QBLUE Quick Start

Version 0.6

Table of Contents

1. Introduction

QBLUE Development Kit is a complete solution for Quintic's QBLUE Bluetooth series development, testing and evaluation. This user guide contains reference information of hardware and software contained in QBLUE Bluetooth QN902x Development Kit. It starts with an overview of hardware，system requirements and software architecture, followed by a quick introduction on how to get started with the DK and demo applications. Programming and debugging steps are illustrated in detail to help users during application development.

1.1 DK Hardware

The QN902x Development Kit contains the following hardware components:

- QBLUE Development Platform Motherboard
- QN902x EVB
- Pulse Antenna
- USB Cable

1.2 System Requirements

To use the QN902x SDK, please follow the system requirements below:

- A PC running with Windows XP or Windows 7
- Keil MDK-ARM
- J-Link Software

1.3 Software Architecture

The software platform of QBLUE consists of two main parts: Firmware and Application project. The code executed in the ROM is called Firmware. The code executed in the SRAM is called Application project.

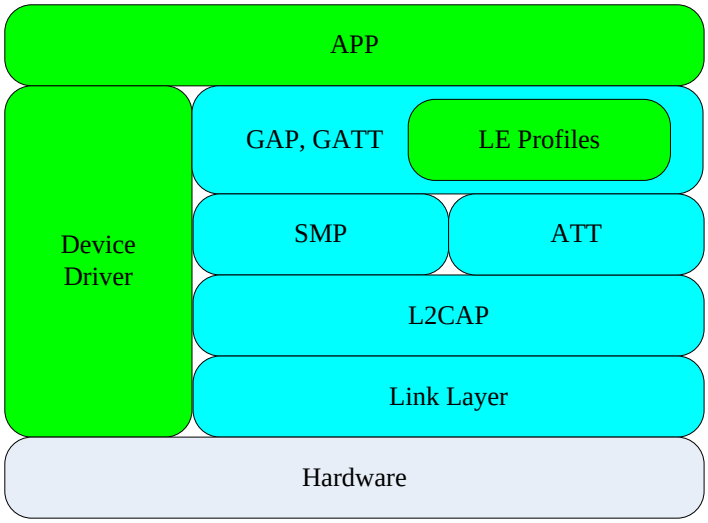

Figure 1 Software Architecture

2. Getting Started

2.1 Install Keil MDK-ARM, Setup_JLinkARM and QBlue

Before plugging the DK platform into the PC via the USB cable, it is highly recommended to install Keil MDK-ARM, QBlue-x.x.x.exe and Setup_JLinkARM.exe to your PC first.

1. Download and install Keil MDK-ARM from https://www.keil.com/demo/eval/arm.htm. If you have Keil MDK-ARM already installed, go to step 2.
2. Install QBlue-x.x.x.exe from Quntic released SDK package. If you choose to add Quintic device database to Keil manually, please install the tool "QnDevDBforKeil.exe" located at the default installation path "C:\Quintic Corporation\QBlue-x.x.x\Tools\QnDevDBforKeil". Otherwise, it will show some error when compiling project.
3. Download and install Setup_JLinkARM.exe from www.segger.com. The serial number as Segger ID for each board is needed, which is already attached at the back side of the motherboard.

2.2 Hardware Setup

1. Connect the antenna to the SMA connector on the QN902x EVB. Tighten the antenna's screw firmly on to the SMA connector. If not properly connected, you might see reduced RF performance.
2. Next, mount the QN902x EVB on to connectors J1 and J2 on the QBLUE Development Kit Motherboard.
3. Connect a USB cable from the Motherboard to your computer.
4. To activate Virtual COM, connect pin17 of jumper J25 with pin17 of J4, also connect pin 18 of J25 and J4 together.

The hardware setup is completed as shown in Figure 2.

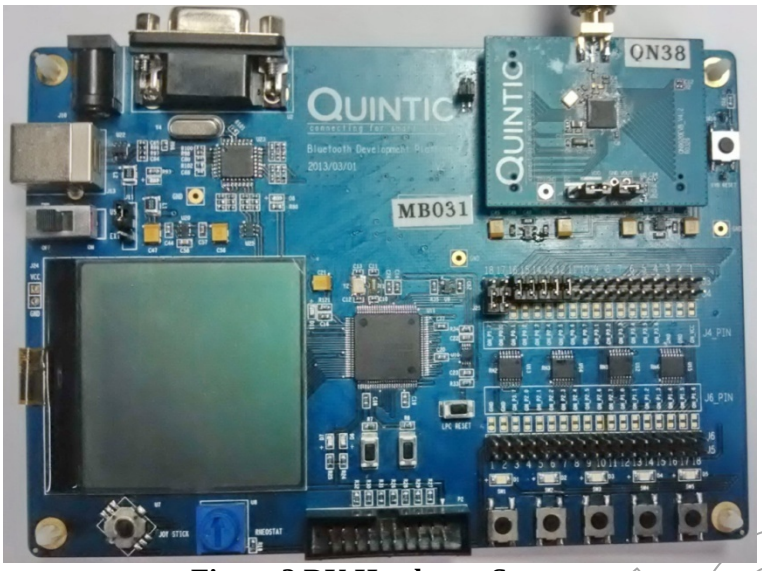

Figure 2 DK Hardware Setup

2.3 Install board drivers

You can now turn the board on after connecting your DK platform to the computer with a USB cable. A “Found new Hardware” dialog box will pop out as shown in Figure 3, for you to locate the driver needed. Please select “No, not this time” and continue with “Next”.

Figure 3 Found New Hardware Window (Windows XP)

After clicking next, the window as shown in Figure 4 will appear. Please select “Install from a list or specific location” to install the driver for the virtual COM.

On the board, virtual COM and JLINK OB debugger use the same USB port through a USB hub. This section deals with installing virtual COM driver. For more information on installing Jlink OB please log on to www.segger.com.

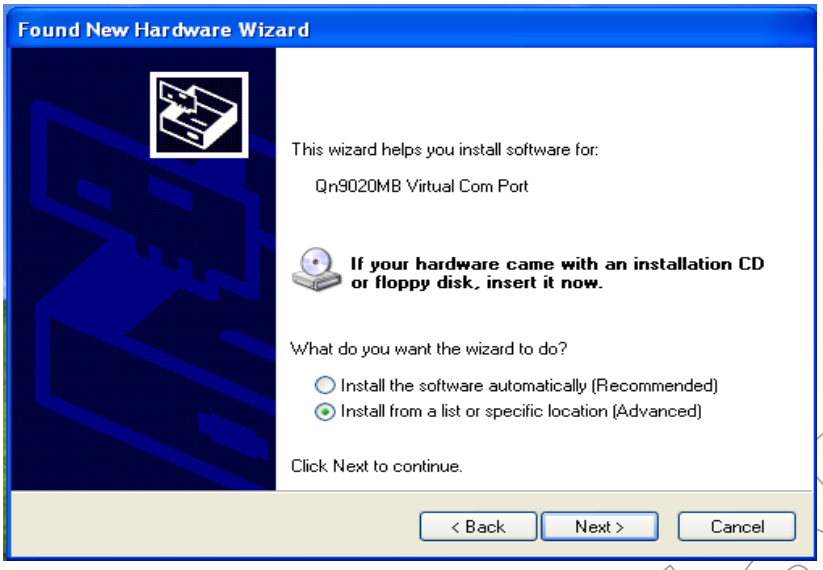

Figure 4 Select automatic installation of software (Windows XP)

By default, drivers for the platform are located at C:\Quintic Corporation\QBlue-x.x.x\USBCdcDrv, as shown in Figure 5 below.

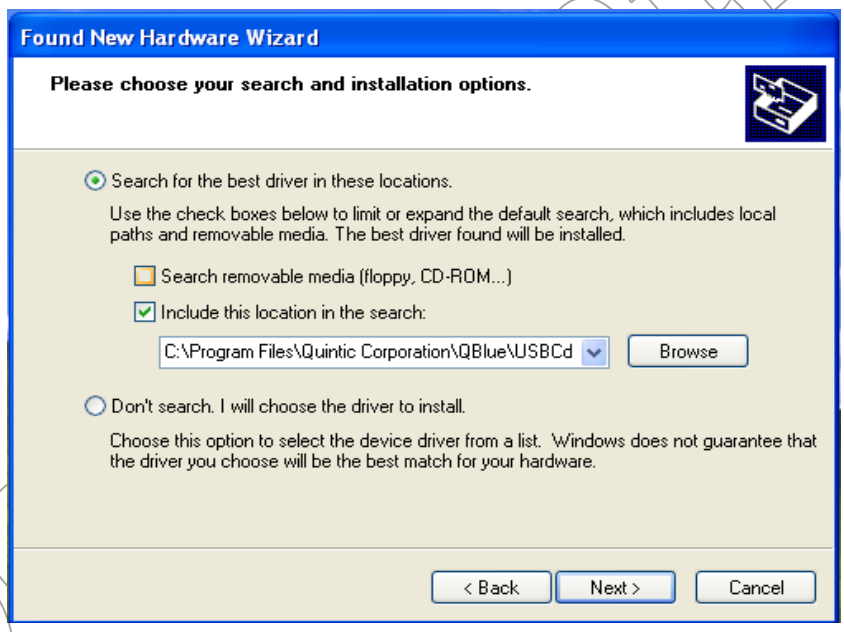

Figure 5 Manually locate driver

The driver is now installed and the computer should be ready to use the virtual COM.

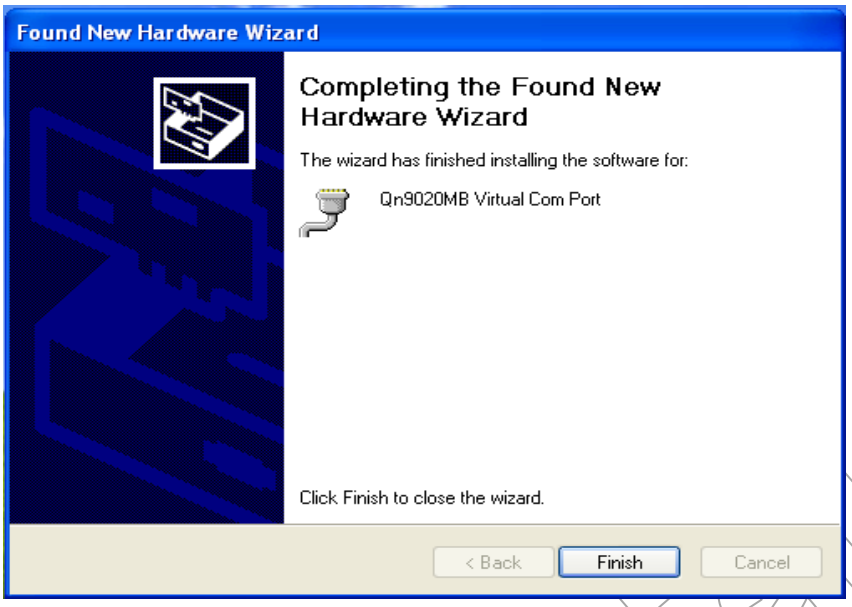

Figure 6 The driver installation is completed (Windows XP)

You can verify that the driver is properly installed by opening the Device Manager (Figure 7). When the platform is connected, the "COM and LPT" list contains "QN9020 virtual COM".

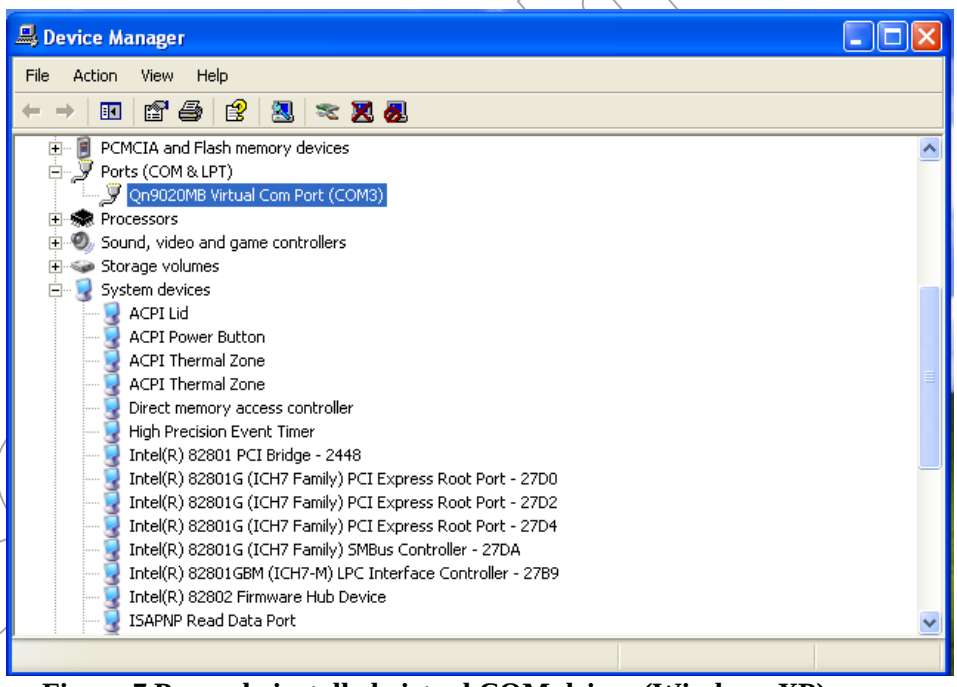

Figure 7 Properly installed virtual COM driver (Windows XP)

3. Working Mode

Quintic's QBLUE provides a flexible platform for wireless applications, by supporting two working modes: Wireless SoC Mode and Network Processor Mode.

3.1 Wireless SoC Mode

In the Wireless SoC Mode the link layer, host protocol, profiles and application all run on the QBLUE as a single chip solution. This is the work mode that the application samples are used.

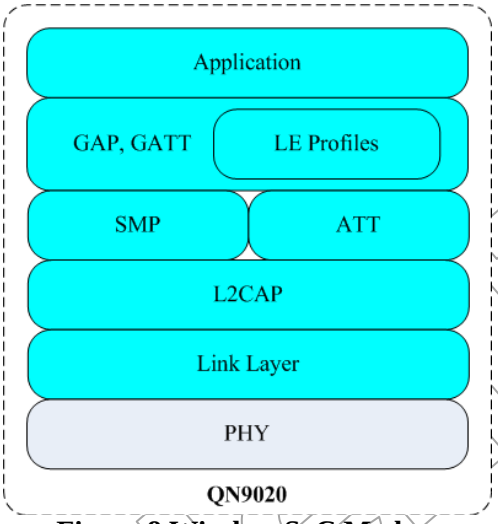

Figure 8 Wireless SoC Mode

3.2 Network Processor Mode

In the Network Processor Mode the link layer, host protocols and profiles run on the QN9020, and the application runs on the external microcontroller or PC. These two components communicate via ACI (Application Control Interface), which are provided on QN9020.

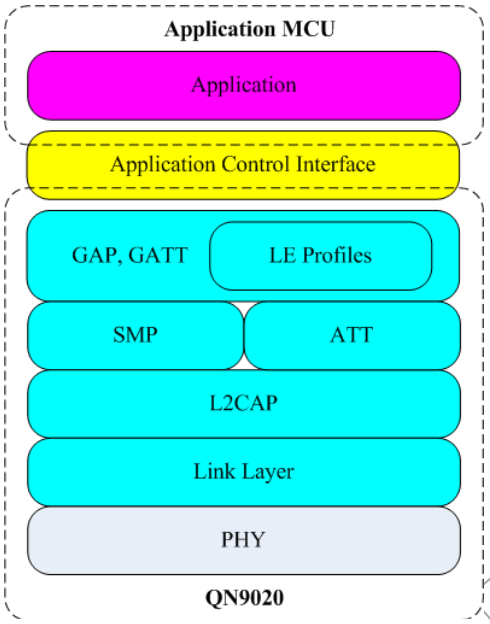

Figure 9 Network Processor Mode

QBLUE also provides EACI (Easy ACI, Easy Application Controller Interface) application. There are two Hardware units in this solution, one is the MCU which acts as Host role and the other is QN9020 which acts as controller role. The hardware interface between the host unit and the controller unit is by UART. The Easy ACI packets will communicate through the UART hardware interface as shown in Figure 10. You can reference the document "QN9020 Easy ACI Programming Guide" for detailed information of EACI.

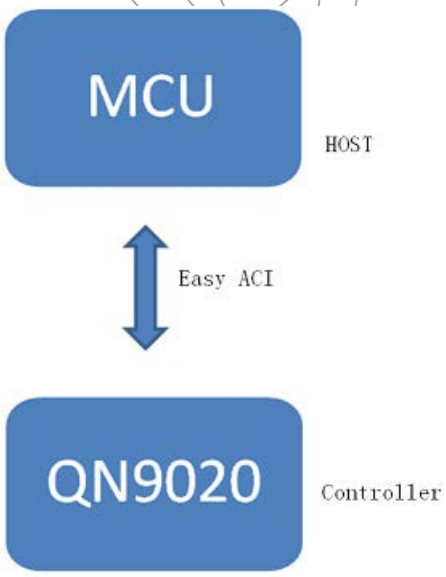

Figure 10 Easy ACI Interface

4. Start with Proximity Demo

QN902x DK is shipped with pre-loaded Proximity profile. The Proximity demo enables an interactive connection with iPhone or iPAD using an iOS App. In this case, QN902x DK works as an accessory of iPhone or iPAD.

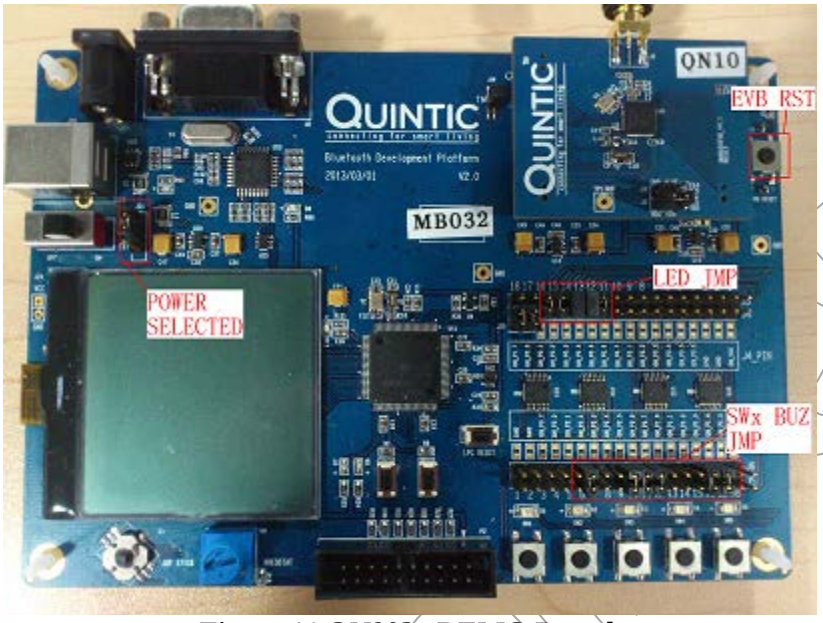

Figure 11 QN902x DEMO Board

4.1 Install APP

Any third party App for proximity can be used to perform the test. Alternately Quintic's proximity app can also be used as well. The App is available at the app store by name of "Find Easy".

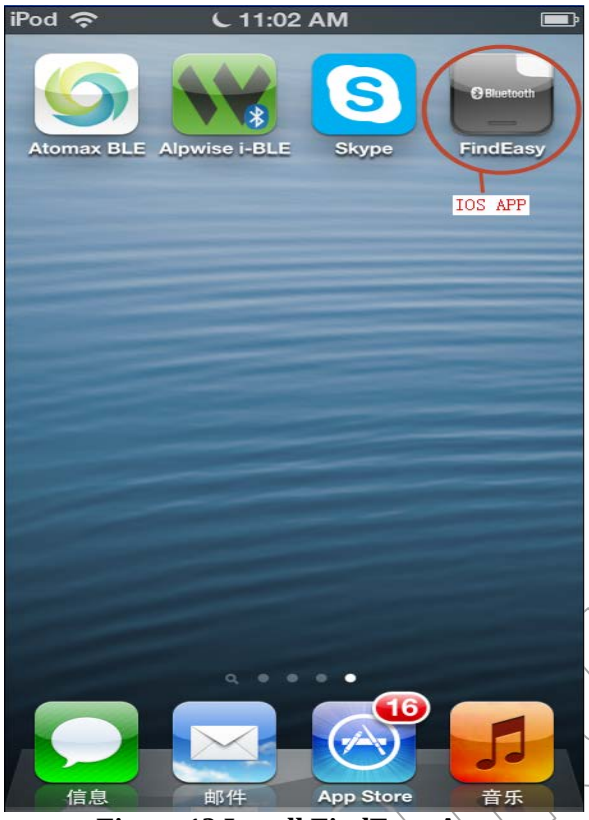

Figure 12 Install FindEasy App

4.2 Power Supply

Jumper J11 above the LCD screen can be used to select the power source. Connect appropriate jumper if using USB cable to power the board. External power sources like 3 AAA batteries or external power supply to the DC jack can also be used to power the board.

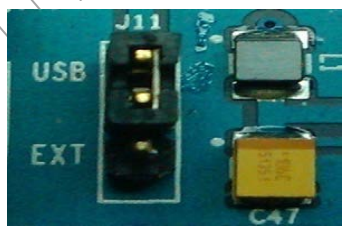

Figure 13 Power Supply

4.3 LED Jumper

Find J3 and J4 port as shown below and connect J3-12 to J4-12, J3-14 to J4-14, J3-15 to J4-15, J3-16 to J4-16, for LED by jumpers.

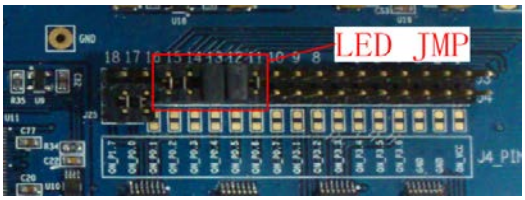

Figure 14 LED Jumpers

4.4 Buzz Jumper

Connect J5-10 and J6-10 is for Buzzer jumper as shown below.

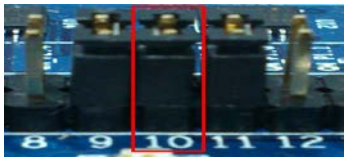

Figure 15 BUZZ Jumper

4.5 Button Jumper

Find J5 and J6 port as shown below and connect J5-16 to J6-16, J5-17 to J6-17, for Button by jumpers.

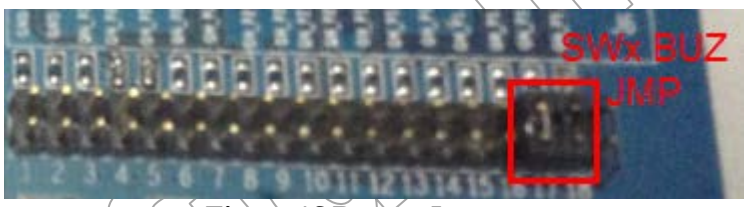

Figure 16 Button Jumpers

4.6 Launch

When power on or EVB reset, the application is in the deep sleep status.

Upon power ON, the EVB is in idle mode, similarly after reset the EVB also enters idle mode.

Once the EVB is ON and in idle mode the LED D2 turns ON, pressing the SW4 will take the EVB into Advertising mode. LEDs D1,D3,D4 and D5 will start to flicker. The flashing frequency depends on advertising interval. In the first 30 seconds the application uses a short advertising interval and then uses a long advertising interval. After the 30sec LED D3 will be solid and D1, D4 and D5 will flash depending on the advertising interval which is around 1sec. If SW4 is pressed again, the application stops advertising and D1, D4 and D5 will stop flashing, D3 will be solid, which means the application is back to deep sleep mode.

When the EVB is advertising, Iphone App can be used to test the functionality. If the Quintic's App is being used, following is a brief intro.

Run the application on the Iphone and click on “scan”, the App will find BLE devices nearby. If “QN PROXR” appears in the window of the App, this shows that you have found the Quintic device. Click the “connect” icon, a green “In Range” icon will appear, when the “connect” icon changes into “Alert”, it means that the connection is complete, as shown below:

Figure 17 Launch

Following are the EVB LED Indicators:

LED1 ON indicates that the application is connected.
LED2 ON indicates QN902x is in the DEEP SLEEP mode.
LED3 ON indicates QN902x is in the SLEEP mode.
LED4 ON indicates QN902x is in the CPU CLOCK OFF mode.
LED5 ON indicates QN902x is in the ACTIVE mode.

Proximity alerts are indicated using the Buzzer, the volume buzz indicates the type of alert.

4.7 Function Operation

When connected, we may do the following interactions between iPhone and device (QN9020 demo) as shown below.

Figure 18 Function Menu

4.7.1 Find me

The App finds the device: Click “Alert” icon on the App, if the buzzer is ON, its notifying the proximity of the device.

4.7.2 Anti-loss

According to the configuration on iPhone App, when the distance of iPhone and device is out of range, device will respond by turning ON the buzzer.

Figure 19 Anti-Loss Menu

4.7.3 Disconnect

1. If the “disconnect” icon is selected, the device will disconnected by no response.
2. If distance is too far for a connection to be retained and the link is disconnected. The device will respond with a buzz.

4.7.4 Others

1. When buzzer is ON, pressing Button 2 once, will turn off the buzzer .

5. Application Development

Here we give you a step-by-step walkthrough of the Proximity Demo to describe how to start with your own application development using Keil MDK-ARM.

5.1 Open a project

Please choose a project file in our SDK sample code as shown in Figure 20, and open it.

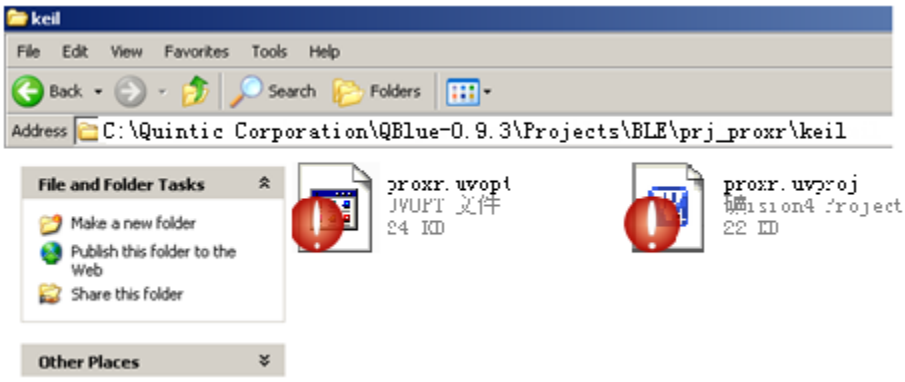

Figure 20 Open a project

5.2 Compiling

Please press the compiling icon which is marked red as shown in Figure 21, and wait until the compiling is finished.

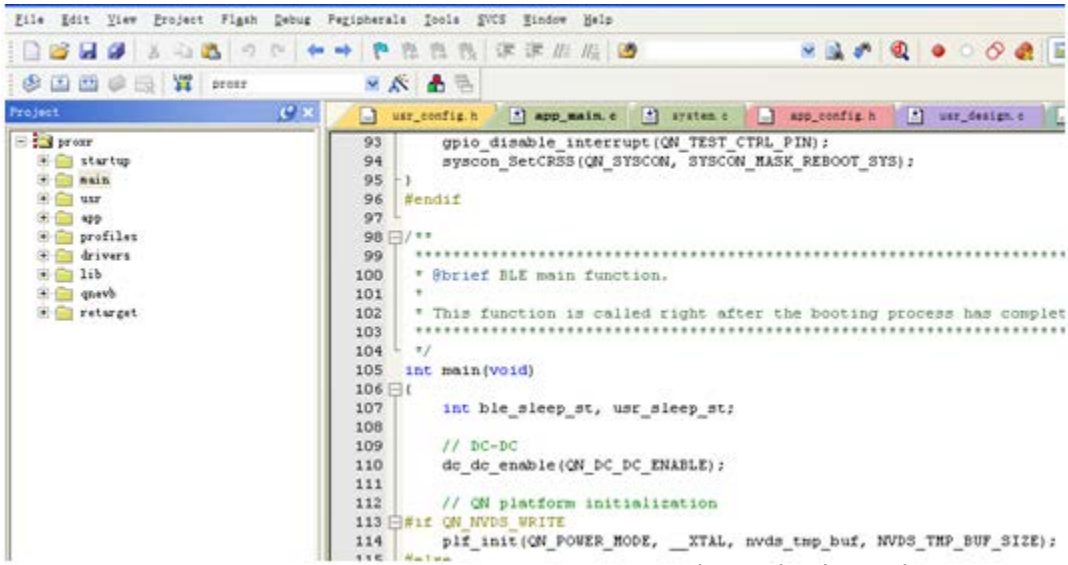

Figure 21 Compiling in Keil

When you see " '.\obj\proxr.axf ' = 0 Error(s), 0 Warning(s)", it means the compiling has finished. Then you can download .bin file which is named as "proxr.bin". By default, the file is located at "C:\Quintic Corporation\QBlue-x.x.x\Projects\BLE\prj_proxr\keil\bin " , as shown in Figure 22.

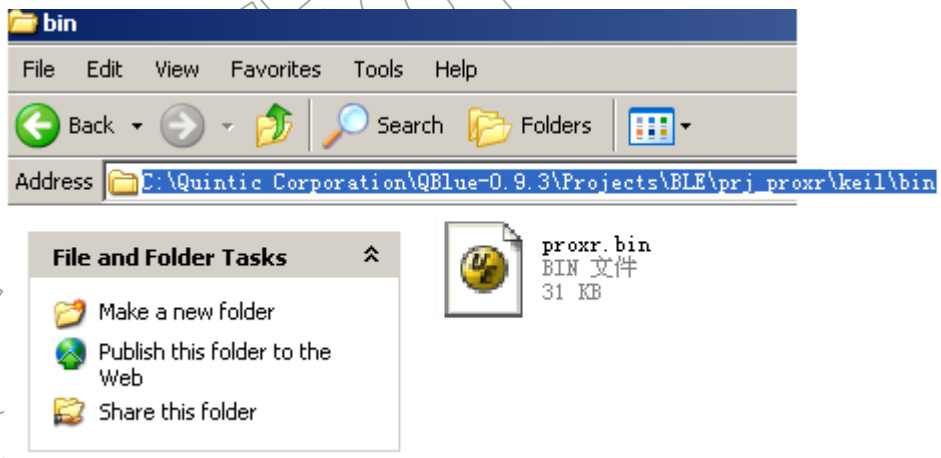

Figure 22 Location of proxr.bin File

5.3 Configure BLE Device

Please use Quintic's NVDS Configurator Tool to configure your BLE device address, name, etc. These configuration data are stored at NVDS area in the flash. The NVDS Configurator can make

it easier to add, edit and delete the configuration data. The tool is also used to burn the configuration data to target chip, or dump it from the chip. Whenever you click the button "Connect" to make a connection, please press the reset button of the target chip on DK. This is to get into the boot mode. Otherwise the connection will not succeed. For more information on this tool, please refer to "NVDS Configurator Manual" for details.

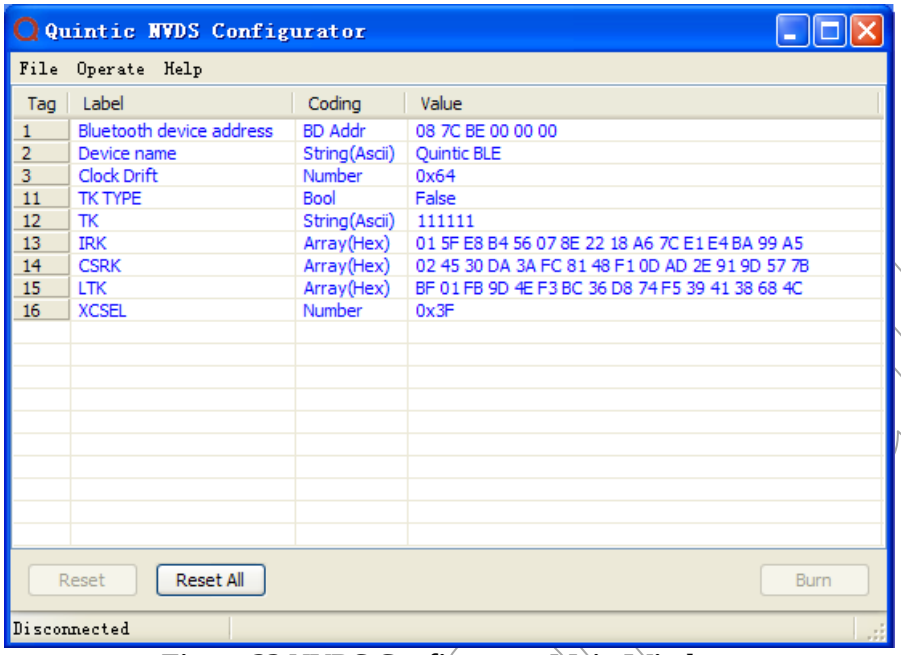

Figure 23 NVDS Configurator Main Window

5.4 Download File

Double click "QnISPStudio.exe", and select the UART interface and corresponding COM port.

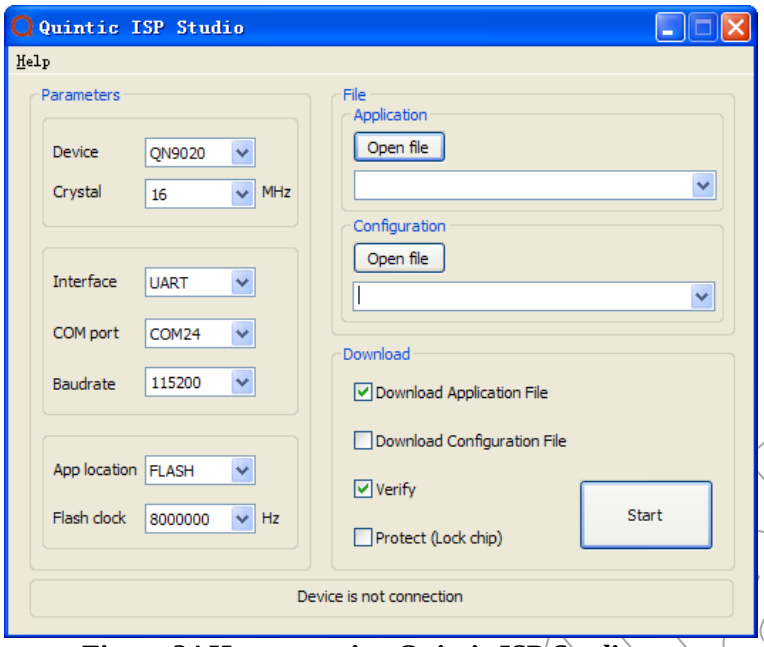

Figure 24 How to setting Quintic ISP Studio

As shown in Figure 24, use “Open file” to locate the bin file and choose “proxr.bin”. The path of proxr.bin is shown in Figure 22. Once the settings for the interface and port are as mentioned above, please click the “Start” button.

Now please press the “EVB RESET” button on the motherboard, the .bin file will be downloaded to EVB automatically. If the download is successful, a window will prompt up as shown in Figure 25 below.

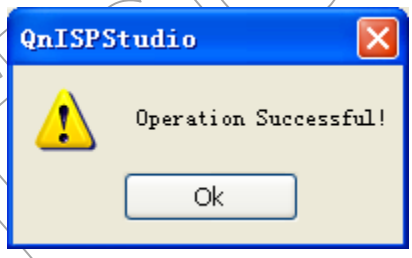

Figure 25 Download Successfully

If you can’t download the bin file successfully, please check the following:

1. Whenever you click the button “Start” to download files, please press the reset button of the target chip on DK into boot mode. Otherwise the connection will not succeed.
2. Don’t select “Protect” box if you would like to debug the program code. It causes the data in flash encrypted and flash download failed when you debug by Keil later.
3. Check whether the USB cable was connected with PC and DK motherboard, and the Platform’s power was ON.
4. Make sure the com port setting is right. You can see com port setting in Figure 24 (in this figure, the com port is COM24).

For more information on the QnISPStudio tool, please refer to the “ISP_Studio_Manual” and “QBLUE_DK_Motherboard_User_Guide” for details.

5.5 Advanced Application

According to your requirement, you may need to add some peripheral function, for example, button, LED, buzzer and so on. We offer some useful tools to help you to realize it. It is very easy to realize such functions with the help of the sample codes that are provided with the SDK.

5.5.1 Add Driver File

In your project file, there might be no related driver files. For example, you want to do button control, which needs GPIO definition to do so. It is easy to add this by using the Keil development tool. By default, the directory at "C:\Quintic Corporation\QBlue-x.x.x\Projects\BLE\src", there is a folder named as "driver". All peripheral function source codes are included here. You can add the related code files to your project in Keil as shown below:

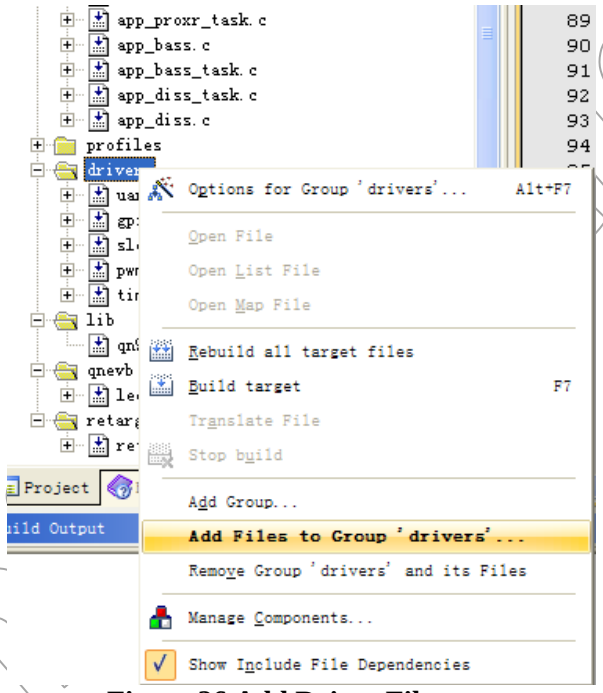

Figure 26 Add Driver File

To study how to use the peripheral function, you may open the dedicated peripheral function. By default, the directory is "C:\Quintic Corporation\QBlue-x.x.x\Projects\Driver". Open the project you need, directly download to your project board and run it. You may try to modify it for learning. For example, gpio function below.

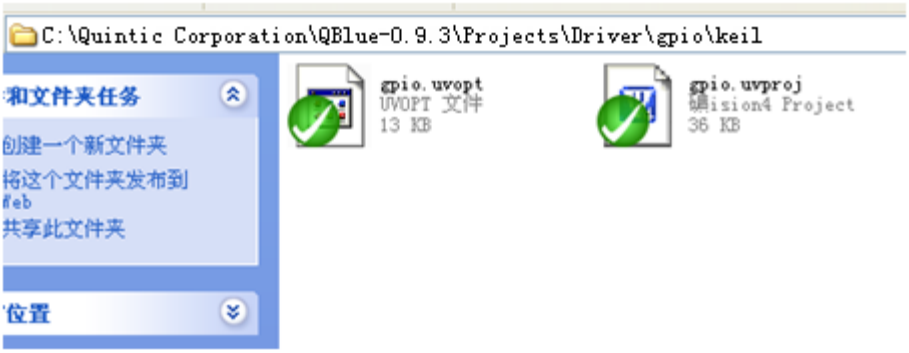

Figure 27 GPIO Dirver

Double click “gpio.uvproj”, open it using Keil as shown below.

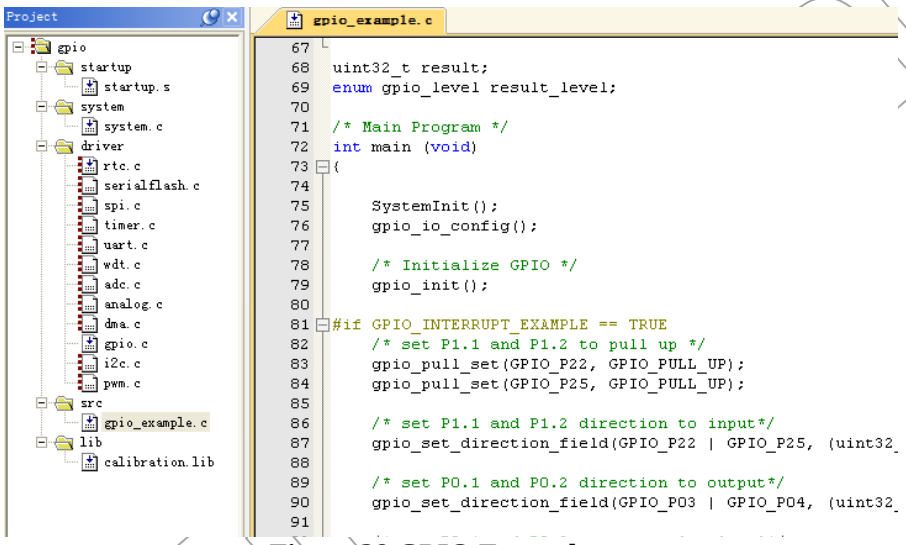

Figure 28 GPIO Example

The main function is in gpio_sample.c.

5.5.2 Special GPIO extend function

1. Change GPIO function.
   a) Open “QnDriverTools”, choose “IO” icon as shown below.

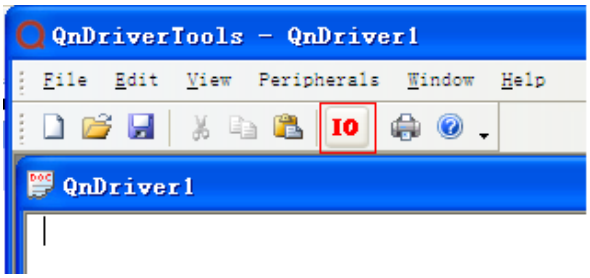

Figure 29 Open QnDriverTools and choose “IO” icon

b) Select P2.6, and set P2.6 as PWM function.

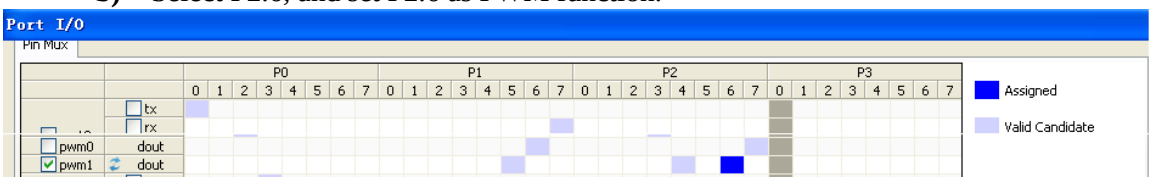

Figure 30 Set P2.6 to PWM function

c) After the setting is finished, please click "OK" button on the bottom right of the "QnDriverTools" window. The source code will be generated automatically as shown in Figure31

```
//Pin Mux Control Register
syscon_SetPMCR0(QN_SYSCON, P00_GPIO_3_PIN_CTRL
                         | P03_GPIO_2_PIN_CTRL
                         | P05_GPIO_15_PIN_CTRL
                         | P06_SW_DAT_PIN_CTRL
                         | P07_SW_CLK_PIN_CTRL
                         | P10_GPIO_8_PIN_CTRL
                         | P11_GPIO_7_PIN_CTRL
                         | P12_GPIO_6_PIN_CTRL
                         | P13_GPIO_5_PIN_CTRL
                         | P17_GPIO_4_PIN_CTRL
                         | P23_GPIO_12_PIN_CTRL
                         | P24_GPIO_11_PIN_CTRL
                         | P26_PWM1_PIN_CTRL
                         | P27_GPIO_9_PIN_CTRL
                         | P31_GPIO_14_PIN_CTRL
                         | P32_GPIO_13_PIN_CTRL);

//Pin Mux Control Register
syscon_SetPMCR1(QN_SYSCON, P01_GPIO_18_PIN_CTRL
                         | P02_GPIO_17_PIN_CTRL
```

Figure 31 Source code of adding pwm.c

d) Open "system.c" which is lacated at "C:\Quintic Corporation\QBlue-x.x.x\Projects\BLE\prj_proxr\src" by default.

Please find "P26_GPIO_10_PIN_CTRL" in function "SystemIOCfg", and replace it with "P26_PWM1_PIN_CTRL". Save this file.

2. Add PWM function to Project. Please refer to the attached "SAMPLE CODE" for the details.

If you have finished adding above code to current project, please compile this project, and download it to EVB board.

Note:

1. More information about how to enable related module, please refer to example code which are located at "C:\Quintic Corporation\QBlue-x.x.x\Projects\Driver\adc\keil" by default.

2. More information of “QnDriverTools”, please refer to “Driver_Tools_Manual” for the details.

6 Application Debugging

6.1 Configure Debugging Environment

Select “Debug” button. Please make sure the following settings are correct.

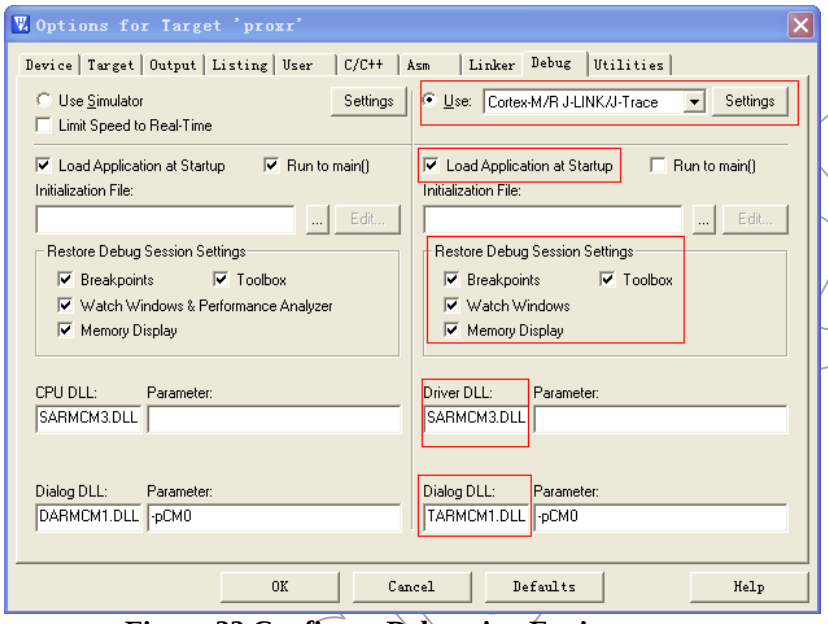

Figure 32 Configure Debugging Environment

The “Utilities” require specific settings, please take Figure33 as reference.

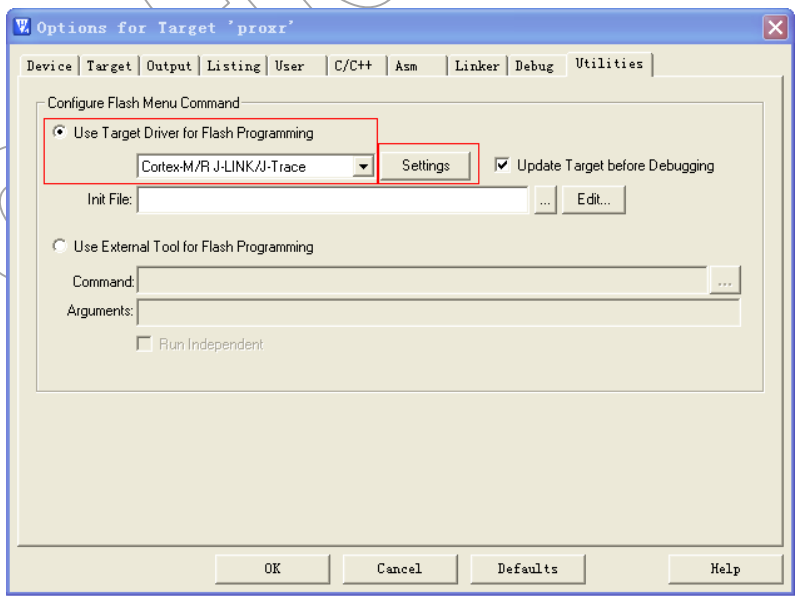

Figure 33 Utilities setting

Press “Settings” button, and then select “debug”, please set Port type to SW. Please look at Figure 34.

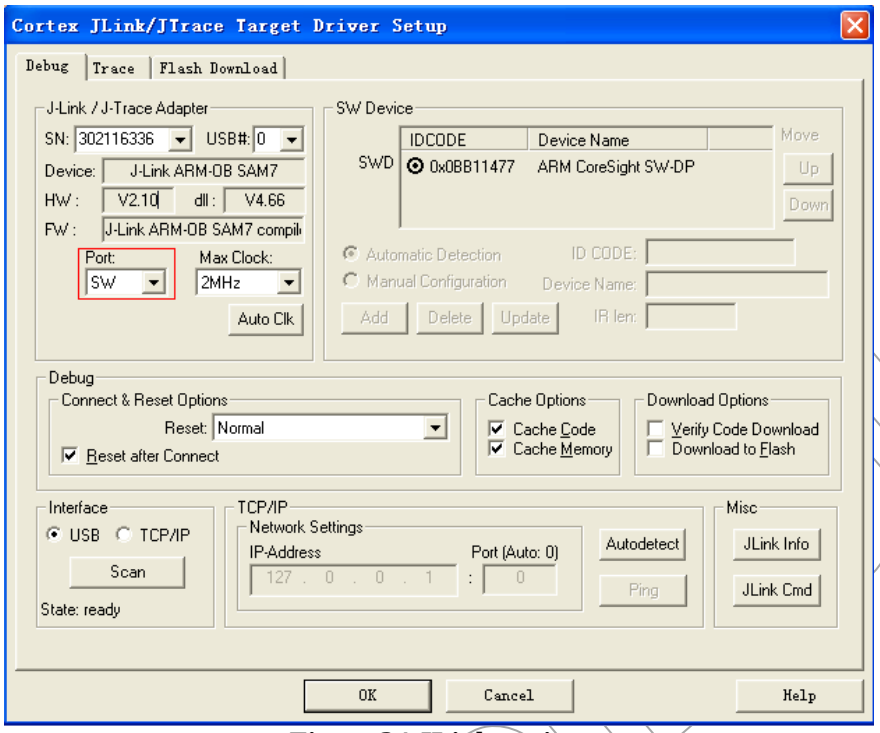

Figure 34 JLink setting

6.2 Start Debugging

Press “Start/Stop” or “Ctrl” + “F5” to start or stop debugging.

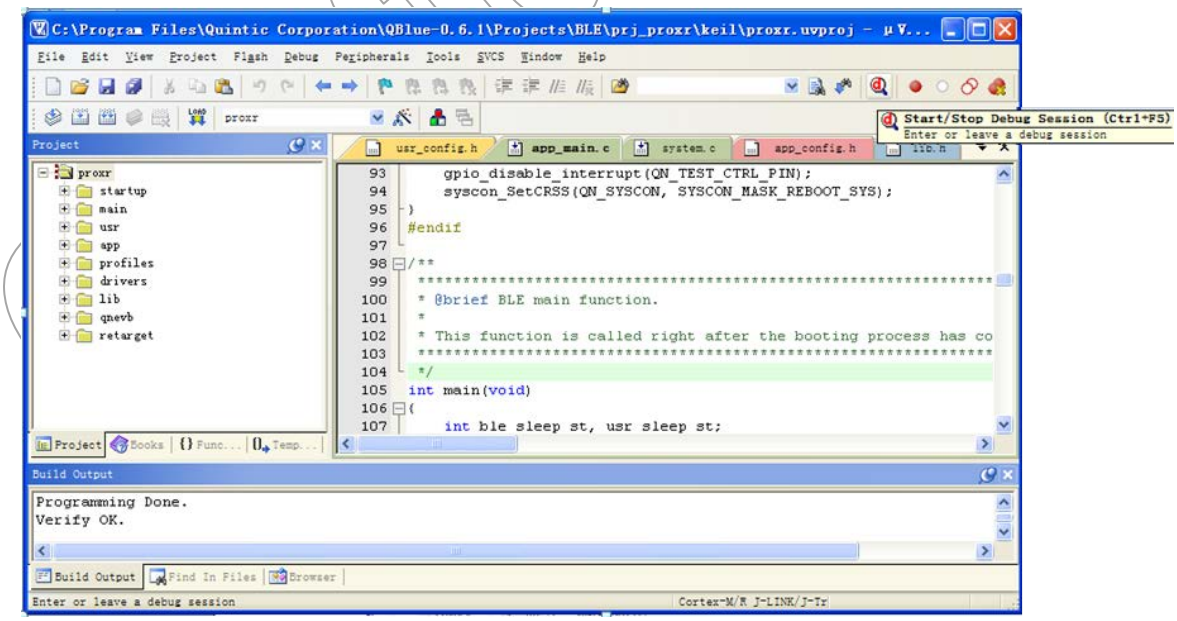

Figure 35 Start Debugging

6.3 Set Breakpoint

Once in debugging mode, you can add breakpoints to project. For example, if you add a breakpoint at function “SystemInit” firstly, and press “Run”, then the code will stop at “SystemInit”.

More information please refer to http://www.keil.com/uvision/debug.asp or choose “Help”.

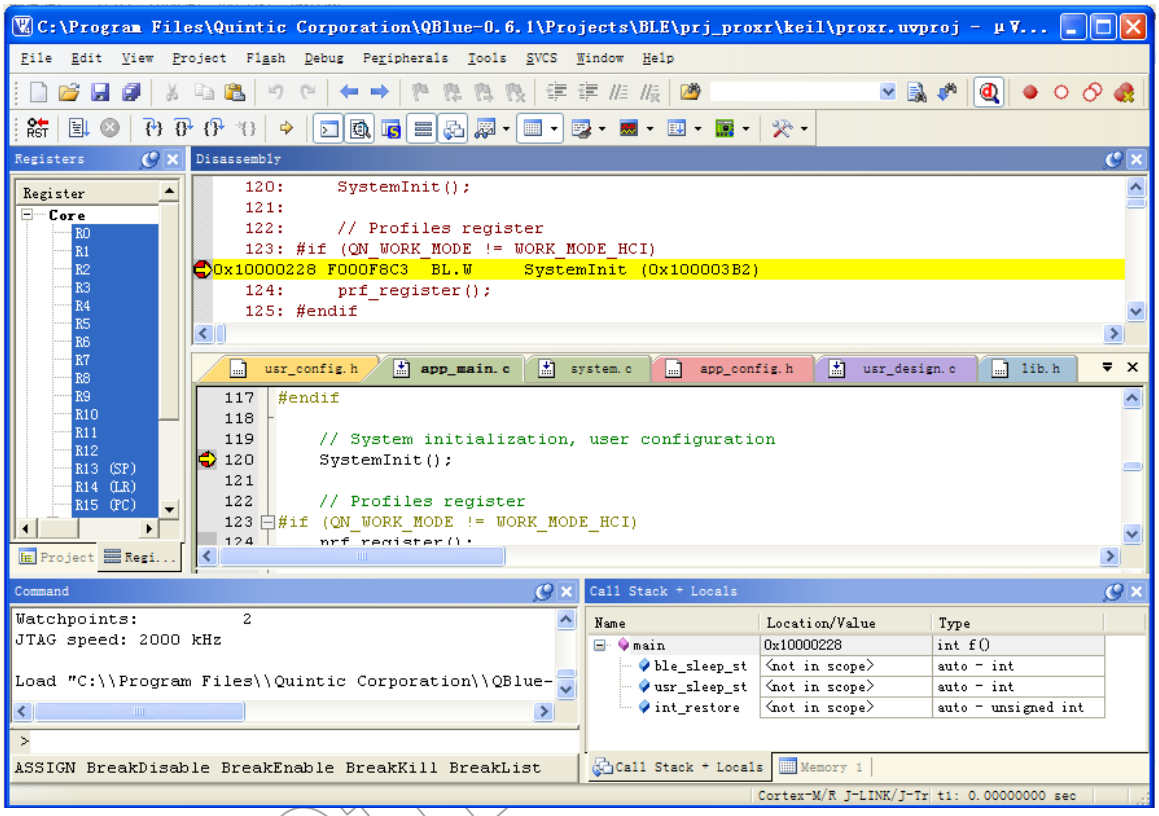

Figure 36 Set Breakpoint

Release History

REVISION	CHANGE DESCRIPTION	DATE
0.1	Initial release	2013-04-10
0.2	Add advance application example for demo	2013-04-18
0.3	Add Proximity Demo	2013-05-17
0.4	Update Proximity demo operation	2013-05-29
0.5	Update some contents and figures	2013-10-8
0.6	Update the contents and figure in chapter 5.4	2013-10-10

