

Getting Started with uCXpresso.BLE

First of all, the uCXpresso.BLE have to run on the LPCXpresso IDE, and you need to install the LPCXpresso IDE in your system first. You can download the LPCXpresso IDE from www.lpcware.com , free license after register.

About LPCXpresso, please refer to <http://www.lpcware.com/lpcxpresso/download>

History:

2014/3/12 Add Serial Terminal & Debug

2014/3/11 First Edition

1. Upgrade Your LPCXpresso IDE

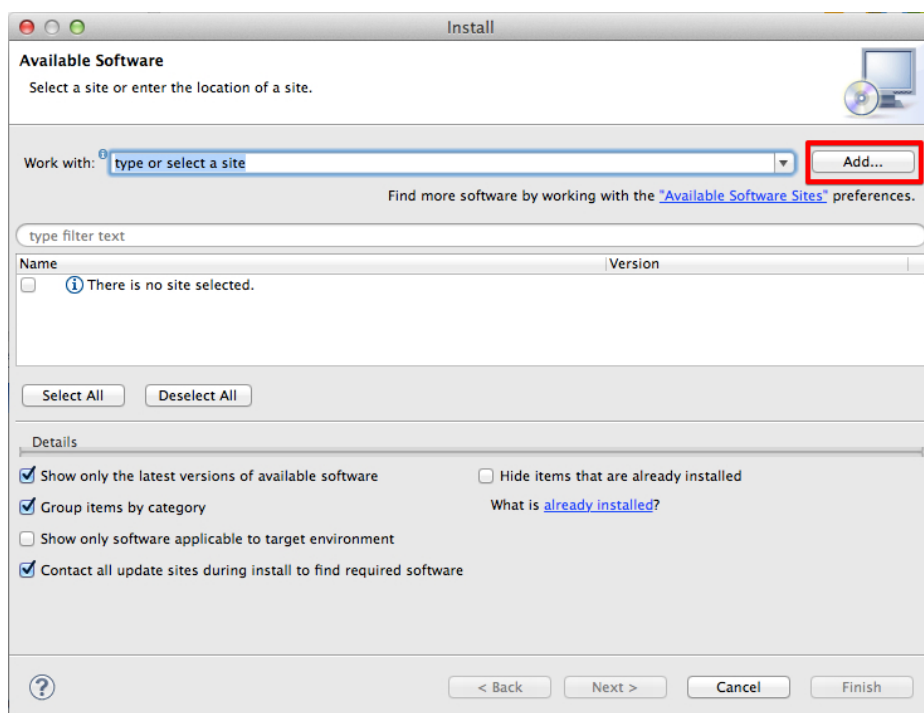
1.1. Increase VM memory for Windows System

- Close LPCXpresso if executing.
- Open `c:/nxp/LPCXpresso_x.x/lpcxpresso/lpcxpresso.ini`
- Modify MaxPermSize to 512

1.2 Install new software (Eclipse Plug-in)

LPCXpresso Main Menu > Help > Install New Software...

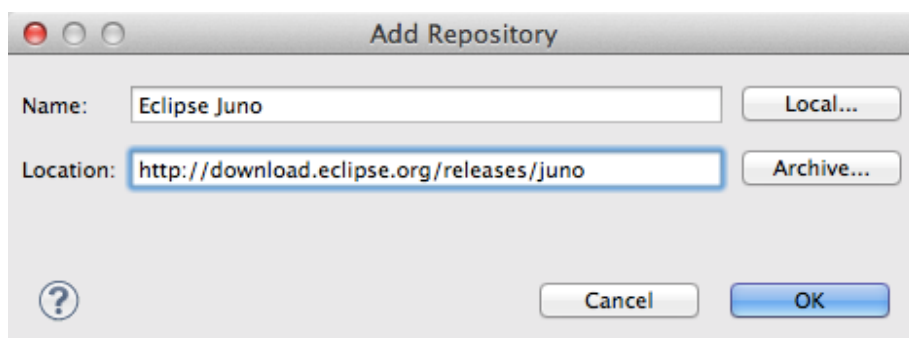
Click [Add]



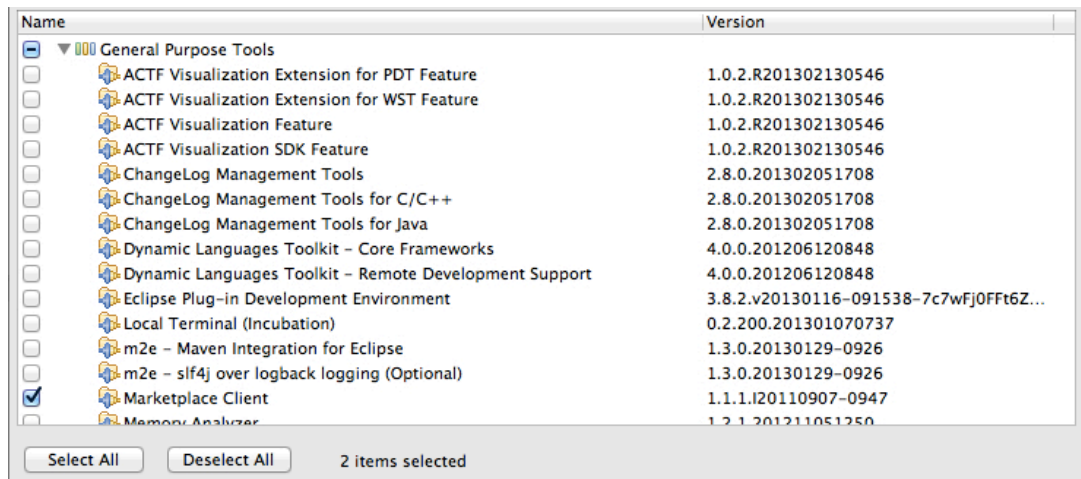
1.3 Install Eclipse Plug-In

Name: Eclipse Juno

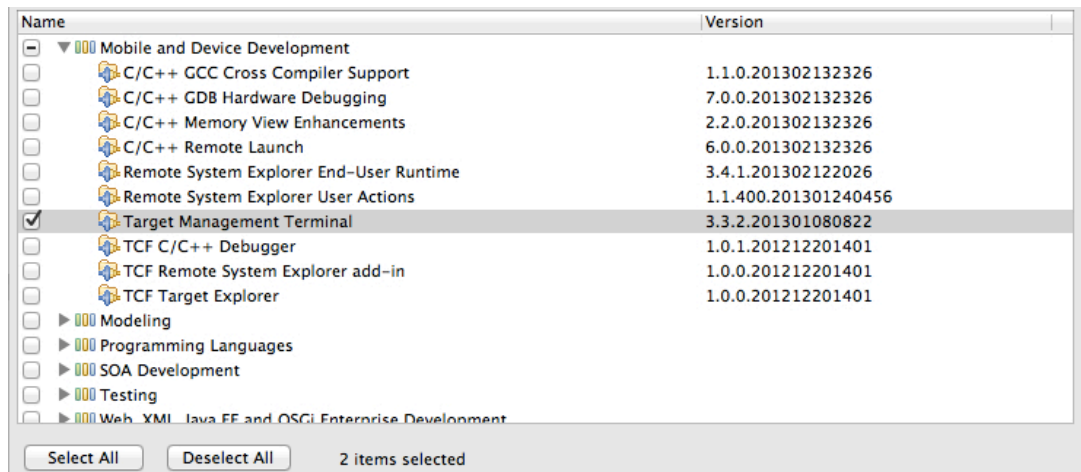
Location: <http://download.eclipse.org/releases/juno>



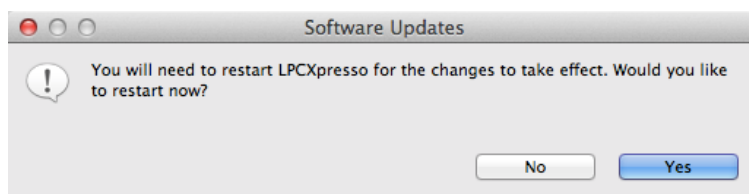
In General Purpose Tools section, select “Marketplace Client”



In Mobile and Device Development section, select “Target Management Terminal”



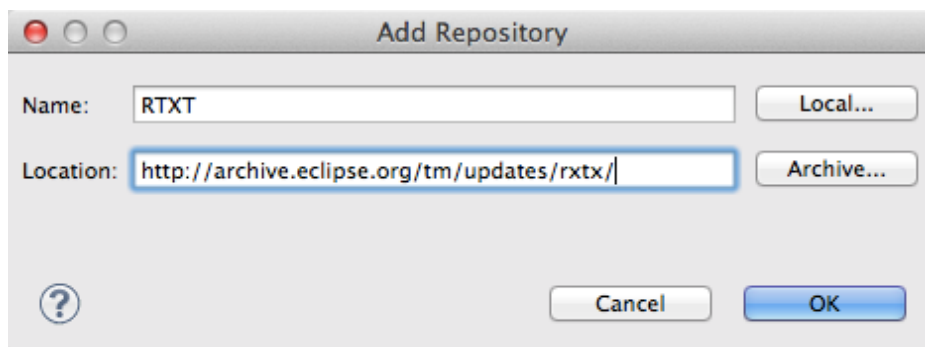
Click [Next] to start the installation. In progress, you need to [Confirm >] the Selected Features, and [accept] the licenses of software, and click [OK] to confirm any security-warning message, click [Yes] to restart and finish the installation if need.



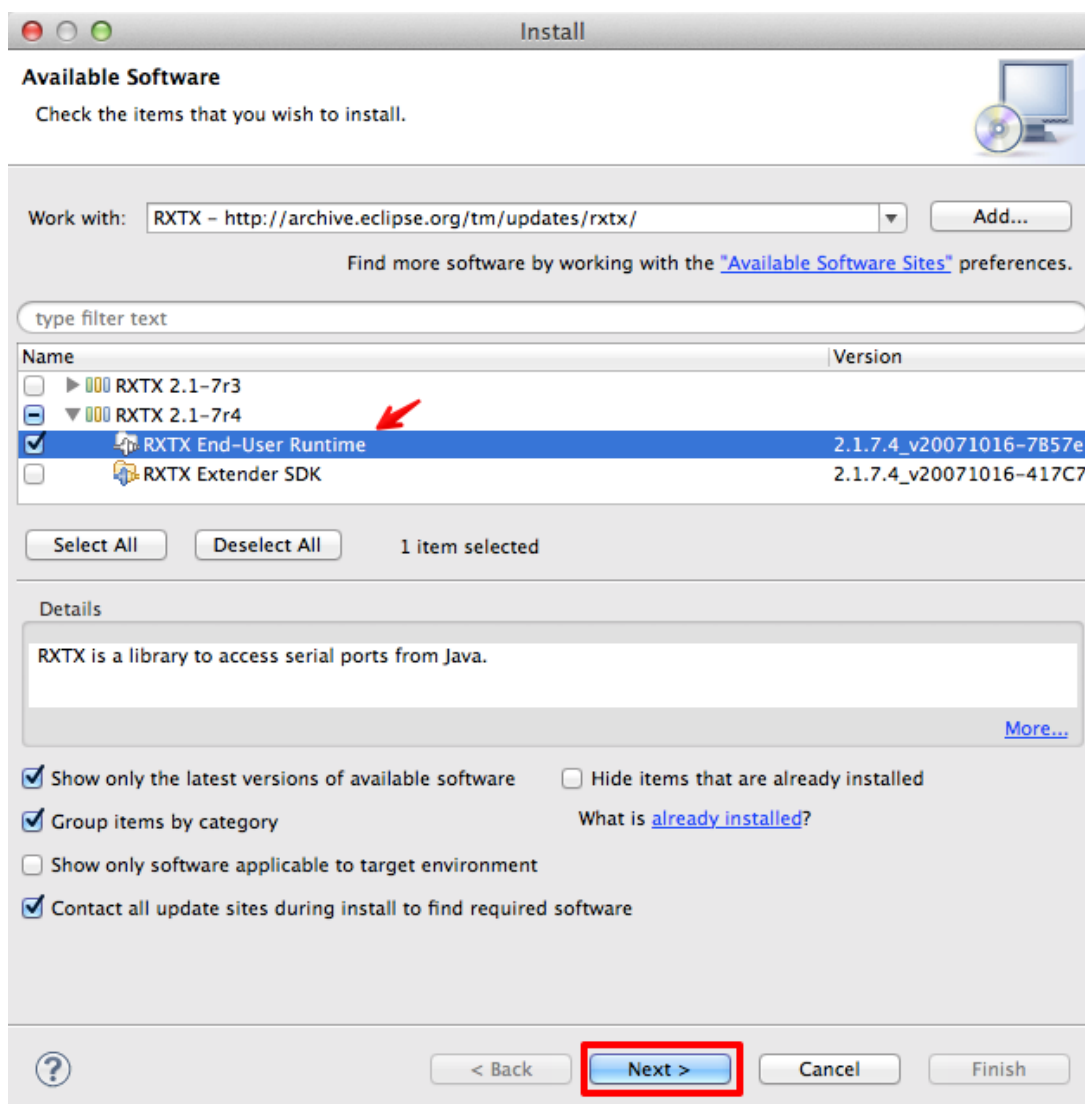
1.4. Install RXTX Plug-in

Name: RXTX

Location: <http://archive.eclipse.org/tm/updates/rxtx/>



Select latest version "RXTX End-User Runtime"



Click [Next] to start the installation.

Remark:

For Mac OS/X user, you have to follow below steps to enable the USB CDC virtual COM. Port:

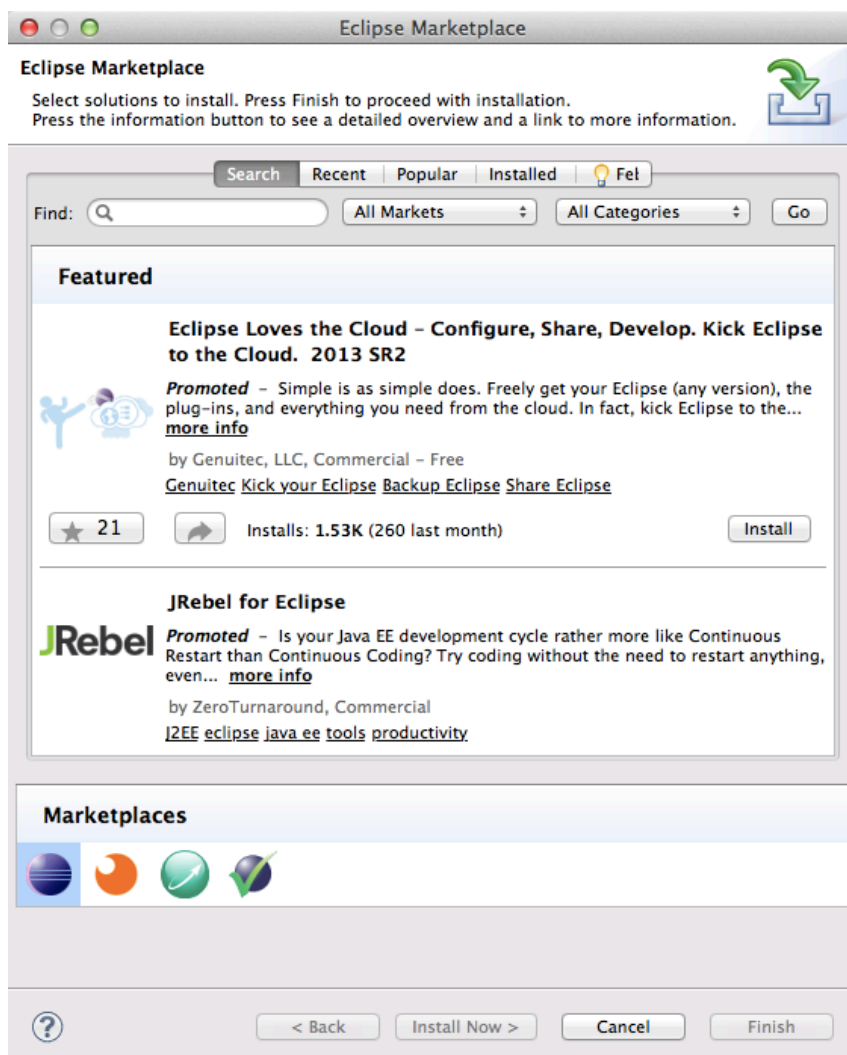
1. Open Terminal App of OS/X
2. Type :
sudo mkdir /var/lock
sudo chmod a+rw /var/lock

In Windows System, The USB CDC driver can be download from below link:

for NANO11Uxx : http://www.embeda.com.tw/tw/wp-content/uploads/2014/01/nano11Uxx_usb_driver.zip
(unzip and copy INF file to desktop, and USBCOM port driver direct to the INF file.)

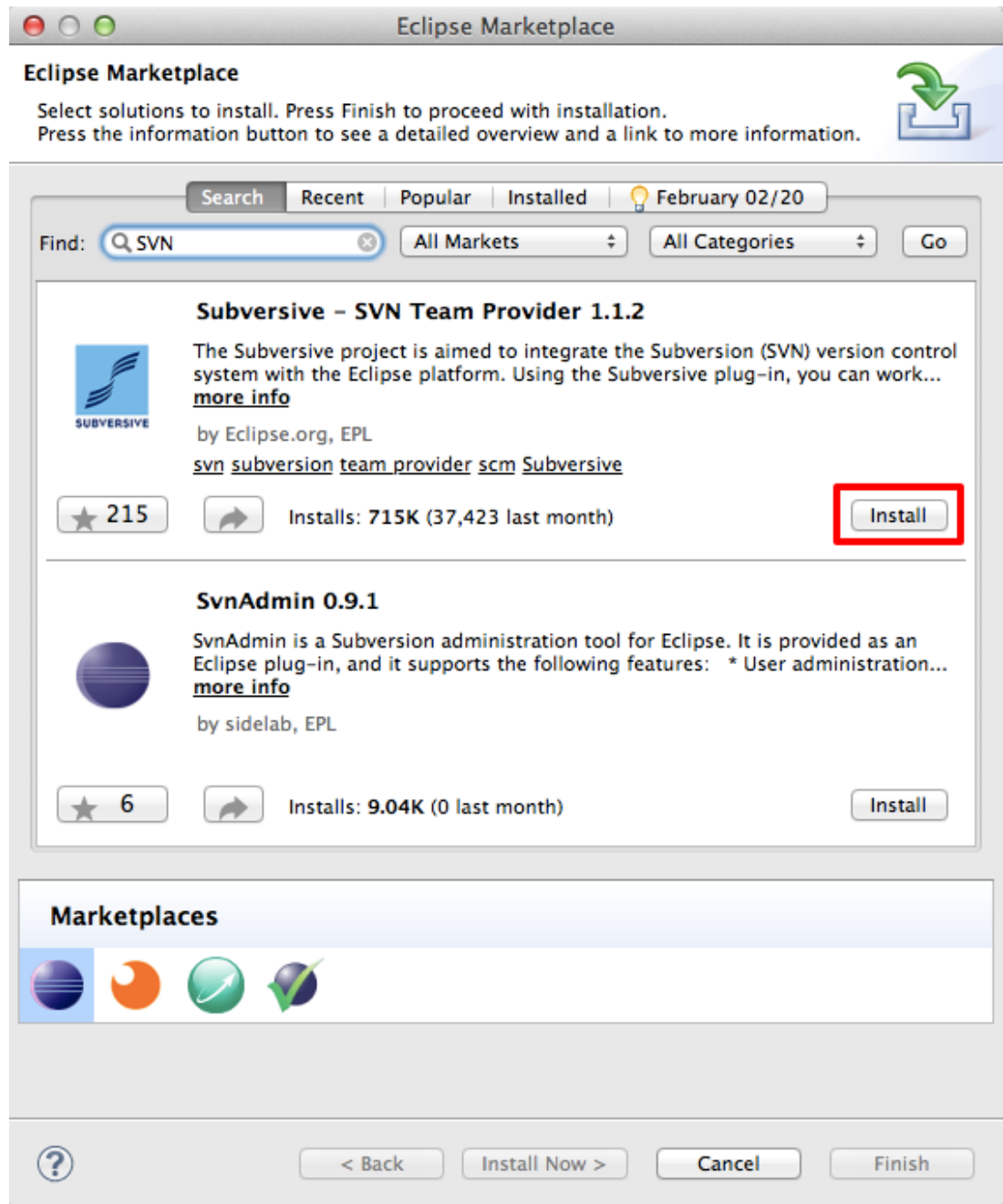
1.5. Install new software from Eclipse Marketplace

Click LPCXpresso Main Menu > Help > Marketplace...

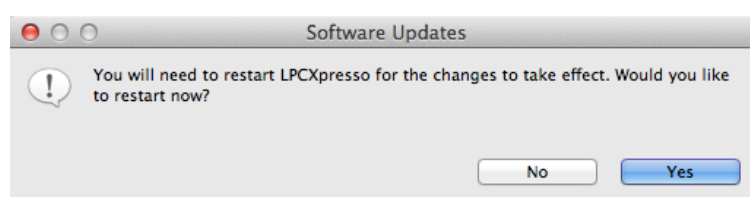


1.6 Install Subversion – SVN Team Provider (Plug-in)

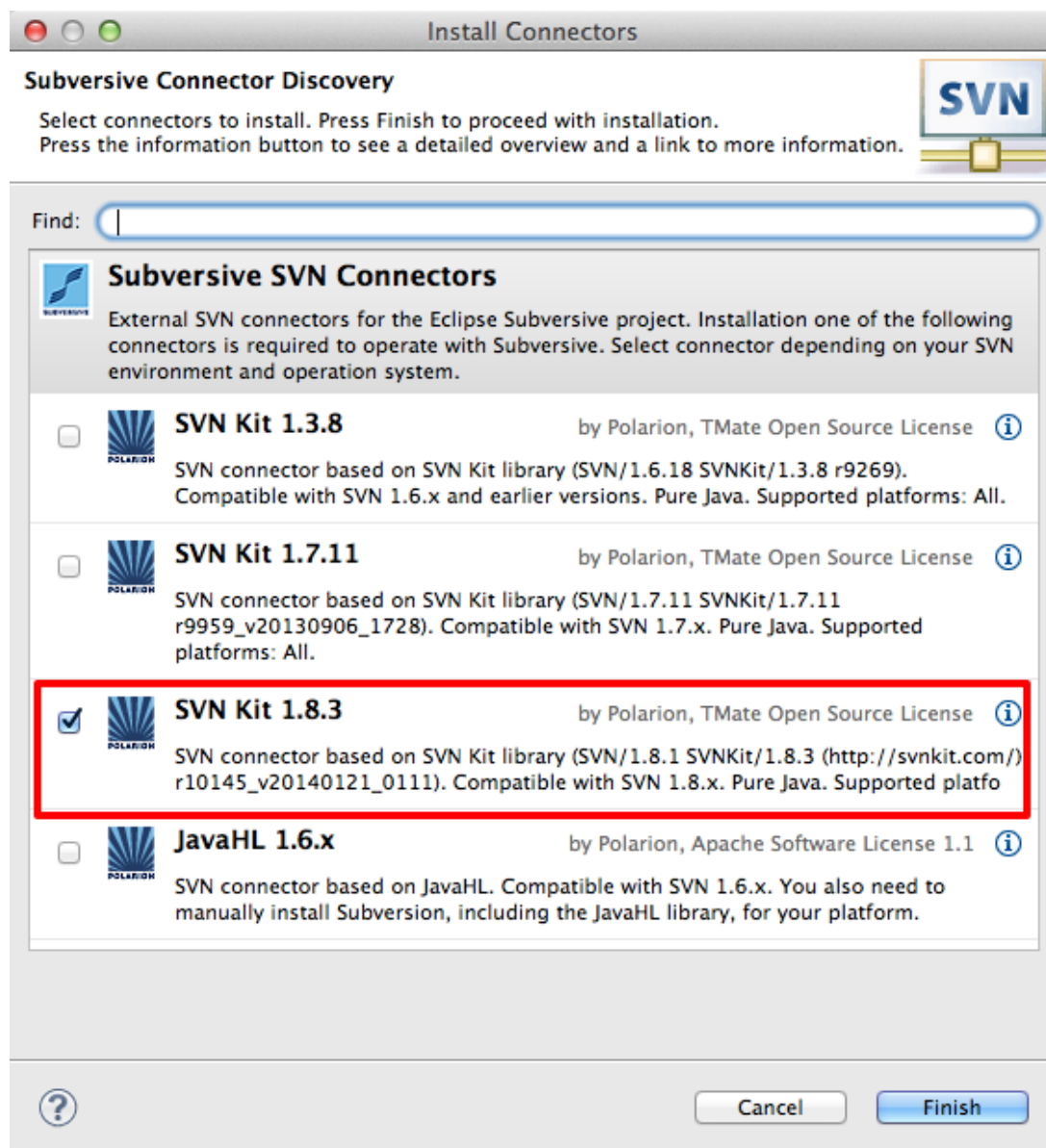
In Find, input “SVN” and click [Go]



Click [Install] to start the installation. In progress, you need to [Confirm >] the Selected Features, and [accept] the licenses of software, and click [OK] to confirm any security-warning message. Click [Yes] to restart and finish the installation if need.



After restart, you may need to install the SVN connector kit:

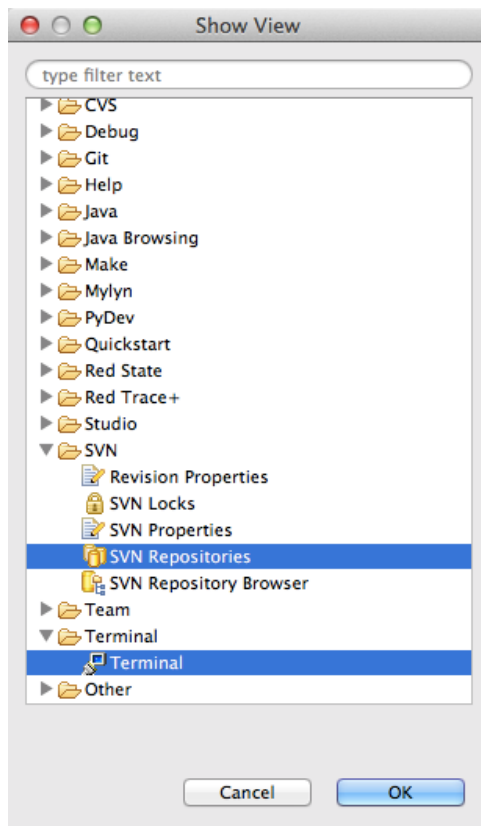


Select the latest version “SVN Kit x.x.x”, and click [Finish] to start the installation. Confirm and click [Next], and [accept] the licenses of software, and click [OK] to confirm any security-warning message. Click [Yes] to restart and finish the installation if need.



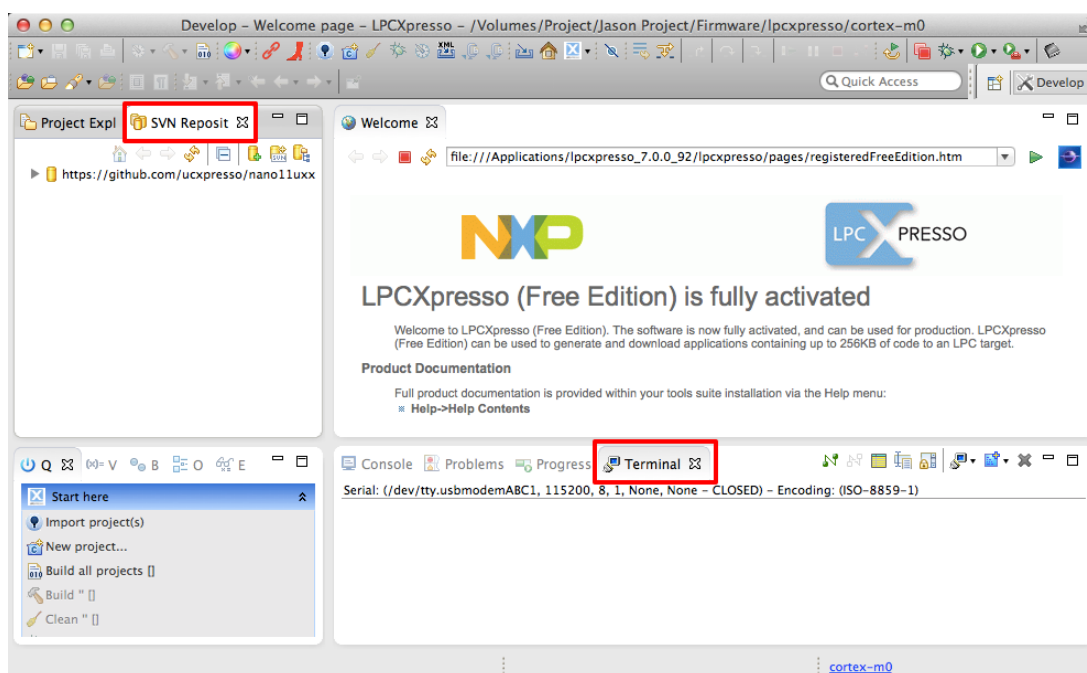
1-7. Show the SVN and Terminal Views

Click LPCXpresso Main Menu > Window > Show View > Other...



Select the “SVN Repositories” and “Terminal”, click [OK]

Move the “SVN Repositories” Tab to “Project Explorer” right side (optional)



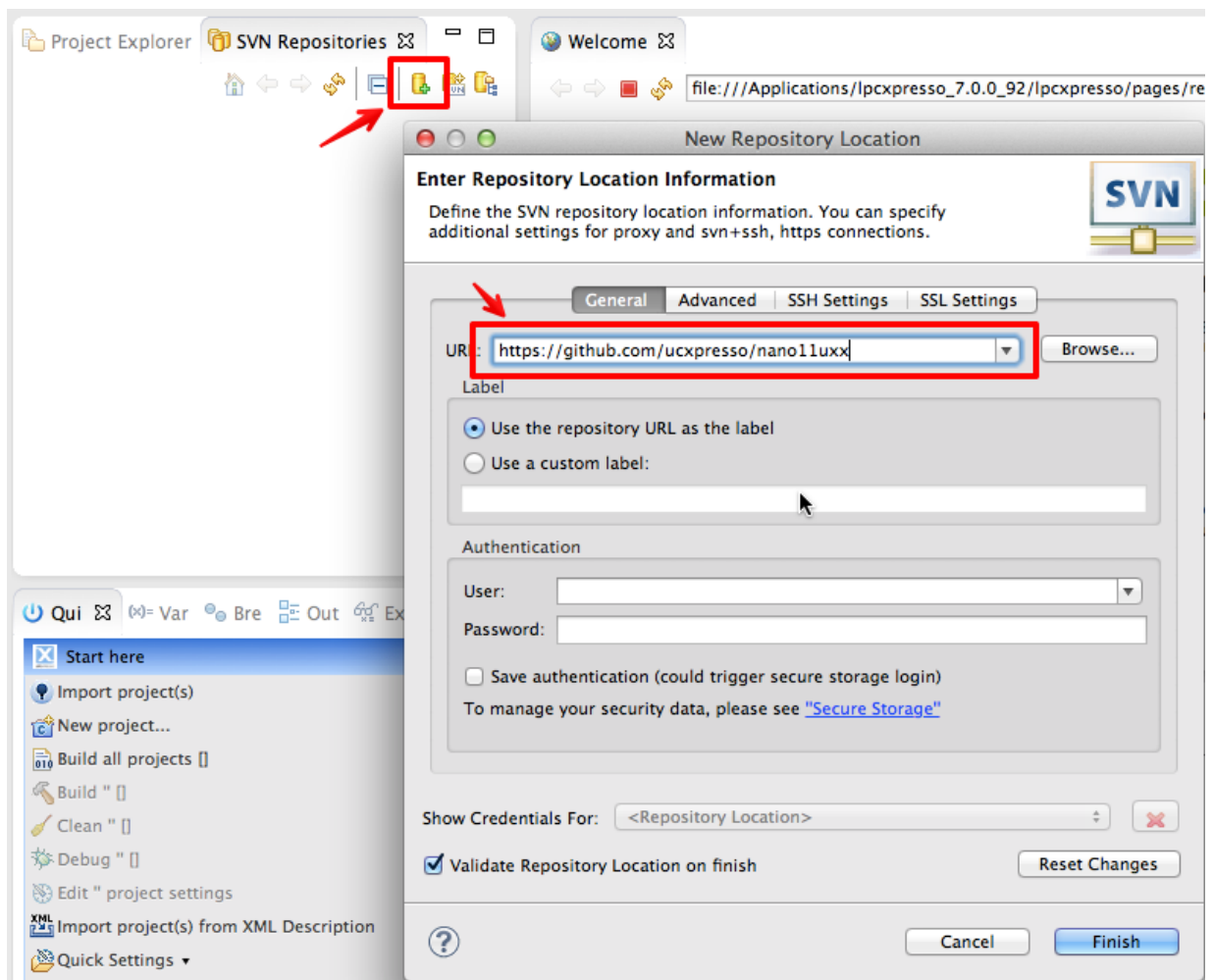
2. Install uCXpresso.BLE Framework

2.1 Using GitHub Repositories

GitHub



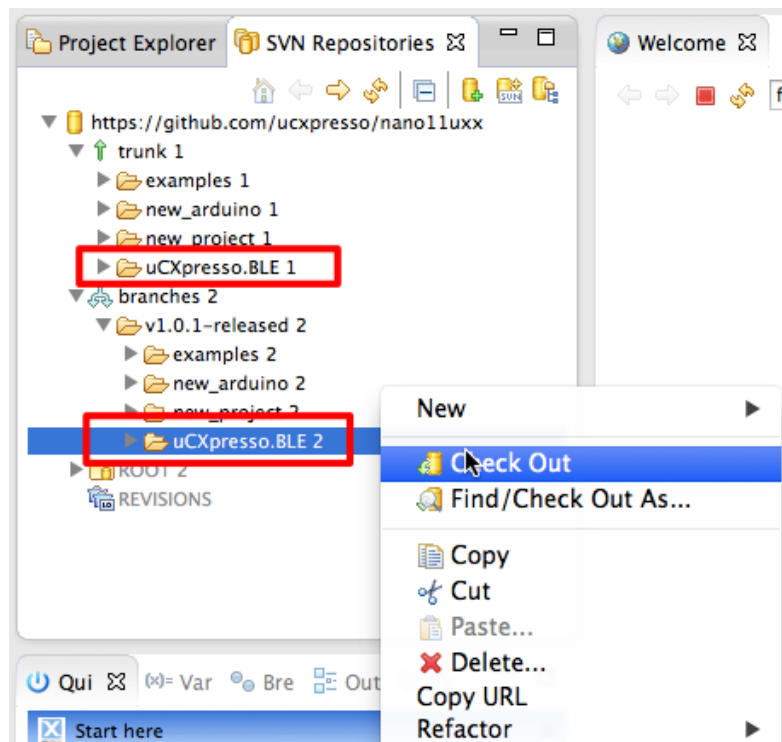
In “SVN Repositories” Tab, Click “New Repository Location”,



and URL Input: `https://github.com/ucxpresso/nano1luxx`

2.2 Check Out uCXpresso.BLE framework

Expand the uCXpresso.BLE SVN repository

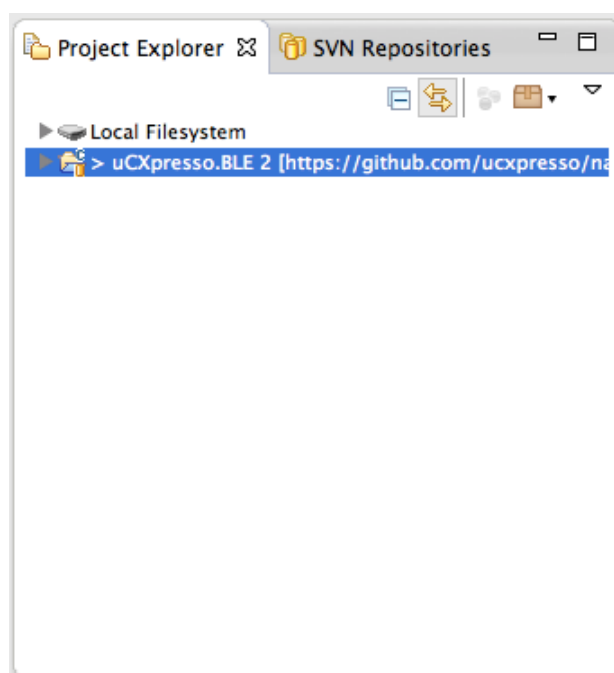


trunk: uCXpresso.BLE RC version

branches: All released version

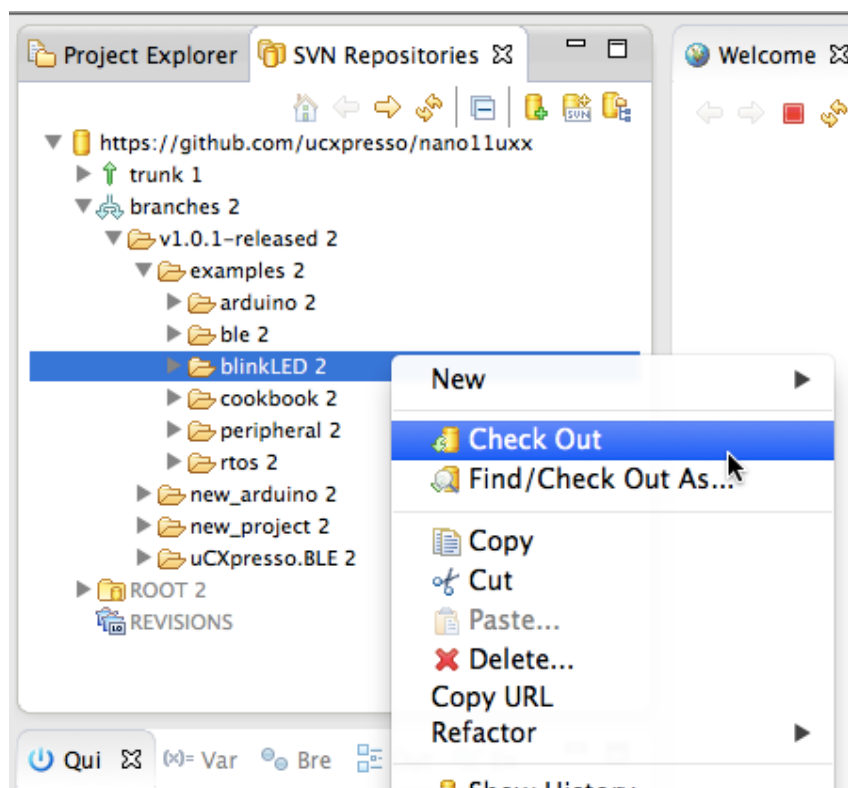
Select “uCXpresso.BLE” folder, and click right button of mouse, and click “Check Out” in drop down menu.

The uCXpresso.BLE framework show in your workspace after check out.



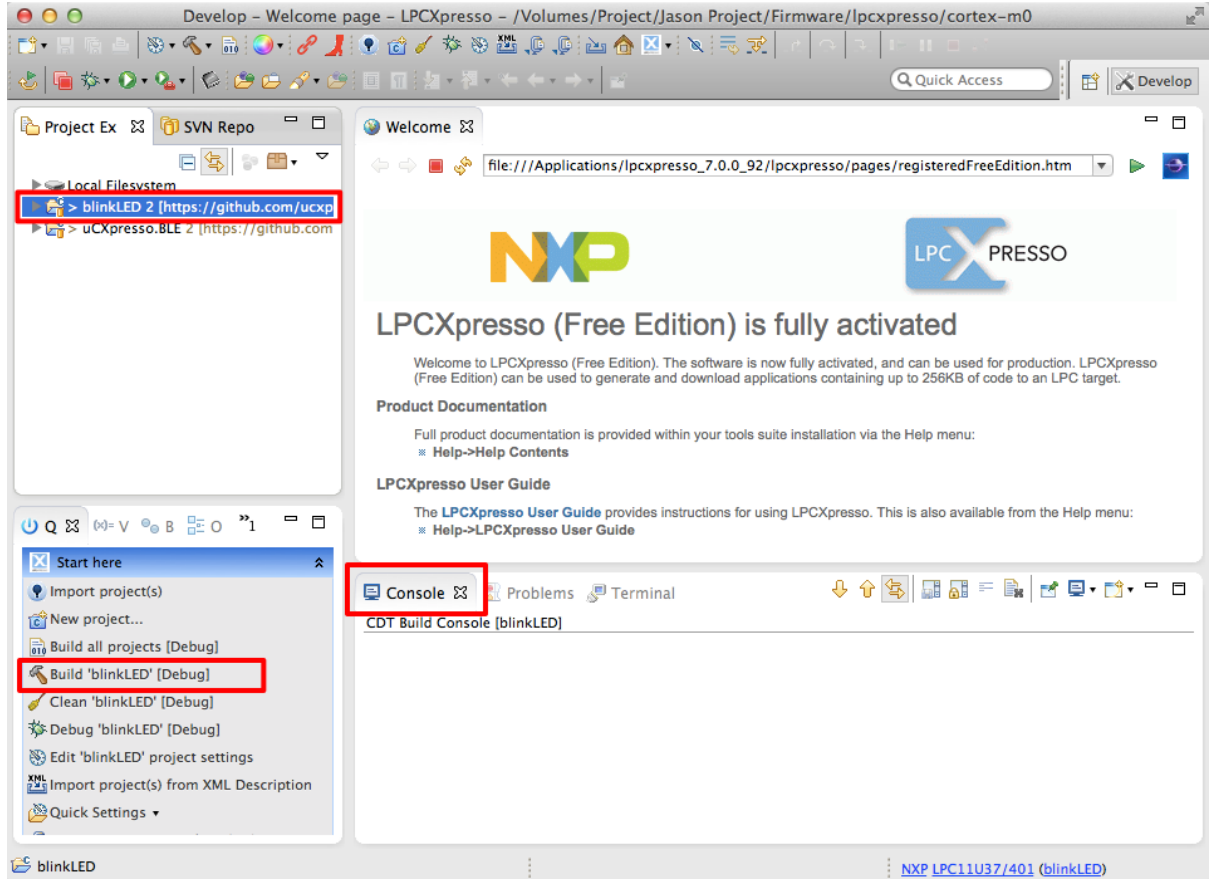
2.3 Check Out example “blinkLED”

In SVN Repositories Tab, select the “blinkLED” folder in “/examples”, then Check Out it.

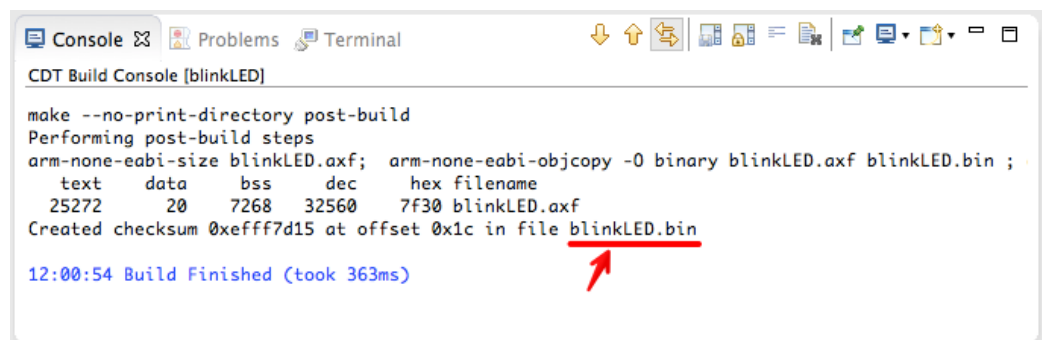


2.4 Build the “blinkLED” example project

1. Click “blinkLED” Project on Project Explorer View.
2. Click “Console” View
3. Click “Build ‘blinkLED’ [Debug] on Quick Start View.



In Console View, the compiler will create a “blinkLED.bin” image file.



Note:

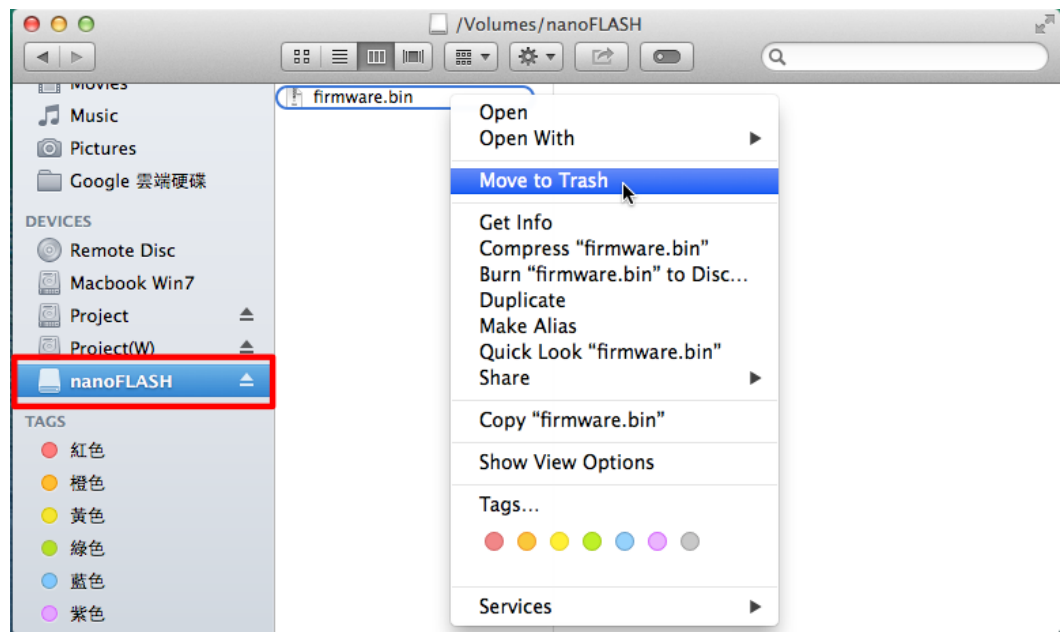
code size = text + data = 25292 bytes

ram size = data + bss = 7288 bytes

3. Download Image File

3.1 Erase Flash Code Memory

First, you need to erase the flash memory for new binary image. Click the “ISP” button on the nano11U37-BLE. In Finder (or File Manager) will creates an USB Disk and called “nanoFLASH”



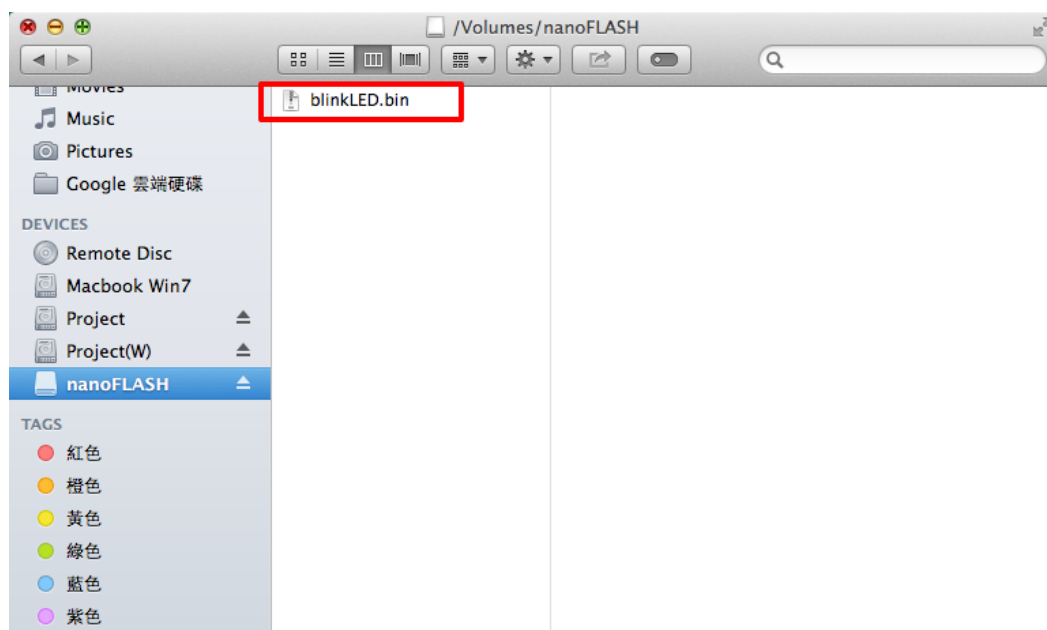
Remove the “firmware.bin” and move to trash.

Note:

In ISP (bootloader) mode: The LED1, LED2, LED3 and LED4 are blink together.

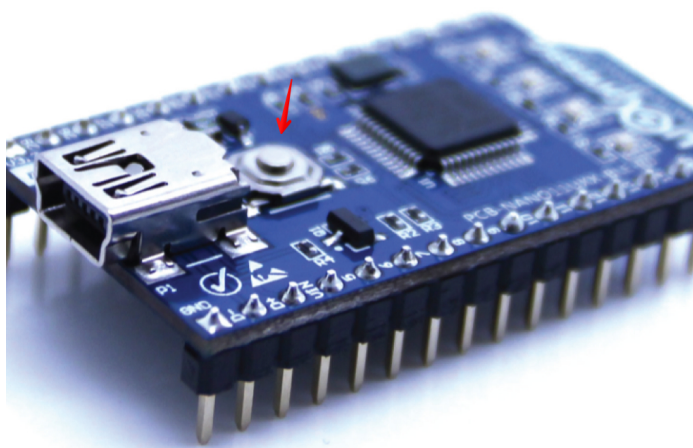
3.2 Program Flash Code Memory

Copy the “blinkLED.bin” from “blinkLED\Debug” folder to nanoFLASH



Eject the “nanoFLASH” disk and click the “ISP” button of NANO11U37-BLE again to execute your binary code. (In App Mode)

Note:

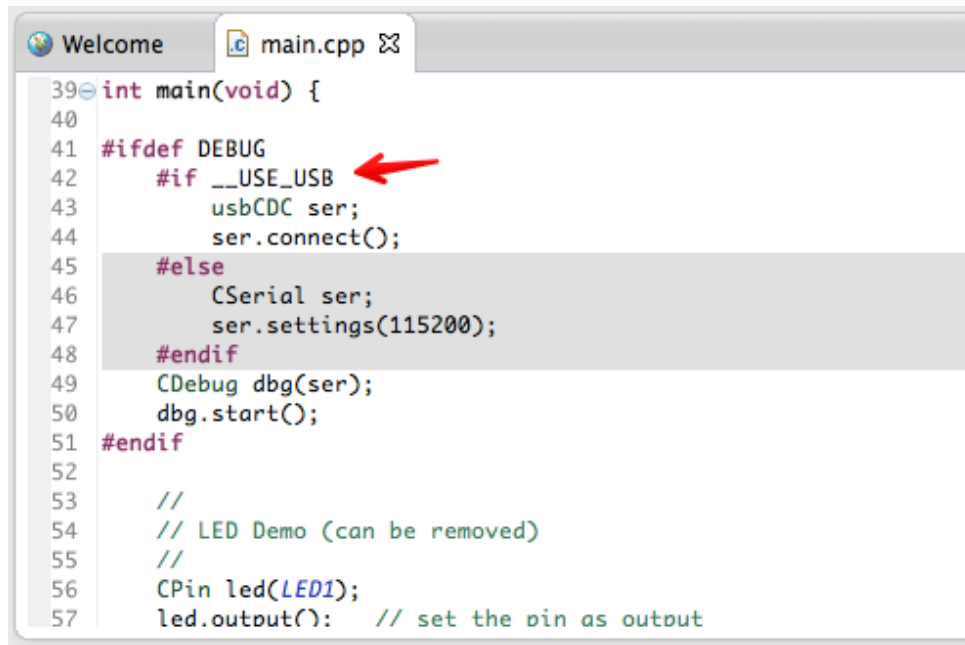


The “ISP” button.

4. Serial Terminal & Debug

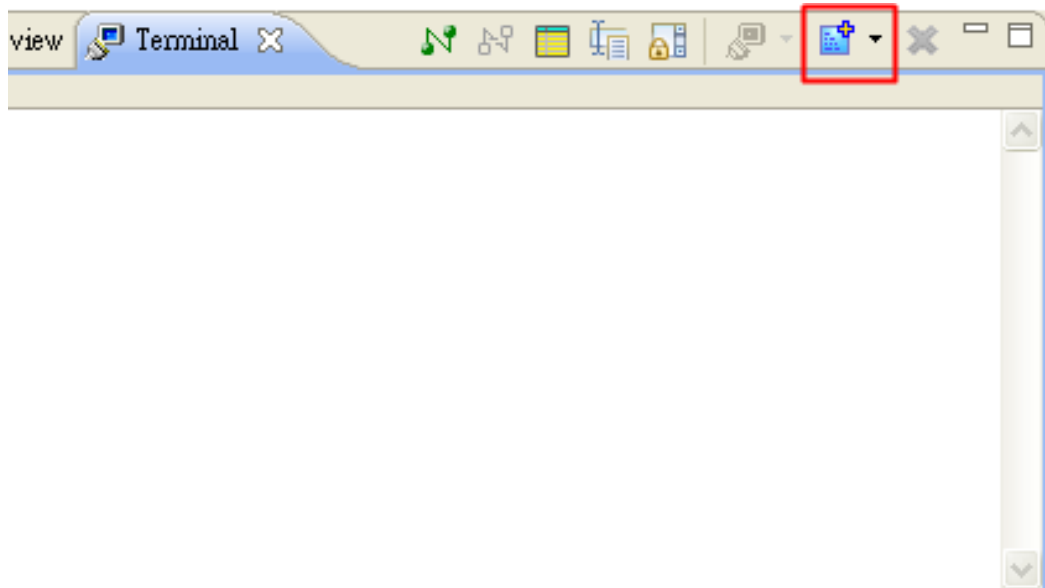
4.1 Serial Terminal (Use USB CDC Virtual COM. Port)

In main.cpp of blinkLED, the usbCDC will be enabled.



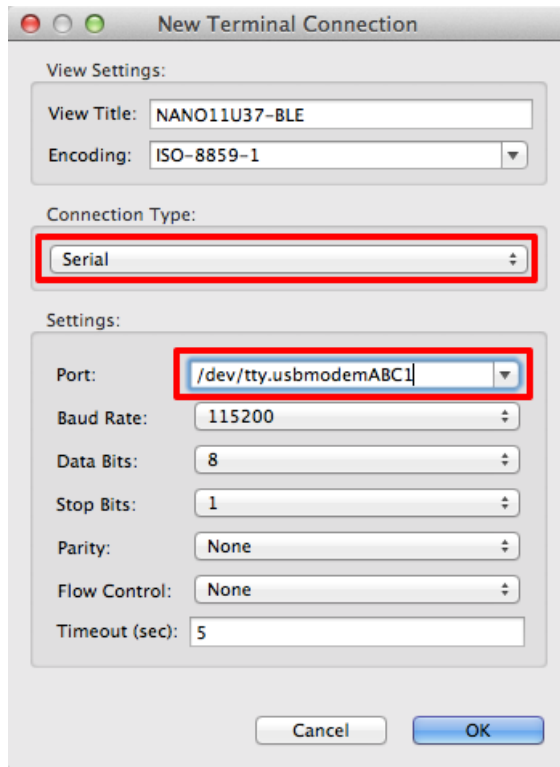
```
39 int main(void) {
40
41     #ifdef DEBUG
42         #if __USE_USB
43             usbCDC ser;
44             ser.connect();
45         #else
46             CSerial ser;
47             ser.settings(115200);
48         #endif
49         CDebug dbg(ser);
50         dbg.start();
51     #endif
52
53     //
54     // LED Demo (can be removed)
55     //
56     CPin led(LED1);
57     led.output(); // set the pin as output
```

In Terminal View, click “New Terminal Connection” to add a new connection.

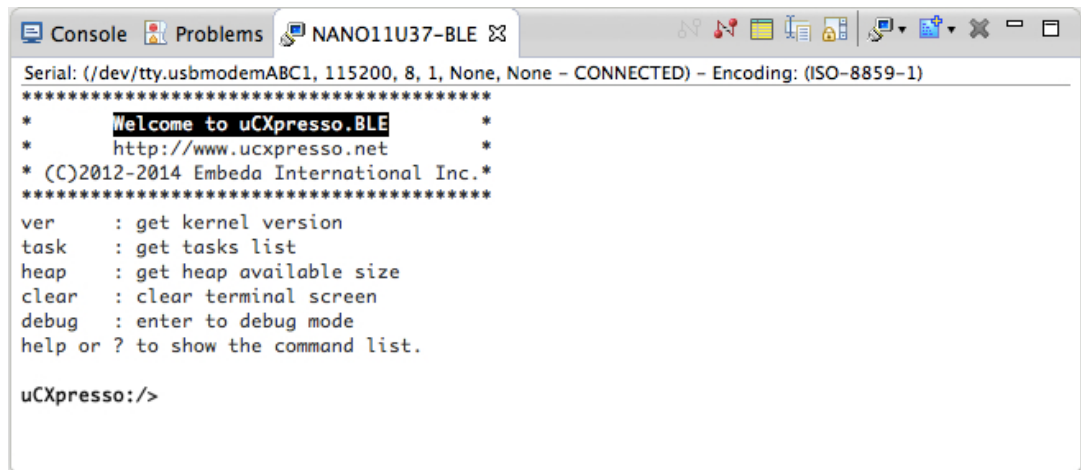


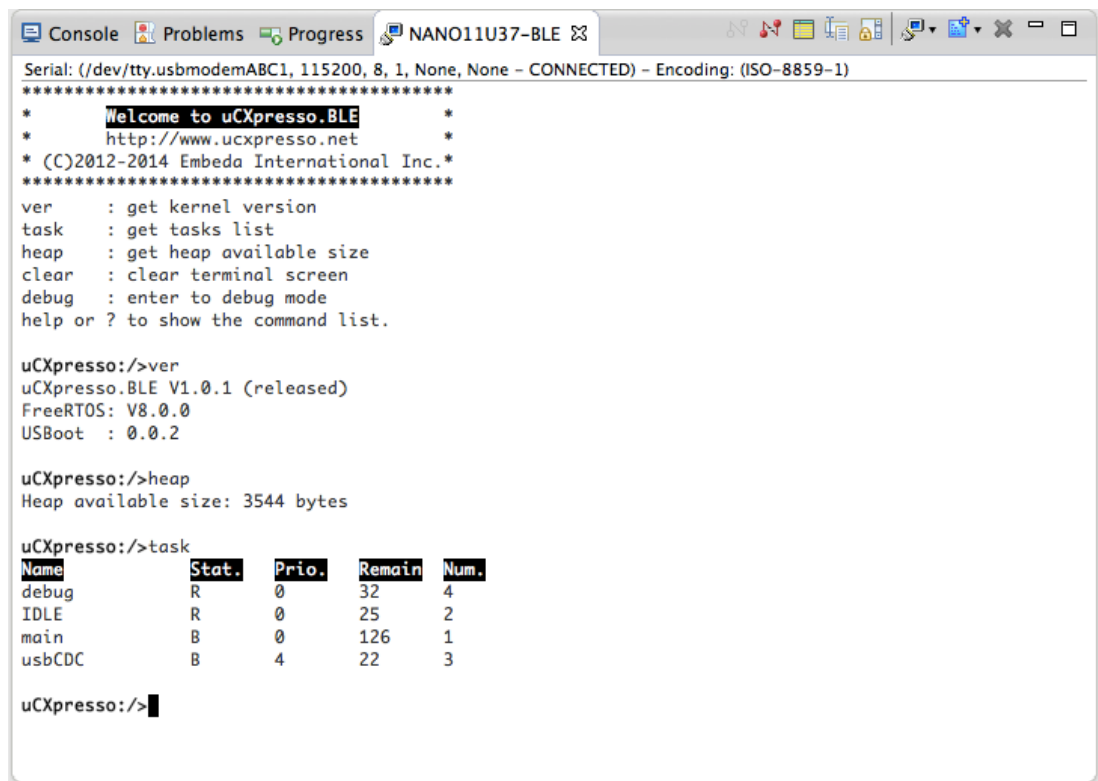
In Terminal Settings:

1. View Title: NANO11U37-BLE
2. Connection Type: Serial
3. Port : /dev/tty.usbmodemABC1 (for Mac OS/X)



The blinkLED serial shell will run in the Terminal View





```
Serial: (/dev/tty.usbmodemABC1, 115200, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-1)
*****
*      Welcome to uCXpresso.BLE      *
*      http://www.ucxpresso.net      *
* (C)2012-2014 Embeda International Inc.*
*****
ver      : get kernel version
task     : get tasks list
heap     : get heap available size
clear    : clear terminal screen
debug    : enter to debug mode
help or ? to show the command list.

uCXpresso:/>ver
uCXpresso.BLE V1.0.1 (released)
FreeRTOS: V8.0.0
USBoot   : 0.0.2

uCXpresso:/>heap
Heap available size: 3544 bytes

uCXpresso:/>task


| Name   | Stat. | Prio. | Remain | Num. |
|--------|-------|-------|--------|------|
| debug  | R     | 0     | 32     | 4    |
| IDLE   | R     | 0     | 25     | 2    |
| main   | B     | 0     | 126    | 1    |
| usbCDC | B     | 4     | 22     | 3    |



uCXpresso:/>
```

Command:

[ver](#), to show the kernel & module version.

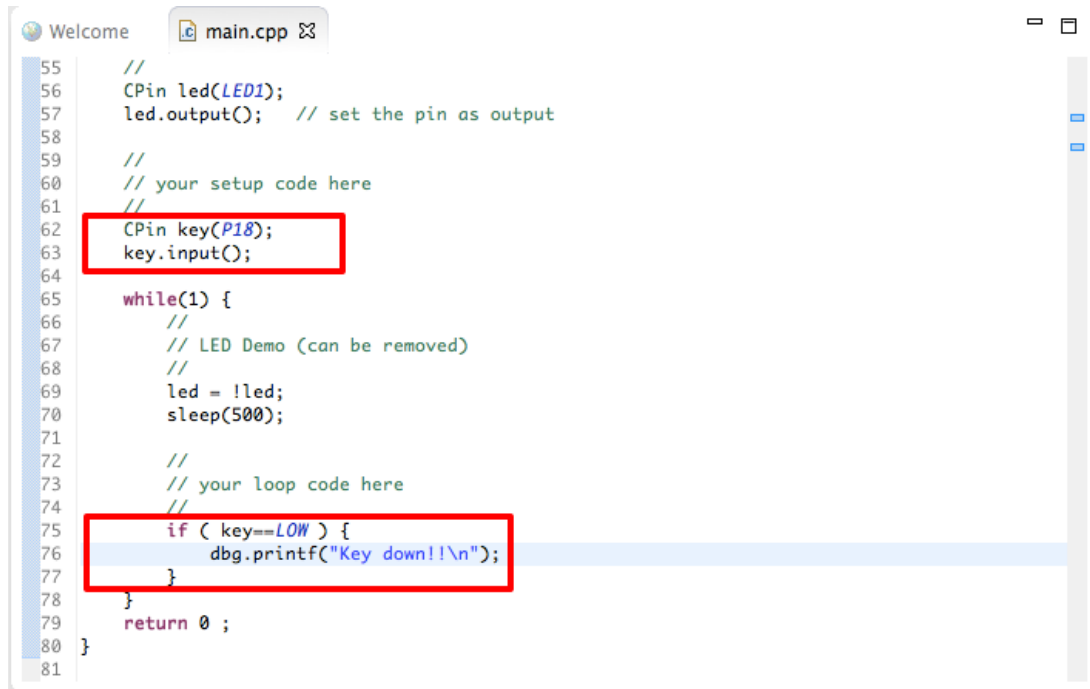
[heap](#), to check the available size of heap memory.

[task](#), to check all of tasks in the system.

[debug](#), to enter to debug mode to view the DBG(...) message from user's program.

4.2 Debug

Add 'key' object in main.cpp of blinkLED project.



```

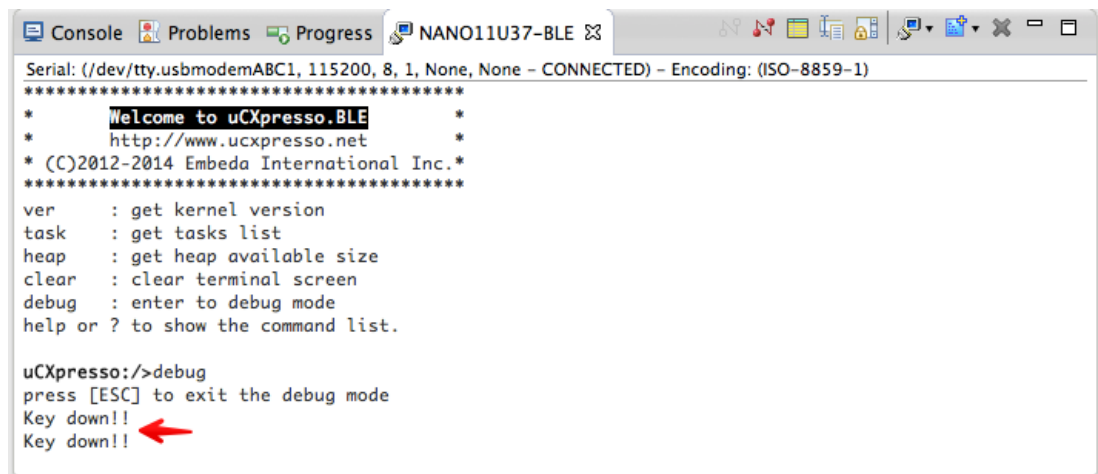
55 //
56 CPin led(LED1);
57 led.output(); // set the pin as output
58
59 //
60 // your setup code here
61 //
62 CPin key(P18);
63 key.input();
64
65 while(1) {
66 //
67 // LED Demo (can be removed)
68 //
69 led = !led;
70 sleep(500);
71
72 //
73 // your loop code here
74 //
75 if ( key==LOW ) {
76     dbg.printf("Key down!!\n");
77 }
78
79 return 0 ;
80 }
81

```

And add the key check in while loop.

Build Debug and download blinkLED.bin to nanoFLASH, then execute it.

In Terminal View, input 'debug' command to enter the debug mode:



```

Serial: (/dev/tty.usbmodemABC1, 115200, 8, 1, None, None - CONNECTED) - Encoding: (ISO-8859-1)
*****
*      Welcome to uCXPRESSO.BLE      *
*      http://www.ucxpresso.net      *
* (C)2012-2014 Embeda International Inc.*
*****
ver      : get kernel version
task     : get tasks list
heap     : get heap available size
clear    : clear terminal screen
debug    : enter to debug mode
help or ? to show the command list.

uCXPRESSO: /> debug
press [ESC] to exit the debug mode
Key down!!
Key down!!

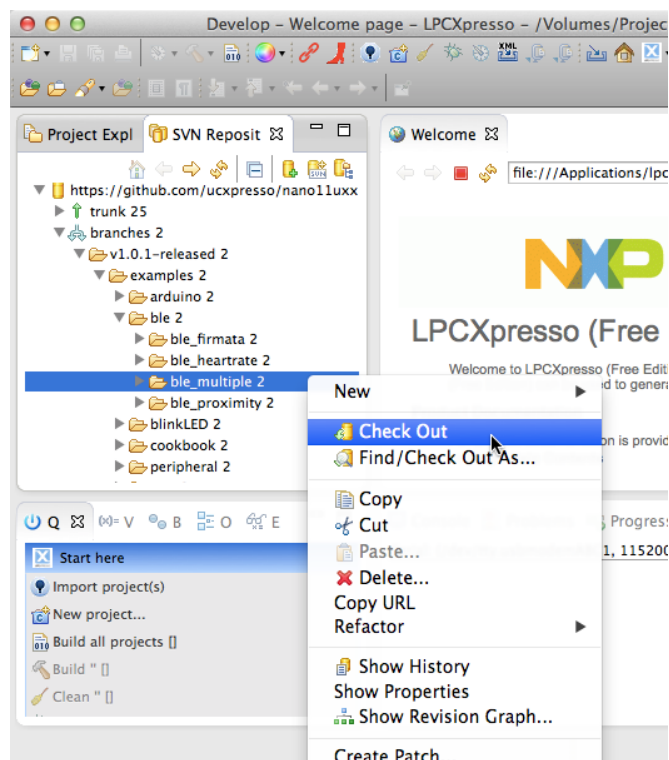
```

Try to push the key (pin 18) to LOW, and you can see the debug message to show in the terminal view.

5. Example ble_multiple Project

The ble_multiple project included:

1. Tickless Power Save Feature
2. BLE Arduino Firmata Test
3. BLE Battery Level Service
4. BLE Heath Thermometer Service
5. BLE Proximity Service



iOS App : BLE Arduino

<https://itunes.apple.com/tw/app/ble-arduino/id547628998>



Youtube Firmata Demo: http://youtu.be/7v7_mqfynRA

6. Class Manual

6.1 Online Class Manual

<https://rawgithub.com/ucxpresso/nano11uxx/master/uCXpresso.BLE/doc/doxygen/html/index.html>

uCXpresso.BLE v1.0.2
RTOS C++ Framework for Bluetooth Low Energy

Navigation: Main Page | Modules | **Classes** | Search

Class List | Class Index | Class Hierarchy | Class Members

Public Member Functions | List of all members

bleSerial Class Reference
BLE

bleSerial class is a ble core, and inherits from **CStream** class to provide the stream virtual functions for serial input and output. the **bleSerial** class also inherits from the **CThread** class and can be work in background.
More...

```
#include "class/ble_serial.h"
```

Inheritance diagram for bleSerial:

```

graph BT
    bleSerial --> CStream
    bleSerial --> CThread
    CStream --> CObject
    CThread --> CObject
  
```

Generated on Tue Mar 11 2014 10:13:40 for uCXpresso.BLE by **doxygen** 1.8.6