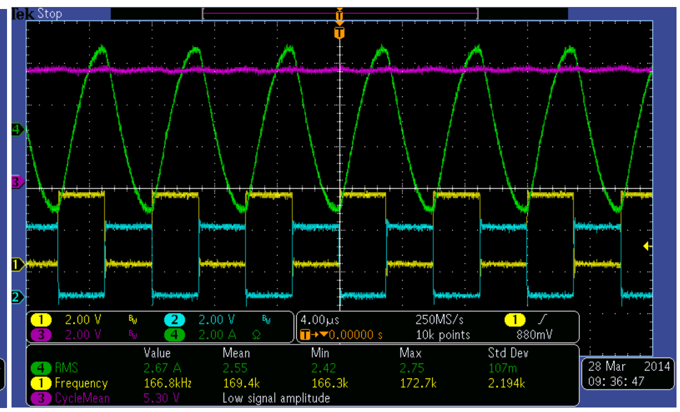


Figure 57. Full bridge, frequency control

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

   Ch1: PWM1 Ch2: COMM

   Ch3: RX bus voltage Ch4: Coil current

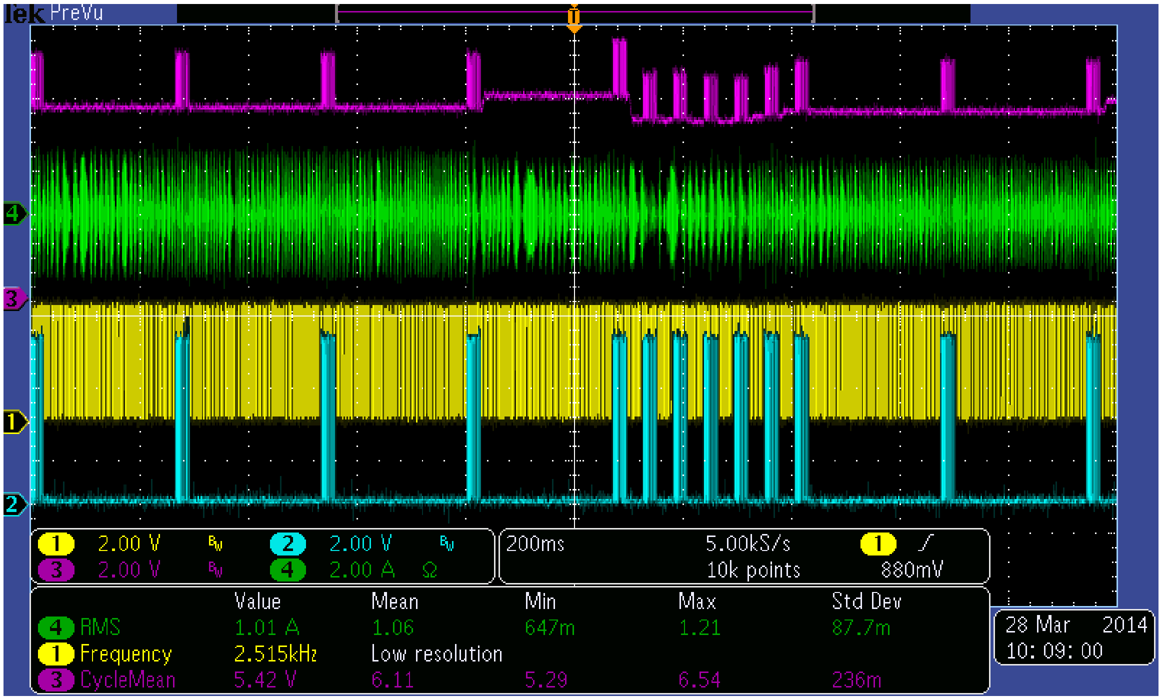


Figure 58. System response on load dump

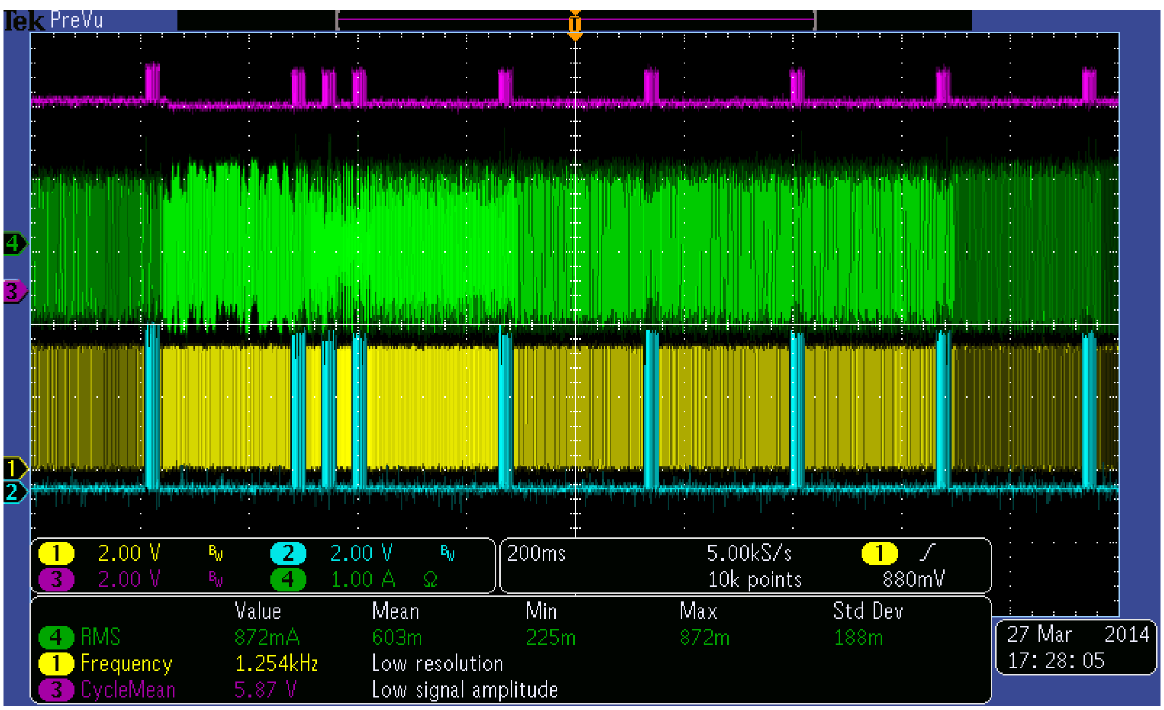


Figure 59. System response on load step

7.3.5 Q measurement

7.3.5.1 Free Resonance Q factor detection signal

Free resonance Q detection includes one pre-charge signal and several resonance signals to get the decay rate. Figure 60 shows the pre-charge signals. CH1 is PWM1 and CH2 is sampling voltage of resonance signal (TP23).

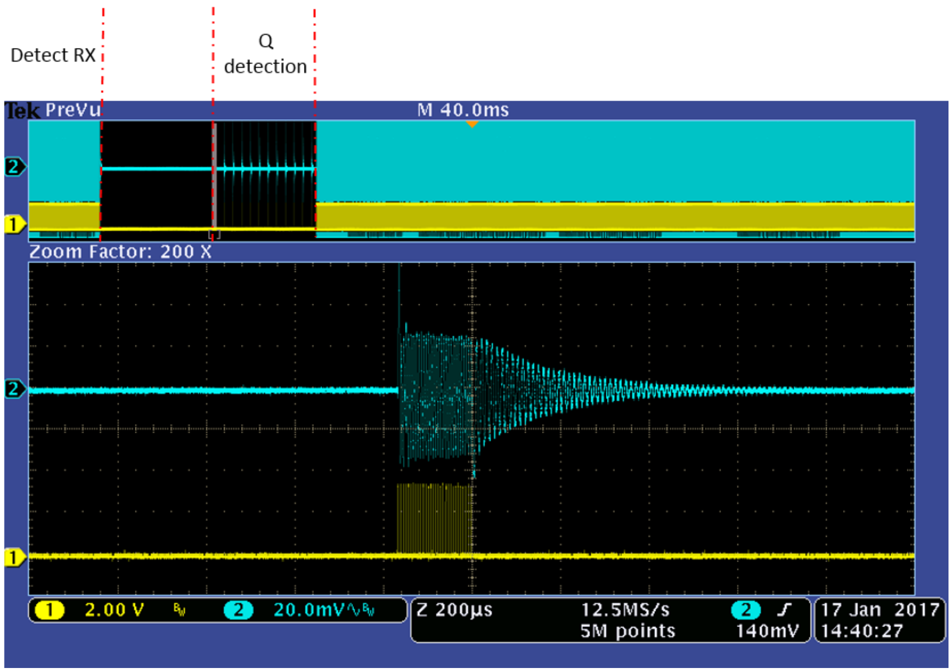


Figure 60. Free resonance Q detection pre-charge signal

The following figure shows one of the resonance signals (TP23).CH1 is PWM1 and CH2 is sampling voltage of resonance signal (TP23).

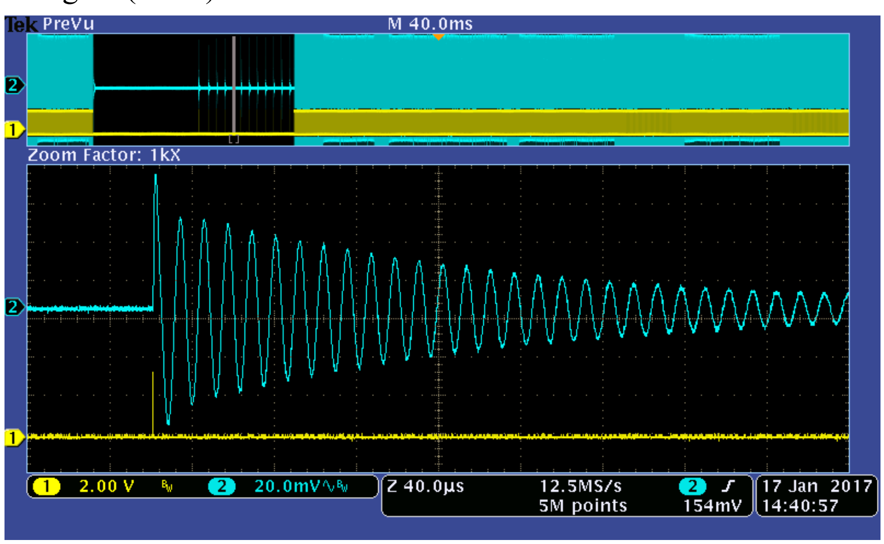


Figure 61. Resonance signal during free resonance Q detection

7.3.5.2 External driver Q factor detection signal

The rail voltage of TX switches to low voltage generated by LDO to make sure not to wake up RX during Q factor measurement. In Figure 62, CH1 is PWM1, CH2 is the sampling driver signal, and CH4 is the sampling resonance signal. The driver signal near resonant frequency is relatively low.

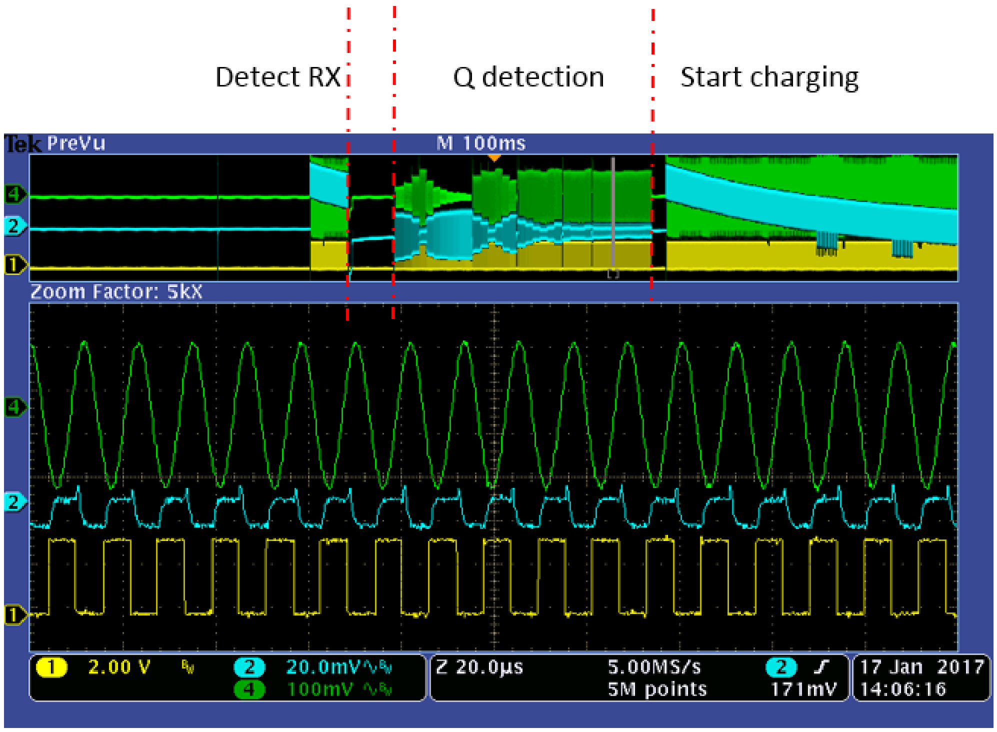


Figure 62. External driver Q detection signals

8 Flash Extension of WCT1011

MWCT1011CFM is the premium version of MWCT1012CFM with 64 KB flash, which can replace MWCT1012CFM directly. The V3.1 release supports IC MWCT1011CFM and MWCT1012CFM with 48 KB flash at the same time. If MWCT1011CFM is used for additional design and product differentiation, some modifications are needed to extend flash to 64 KB based on the V3.1 release.

1. Change memory configuration in the command file:

`15W_MP\build\demo\wct1012demo\Project_Settings\Linker_Files\ MWCT1012_Internal_PFlash_SDM.cmd`

```
.p_flash_ROM          (RX)  : ORIGIN = 0x0208,   LENGTH = 0x7BF8   # reserved for program code
.p_dflash_DATA         (RX)  : ORIGIN = 0x7E00,   LENGTH = 0x0200   # reserved for EEPROM
emulation (2 1024 byte sectors)
```

2. Change flash configuration in files:
   - `15W_MP\example\wct1012\driver\flash.h`

     ```
     #define NUM_FLASH_SECTORS                       64
     ```
   - `15W_MP\example\wct1012\hal\wct_hal_cfg.h`

     ```
     #define DATA_FLASH_BASE_SECTOR_NUMBER           63U
     ```
3. Change the chip type and replace the .mem file and .tcl file in debugger configuration:
   - Find MWCT1011.mem and MWCT1012.tcl in the installation path of CW 10.7:

     .mem file is in path `..\ CW 10.7\CW MCU v10.7\MCU\lib\wizard_data\DSC\DataBase\mem_files`

     .tcl file is in path `..\CW 10.7\CW MCU v10.7\MCU\lib\wizard_data\DSC\DataBase\init_files`
   - Copy the file to the location: `\15W_MP\build\demo\wct1012demo\Project_Settings\Debugger`
   - Replace the .mem and .tcl file in debugger configurations. Take "WCT_MPTX_1COIL_Demo_Release_SDM_OSJTAG" for example. The steps are as shown in Figure 63 to Figure 66.

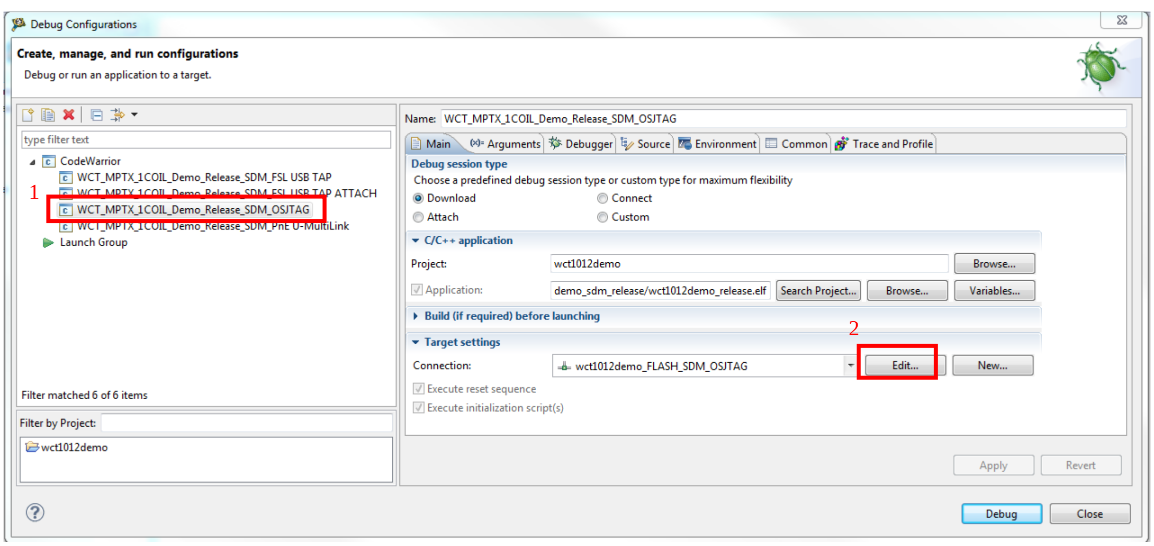


Figure 63. Changing the peripheral settings of debugger (1)

Select the chip type MWCT1011.

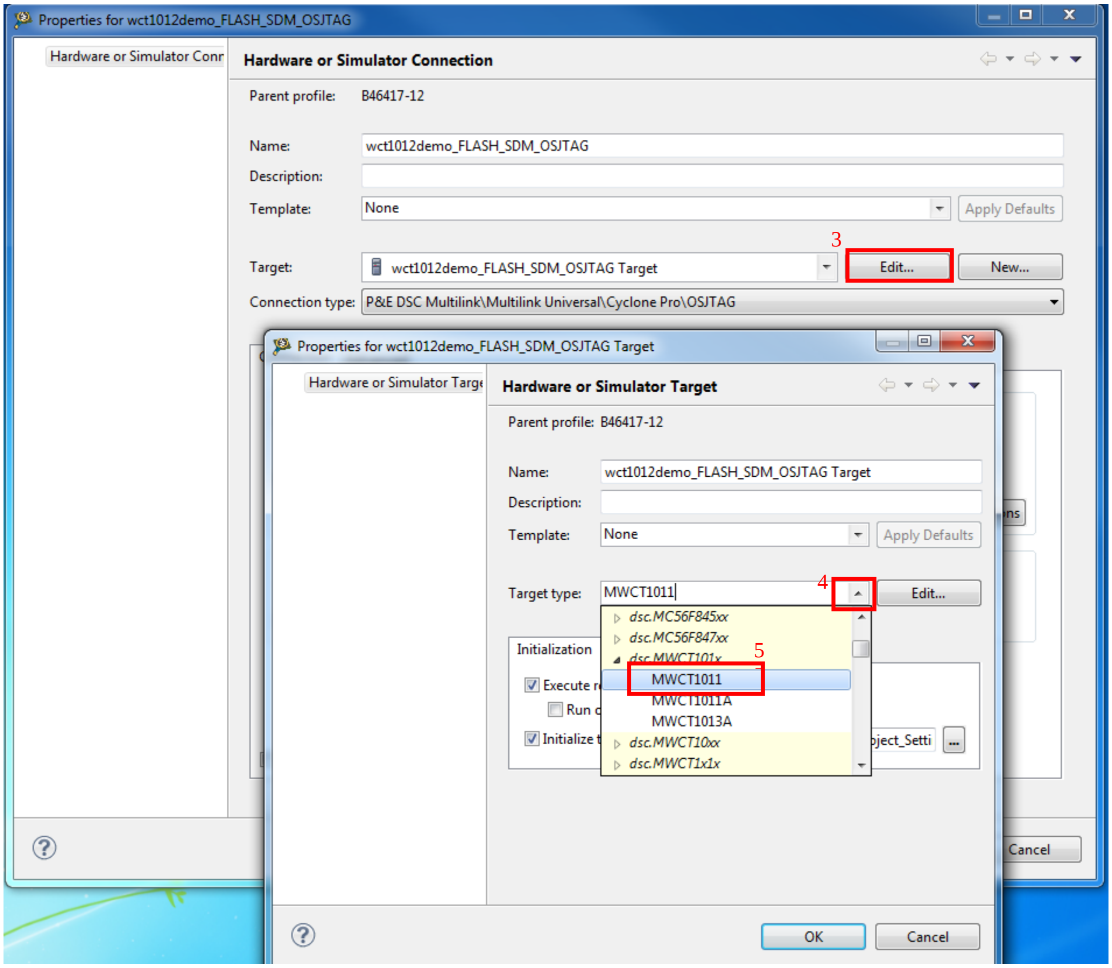


Figure 64. Changing the peripheral settings of debugger (2)

Replace the .tcl file.

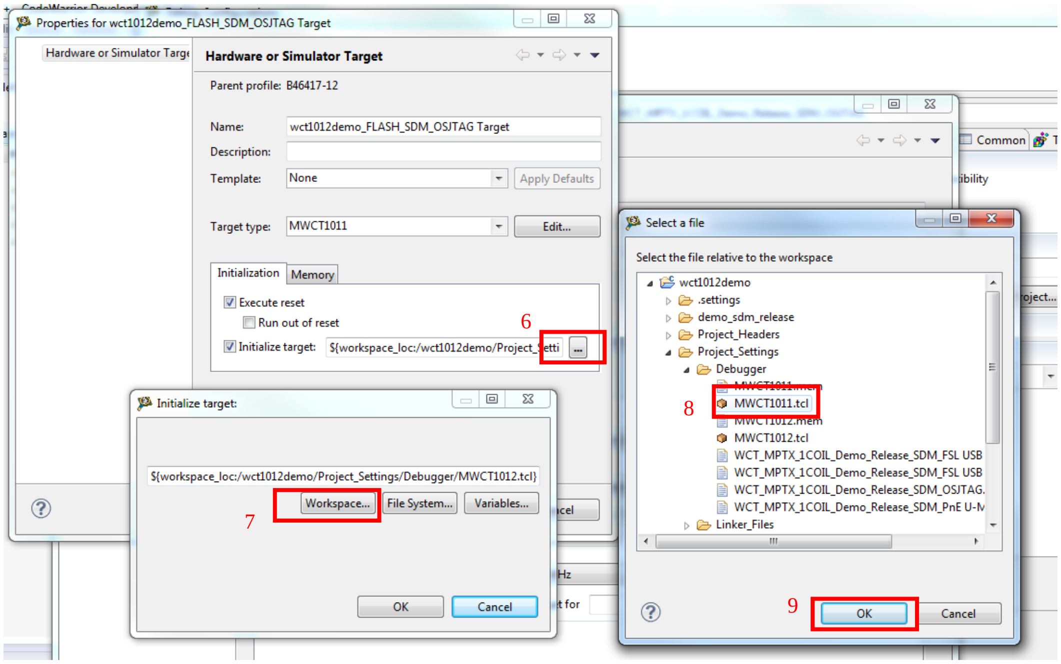


Figure 65. Changing the peripheral settings of debugger (3)

Replace the .mem file.

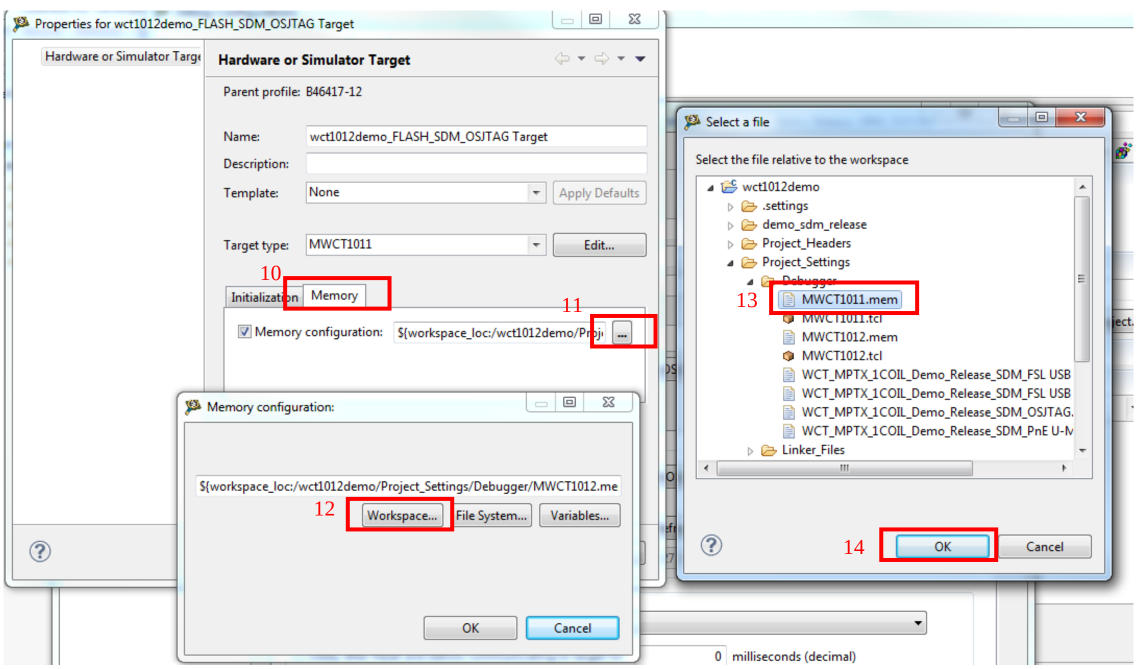


Figure 66. Change the peripheral settings of debugger (4)

Click **OK** to save these changes.

After finishing these steps, the 64KB flash of MWCT1011CFM is available.

9 References

- NXP wireless charging solution page:
  www.nxp.com/products/power-management/wireless-charging-ics
- NXP Codewarrior 10 IDE page:
  www.nxp.com/products/software-and-tools/software-development-tools/codewarrior-development-tools:CW_HOME
- NXP FreeMASTER tool page:
  www.nxp.com/products/power-management/wireless-charging-ics/freemaster-run-time-debugging-tool:FREEMASTER
- WPC page:
  www.wirelesspowerconsortium.com
- WCT1012 Documents:
  - *WCT1012 15W Single Coil TX V3.1 Reference Design System User's Guide* (WCT1012V31SYSUG)
  - *WCT1012 TX V3.1 Library User's Guide* (WCT1012V31LIBUG)
  - *WCT1012 15W Single Coil TX V3.1 Runtime Debugging User's Guide* (WCT1012V31RTDUG)
  - *WCT-15W1COILTX V3.1 Release Notes* (WCT1012V31RN)
  - *Rework List for the WCT-15W1COILTX Rev.3 Board* (WCT1012V31RL)

10 Revision History

Table 3. Revision history

Revision number	Date	Substantive changes
0	02/2017	Initial release.

How to Reach Us:

Home Page:

nxp.com

Web Support:

nxp.com/support

Document Number: WCT1012V31SYSUG
Rev. 0
02/2017