G3 를 이용한 TLS 환경 셋팅 및 실행 가이드

# 개요

라즈베리파이 환경에서 IEVB-100을 사용하여 G3와 통신하며 TLS demo를 수행하고자 할 때, G3\_API를 통한 IEVB-100의 준비 과정과 이후의 TLS demo를 수행하기까지의 과정을 설명한다.

# 프로젝트 설치

## 다운로드 위치:

<https://github.com/ictk/tls_wolfssl_with_g3>

## 설치 방법 :

1. 깃 클론.   
   $ git clone <https://github.com/ictk/tls_wolfssl_with_g3.git>
2. 해당 사이트에 들어가서 ‘Clone or download’ 클릭후 ‘Download ZIP’ 버튼을 눌러서 zip 파일로 받은 후 압축을 푼다.

# 빌드 방법

## 라이브러리 빌드

1. 해당 디렉토리로 이동 (cd $(workspace)/libneo\_wolf\_ssl)
2. >cd libneo\_wolf\_ssl
3. Makefile 안에 PLATFORM=”플랫폼 이름”(gnu-armhf, gnu-x86, gnu-x64)  
   라즈베리 파이는 gnu-armhf, 을 설정한다.
4. >make
5. $(workspace)/example/reflib/ 밑에 libneo\_wolf\_ssl.so 생성 확인

## 실행 프로그램 빌드

1. 해당 디렉토리 로 이동 (cd $(workspace)/console\_neo\_wolf\_ssl)
2. Makefile 안에 PLATFORM=”플랫폼 이름”(gnu-armhf, gnu-x86, gnu-x64)  
   라즈베리 파이는 gnu-armhf, 을 설정한다.
3. >make
4. $(workspace)/bin/neo\_wolf\_ssl 생성 확인

## tls setup 빌드

1. 해당 디렉토리 로 이동 (cd $(workspace) /etc/tls\_setup)
2. Makefile 안에 PLATFORM=”플랫폼 이름”(gnu-armhf, gnu-x86, gnu-x64)  
   라즈베리 파이는 gnu-armhf, 을 설정한다.
3. >make
4. $(workspace)/bin/tls\_setup 생성 확인

# 라이브러리 구조

## 라이브러리 설명

|  |  |
| --- | --- |
| 모듈이름 | 설명 |
| libneo\_wolf\_ssl.so | tls 주 기능 |
| libg3\_api\_lib.so | g3 핸들링 |
| libg3\_io\_lib.so | g3 의 실제 io 제어 |
| libneo\_c\_lib.so | 기타 유틸 uart 제어 |
| libft4222.so | ievb-100 제어 |
| libwiringPi.so | 라즈베리파이 네 i2c 및 gpio 이용 |
| libcrypt.so.1 | openssl 암호 모듈 사용 |

# IEVB-100 설치

## FT4222 라이브러리 설치

라즈베리파이 환경에서 IEVB-100 보드를 통해 G3와 통신하여 g3\_api의 실행 및 TLS DEMO 수행을 위해서는 별도의 FT4222 library의 설치가 필요.

|  |
| --- |
| >cd $(workspace)/tc/ft4222  >sudo ./install4222.sh |

## rules file 설치

1. ft4222 장치 연결 시 root 권한 없어도 접근 수행 허용

|  |
| --- |
| > sudo cp $(workspace)/install/50-ft422.rules /etc/udev/rules.d/ |

1. 장치를 제거 후 다시 연결한다.

## 디바이스 연결

해당 디바이스를 본체 단말기와 연결한다.

다음 과 같이 연결 여부를 확인 한다.

|  |
| --- |
| >lsusb  Bus 001 Device 005: ID 0403:601c Future Technology Devices International, Ltd  Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter  Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.  Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub |

# Provisioning

TLS demo를 수행하기 위해선 demo 시나리오에 맞는 provisioning 과정이 필요하다. 이는 demo scenario에 따라 달라질 수 있으며, Key 및 Certificate 주입과 사용 환경 구성 등의 과정을 포함한다.

## 셋업 영역

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TITLE | Sector | TYPE | AND/OR | AC COND | AC[0~6] |
| ROOT |  | VOID | OR|OR|OR | COND|COND|COND | 3,1,3,1,0,0 |
| DATA\_0 |  | VOID | OR|OR|OR | FREE|FREE|FREE | 3,1,3,1,0,0 |
| DATA\_1 |  | VOID | AND|AND|AND | FREE|FREE|FREE | 0,0,0,0,0,0 |
| PUB MANAGE | 1 | ECC\_PUB | AND|AND|AND | FREE|FREE|FREE | 1,3,1,3,1,3 |
|  | 2 | VOID | AND|AND|AND | FREE|FREE|FREE | 0,0,0,0,0,0 |
| PASSWORD | 3 | Password | OR|OR|OR | FREE|FREE|FREE | 1,3,1,3,0,0 |
| PRV KEY | 4 | ECC\_PRV | OR|OR|OR | COND|COND|COND | 1,3,1,3,1,3 |
| PUF | 5 | ECC\_PUF | OR|OR|OR | COND|COND|COND | 1,3,1,3,1,3 |
| PUB CA | 6 | ECC\_PUB | OR|OR|OR | COND|COND|COND | 1,3,1,3,0,0 |
|  | 7 | VOID | AND|AND|AND | FREE|FREE|FREE | 0,0,0,0,0,0 |
| DEVICE PUB KEY | 8 | ECC\_PUB | OR|OR|OR | COND|COND|COND | 1,3,1,3,0,0 |
|  | 9 | VOID | AND|AND|AND | FREE|FREE|FREE | 0,0,0,0,0,0 |
| DEVICE PRV KEY | 10 | ECC\_PRV | OR|OR|OR | COND|COND|COND | 1,3,1,3,1,3 |
| AES128 KEY | 11 | AES128 | OR|OR|OR | COND|COND|COND | 1,3,1,3,1,3 |
| SHA256 KEY | 12 | SHA256 | OR|OR|OR | COND|COND|COND | 1,3,1,3,1,3 |

## 데이터 영역 및 실제 키

키 영역에는 해당 키 등을 입력 한다. 각 키에 해당하는 값들은 아래와 같다.

|  |
| --- |
| ##################################################  CA CERT  ##################################################    ##################################################  CA KEY  ##################################################  PRV KEY:  CD26689A25B034C9165A3527FEC74F2A4133A946243C63103884F7CAA06CA5F1  PUB KEY:  D57CBAB7682AE97681C72FAB34AF2873CDBB49E442F85EFEA4DE55C51ED093EE7E1AD8C3E4025E80F9C4F866B26000D5C8E31C838070D9C0A18FF75AA0EC9B4B  ##################################################  SERVER CERT  ##################################################    ##################################################  SERVER KEY  ##################################################  PRV KEY:  59FBE7E12BDC99DD451446473D3B91A5C7B7D475839A76C6481D013CA887D346  PUB KEY:  557C3311C1CAB0782A690E8DEA04FD7504CEE338666D8942192A41E06A77035D7DB3704B0353C38A8EAA7D3CD0AB0E629A662EDA06B3BB8E11B4221ADE24FA5A  ##################################################  \*CLIENT CERT  ##################################################    ##################################################  CLIENT KEY  ##################################################  PRV KEY:  B4083C7036419F8CB7BA8368313F83F07DEFF63EDC87AEB9C03E8EA211F29A2B  PUB KEY:  4A5B42C20E614F27DB01E7AC348BE234F9CB0E79BA3C08C7415A22F8C9EA7A3D24B9E008B8C7F37C775BA611C09236A7315C50BB1F5D1A64C8548CFCF5BDF90F |

## 설정 방법

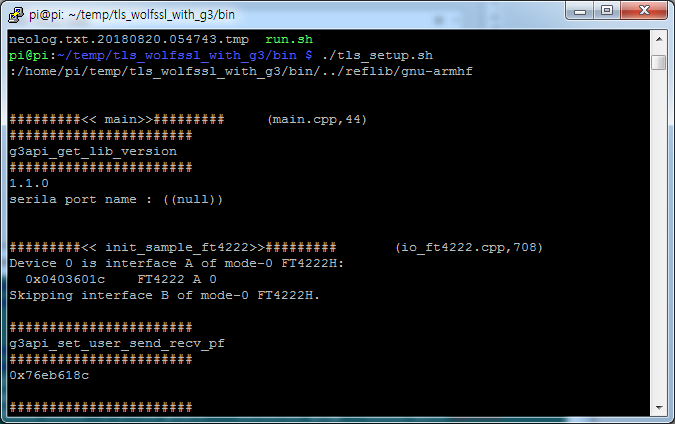
1. sector에 접근하기 위한 권한 설정
2. Device cert & CA cert 저장 ( Data 1 area )
3. 두 Cert는 DATA\_AREA\_1에 저장되어 있으며 각 Cert가 시작되는 Sector number와 Cert의 크기는 Cert Header를 통해 명시해 주어야 함. Cert Header는 Data\_area\_1의 sector 0에 지정되어 저장됨. 4byte로 이루어지며 index, sector number, Size 값으로 구성.
4. 각 cert를 verify하기 위한 public key 저장
5. 각 key 값이 저장된 sector에 key 저장
6. root에 대한 권한 설정

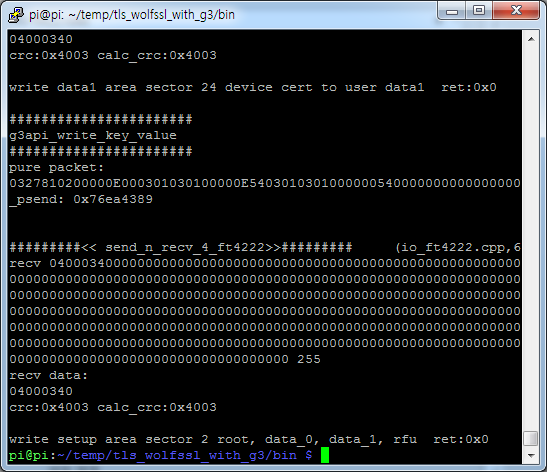
## Provisioning 실행

1. tls\_setup.sh 실행

|  |
| --- |
| ~>cd $(workspace)/in  ~$./tls\_setup.sh |

1. 정상 동작시 다음과 같은 화면이 나온다.





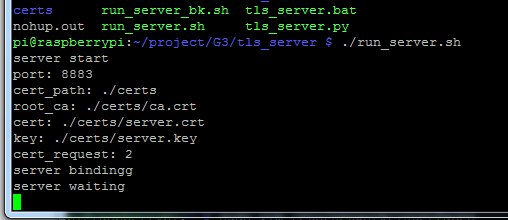
# TLS DEMO

## 서버 실행

1. 다음 명령을 실행

|  |
| --- |
| ~>cd $(workspace)/etc/tls\_server  ~>./run\_server\_con.sh (port 8883 으로 실행 하고 certs 폴더에 위에 언급한 인증서와 키를 설정한다.) |

1. 서버가 정상 동작시 화면

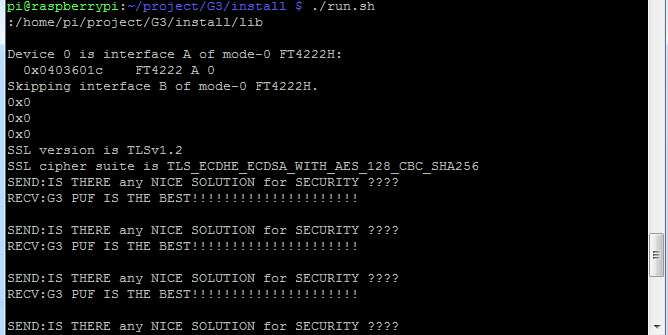


## 클라이언트 실행

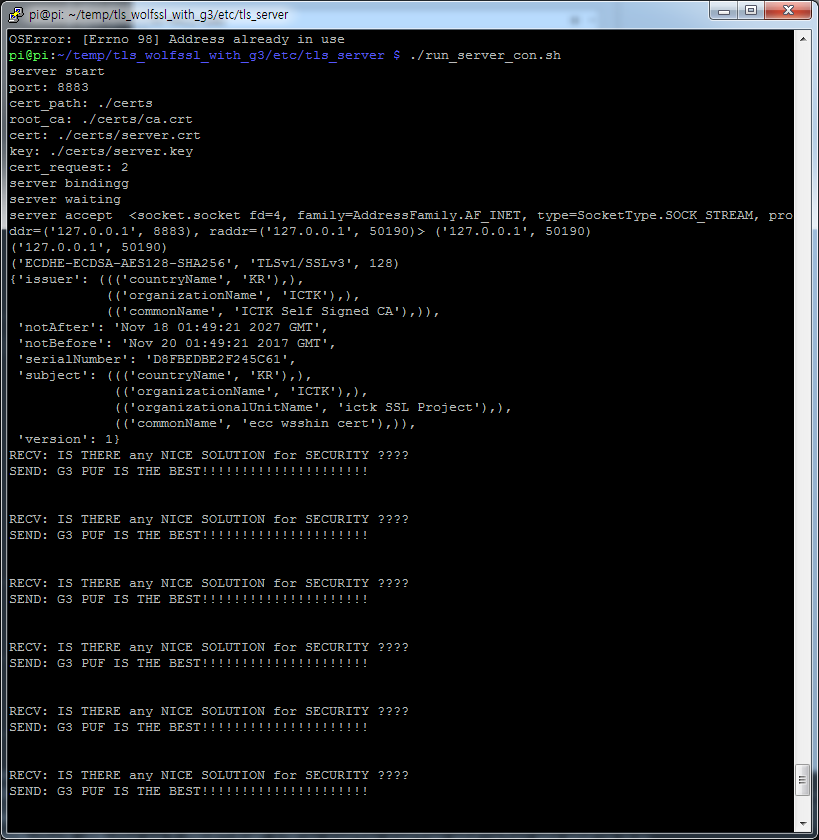
1. IEVB-100을 연결 한다.
2. 실행

|  |
| --- |
| ~>cd $(workspace)/bin  ~>./run.sh |

1. 실행 화면



1. 정상 동작시 서버 화면



# 임의 Certificate 생성 및 이를 통한 DEMO 진행

* 현 데모에서는 미리 만들어저 지정해 둔 키와 인증서를 tls\_setup을 통해서 주입되어 Scenario를 진행한다.
* CA 인증서와 키는 자체적으로 발생 했고 서버인증서와 디바이스 인증서를 서명 했다.
* Certificate는 외부에서 제작하여도 무관하나 이럴 경우는 G3칩을 바탕으로 CSR 을 만들어줘야 한다.
* 현재 시나리오는 디바이스 키를 키 영역에 입력 하는 방식으로 사용되었다. 만약 PUF key를 사용할 경우는 퍼블릭 키를 Tbs에 넣어서 ca를 통해 서명을 받아야 한다.