Application Report
SPRACM5–April 2019

How to Port WOLFSSL Onto TI Sitara AM335 Starterkit

Rio Chan

ABSTRACT

This application report introduces how to integrate the wolfSSL onto TI Sitara RTOS.

Project collateral discussed in this application report can be downloaded from the following URL: http://www.ti.com/lit/zip/spracm5.

Contents

List of Figures

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

WolfSSL is a famous TLS/SSL software solution and it is proven by many worldwide customers. Its quality is robust and the WolfSSL company maintains the security of their product each year.

This document contains:

- Where to get the right WolfSSL code versions
- Which TI RTOS version will be the suitable base for porting
- Step-by-step porting
- Code building
- How to run the demo

2 Hardware and Software Required Stuffs

- Hardware:
  - TI AM335 Starter Kit
- Software:
  - TI RTOS SDK for AM335
- WolfSSL for TI TivaC:
  - wolfSSL/wolfssl-examples
- WolfSSL main release
- TI CCS 7.4
  - Download CCS
- Microsoft Virtual Studio Express 2012
  - en_visual_studio_express_2012_for_windows_desktop_x86_web_installer_1001991.exe

3 Step-by-Step Porting

Follow these steps for porting:

1. Download the AM335 RTOS SDK.
2. Create the example by referencing this: http://processors.wiki.ti.com/index.php/Rebuilding_The_PDK
3. Build the PDK.
4. Follow the information in Section 4 to merge the WolfSSL required code.
5. The pictures in Section 4 have the WolfSSL original code.
6. Add the must-have compile option.
7. Rebuild the entire project.

4 Merging the WolfSSL Code and Building Regarding the NIMP FTP Example

Create the three sub-folders (see Figure 1) in your code base, then put those folders under the parent folder named: wolfSSL.

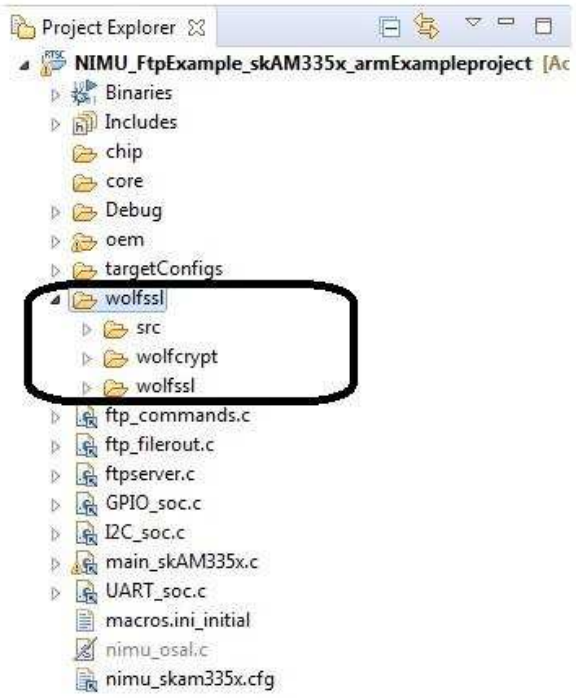

Figure 1. Project Arch to Create Three Subfolders for wolfSSL

1. Copy wolfssl-master\src to (for example):

   If you are installing all of the TI packages, then, go to pdk_am335x_1_0_10\packages\MyExampleProjects\NIMU_FtpExample_skAM335x_armExampleproject\wolfssl\src.

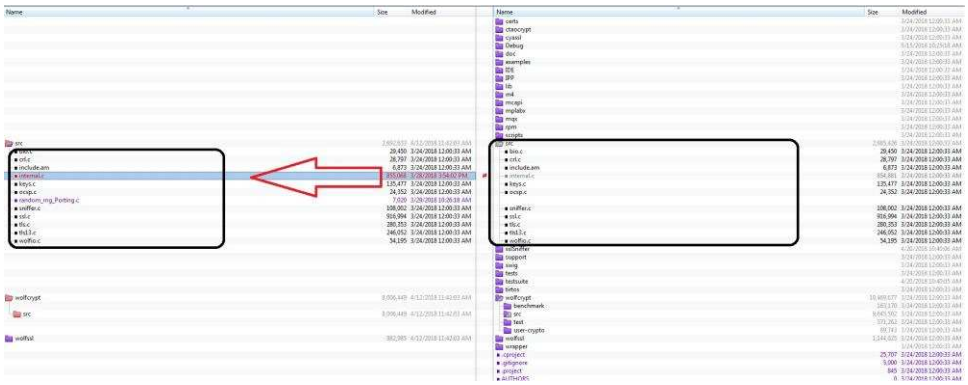

Figure 2. Copy wolfssl\src

2. Copy wolfssl-master\wolfcrypt\src to (for example):

   If you are installing all of the TI packages, then, go to pdk_am335x_1_0_10\packages\MyExampleProjects\NIMU_FtpExample_skAM335x_armExampleproject\wolfssl\wolfcrypt\src.

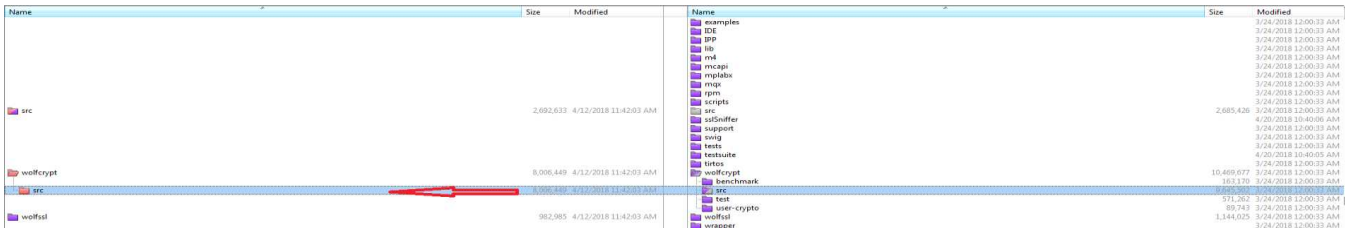

Figure 3. Copy wolfssl\wolfcrypt\src

Do not copy the “port” dir.

Do not copy the .asm .s file.

Figure 4 is marked with “X”.

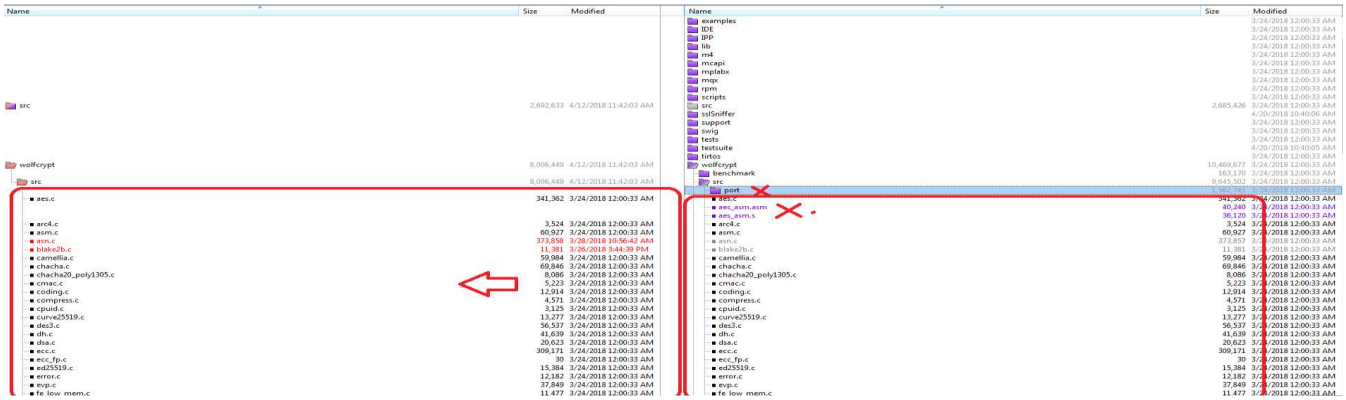

Figure 4. Ignore the .s/.asm Files

3. Copy the wolfssl-master\wolfssl\ files to (for example):

   If you are installing all the TI packgeI, then, go to pdk_am335x_1_0_10\packages\MyExampleProjects\NIMU_FtpExample_skAM335x_armExampleproject\wolfssl\wolfSSL.

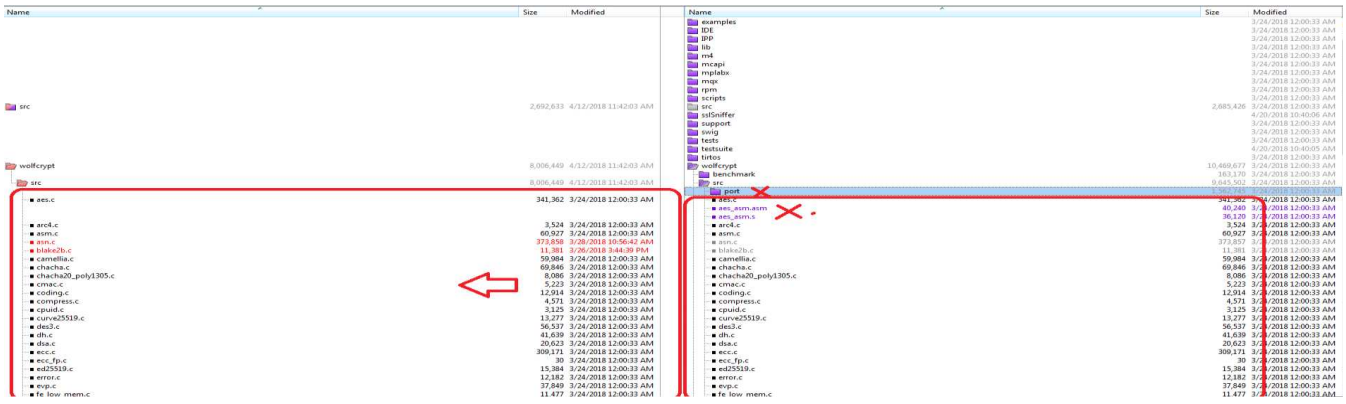

Figure 5. Copy wolfssl\wolfSSL

4. Copy the wolfssl-master\wolfssl\wolfcrypt code (without “port” folder”) to (for example):

   If you are installing all the TI packge, then, go to pdk_am335x_1_0_10\packages\MyExampleProjects\NIMU_FtpExample_skAM335x_armExampleproject\wolfssl\wolfSSL\wolfcrypt.

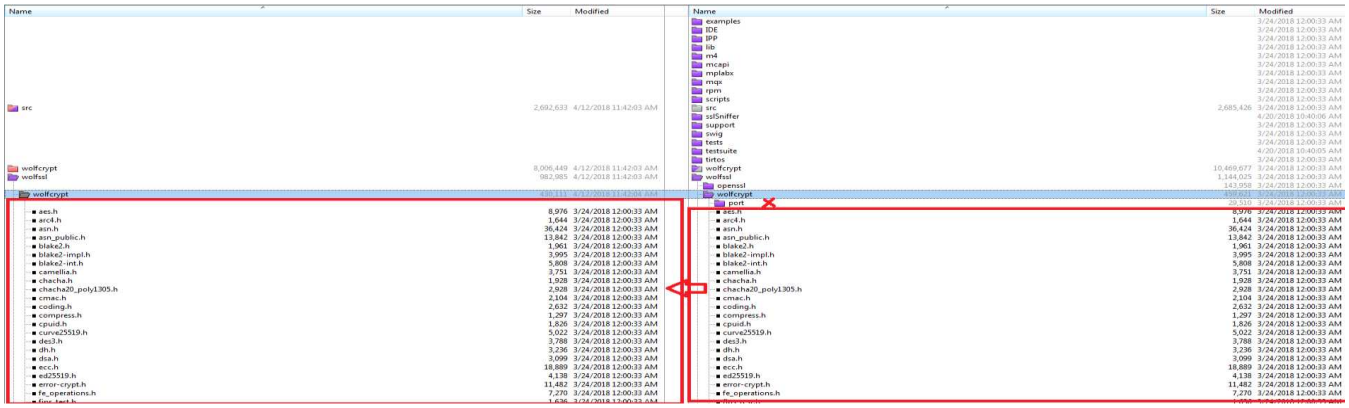

Figure 6. Copy wolfssl\wolfSSL\wolfcrypt

5. Modify some codes:

   In the wolfssl\wolfcrypt\src\random.c, add the code as shown in Figure 7.

Figure 7. random.c Modification

6. In the wolfssl\src\internal.c, add the code as shown in Figure 8.

Figure 8. internal.c Modification

Add a new file (for example):

- wolfssl\src\random_rng_Porting.c

Reference the random_rng_Porting.c file in this zip file: 2018_5_15_WolfSSL_Importan_Temp_Backup_Client_Server_All_Okay_Release.

Download from here.

```
25 //*****************************************************************************
26
27 #include <stdint.h>
28
29 #if 0//Rio
30 //#include "ustdlib.h"
31 #else
32 #include <wolfssl/wolfcrypt/types.h>
33 #endif
34
35 #include "random_rng_Porting.h"
36
37 //*****************************************************************************
38 //
39 //! \addtogroup random_api
40 //! @{
41 //
42 //*****************************************************************************
43
```

Figure 9. random_rng_Porting.c Modification

7. Add the two parts shown in Figure 10 into the file: nimu_skam335x.cfg.

```
/*
*   file name: nimu_skam335x.cfg
*   This file is included in the ethernet switch example
*
*****************************************************************************/

/* ================= General configuration ================= */

var enableStaticIP          = 1;

var Defaults = xdc.useModule('xdc.runtime.Defaults');
var Diags = xdc.useModule('xdc.runtime.Diags');
var Error = xdc.useModule('xdc.runtime.Error');
var Main = xdc.useModule('xdc.runtime.Main');
var Memory = xdc.useModule('xdc.runtime.Memory')
var SysMin = xdc.useModule('xdc.runtime.SysMin');
var System = xdc.useModule('xdc.runtime.System');
var Text = xdc.useModule('xdc.runtime.Text');
var Clock = xdc.useModule('ti.sysbios.knl.Clock');
var Task = xdc.useModule('ti.sysbios.knl.Task');
var Semaphore = xdc.useModule('ti.sysbios.knl.Semaphore');
var Hwi = xdc.useModule('ti.sysbios.hal.Hwi');
var Timer = xdc.useModule('ti.sysbios.hal.Timer');
var HeapMem = xdc.useModule('ti.sysbios.heaps.HeapMem');
var SemihostSupport = xdc.useModule('ti.sysbios.rts.gnu.SemiHostSupport');

//CS: 2018/4/24, solve Seconds_set in the main.c
var Seconds = xdc.useModule('ti.sysbios.hal.Seconds');


/*
 * Program.argSize sets the size of the .args section.
 * The examples don't use command line args so argSize is set to 0.
 */
Program.argSize = 0x0;
```

Figure 10. nimu_skam335x.cfg Modification

8. In the wolfssl\src\internal.c file, add the two parts:
   The first part is to register the user I/O call back:
   - wolfSSL_SetIORecv
   - wolfSSL_SetIOSend

```
151 //Rio: Thos include will solve the my_IORecv/Send error
152 #include <ti/ndk/inc/usertype.h>
153 #include <ti/ndk/inc/socketndk.h>
154 #include <ti/ndk/inc/socket.h>
155 #if 1//Rio: Porting my own IO Send/Recv
156     int my_IORecv(WOLFSSL* ssl, char* buff, int sz, void* ctx)
157     {
158         /* By default, ctx will be a pointer to the file descriptor to read from.
159          * This can be changed by calling wolfSSL_SetIOReadCtx(). */
160         int sockfd = *(int*)ctx;
161         int recvd;
162
163
164         /* Receive message from socket */
165         if ((recvd = recv(sockfd, buff, sz, 0)) == -1) {
166             /* error encountered. Be responsible and report it in wolfSSL terms */
167
168             //WOLFSSL_ENTER(stderr, "IO RECEIVE ERROR: \n");
169             WOLFSSL_ENTER("IO RECEIVE ERROR: \n");
```

Figure 11. My Own IO Callback Regs

Another one is to get the system time, this is related with the NO_ASN_TIME/ASN_TIME config.

```
5937     return (unsigned long)seconds + (unsigned long)ticksWithinSecond / 1000;
5938 }
5939
5940 #if 1//Rio
5941 unsigned long my_time(unsigned long* timer)
5942 {
5943 #if 0//Rio: This is not correct time stamp that WolfSSL needs
5944     //(void)timer;
5945     return Timestamp_get32(); /* use your own code to get time */
5946 #else
5947 #if 0
5948 //Rio: Time test, below is working, but the format is not correct.
5949     time_t t;
5950     struct tm *ltm;
5951     char *curTime;
5952
5953     //https://www.epochconverter.com/ Rio: Go to this web to convert EPOCH time
5954     //https://www.wolfssl.com/forums/topic862-solved-asnbeforedatee-when-calling-ctxloadverifybuffer.html
5955
5956     Seconds_set(1476377542);
5957     t = time(NULL);
5958     ltm = localtime(&t);
5959     curTime = asctime(ltm);
5960     UART_printf("Rio: Asn.c my_time return time : %s\n", curTime);
5961 #endif
5962     //Rio: This is the right solution
5963     return getSeconds();
5964 #endif
5965 }
5966 #endif
5967
5968
```

Figure 12. My Time Modification

9. Add the “must-have” compile options for wolfSSL.
   For example, xNO_FILESYSTEM is to disable the “NO_FILESYSTEM”.

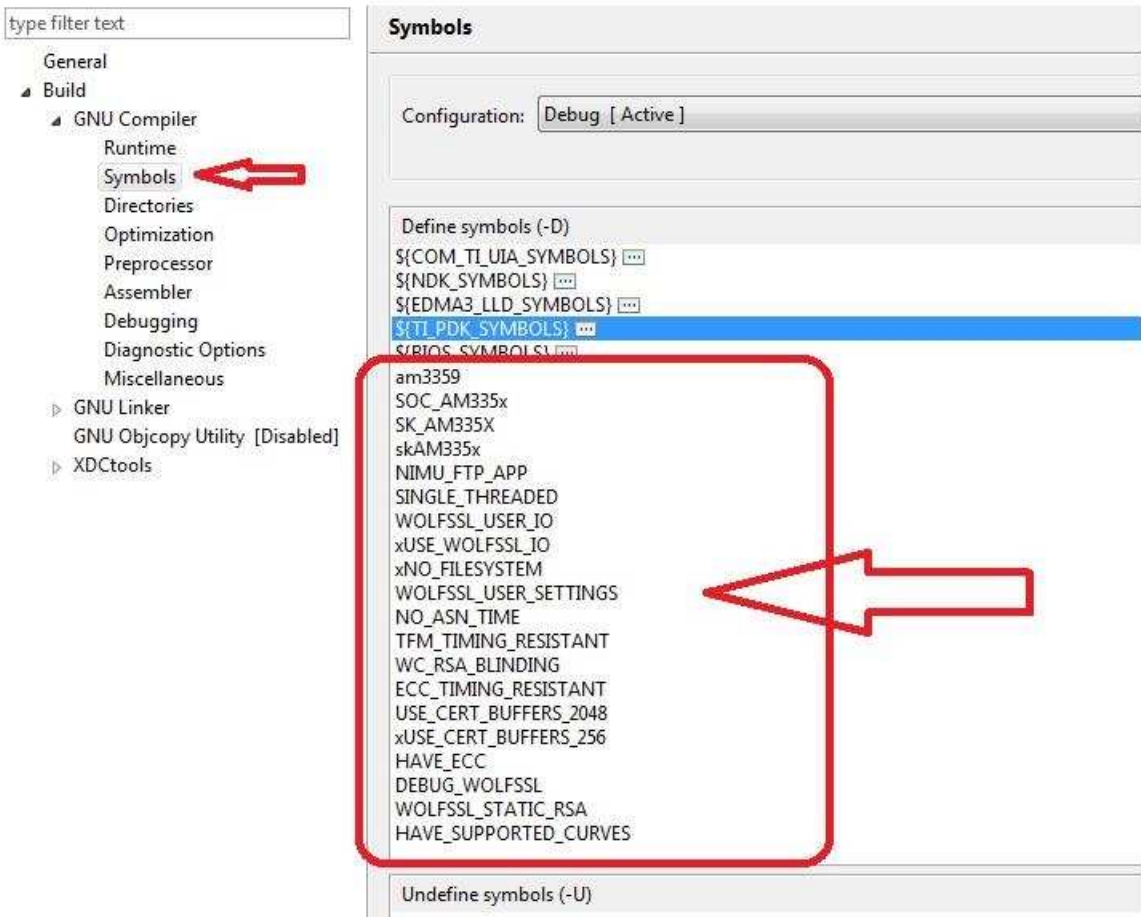

Figure 13. Compile Definition

Add the included folder for the wolfSSL used header file.

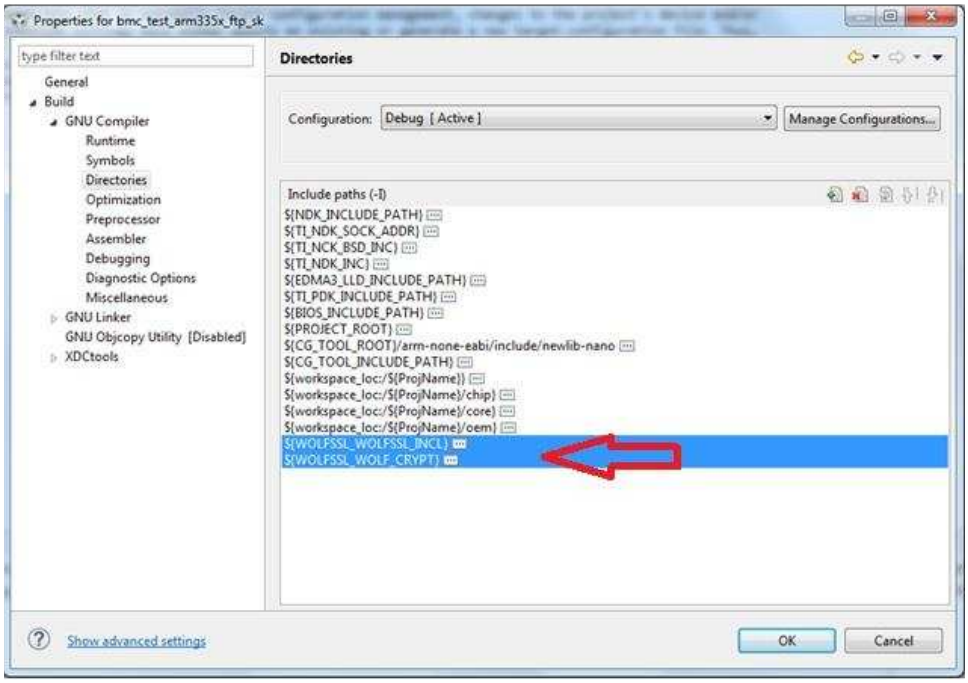

Figure 14. Include Path Setting Part 1

10. Add the Variables for environment including use.

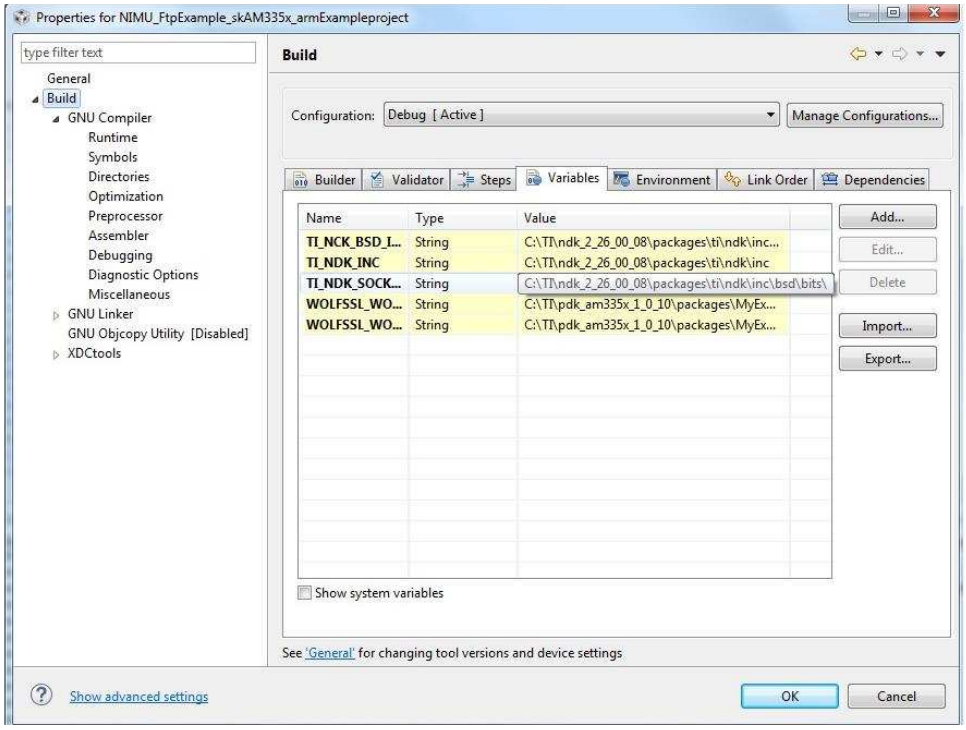

Figure 15. Include Path Setting Part 2

11. Adding the variable to the environment.

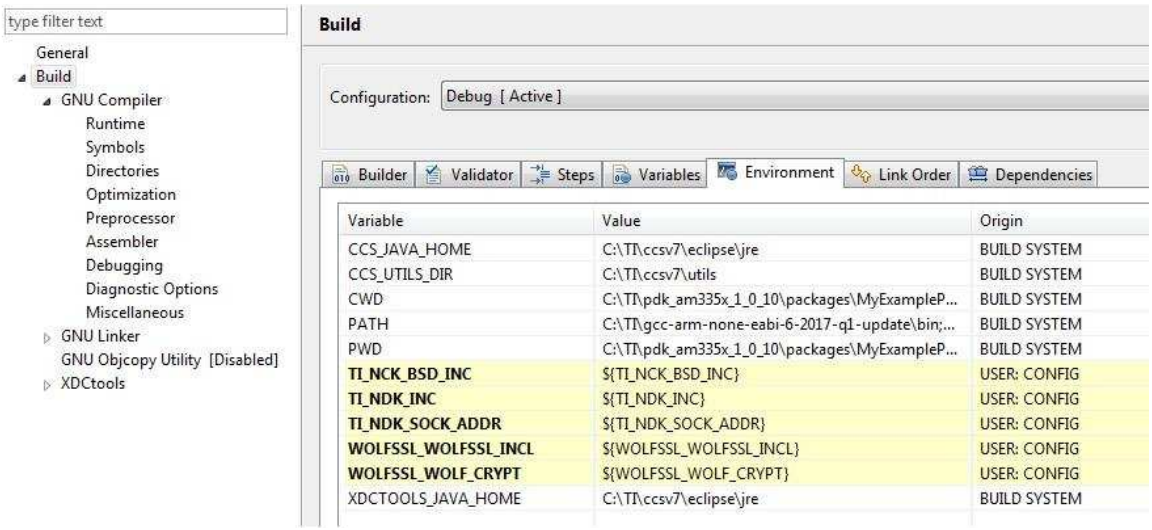

Figure 16. Include Path Setting Part 3

12. Project settings:

    Select the right compiler version and the boards. AM335SK can use the ICE_AM3359.

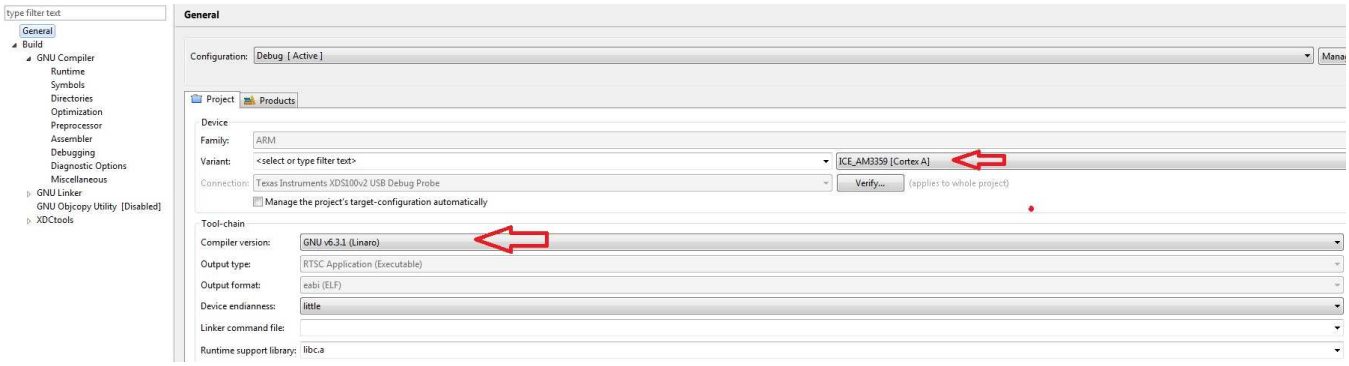

Figure 17. Project Setting

13. Product settings:

    Please make sure the right versions of:

    - XDCtools
    - SysBios
    - PDK
    - NDK

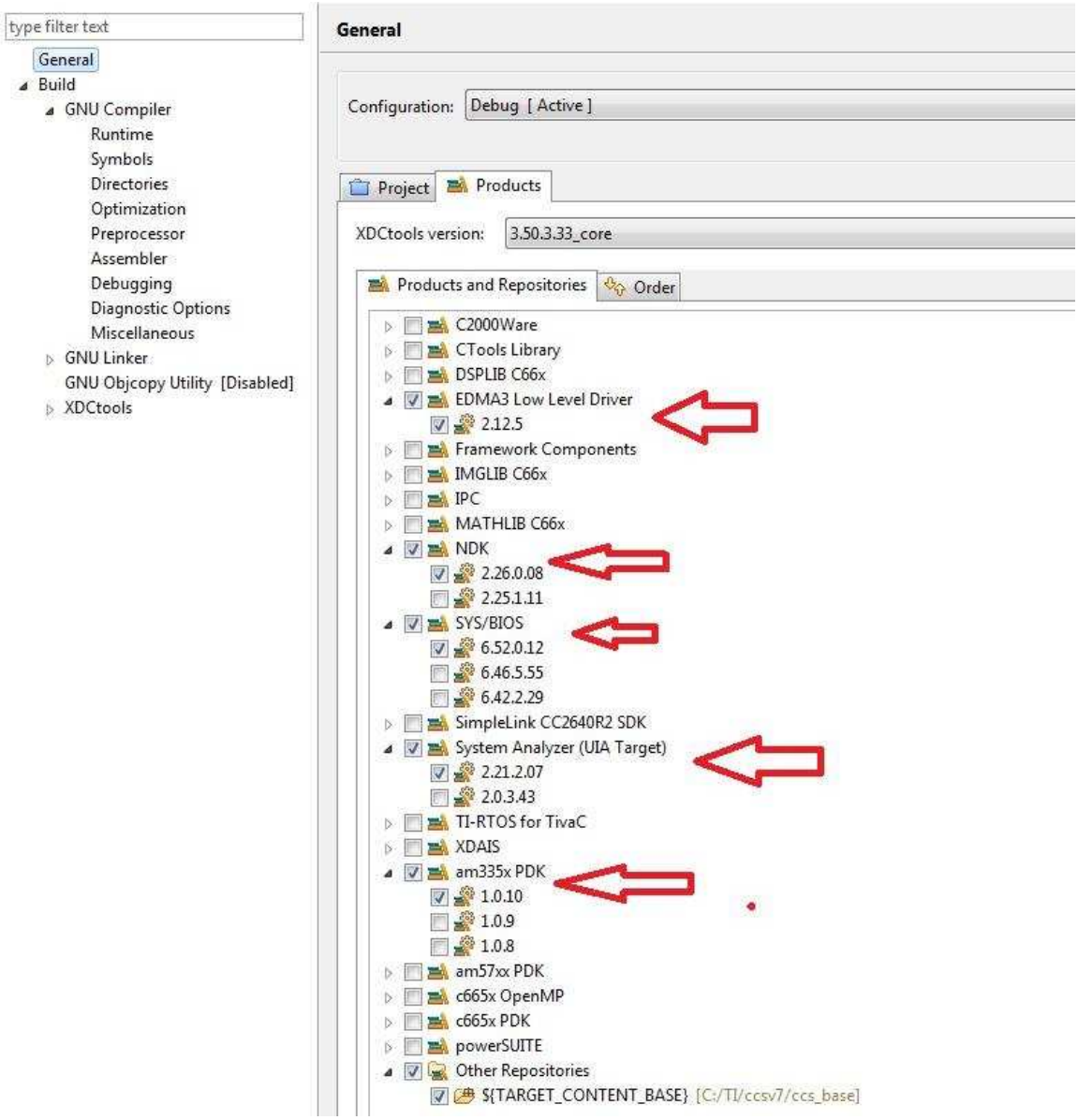

Figure 18. Product Setting

14. Target config:

    The important key for the download image and debug is the JTAG.

    - It needs to choose XDS100V2 USB
    - Board of device is: SK_AM3358

    You can test the connection while finishing your own setting of the “ccxml.

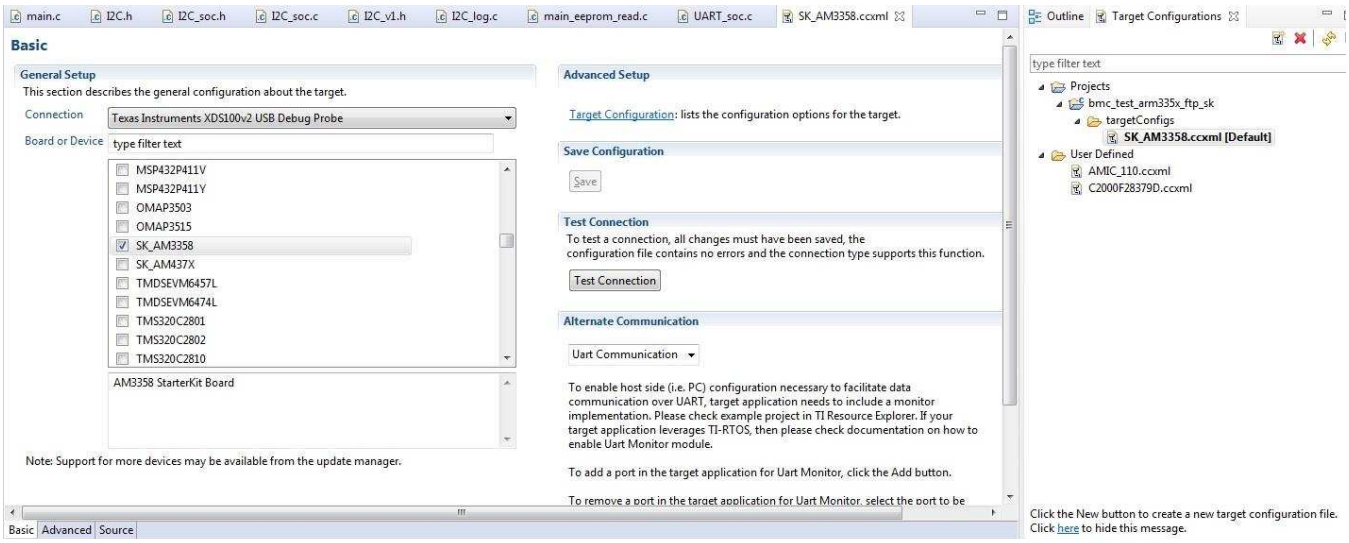

Figure 19. Target Config

15. Build should be successful.

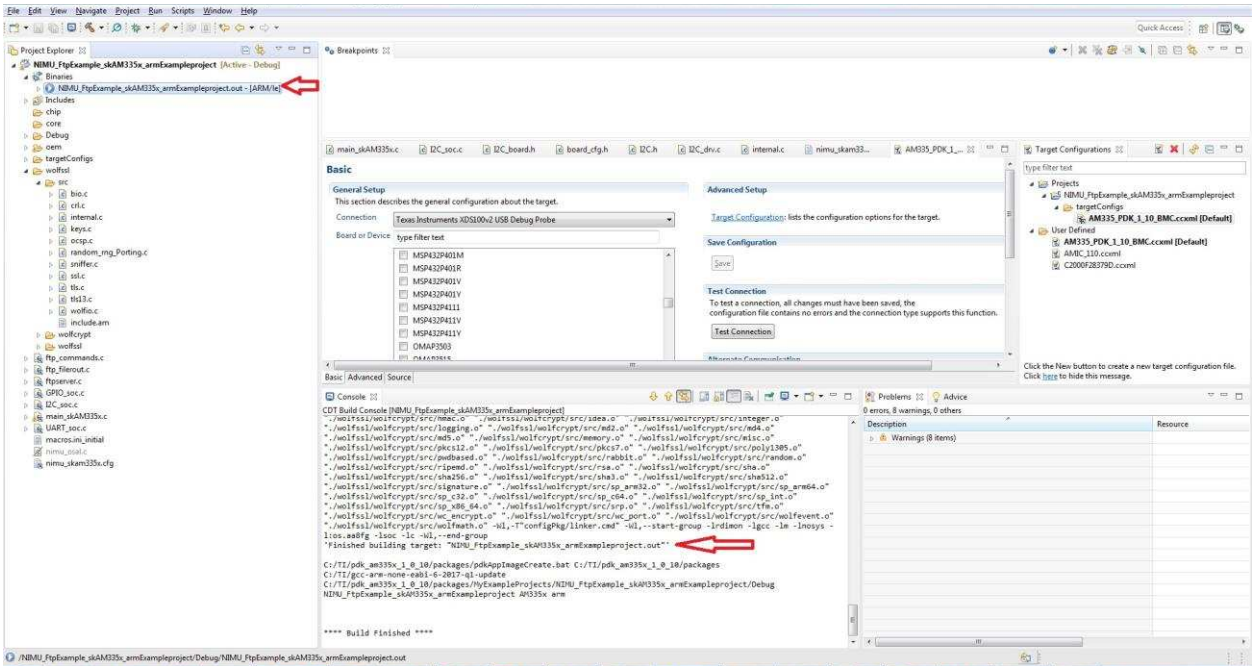

Figure 20. Project Building

5 How to Verify?

You can reference this article:

- USING WOLFSSL WITH VISUAL STUDIO

You can access the two exe files that are listed under this path: : wolfssl-master\Debug.

- Client.exe
- Server.exe
- Run the WolfSSL Client on the NB to verify your server code (the Server IP/port is depended on the code).
  - Client.exe –h 192.168.1.4 –p 2000
- Run the WolfSSL Server on the NB to verify your WolfSSL client code (the port is dependent on the code).
  - Server.exe –b –p 1000
- Use the NB with Win7 and Virtual studio express.
  - en_visual_studio_express_2012_for_windows_desktop_x86_web_installer_1001991.exe

6 Testing environment

The testing environment is as shown in Figure 21; each node will communicate with the Ethernet.

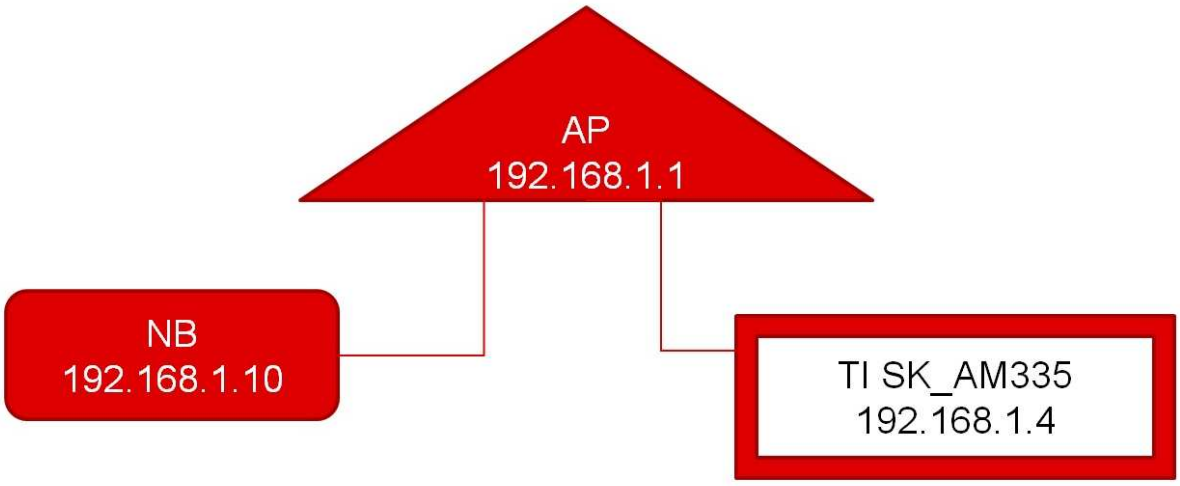

Figure 21. Demo Setting

7 Demo Movie

Search on YouTube for “TI Rio WolfSSL”. Or, visit the link here.

8 Function API

You can reference this article:

- wolfSSL-Porting-Guide.pdf

The four API are the basic soul for the entire demo. All of the important API are listed as shown below.

- wolfSSL_CTX_new
- wolfSSL_CTX_load_verify_buffer
- wolfSSL_CTX_use_certificate_buffer
- wolfSSL_CTX_use_PrivateKey_buffer

The two calls are user configured for your own code; you can refer to the wolSSL porting guide.

- wolfSSL_SetIORecv
- wolfSSL_SetIOSend

The following are the APIs used after the TCP socket is configured and connected. The wolfSSL data transmission on the TCP socket will rely on those APIs.

- wolfSSL_new
- wolfSSL_set_fd
- wolfSSL_connect
- wolfSSL_get_fd
- wolfSSL_write
- wolfSSL_read
- wolfSSL_free

9 Test Pass Logs

Two cases were tested. Only the important logs were captured on the AM335 server role.

- Case1: AM335 is the server role, and NB is the client role
- Case2: AM335 is the client role, and NB is the server role.

Case 1

```
AM335 Server Role _ AM335 Side(Partial part only):

tcpServerHandler_worker: start clientfd = 0x80096264
wolfSSL_read()
wolfSSL_read_internal()
ReceiveData()
Handshake not complete, trying to finish
wolfSSL_negotiate
SSL_accept()
my_IORecv: received
growing input buffer

my_IORecv: received
received record layer msg
DoHandShakeMsg()
DoHandShakeMsgType
processing client hello
Matched No Compression
Adding signature algorithms extension
Signature Algorithms extension received
MatchSuite
VerifyServerSuite
Requires RSA
Verified suite validity
accept state ACCEPT_FIRST_REPLY_DONE
growing output buffer
```

```
…….
BuildMessage
my_IOSend: sent
Shrinking output buffer

wolfSSL_read()
wolfSSL_read_internal()
ReceiveData()
my_IORecv: received
growing input buffer

my_IORecv: received
received record layer msg
got ALERT!    Mps. This one is correct, that means the NB (Client has no more data to send to
AM335)
Got alert
        close notify
```

AM335 Server Role : NB Side:

You can see the NB is running as Client.

```
E:\TLS_Wolf_64_Bit\wolfssl-master\Debug>client -h 192.168.1.4 -p 2000
peer's cert info:
 issuer : /C=US/ST=Montana/L=Bozeman/O=Sawtooth/OU=Consulting/CN=www.wolfssl.com
/emailAddress=info@wolfssl.com
 subject: /C=US/ST=Montana/L=Bozeman/O=wolfSSL/OU=Support/CN=www.wolfssl.com/ema
ilAddress=info@wolfssl.com
 serial number:01
SSL version is TLSv1.2
SSL cipher suite is TLS_RSA_WITH_AES_256_CBC_SHA256
Client Random : B42BCE30A6B622E3D9EFE0A6455265F1E86447917CC441DECFB7A1243B84F9CB

wolfSSL's AM335 SK  Series Connected Launchpad Heard you loud and clear!!!

E:\TLS_Wolf_64_Bit\wolfssl-master\Debug>
```

Case 2

AM335 Client Role : AM335 Side(Partial part only). Only the important logs were captured on the AM335 side.

```
---------- ps: tcpClientHandler:wolfSSL_connect success ----------.
SSL_get_fd
SSL_write()
growing output buffer

BuildMessage
my_IOSend: sent
Shrinking output buffer

wolfSSL_read()
wolfSSL_read_internal()
ReceiveData()
my_IORecv: received
growing input buffer

my_IORecv: received
received record layer msg
got app DATA
Shrinking input buffer

---------- ps: tcpClientHandler:wolfSSL_Heard: "I hear you fa shizzle!" ----------.
SSL_free
CTX ref count not 0 yet, no free
---------- ps. tcpClientHandler:wolfSSL Memory_free ----------.
SSL_CTX_free
CTX ref count down to 0, doing full free
wolfSSL_CertManagerFree
```

```
wolfSSL_Cleanup
wolfCrypt_Cleanup
```

AM335 Client Role :NB Side:

```
E:\TLS_Wolf_64_Bit\wolfssl-master\Debug>Server -b -p 1000
peer's cert info:
 issuer : /C=US/ST=Montana/L=Bozeman/O=wolfSSL_2048/OU=Programming-2048/CN=www.w
olfssl.com/emailAddress=info@wolfssl.com
 subject: /C=US/ST=Montana/L=Bozeman/O=wolfSSL_2048/OU=Programming-2048/CN=www.w
olfssl.com/emailAddress=info@wolfssl.com
 serial number:b9:bc:90:ed:ad:aa:0a:8c
SSL version is TLSv1.2
SSL cipher suite is TLS_RSA_WITH_AES_256_CBC_SHA256
Server Random : 2368E8B669D5D3CA1706F90292914C8A072135D3DB7BE6DCB55003ADD597D550

Client message: Hello from TI AM335 SK EVM

E:\TLS_Wolf_64_Bit\wolfssl-master\Debug>
```

10 References

- Using WolfSSL to add TLS/SSL security to a TCP/IP server: (https://www.freertos.org/FreeRTOS-Plus/WolfSSL/Using-SSL-TLS-in-a-server-site-application.shtml)
- Using WolfSSL to add TLS/SSL security to a TCP/IP client: (https://www.freertos.org/FreeRTOS-Plus/WolfSSL/Using-SSL-TLS-in-a-client-site-application.shtml)
- Using wolfSSL with TI-RTOS - Texas Instruments Wiki: (http://processors.wiki.ti.com/index.php/Using_wolfSSL_with_TI-RTOS)
- wolfSSL User Manual | Chapter 11: SSL/TLS Tutorial | Documentation: (https://www.wolfssl.com/docs/wolfssl-manual/ch11/)
- wolfSSL User Manual | Chapter 17: wolfSSL API | Documentation: (https://www.wolfssl.com/docs/wolfssl-manual/ch17/)
- Using wolfSSL with Visual Studio | wolfSSL Embedded SSL/TLS Library: (https://www.wolfssl.com/docs/visual-studio/)
- TCP socket error codes https://gist.github.com/gabrielfalcao/4216897
- Udp error codes listed anywhere_ - Troubleshooting – Particle: (https://community.particle.io/t/udp-error-codes-listed-anywhere/18775/3)
- Processor SDK RTOS NDK - Texas Instruments Wiki: (http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_NDK#Examples)

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265