PLUG-AND-TRUST-NANO-PACKAGE

Release 1.0.0

Mar 31, 2022

CONTENTS

Content

CHAPTER

ONE

CHANGELOG

Release v1.0.0

- Initial commit
- **Features**
  - ECDSA and ECDH with NIST P256
  - AES Encrypt / Decrypt (ECB,CBC,CTR)
  - Binary Objects
  - Encrypted I2C communication using PlatformSCP channel based on Global Platform SCP03
  - Platforms - Linux, frdm-k64 bare metal, Zephyr OS

CHAPTER

TWO

INTRODUCTION

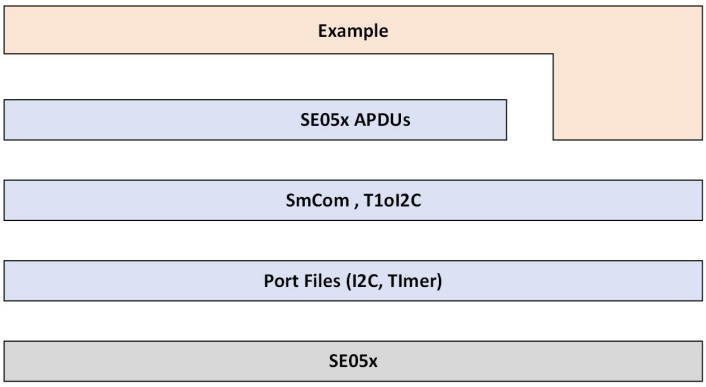
2.1 Introduction to the Plug & Trust Nano Package

The Plug & Trust Nano package is an optimized middleware for communicating between a host processor or microcontroller and the EdgeLock SE05x and A5000 secure elements and authenticators. The Plug & Trust Nano Package has been designed for memory constrained devices and consumes only ~1KB of RAM for SCP03 encrypted communication over I2C.

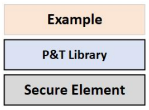
Note that the examples and libraries contained in the Plug & Trust Nano package have been specifically designed to fit into constrained devices and are not compatible with examples and libraries available in the standard Plug & Trust package.

The standard Plug and Trust middleware package can be downloaded from https://www.nxp.com/products/:SE050. The package has support for more crypto curves, plugins, examples and more platforms.

Nano Package -



PLUG AND TRUST NANO PACKAGE



Nano package Features

- ECDSA and ECDH with NIST P256
- AES Encrypt / Decrypt (ECB,CBC,CTR)
- Binary Objects
- Encrypted I2C communication using PlatformSCP channel baed on Global Platform SCP03 channel
- Platforms - Linux, frdm-k64 bare metal, Zephyr OS

Folder structure

```
example --- se05x examples
lib  ------ se05x library files
 |
 |__
        apdu -------- Contains se05x apdu apis with scp03 support
        platform ---- Platform specific files. Modify / add the files here to support␣
↪other platform
        t1oi2c ------ T1OI2C files
        mbedtls_alt - Mbedtls ALT files to access SE05x
```

2.2 PlatformSCP03

Using nano package, host can establish encrypted I2C communication via PlatformSCP channel (based on Global Platform SCP03). This requires some host crypto operations.

Plug and Trust Nano package has these host crypto apis implemented using

- openssl (version 1.1.1). (simw-nanopkg/lib/apdu/scp03/openssl)
- tinyCrypt (simw-nanopkg/lib/apdu/scp03/tc)

To use a different host crypto, re-implement the host crypto apis - *simw-nanopkg/lib/apdu/scp03/se05x_scp03_crypto.h*

When building the example with 'Platform SCP' enabled, make sure to assign valid scp03 keys to session context. (DEK key is required only for key rotation - se05x_rotate_scp03_keys).

Note: Product Deployment => Make sure to store the SCP03 keys securly.

The Default Platform SCP keys for ease of use configurations are present in

- SE050 Configuration: https://www.nxp.com/docs/en/application-note/AN12436.pdf
- SE051 Configuration: https://www.nxp.com/webapp/Download?colCode=AN12973

```
void ex_set_scp03_keys(pSe05xSession_t session_ctx)
{
    session_ctx->pScp03_enc_key     = &scp03_enc_key[0];
    session_ctx->pScp03_mac_key     = &scp03_mac_key[0];
    session_ctx->pScp03_dek_key     = NULL;
    session_ctx->scp03_enc_key_len = 16;
    session_ctx->scp03_mac_key_len = 16;
    session_ctx->scp03_dek_key_len = 0;
    return;
}
```

2.3 Build option

Platform SCP03

```
-DPLUGANDTRUST_SCP03=ON : Build with Platform SCP03 enabled
-DPLUGANDTRUST_SCP03=OFF : Build with Platform SCP03 disabled
```

Debug Logs

```
-DPLUGANDTRUST_DEBUG_LOGS=ON : Build with Debug logs enabled
-DPLUGANDTRUST_DEBUG_LOGS=OFF : Build with Debug logs disabled
```

2.4 Examples

Examples on linux

Refer *simw-nanopkg/examples/<example>/readme.rst*. Section 3 *Getting Started (Linux)*

Examples on k64

Refer *simw-nanopkg/se05x_sign/k64f/readme.rst*. Section 4.2 *SE05x Sign Example - frdm-k64*

Refer *simw-nanopkg/se05x_crypto/k64/readme.rst*. Section 4.1 *SE05x Crypto Example - frdm-k64*

Examples on Zephyr OS

Integration of nano package in Zephyr OS is maintained in branch - **int/zephyr**

Refer *simw-nanopkg/zephyr/readme.rst*. Section 5 *Getting Started with Zephyr + SE05x*

Note: To use policies with objects refer 'test_nist256_sign_policy' in 'Se05x Crypto' example. For more details on policies, Refer Section '3.7 Policies' in https://www.nxp.com/docs/en/application-note/AN12413.pdf

---

2.5 Porting

Platform specific files are maintained in **simw-nanopkg/lib/platform** folder.

Modify / add the files here to support other platforms. By default port files are available for Linux, Zephyr and K64 MCU.

2.6 Mbedtls Alt files

Nano package provides MbedTLS Alt files as an alternative/additional approach to access the secure element using mbedTLS.

In the current implementation only ECDSA Sign is supported via MbedTLS ALT files.

Using Mbedtls Alt files in Zephyr OS

Set **CONFIG_PLUGANDTRUST_MBEDTLS_ALT** to build Plug and Trust with Mbedtls Alt files.

GCP cloud example in Zephyr OS is modified to use SE05x for ECDSA sign.

Prerequisite - SE05x provisioned with private key at location (say 0x11223344).

Replace the private key in *zephyr/samples/net/cloud/google_iot_mqtt/src/private_info/key.c* with the reference to provisioned private key.

The following provides an example of an EC reference key. The value reserved for the private key has been used to contain:

- a pattern of `0x10..00` to fill up the datastructure MSB side to the desired key length

- a 32 bit key identifier (in the example below `0x11223344`)
- a 64 bit magic number (always `0xA5A6B5B6A5A6B5B6`)
- a byte to describe the key class (`0x10` for Key pair)
- a byte to describe the key index (use a reserved value `0x00`)

```
Private-Key: (256 bit)
priv:
    10:00:00:00:00:00:00:00:00:00:00:00:00:00:00:
    00:00:00:11:22:33:44:A5:A6:B5:B6:A5:A6:B5:B6:
    10:00
```

Refer *zephyr/samples/net/cloud/google_iot_mqtt/README.rst* to build GCP cloud example.

CHAPTER

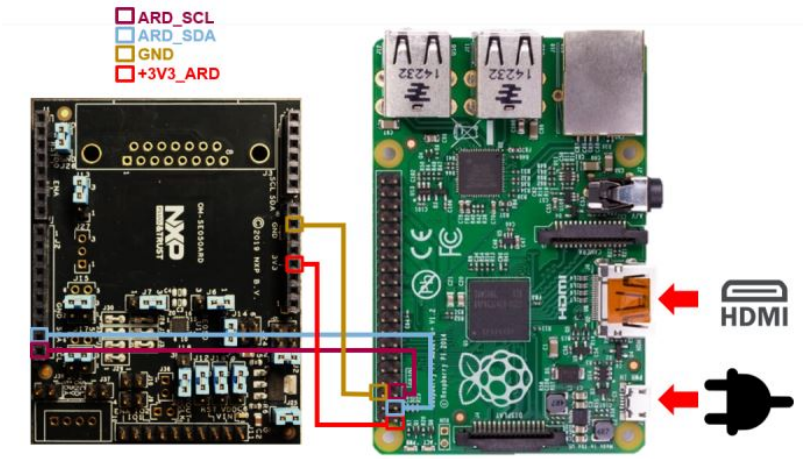
THREE

GETTING STARTED (LINUX)

Prerequisite

Raspberry Pi with raspbian OS installed.

Rpi Connections -



OM-SE05xARD wiring to the Raspberry Pi board

3.1 SE05x Crypto Example

Overview

This example demonstrates SE05x crypto functionality using se05x APIs.

Refer file - 'simw-nanopkg/examples/se05x_crypto/src/ex_se05x_crypto.c'

Note: When building the example with 'Platform SCP' enabled, make sure to assign valid scp03 keys to session context.

Linux Prerequisite

Install cmake , Openssl 1.1.1

```
sudo apt-get install cmake cmake-curses-gui cmake-gui libssl-dev
```

Linux build

To build example run

```
cd simw-nanopkg/examples/se05x_crypto/linux
mkdir build
cd build
cmake ../
make
./ex_se05x_crypto
```

Build options

Platform SCP03

```
-DPLUGANDTRUST_SCP03=ON -- Build with Platform SCP03 enabled
-DPLUGANDTRUST_SCP03=OFF -- Build with Platform SCP03 disabled
```

Debug Logs

```
-DPLUGANDTRUST_DEBUG_LOGS=ON -- Build with Debug logs enabled
-DPLUGANDTRUST_DEBUG_LOGS=OFF -- Build with Debug logs disabled
```

Sample Output

If everything is successful, the output will be similar to:

```
./ex_se05x_crypto
Plug and Trust nano package - version: 1.0.0
Establish Secure Channel to SE05x !
Get Version ==>
Applet Version 6.0.0
test_get_version complete
test_generate_nist256_key complete
test_set_get_nist256_key complete
test_nist256_sign_verify complete
test_set_certificate complete
test_ecdh complete
test_aes_ECB_NOPAD complete
test_aes_CBC_NOPAD complete
test_aes_CTR complete
test_nist256_sign_policy complete
```

3.2 SE05x Sign Example

Overview

This example demonstrates signing a data using nist256 key.

Refer file - 'simw-nanopkg/examples/se05x_crypto/src/ex_se05x_sign.c'.

Note: When building the example with 'Platform SCP' enabled, make sure to assign valid scp03 keys to session context.

Linux Prerequisite

Install cmake , Openssl 1.1.1

```
sudo apt-get install cmake cmake-curses-gui cmake-gui libssl-dev
```

Linux build

To build example run:

```
cd simw-nanopkg/examples/se05x_crypto/linux
mkdir build
cd build
cmake ../
make
./ex_se05x_sign
```

Build options

Platform SCP03

```
-DPLUGANDTRUST_SCP03=ON : Build with Platform SCP03 enabled
-DPLUGANDTRUST_SCP03=OFF : Build with Platform SCP03 disabled
```

Debug Logs

```
-DPLUGANDTRUST_DEBUG_LOGS=ON : Build with Debug logs enabled
-DPLUGANDTRUST_DEBUG_LOGS=OFF : Build with Debug logs disabled
```

Sample Output

If everything is successful, the output will be similar to:

```
Plug and Trust nano package - version: 1.0.0
Generate ecc key
Signature ==>
0X30 0X46 0X2 0X21 0 0X84 0X69 0X1F 0XFE 0XBC 0XC2 0X2F 0X10 0XBB 0X8D 0X95 0X9A 0X26␣
↪0XD3 0XE2 0X11 0X62 0X81 0XF2 0X7D 0X3 0X5E 0X6B 0X42 0XC8 0X63 0X4F 0XC3 0X50 0XFF␣
↪0XE2 0X3 0X2 0X21 0 0XD6 0XEB 0XD3 0X8D 0X83 0XEB 0XF7 0X6F 0X46 0XF1 0XFA 0XF5␣
↪0XF5 0X24 0XFA 0X26 0X98 0X7A 0X92 0X79 0XBA 0X22 0XAE 0X11 0X1D 0X64 0X8E 0XB0␣
↪0XFD 0X48 0X3C 0XB7
```

3.3 SE05x Mandate SCP03 Example

Overview

This example demonstrates how to mandate the use of Platform SCP by calling SetPlatformSCPRequest.

This is a persistent state.

SetPlatformSCPRequest APDU can be sent in session authenticated with 'RESERVED_ID_PLATFORM_SCP' user id.

The example can be used to either 'Mandate PlatformSCP' or remove the 'Mandate PlatformSCP' state.

When example is built with -DPLUGANDTRUST_SCP03=OFF, ex_se05x_mandate_scp03_set is built. (Set Mandate PlatformSCP).

When example is built with -DPLUGANDTRUST_SCP03=ON, ex_se05x_mandate_scp03_remove is built. (Removes Mandate PlatformSCP).

Refer file - 'simw-nanopkg/examples/se05x_mandate_scp03/src/ex_se05x_mandate_scp03.c'.

Note: When building the example with 'Platform SCP' enabled, make sure to assign valid scp03 keys to session context.

Linux Prerequisite

Install cmake , Openssl 1.1.1

```
sudo apt-get install cmake cmake-curses-gui cmake-gui libssl-dev
```

Linux build

To build example run

```
cd simw-nanopkg/examples/se05x_mandate_scp03/linux
mkdir build
cd build
cmake ../ -DPLUGANDTRUST_SCP03=OFF
make
./ex_se05x_mandate_scp03_set

cd simw-nanopkg/examples/se05x_mandate_scp03/linux
mkdir build
cd build
cmake ../ -DPLUGANDTRUST_SCP03=ON
make
./ex_se05x_mandate_scp03_remove
```

Build options

Platform SCP03

```
-DPLUGANDTRUST_SCP03=ON : Build with Platform SCP03 enabled
-DPLUGANDTRUST_SCP03=OFF : Build with Platform SCP03 disabled
```

Debug Logs

```
-DPLUGANDTRUST_DEBUG_LOGS=ON : Build with Debug logs enabled
-DPLUGANDTRUST_DEBUG_LOGS=OFF : Build with Debug logs disabled
```

Sample Output

If everything is successful, the output will be similar to:

```
./ex_se05x_mandate_scp03_set
Plug and Trust nano package - version: 1.0.0
Sending PlatformSCPRequest_REQUIRED command
Example successful

./ex_se05x_mandate_scp03_remove
Plug and Trust nano package - version: 1.0.0
Establish Secure Channel to SE05x !
Sending PlatformSCPRequest_NOT_REQUIRED command
Example successful
```

3.4 SE05x Rotate SCP03 Example

Overview

This example demonstrates how to update SCP03 keys in SE05x.

On running the example, it will update the SCP03 keys and revert back to original keys.

The example works only with Platform SCP03 enabled.

Refer file - 'simw-nanopkg/examples/se05x_rotate_scp03_keys/src/ex_se05x_rotate_scp03_keys.c'.

Note: When building the example with 'Platform SCP' enabled, make sure to assign valid scp03 keys to session context. All keys are necessary for this operation - enc, mac, dek.

Linux Prerequisite

Install cmake , Openssl 1.1.1

```
sudo apt-get install cmake cmake-curses-gui cmake-gui libssl-dev
```

Linux build

To build example run

```
cd simw-nanopkg/examples/se05x_rotate_scp03_keys/linux
mkdir build
cd build
cmake ../ -DPLUGANDTRUST_SCP03=ON
make
./ex_se05x_rotate_scp03_keys
```

Build options

Platform SCP03

```
-DPLUGANDTRUST_SCP03=ON : Build with Platform SCP03 enabled
-DPLUGANDTRUST_SCP03=OFF : Build with Platform SCP03 disabled
```

Debug Logs

```
-DPLUGANDTRUST_DEBUG_LOGS=ON : Build with Debug logs enabled
-DPLUGANDTRUST_DEBUG_LOGS=OFF : Build with Debug logs disabled
```

Sample Output

If everything is successful, the output will be similar to:

```
./ex_se05x_rotate_scp03_keys
Plug and Trust nano package - version: 1.0.0
Establish Secure Channel to SE05x !
Changing SCP03 keys(version - 0b) to NEW KEYS
Congratulations !!! Key Rotation Successful!
Reverting SCP03 keys(version - 0b) to OLD KEYS
Congratulations !!! Key Rotation Successful!
```

3.5 SE05x SCP03 Resume example

Overview

This example demonstrates SCP03 session resumption with SE05x.

Refer file - 'simw-nanopkg/examples/se05x_resume_scp03/src/ex_resume_scp03.c'.

Note: Make sure to assign valid SCP03 keys to session context.

Linux build

To build example run:

```
cd simw-nanopkg/examples/se05x_resume_scp03/src
mkdir build
cd build
cmake ../ -DPLUGANDTRUST_SCP03
make
./ex_establish_scp03
./ex_resume_scp03
```

Build options

Debug Logs

```
-DPLUGANDTRUST_DEBUG_LOGS=ON : Build with Debug logs enabled
-DPLUGANDTRUST_DEBUG_LOGS=OFF : Build with Debug logs disabled
```

Sample Output

If everything is successful, the output will be similar to:

```
SE05x SCP03 Establish !
Plug and Trust nano package - version: 1.0.0
Establish Secure Channel to SE05x !
Simply writing the session keys to the file system is not a secure implementation. It␣
↪must not be used in production !!!...
SE05x SCP03 Establish Success !

SE05x SCP03 Resume Example !
Plug and Trust nano package - version: 1.0.0
Resuming Secure Channel to SE05x !
Simply writing the session keys to the file system is not a secure implementation. It␣
↪must not be used in production !!!...
SE05x SCP03 Resume Success !
```

CHAPTER

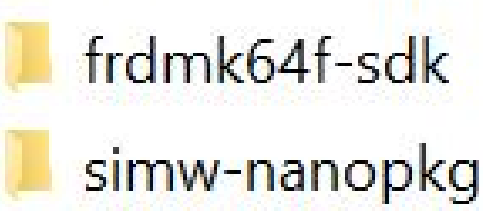
FOUR

GETTING STARTED WITH K64

4.1 SE05x Crypto Example - frdm-k64

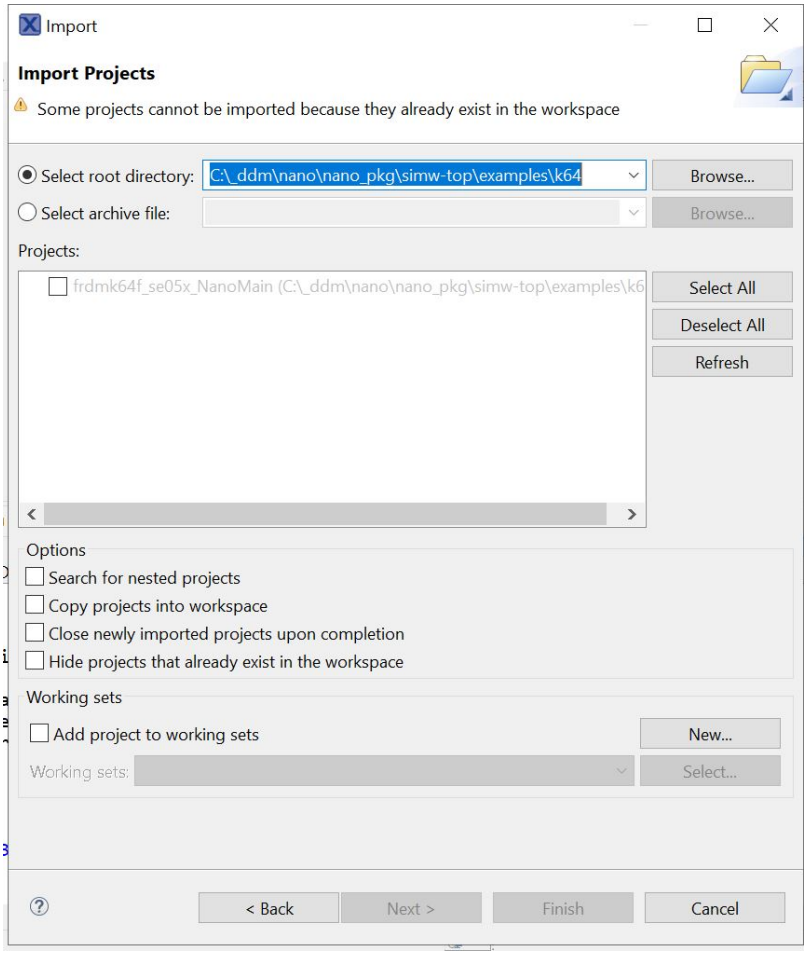
Prequisite

1. Download the frdmk64f SDK from https://mcuxpresso.nxp.com/en/select.
2. unzip and place the sdk in parallel to the nano package as shown in the image below. Rename the sdk folder to "frdmk64f-sdk".



Import the project

1. Click on File, Import, Existing project to workspace and click on next.
2. Point to the "simw-nanopkg/examples/se05x_crypto/k64" folder.
3. Select the Project and click on Finish.



Build and Debug

1. Click on Build and then Debug on the Quickstart panel to Build and Debug your project

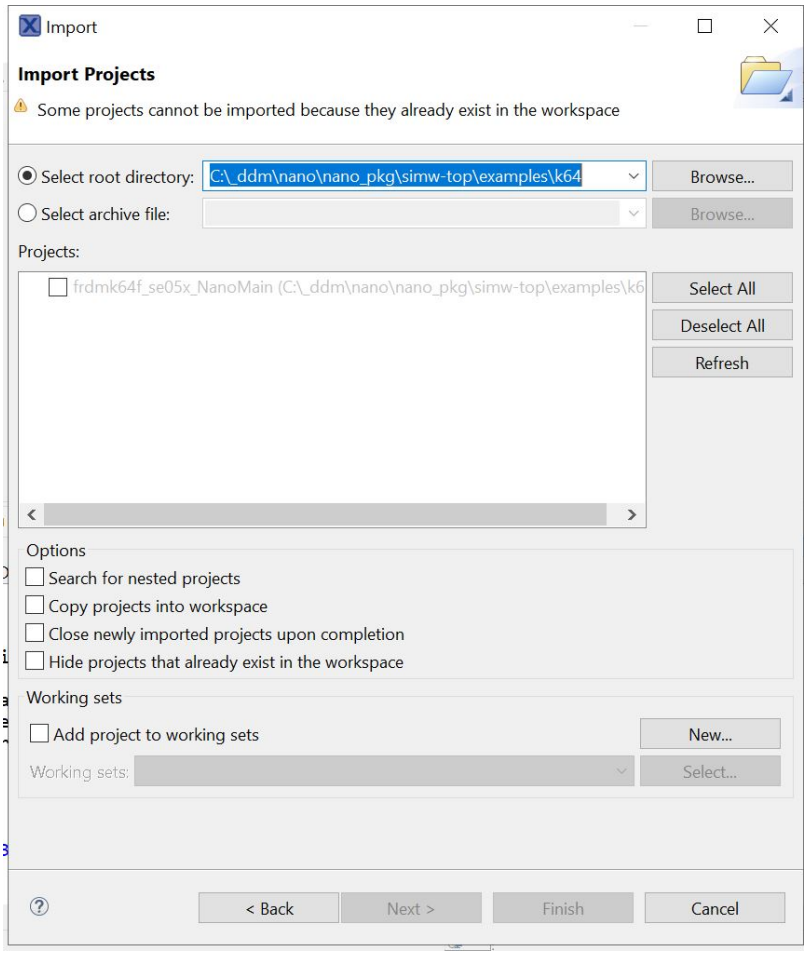
4.2 SE05x Sign Example - frdm-k64

Prequisite

1. Download the frdmk64f SDK from https://mcuxpresso.nxp.com/en/select.
2. unzip and place the sdk in parallel to the nano package as shown in the image below. Rename the sdk folder to "frdmk64f-sdk".

Import the project

1. Click on File, Import, Existing project to workspace and click on next.
2. Point to the “simw-nanopkg/examples/se05x_sign/k64f” folder
3. Select the Project and click on Finish



Build and Debug

1. Click on Build and then Debug on the Quickstart panel to Build and Debug your project

CHAPTER

FIVE

GETTING STARTED WITH ZEPHYR + SE05X

Overview

Plug-and-trust nano package can be used to add the EdgeLock SE05x and A5000 secure elements and authenticators support in Zephyr OS.

Refer 'modules/crypto/nxp-plugandtrust/doc/plug-and-trust-nano-package-api-doc.pdf' for Plug and Trust Crypto APIs.

Zephyr Integration / Build

Clone Plug-and-Trust nano package in Zephyr crypto modules – *<ZEPHYR_PROJECT>/modules/crypto*.

Update west.yml file with Plug-and-Trust module path

```
name: simw-nanopkg
path: modules/crypto/nxp-plugandtrust
revision: -
remote: -
```

Build Options

Use the below options in prj.conf file of the example.

```
CONFIG_PLUGANDTRUST=y/n   ==================> Enable / Disable Plug and Trust lib␣
↪support.
CONFIG_PLUGANDTRUST_SCP03=y/n ==============> Enable / Disable Platform SCP03 support.
CONFIG_PLUGANDTRUST_I2C_PORT_NAME="I2C_0" => Set I2C port used by SE05x on host.
CONFIG_PLUGANDTRUST_LOG_LEVEL_DBG=y/n =====> Enable / Disable Plug and Trust logs.
```

Examples

Build Plug and Trust examples on Zephyr OS as

```
cd <ZEPHYR_PROJECT>/zephyr
west build -b <BOARD> ../modules/crypto/nxp-plugandtrust/examples/<EXAMPLE_NAME>/
↪zephyr/ --pristine
```

Note: Currently examples are tested with Frdm-k64 board.