SAMSUNG Research

Kick off
Tizen Project

Aug. 22, 2018

SAMSUNG Research

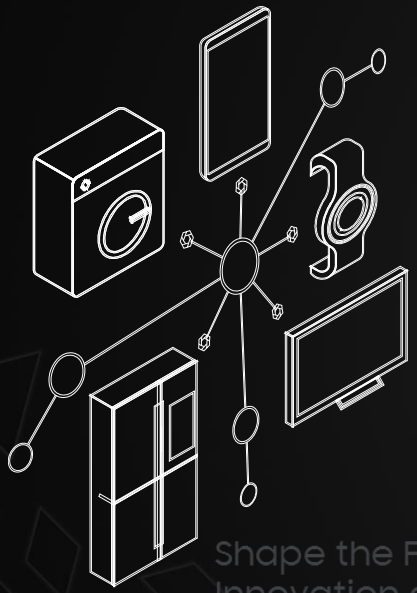
SAMSUNG Research

천리 길도 한걸음부터

걷기도 전에 뛰려고 한다

등고자비

SAMSUNG Research



Shape the Future with
Innovation and Intelligence

Contents

Ⅰ Shape of you
(Tizen Project 외부 구조)

Ⅱ Just the way you are
(Tizen Project 동작 방식)

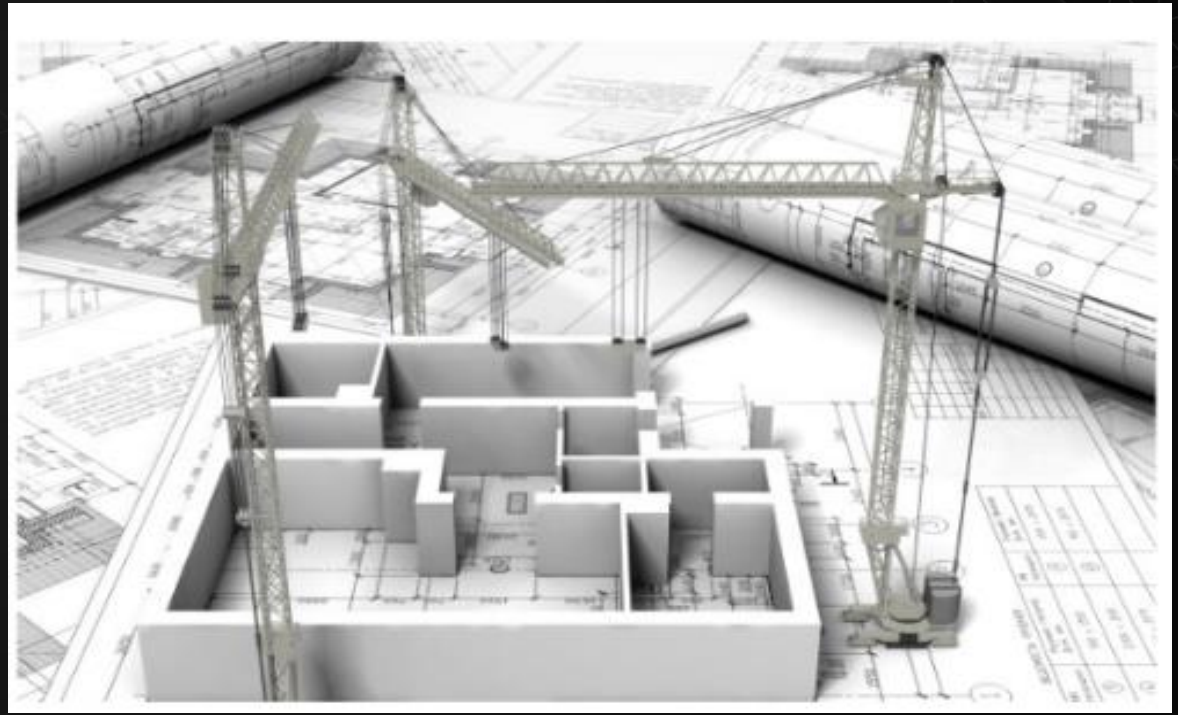
Ⅲ Handclap
(Tizen Project 유용한 개발 Tip)

Tizen Project 외부 구조

Tizen Project 생성하기

Do it my self

- 폴더 구조
- 빌드 환경
- 패키징을 위한 링크



Tizen Project 외부 구조

Tizen Project 생성하기

Interior Tizen Project

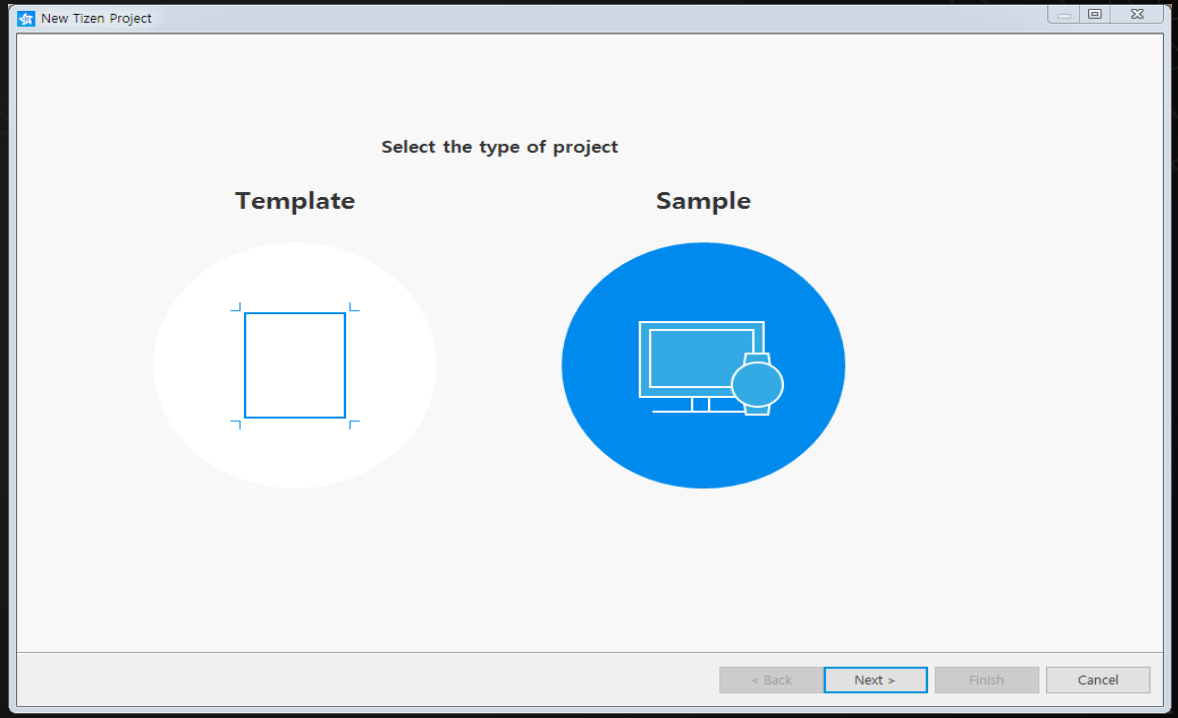
- Templates
- Samples
- 기존 Tizen Project

Tizen Project 외부 구조

Tizen Project 생성하기

Templates & Samples

- 좌측 상단 +아이콘
- Project Explorer 창 가운데 + 아이콘
- File -> New -> Tizen Project

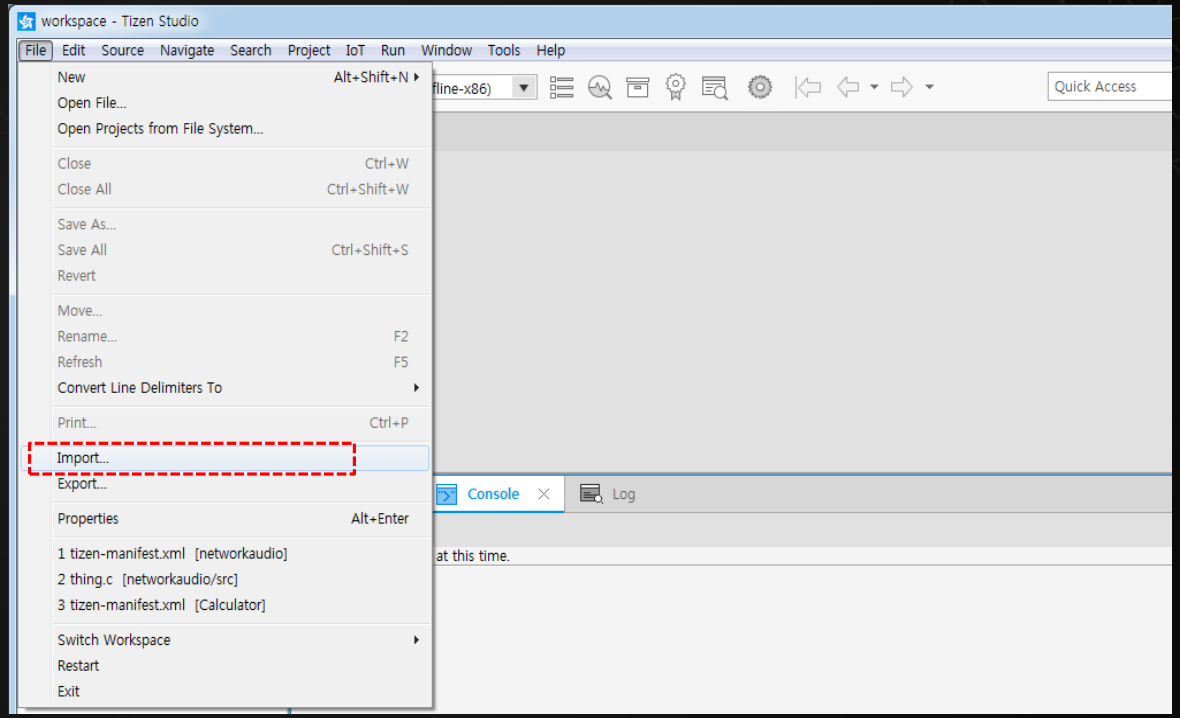


Tizen Project 외부 구조

Tizen Project 생성하기

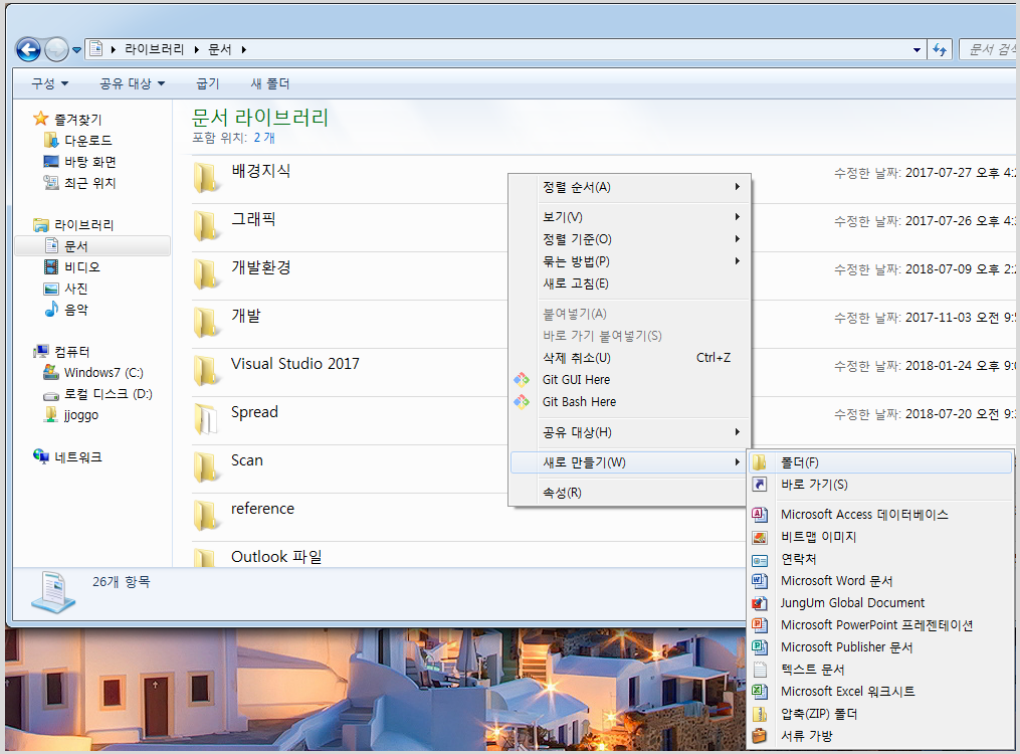
Existent Tizen Project

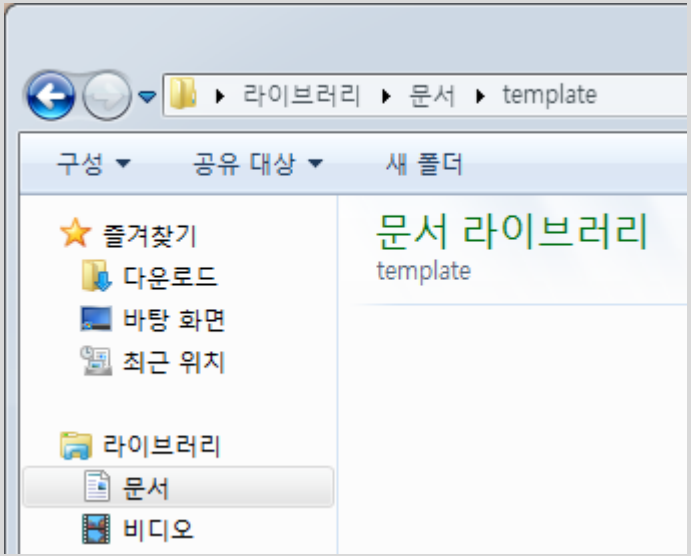
- File -> Import
- Git 으로부터 다양한 Tizen Project 들을 가져 올 수 있습니다



Tizen Project 외부 구조

Tizen Project 생성하기



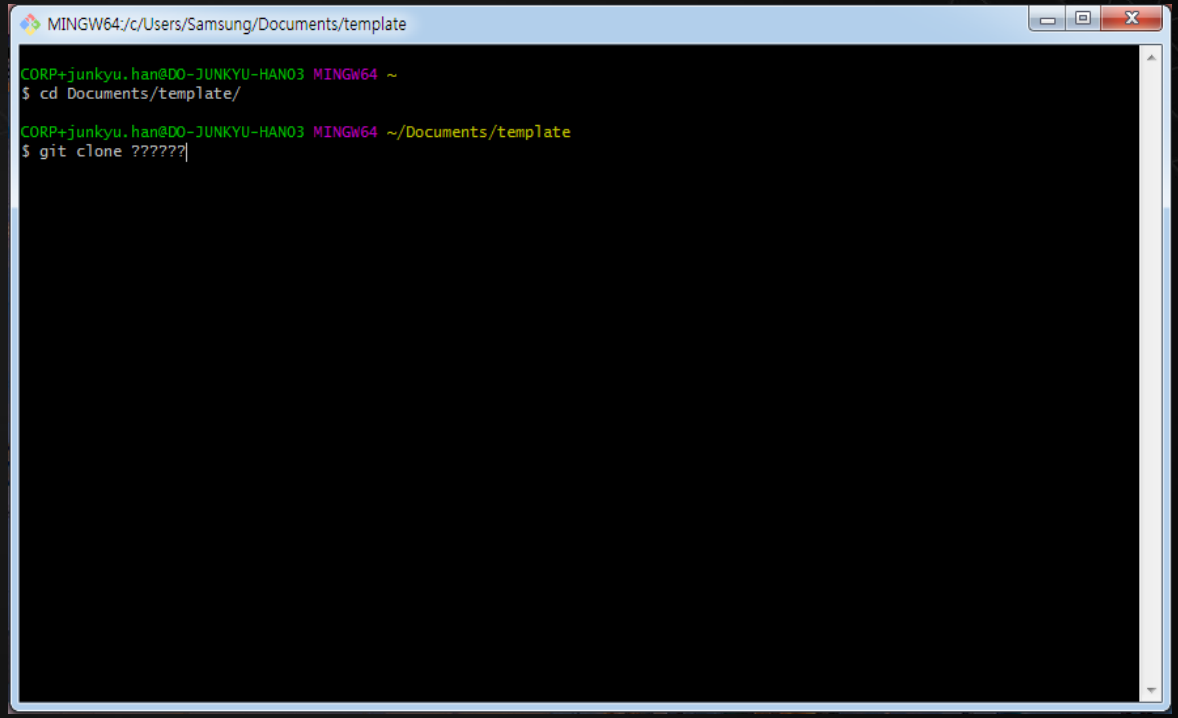


Tizen Project 외부 구조

Tizen Project 생성하기

Git 으로부터 가져오기

- Git bash
- Tizen Project 를 가져올 폴더로 이동해줍니다
- 폴더를 생성하는 명령어는 mkdir 입니다
- git clone 명령어 뒤에 적어줄 project 의 주소는 어떻게 알 수 있을까요?

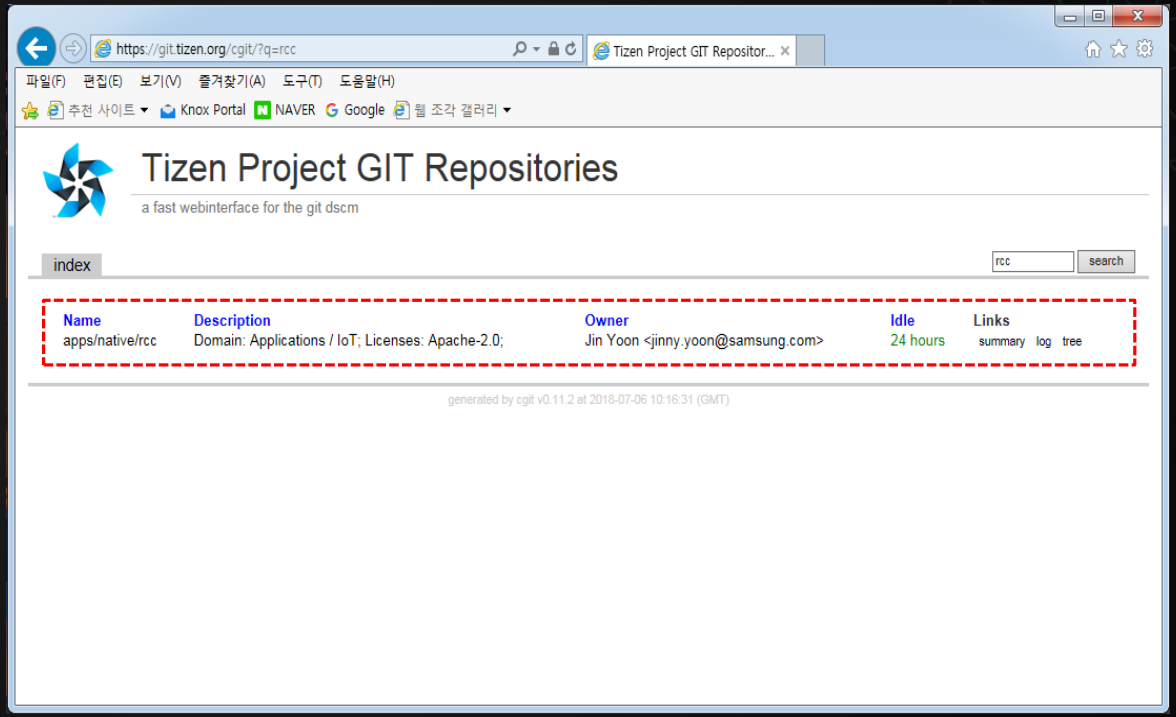


Tizen Project 외부 구조

- Tizen Project 생성하기

Git 으로부터 가져오기

- https://git.tizen.org
- 모든 profile 의 Tizen Project 들이 보입니다
- 스마트하게 search 를 사용해요
- 'rcc' 를 찾아줍니다
- apps/native/rcc 라는 아이가 보여요
- 문제가 있으면 Owner에게....

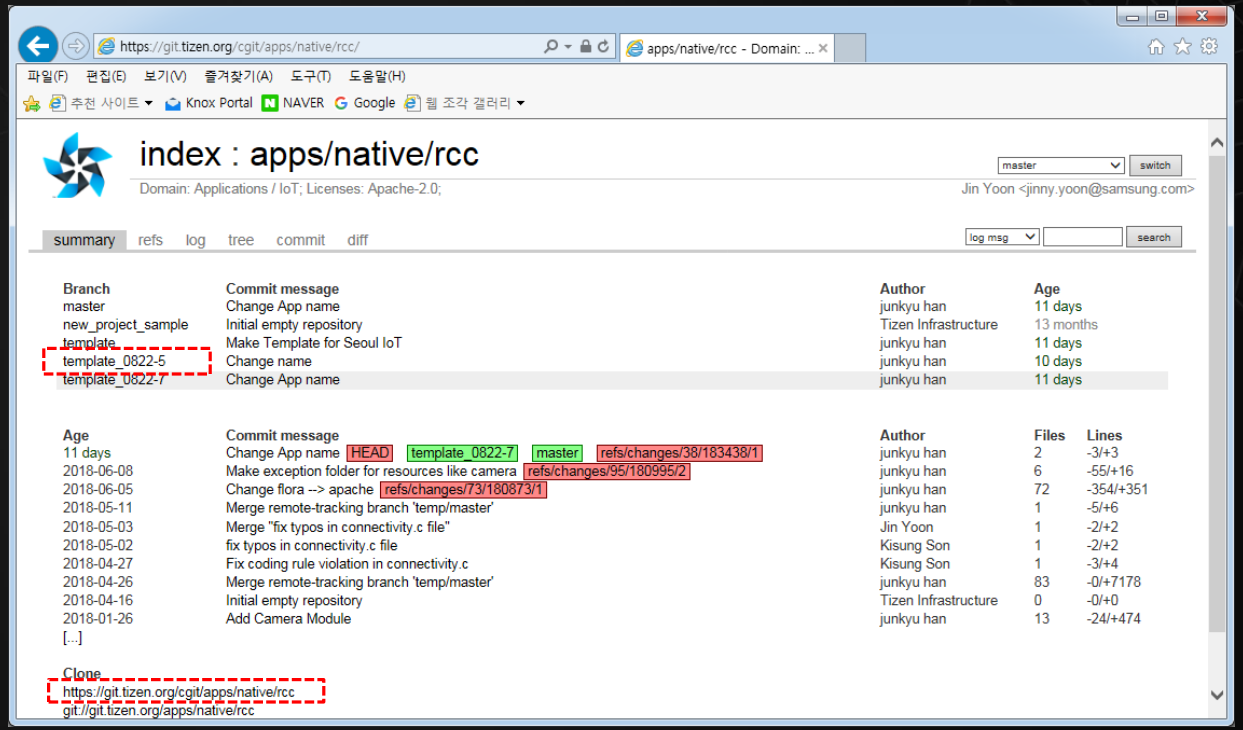


Tizen Project 외부 구조

- Tizen Project 생성하기

Git 으로부터 가져오기

- 프로젝트 rcc 에 대한 많은 정보들을 볼 수 있습니다
- Branch 와 Clone 에 집중하세요
- template_0822-5 라는 branch 만 가져오겠습니다
- Clone 에서는 https 를 사용할게요

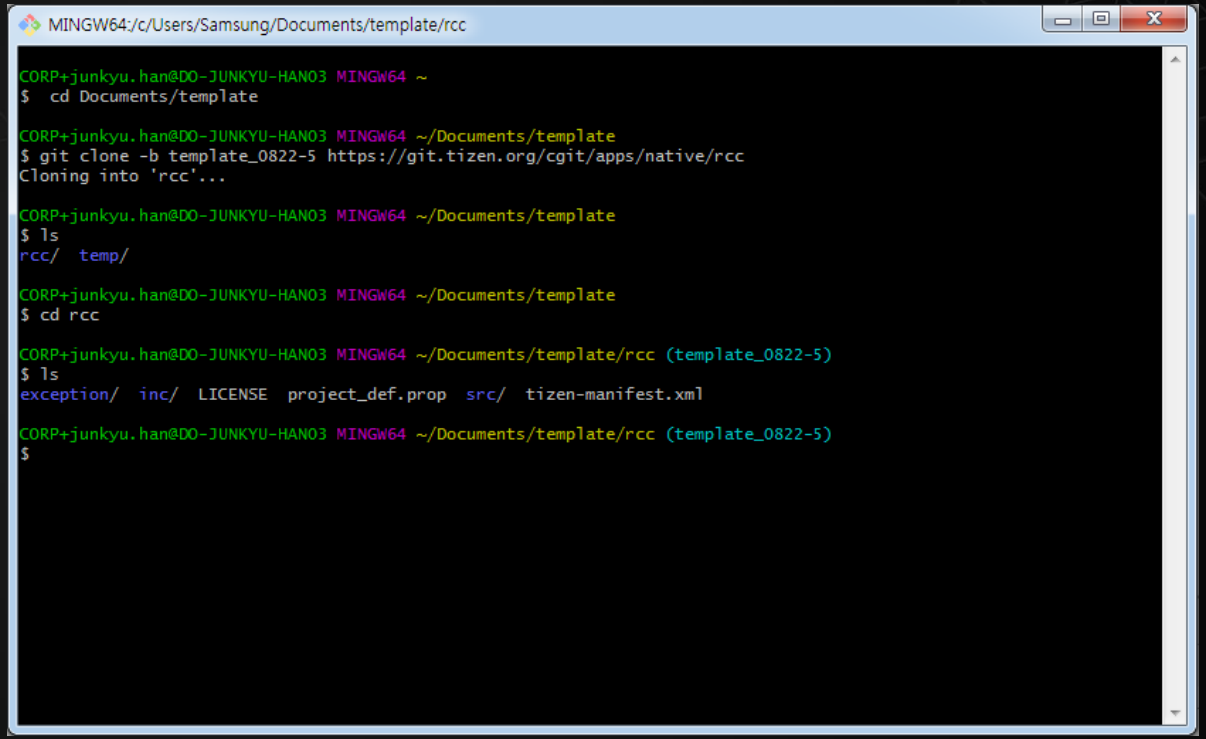


Tizen Project 외부 구조

Tizen Project 생성하기

Git 으로부터 가져오기

- git clone -b template_0822-5 https://git.tizen.org/cgit/apps/native/rcc
- 현재 폴더에 rcc/ 폴더가 생성되었음을 확인합니다
- rcc 프로젝트로 이동해 볼게요
- 무언가 들어있는 빈 폴더가 아님을 확인해줍니다

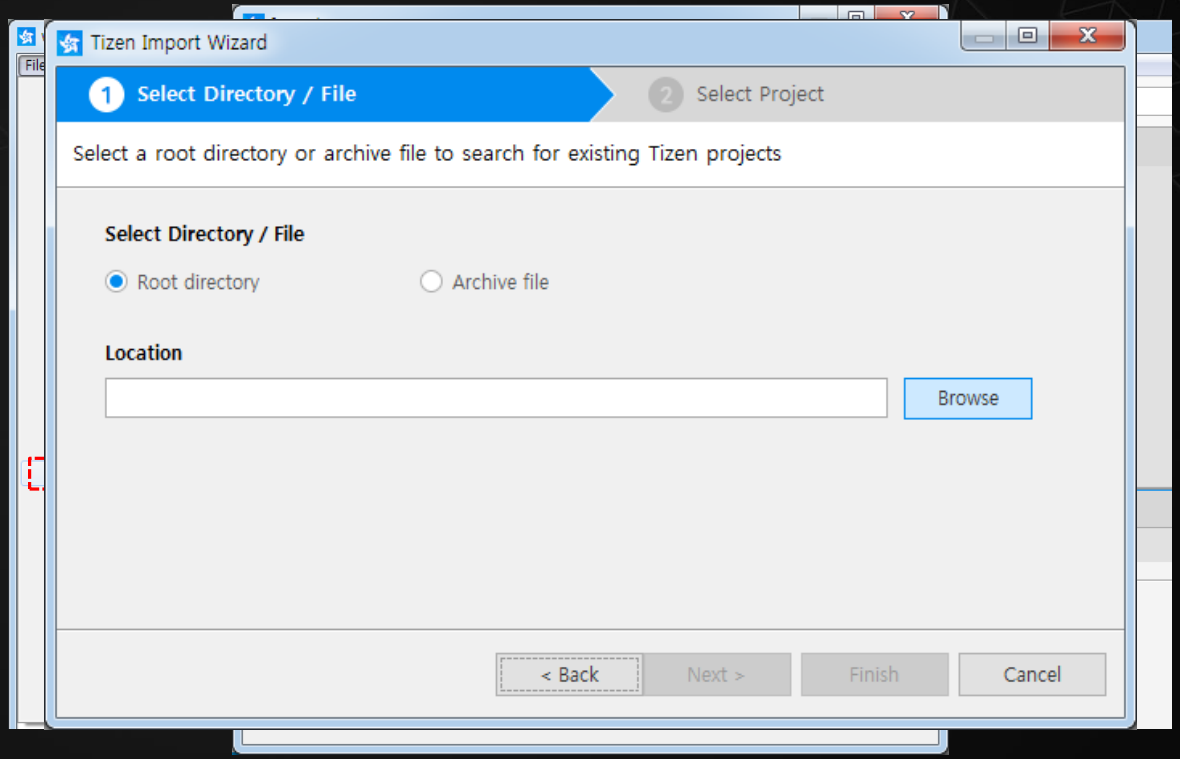


Tizen Project 외부 구조

Tizen Project 생성하기

rcc 프로젝트 가져오기

- File -> Import
- Tizen -> Tizen Project
- Root directory -> Browse

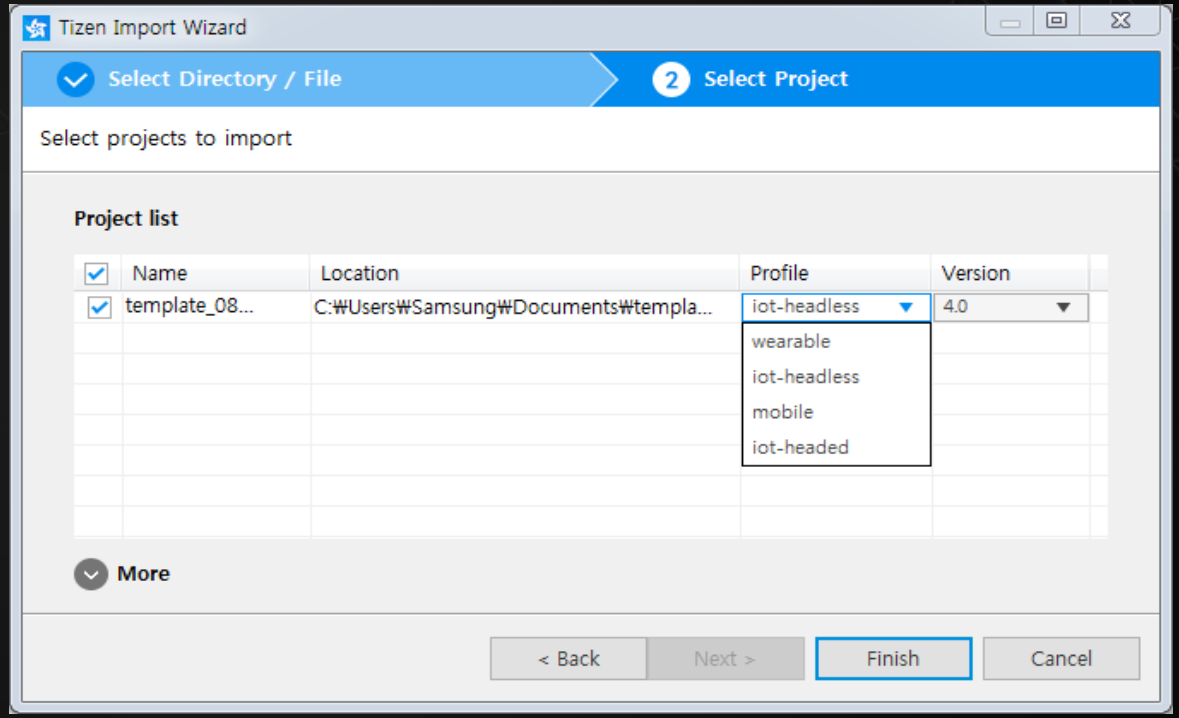


Tizen Project 외부 구조

Tizen Project 생성하기

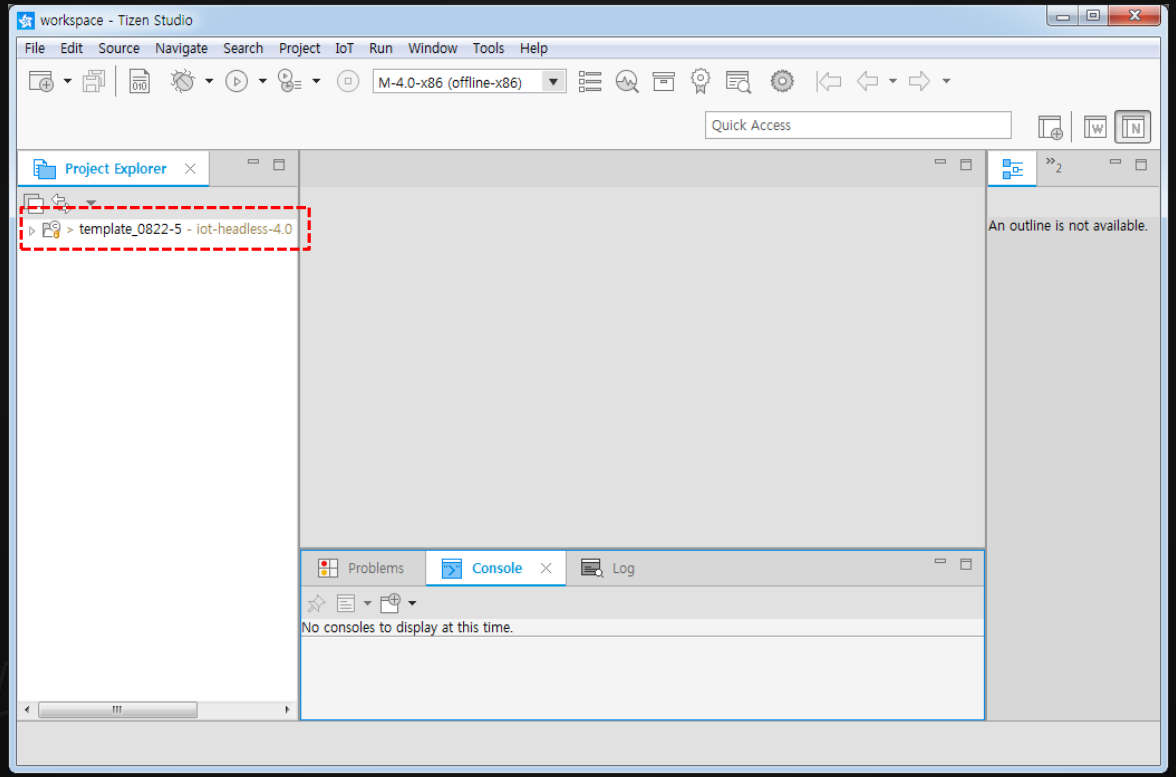
rcc 프로젝트 가져오기

- rcc 프로젝트를 저장해 둔 위치로 가서 rcc 폴더를 선택합니다
- 위치를 다시 한번 확인하고 Next 를 눌러주세요
- 해당 프로젝트가 다양한 profile 을 지원한다면 원하는 profile 을 선택할 수 있습니다



Tizen Project 외부 구조

- Project ‘template_0822-5’ 가 뙇!

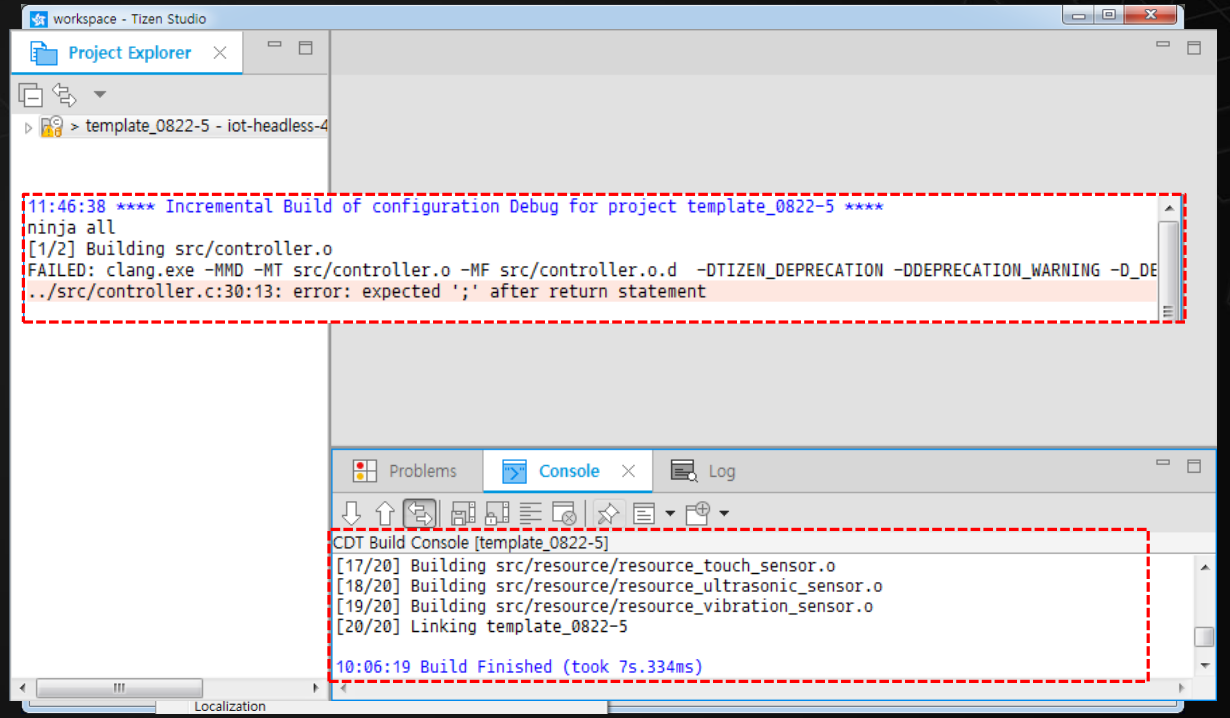


Tizen Project 외부 구조

Tizen Project 폴더 구조

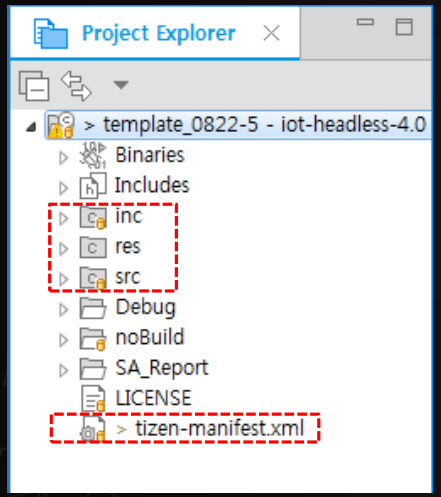
빌드로 유효성 확인

- 프로젝트 이름 우 클릭 -> Build Project
- 하단 Console 창에서 빌드 프로세스 로그 확인 가능합니다
- FAILED 메시지 없이 Build Finished 가 보이면 해당 프로젝트가 성공적으로 빌드 된 것입니다



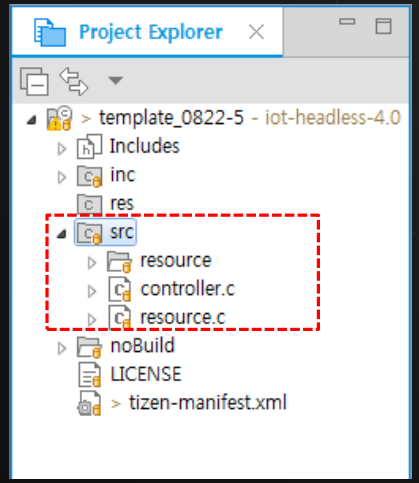
Tizen Project 외부 구조

- Tizen Project 폴더 구조



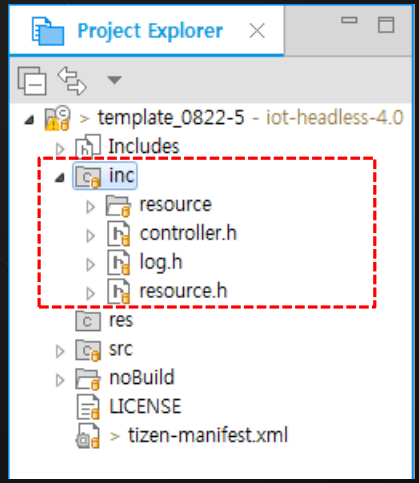
src/

- Source file
- *.c



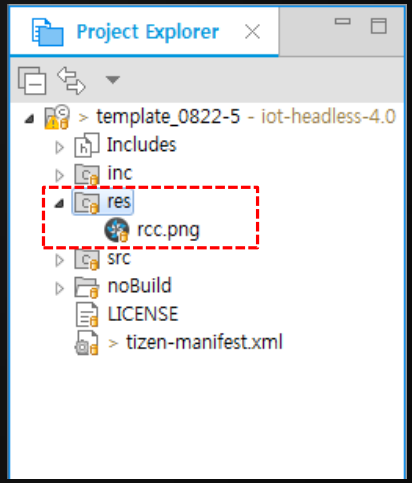
inc/

- Header file
- *.h



res/

- Resource file
- Mp3, Image etc..

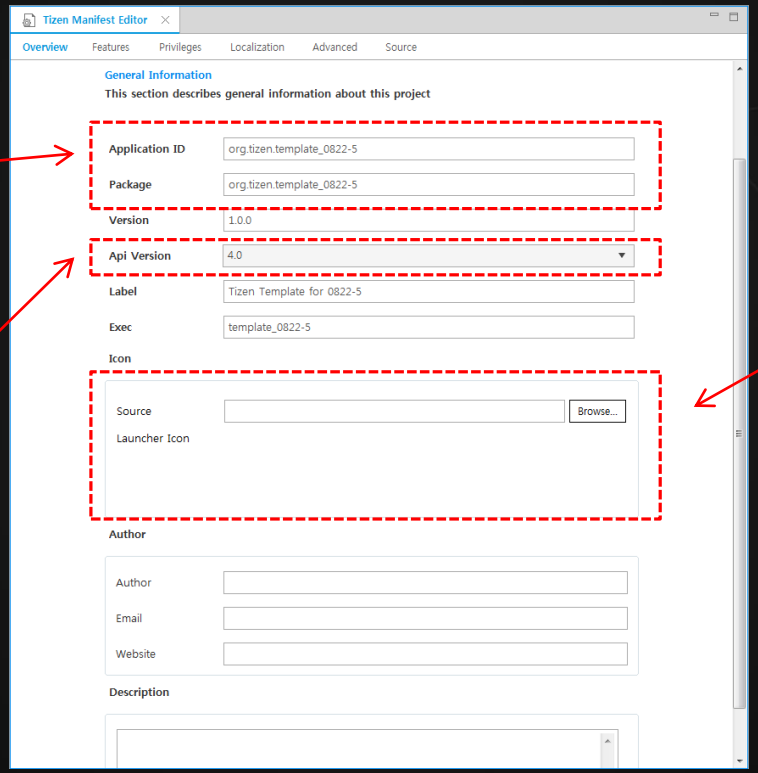


Tizen Project 외부 구조

Tizen Project 특성 설정 – Overview 탭

Application ID 와 Package 이름은 변경 가능하지만!!
Application ID 는 반드시 Package 명 + '.' + 원하는 이름으로 이루어져야 합니다.

API version 은 변경 가능하더라도 매우 신중해주세요
자칫 호출에 실패할 수도 있습니다

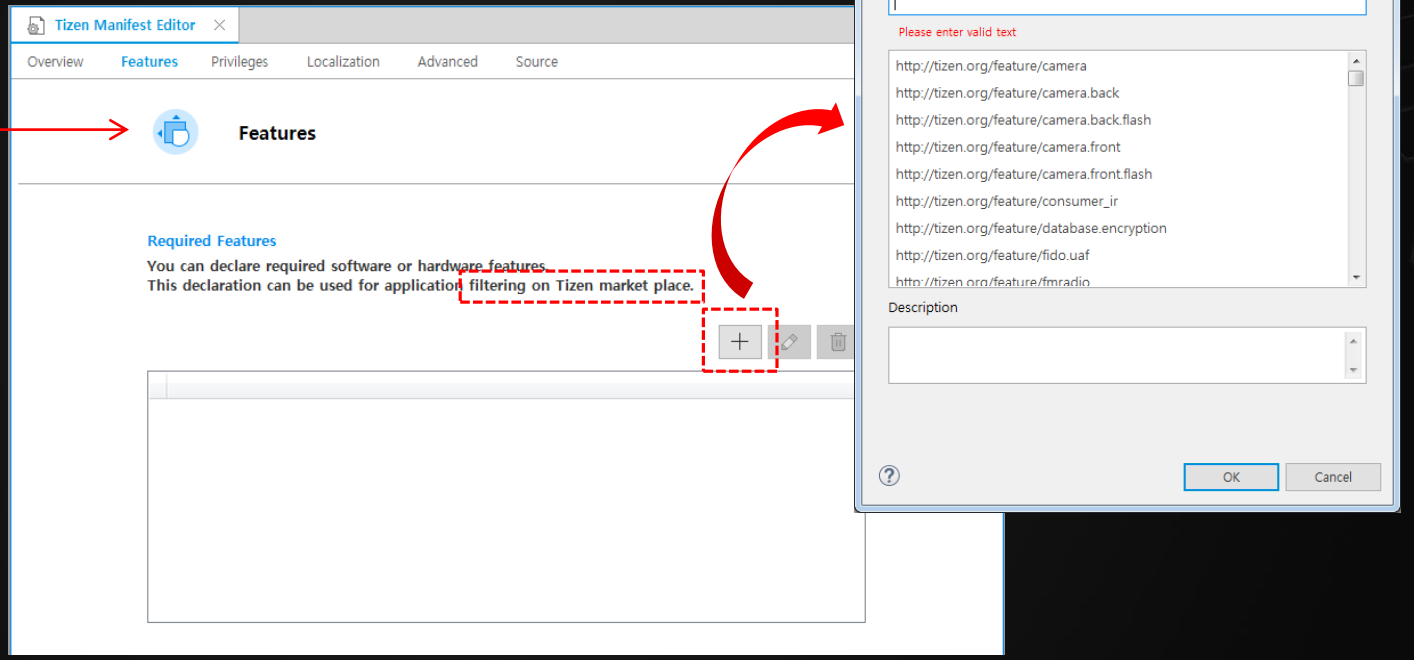


Icon 도 여기서 설정하실 수 있습니다

Tizen Project 외부 구조

Tizen Project 특성 – Features 탭

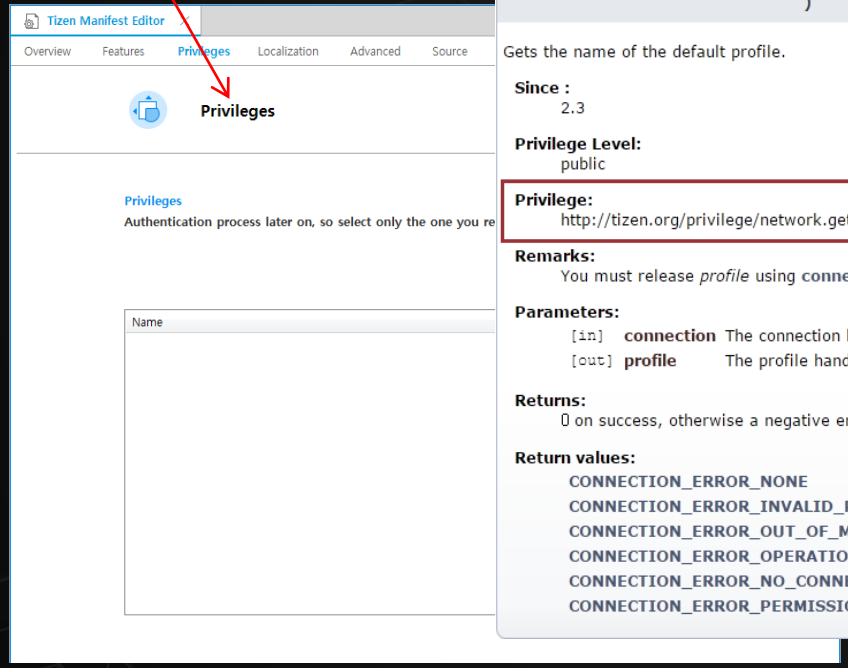
Feature 의 경우 market 에 올릴 때 어느 카테고리로 속하게 할 것인지를 정해주는 것입니다

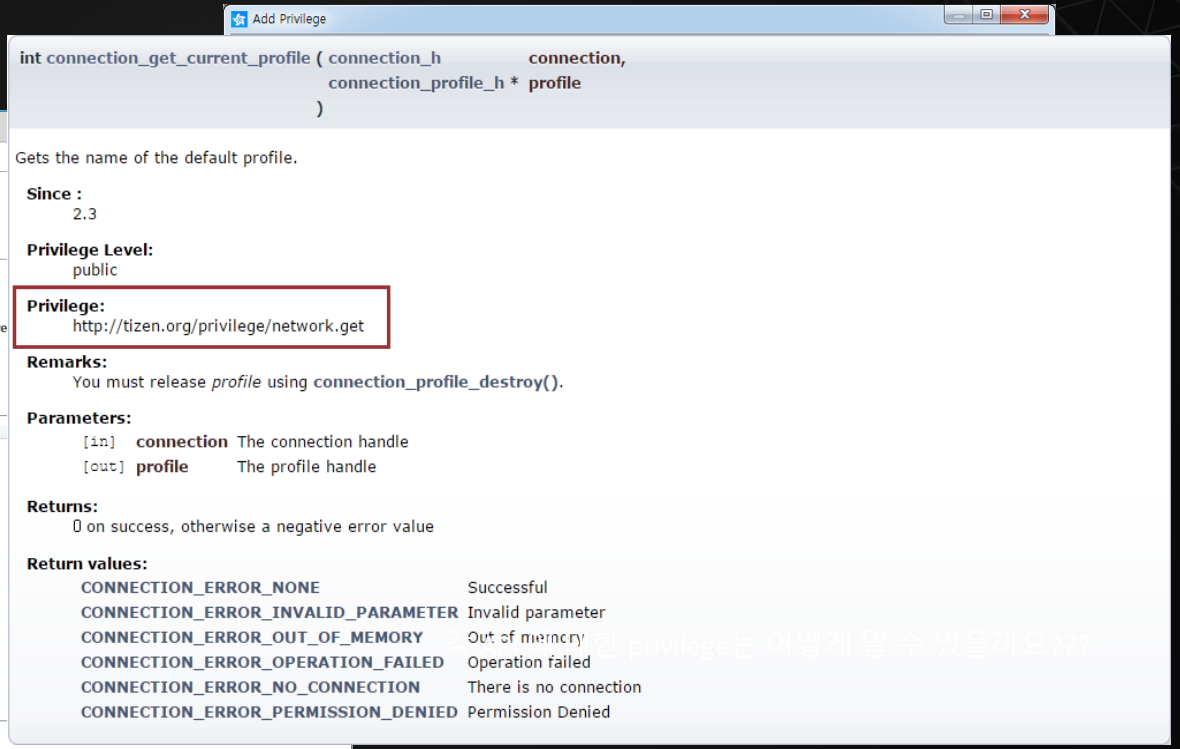


Tizen Project 외부 구조

Tizen Project 특성 설정 – Privilege 탭

Runtime 시 API 호출에 대한 권한을 부여해줍니다



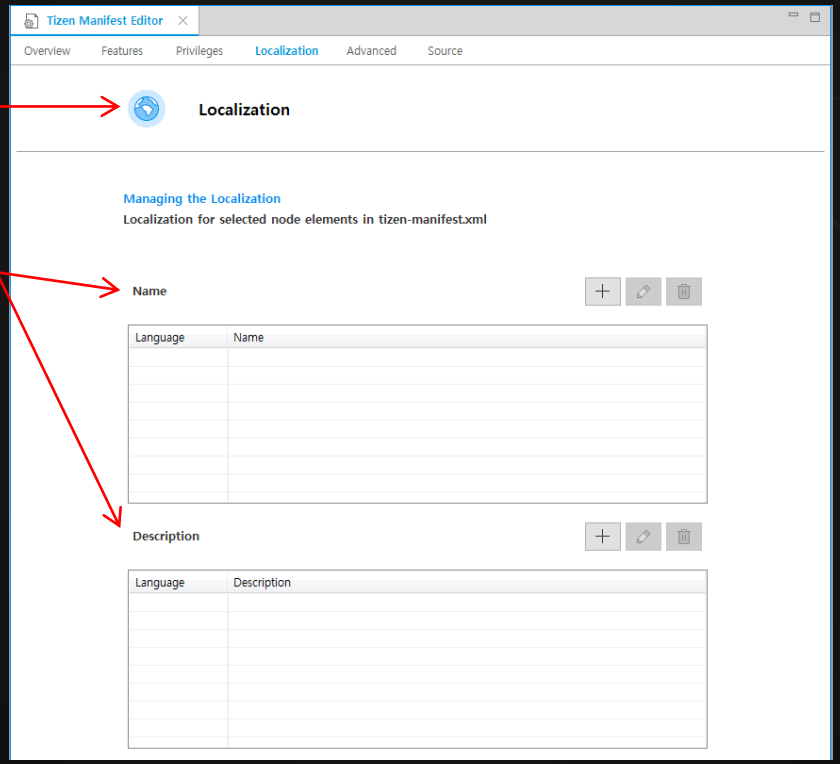


Tizen Project 외부 구조

Tizen Project 특성 – Localization 탭

디스플레이 되는 텍스트의 언어를 추가할 수 있습니다

해당 언어의 이름과, 그에 대한 설명을 추가 합니다



IoT Headless 의 경우 디스플레이 되는 것은 없지만, 로그 또한 원하는 언어로 남길 수 있습니다

A to Z of Tizen Project

구조 of Tizen Project

key-value 쌍으로 지정 해 둔 값은 언제 어느 때고 meta data API 로 가져와서 사용할 수 있습니다

App 간 data share 를 가능하게 하는 부분입니다

집중해 주세요 여긴 좀 중요합니다

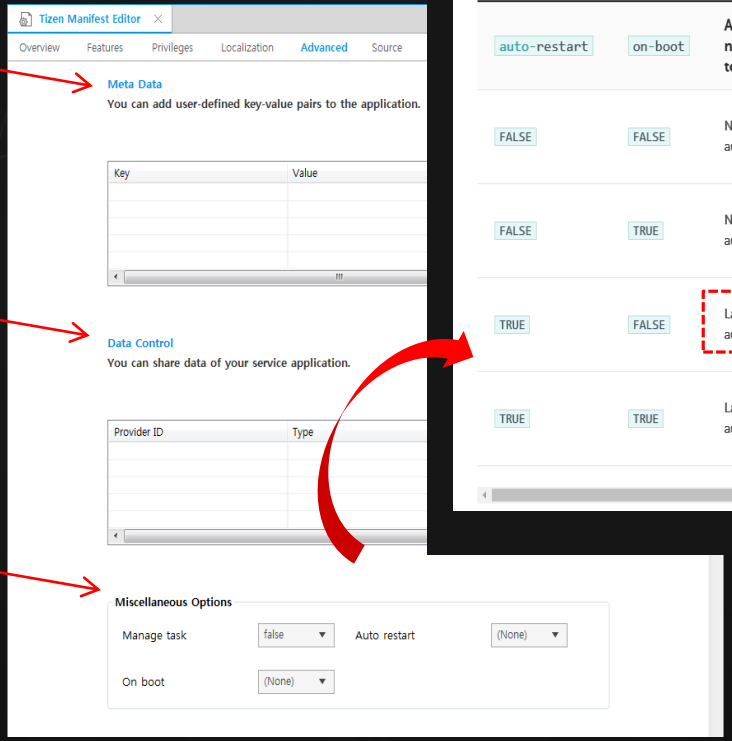


Table: Attribute combinations

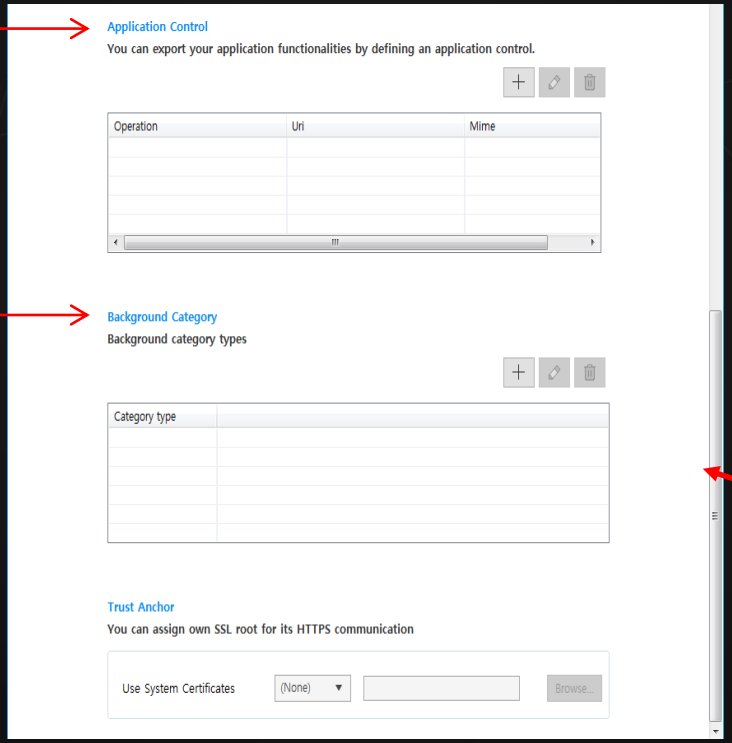
auto-restart	on-boot	After normal termination	On forced close	On Reboot	After package installation	After package update
FALSE	FALSE	Not launched automatically	Not launched automatically	Not launched after reboot	Not launched	Not launched automatically
FALSE	TRUE	Not launched automatically	Not launched automatically	Launched automatically after reboot	Launched	Launched automatically
TRUE	FALSE	Launched automatically	Launched automatically	Not launched after reboot	Not launched	Launched automatically
TRUE	TRUE	Launched automatically	Launched automatically	Launched automatically after reboot	Launched	Launched automatically

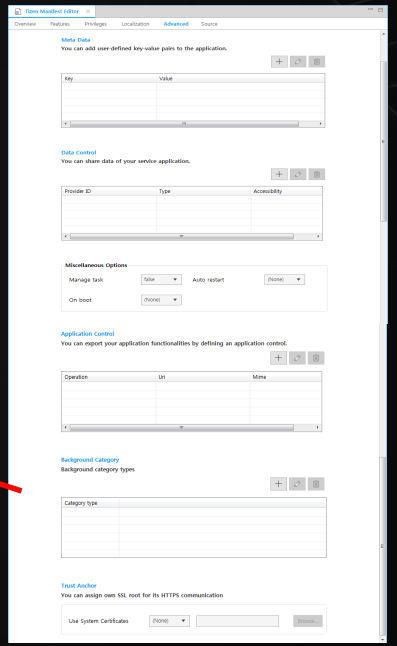
Tizen Project 외부 구조

Tizen Project 특성 – Advanced 탭

해당 App 을 통해 어떤 파일을 실행 시킬 수 있도록 할 것인가 를 설정합니다
예) 어떤 파일의 연결 프로그램에 해당 App이 나오도록 할 것인가

어떤 기능을 Background 에서도 계속 동작하도록 할 것인지 설정해 줍니다





Tizen Project 외부 구조

Tizen Project 특성 – Source 탭

앞서 본 5개의 탭 내용 모두를 xml 코드로
확인 및 설정 할 수 있습니다만..

굳이..이걸로 하신다면..

Tizen Manifest Editor

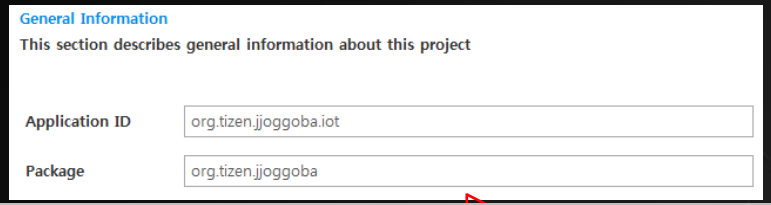
Overview | Features | Privileges | Localization | Advanced | Source

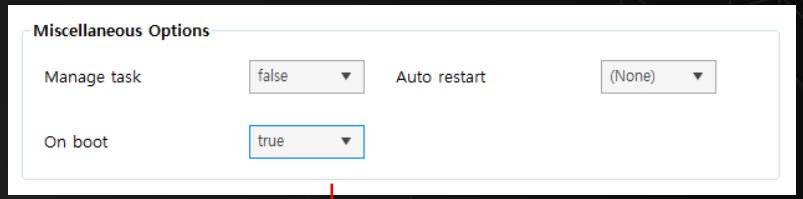
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<manifest xmlns="http://tizen.org/ns/packages" api-version="4.0" package="org.tizen.template_0822-5" version="1.0.0">
    <profile name="iot-headless"/>
    <service-application appid="org.tizen.template_0822-5" exec="template_0822-5" multiple="false" nodisplay="true" taskmanage="false" type="capp">
        <label>Tizen Template for 0822-5</label>
    </service-application>
</manifest>
```

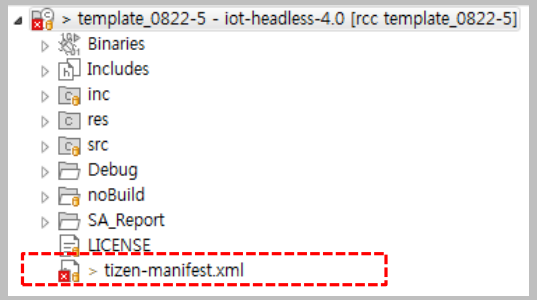
Tizen Project 외부 구조

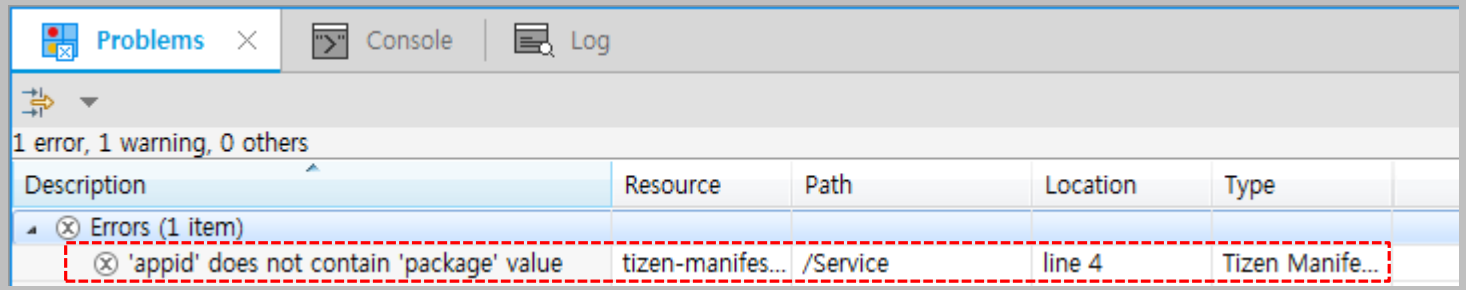
SAMSUNG Research

Tizen Project 특성 설정하기







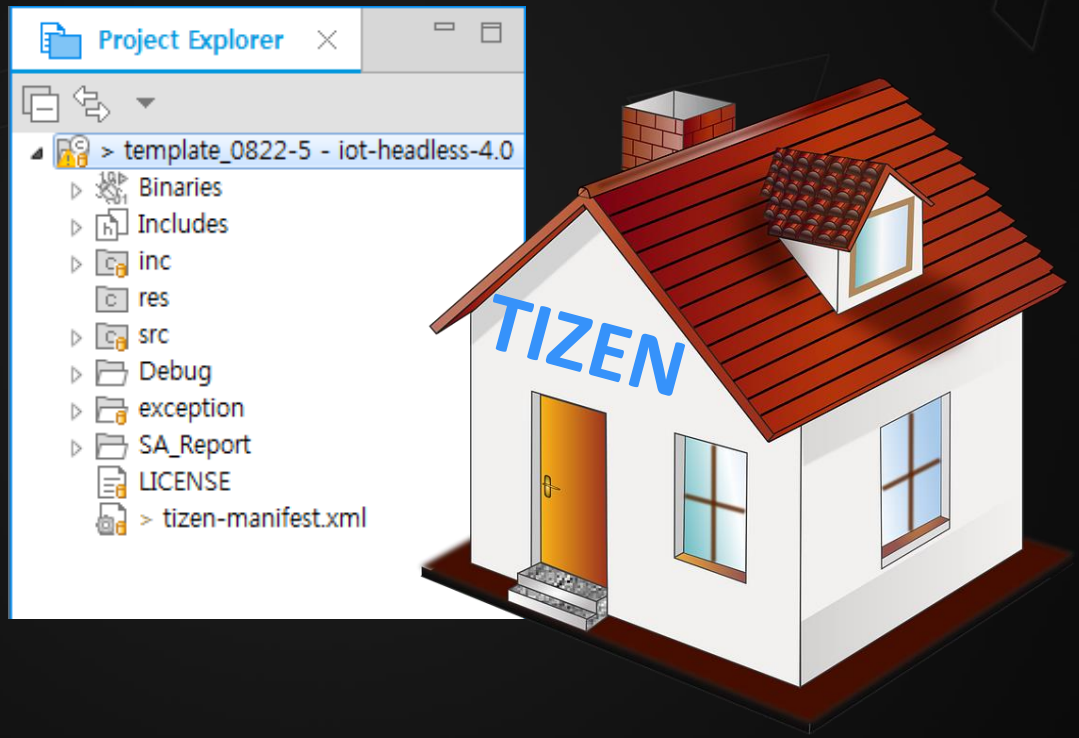


Tizen Project 외부 구조

지금까지는 폴더 구조와 설정 파일에
대해서 알아 보았습니다
집의 외부 구조에 해당합니다

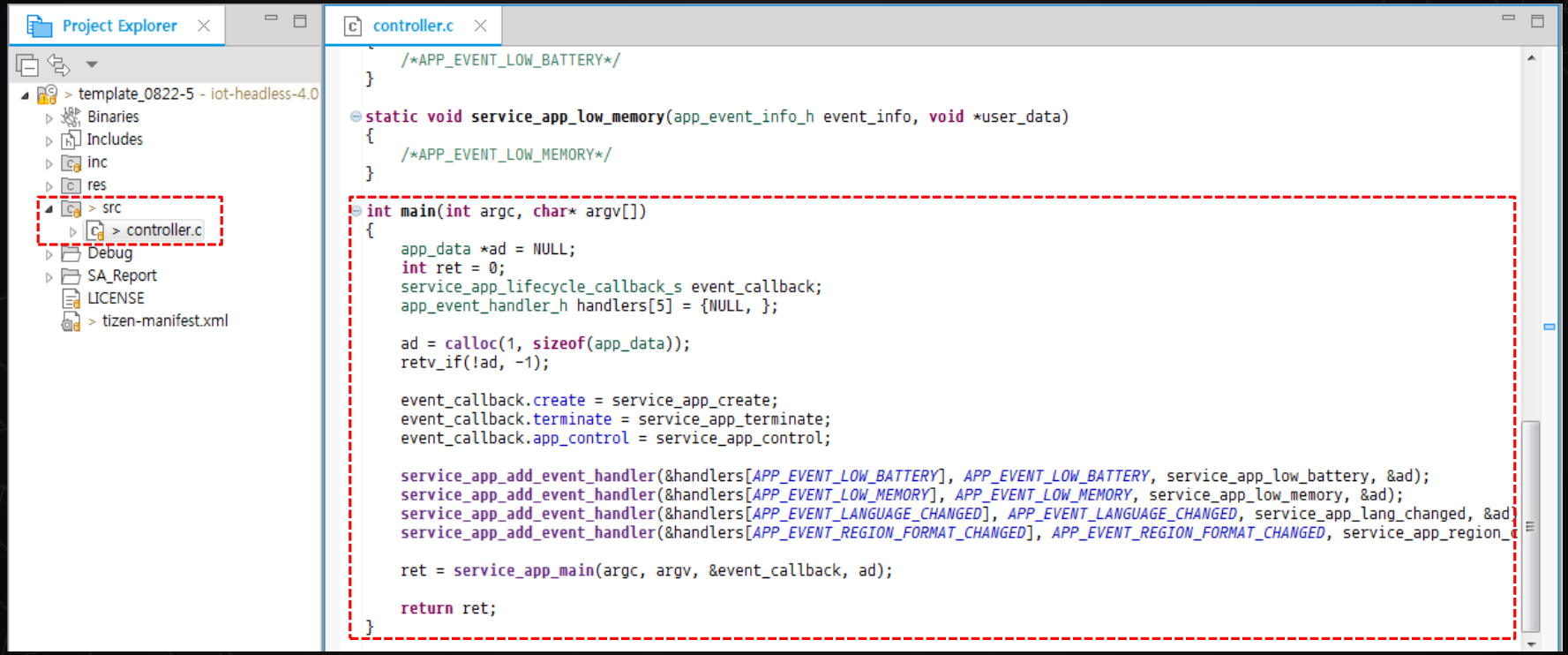
그러나 앞서 말씀 드렸듯이 우리의
목적은 인테리어!

그럼 이제 안으로 들어가 볼까요?



Tizen Project 동작 방식

Life Cycle of Tizen Project



Tizen Project 동작 방식

Life Cycle of Tizen Project

Lifecycle callback

- App의 생애주기 단계마다 main loop 에 의해 자동으로 호출 됩니다

```c
int main(int argc, char* argv[])
{
    app_data *ad = NULL;
    int ret = 0;
    service_app_lifecycle_callback_s event_callback;
    app_event_handler_h handlers[5] = {NULL, };

    ad = calloc(1, sizeof(app_data));
    retv if(!ad, -1);
```

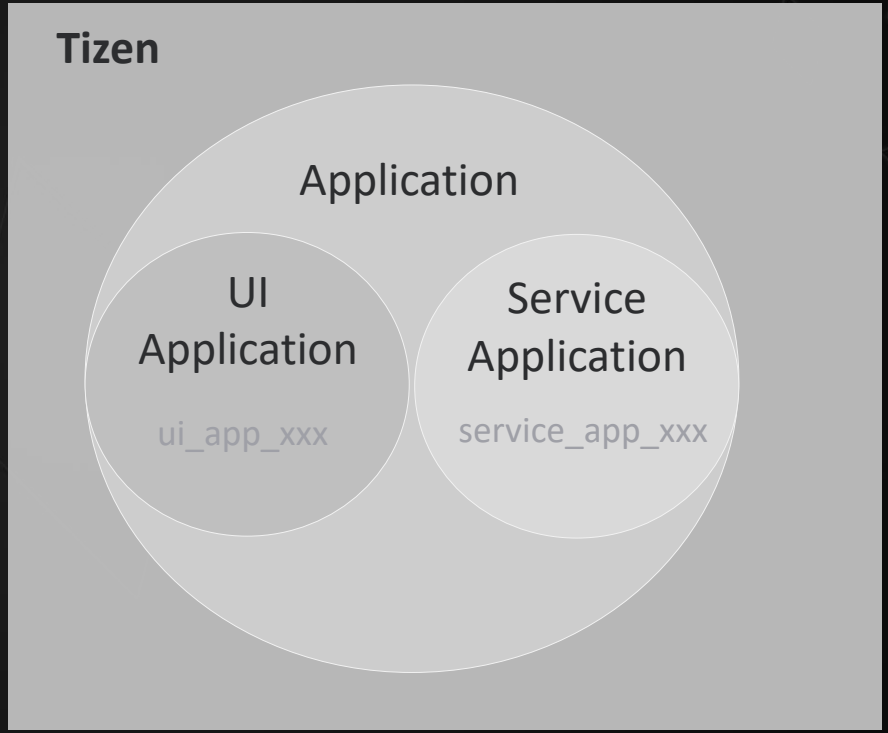
```c
/**
 * @brief The structure type containing the set of callback functions for handling application events.
 * @details It is one of the input parameters of the service_app_efl_main() function.
 * @since_tizen @if MOBILE 2.3 @elseif WEARABLE 2.3.1 @endif
 * @see service_app_main()
 * @see service_app_create_cb()
 * @see service_app_terminate_cb()
 * @see service_app_control_cb()
 */
typedef struct {
    service_app_create_cb create; /**< This callback function is called at the start of the application. */
    service_app_terminate_cb terminate; /**< This callback function is called once after the main loop of the application exits. */
    service_app_control_cb app_control; /**< This callback function is called when another application sends the launch request to the application. */
} service_app_lifecycle_callback_s;
```

Tizen Project 동작 방식

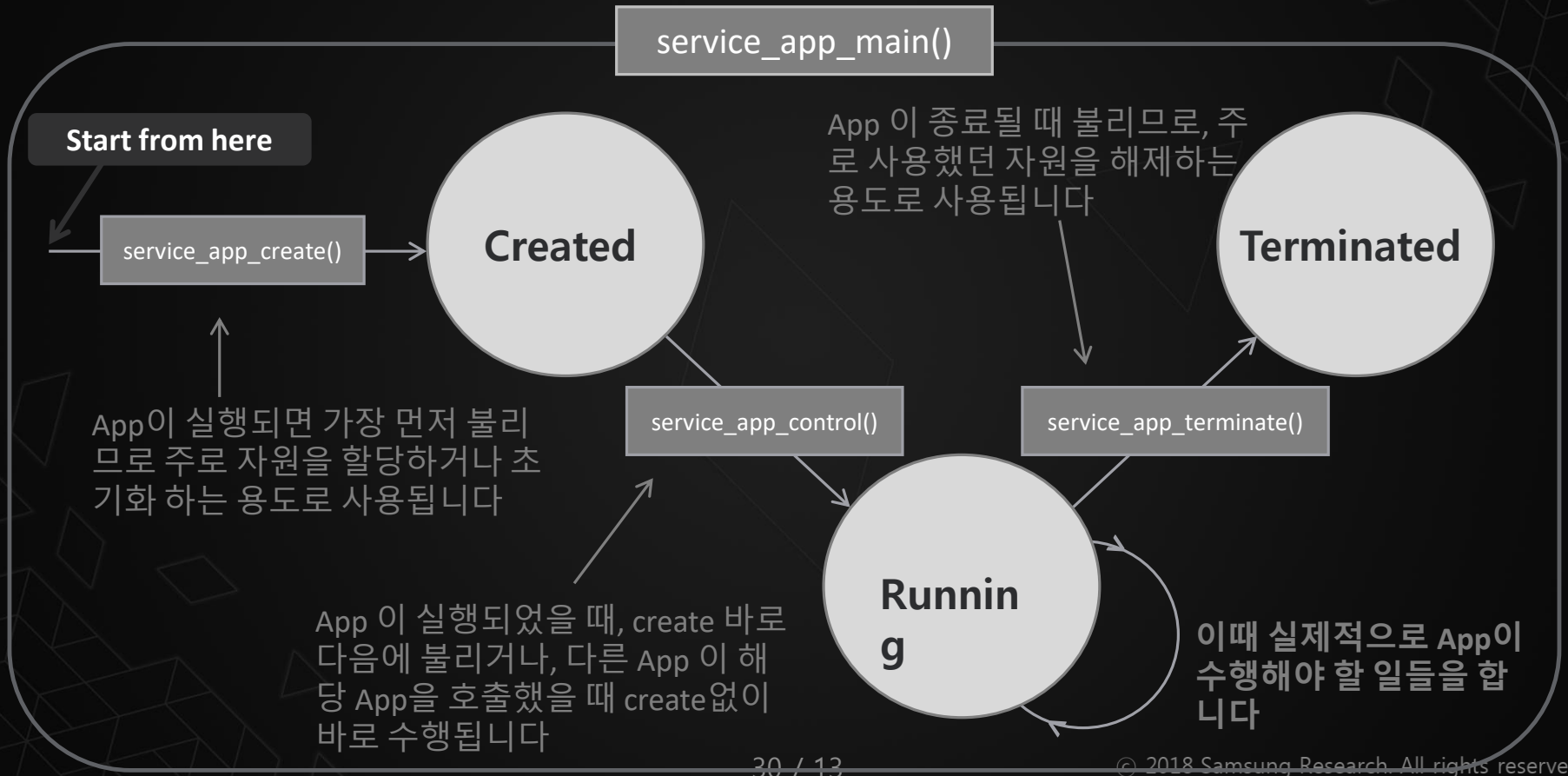
Life Cycle of Tizen Project

Tizen Application

- UI Application 은 화면에 무엇인가 그려지는 Application 입니다
- Service Application 은 무언가를 그리지는 않지만 항상 background 에서 돌아가고 있는 Application 입니다
- 그렇다면 IoT Headless 는??



Tizen Project 동작 방식



Tizen Project 동작 방식

Life Cycle of Tizen Project

Life Cycle 함수 호출 순서

- 호출되는 순서는 번호 순서와 같습니다
- 만약 service_app_create 의 return 이 false 라면??
- app_control_h 구조체는 어디에 쓸까요??
  ctrl + click 해보세요~

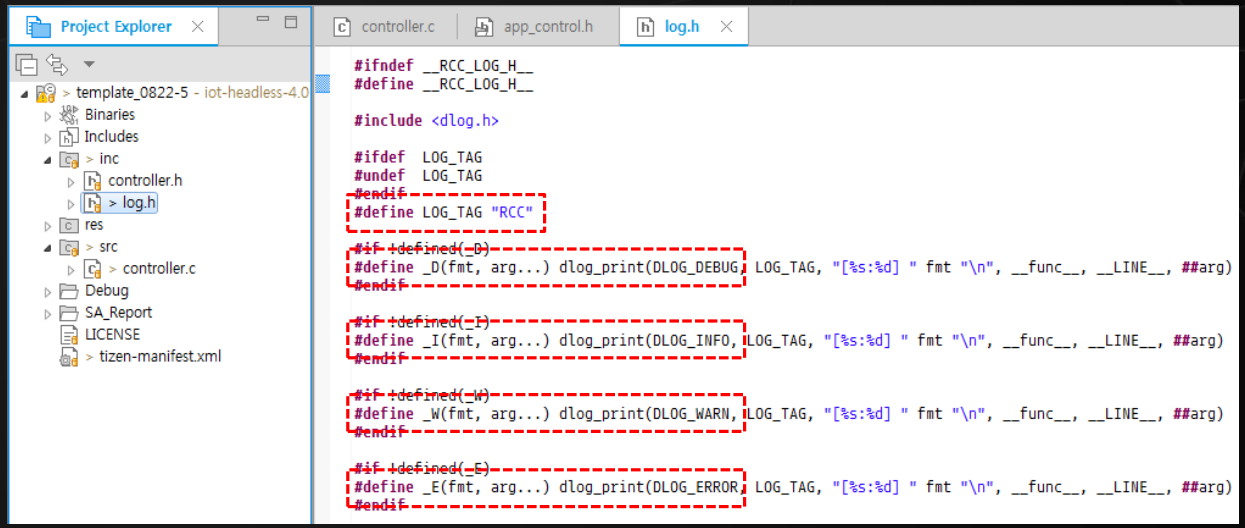
```
static bool service_app_create(void *data) 1
{
    return true;
}

static void service_app_terminate(void *data) 3
{
}

static void service_app_control(app_control_h app_control, void *data) 2
{
}
```

Tizen Project 동작 방식

- Life Cycle of Tizen Project

효과적인 로그 확인

- Log 를 보기 위해 printf ??
- 그러지 말고, inc/log.h 를 봅시다
- LOG_TAG 로 log 를 필터링 해서 볼 수 있습니다
- DLOG_DEBUG, DLOG_INFO, DLOG_WARN, DLOG_ERROR... 느낌 오시죠??



Tizen Project 동작 방식

Life Cycle of Tizen Project

Life Cycle 함수에 로그 넣기

- Create -> control -> terminate 순서를 확인하도록 로그를 넣습니다
- 그런데 이렇게만 하면 terminate 이 불릴까요??
- service_app_exit( ) 는 service_app.h 헤더파일에 있습니다

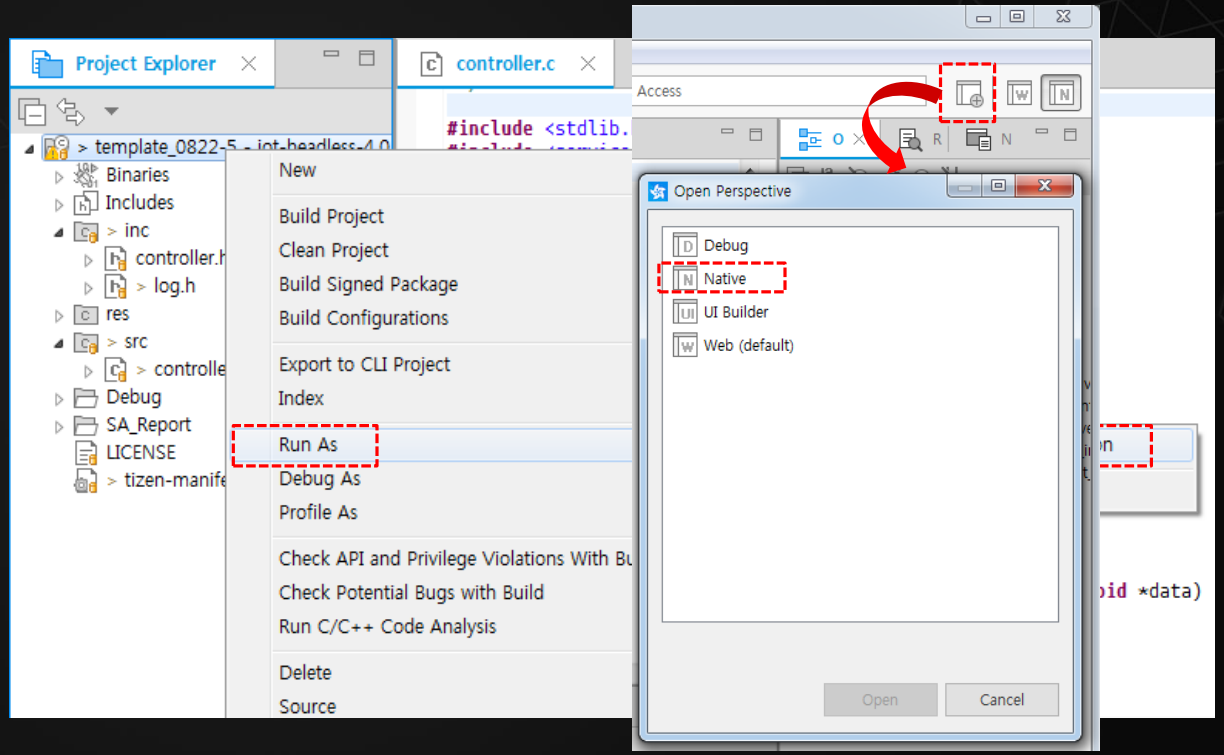
```c
#include <stdlib.h>
#include <service_app.h>

#include "log.h"
#include "controller.h"

typedef struct app_data_s {
} app_data;

static bool service_app_create(void *data)
{
    _D("First");
    return true;
}

static void service_app_terminate(void *data)
{
    _D("Final");
}

static void service_app_control(app_control_h app_control, void *data)
{
    _D("Middle");

}
```

Tizen Project 동작 방식

Tizen Project 실행하기

Run As Tizen Project

- Project 이름 우 클릭 -> Run As -> Tizen Native Application
- 위 항목이 없다면 우측 상단 Open Perpective 에서 Native 선택해 주세요



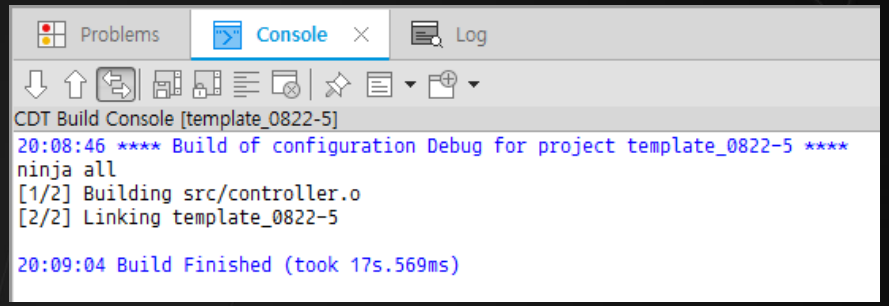
Tizen Project 동작 방식

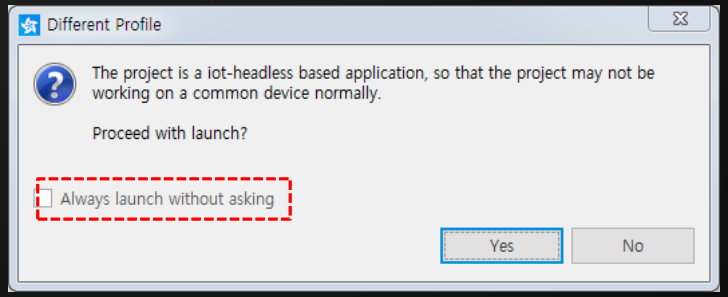
Tizen Project 실행하기

Run As Tize Project

- Run As 는
  - build
  - -> packaging
  - -> install
  - -> launch

  를 순차적으로 수행합니다



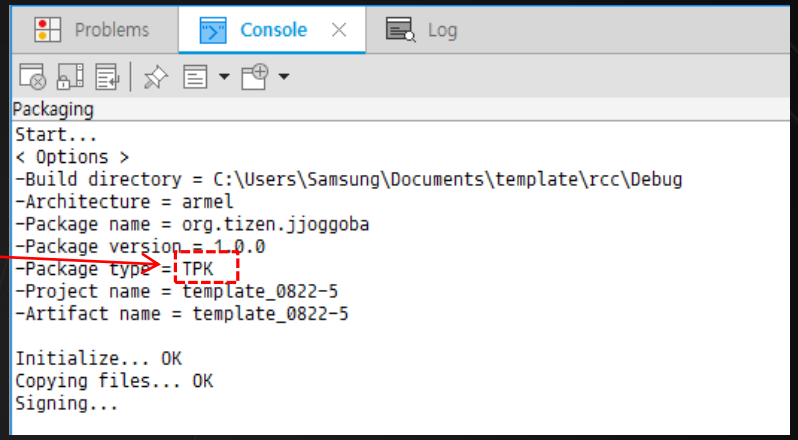


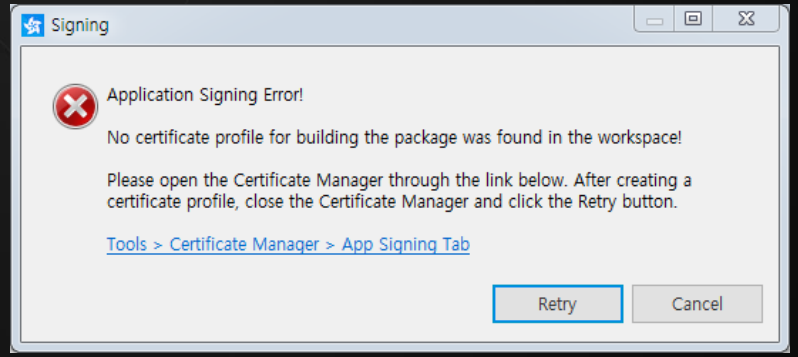
Tizen Project 동작 방식

Tizen Project 실행하기

Certificate Profile

- Packaging 을 통해 tpk(same as apk..) 파일이 생성됩니다
- Tizen Project 는 packaging 단계에서 signing 을 위한 certificate 이 필요합니다



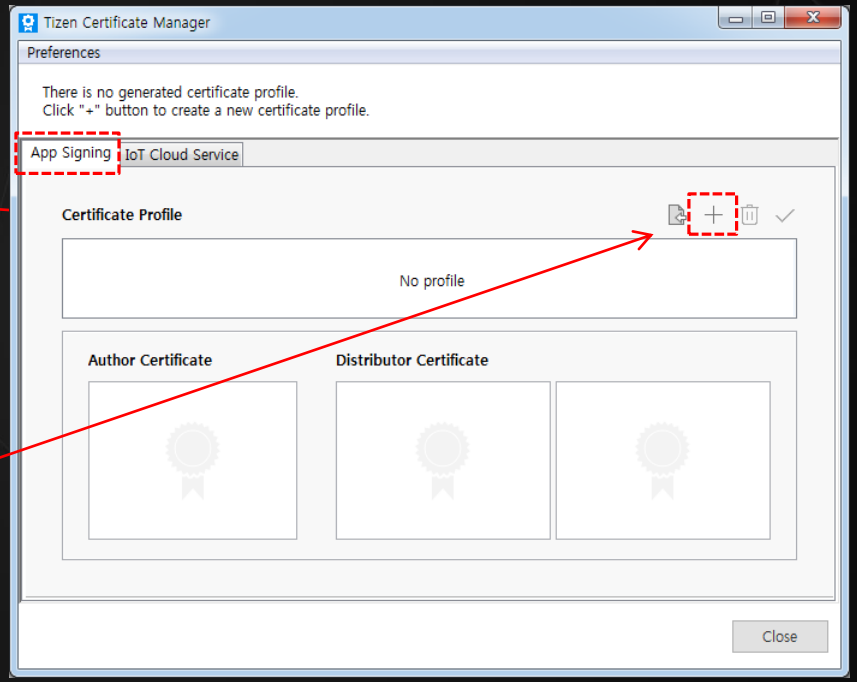


Tizen Project 동작 방식

Tizen Project 실행하기

Certificate Profile

- 기존에 만들어 둔 certificate profile 이 있다면 해당 파일을 선택해 주세요
- IoT Cloud Service 를 위한 Certificate 은 별도로 만들어 줘야 합니다
- 해당 Project로 생성 될 Application 에 대한 Certificate 을 만들어 줍니다

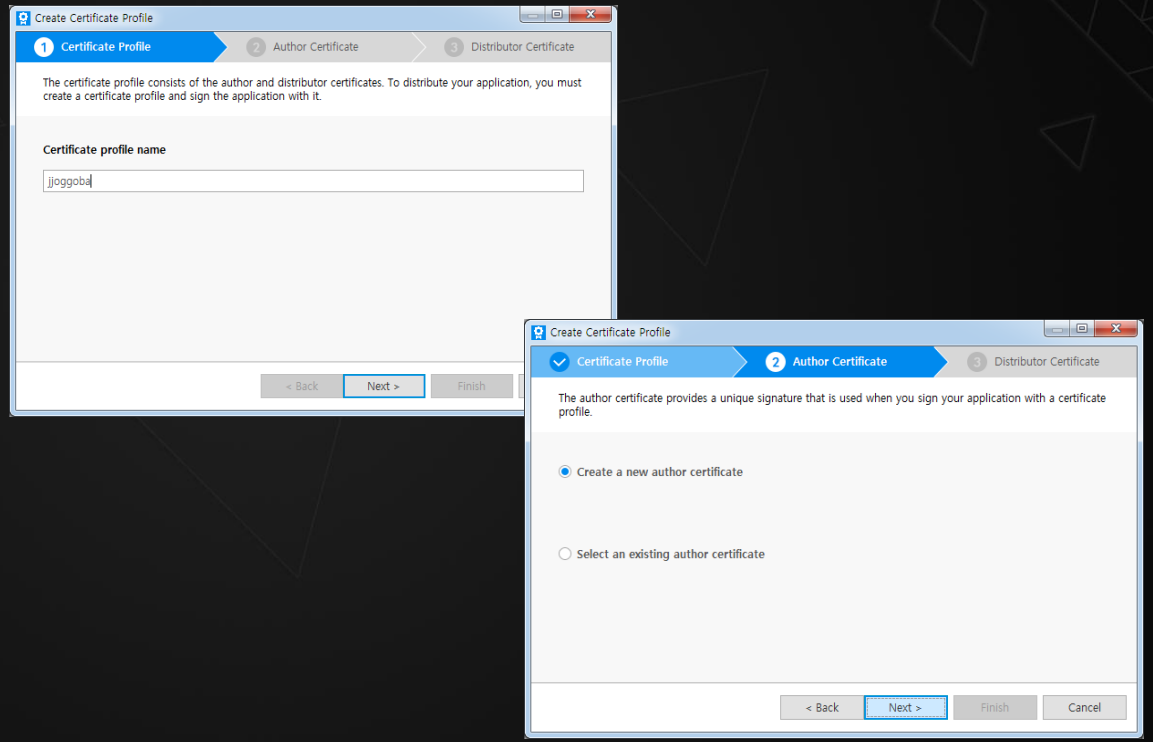


Tizen Project 동작 방식

Tizen Project 실행하기

Certificate Profile

- Certificate profile 의 이름을 적어 줍니다
- Author certificate 과 Distributor certificate 두 certificate 이 하나의 profile 을 구성합니다

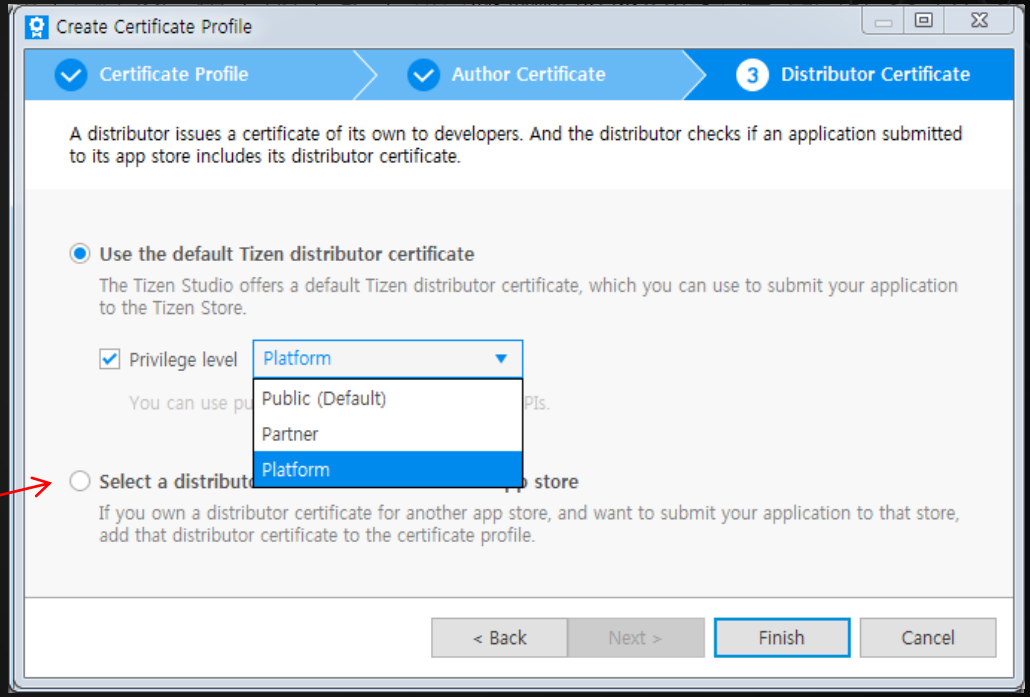


Tizen Project 동작 방식

Tizen Project 실행하기

Certificate Profile

- *는 모두 mandatory 입니다....
- Distributor certificate 에서는 Privilege level 이 중요합니다
- API 사용에 제한이 없도록 Platform level 을 선택해줍니다
- 다른 App Store 를 위한 certificate 을 선택할 수도 있습니다

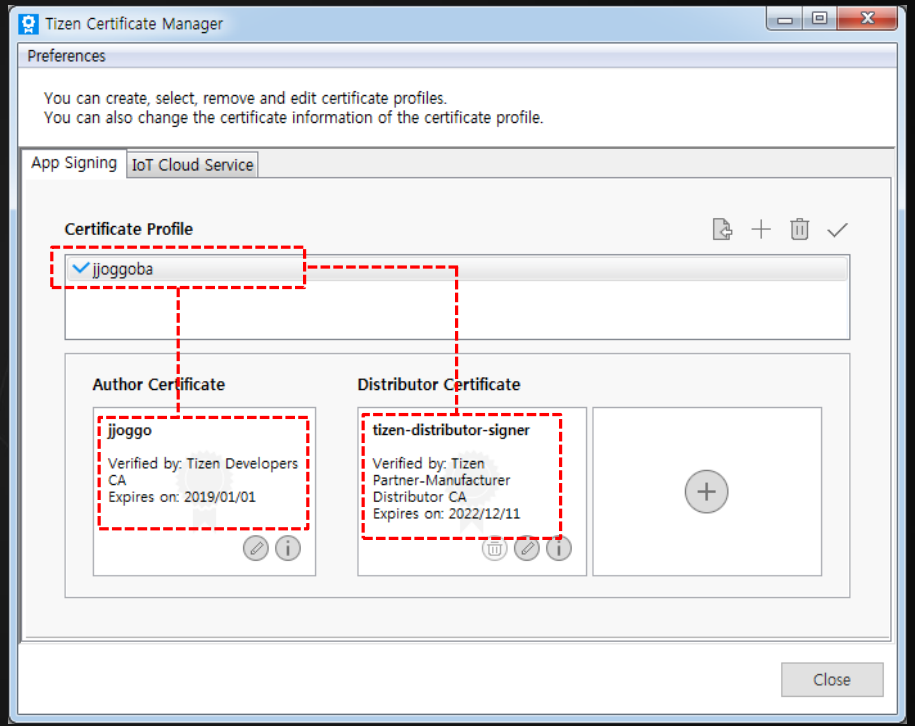


Tizen Project 동작 방식

Tizen Project 실행하기

Certificate Profile

- Certificate profile 이 만들어 졌습니다
- Distributor certificate 은 사실 테스트 용입니다
- Store 에 올라갈 때는 서버에 있는 실제 certificate 으로 패키징 됩니다

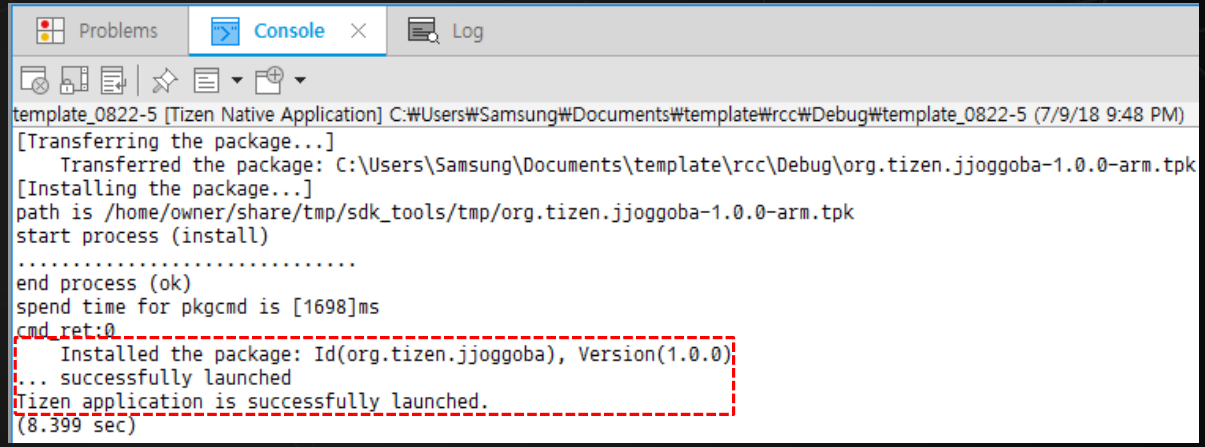


Tizen Project 동작 방식

Tizen Project 실행하기

Install & Launch

- 이제 다시 Run As 해볼까요?
- 오른쪽 로그를 보셨다면 install 과 launch 모두 성공 !!
- 이제 우리의 처음 목표였던 우리가 남긴 로그를 확인할 수 있습니다

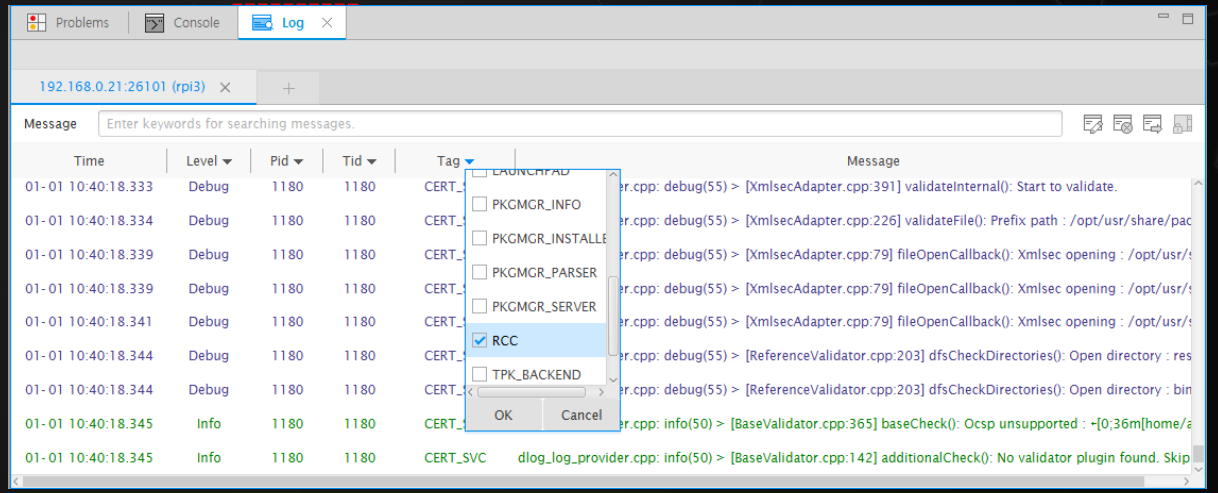


Tizen Project 동작 방식

- Tizen Project 실행하기

Log 확인하기

- Console 옆 Log 탭으로 갑니다
- 혹시 log.h 파일의 LOG_TAG 기억나시나요??
- 보고 싶은 것만 보세요 ^^



Tizen Project 동작 방식

- Tizen Project 실행하기

Log 확인하기

- 계획대로 인가요??
- 왜 로그가 두 번씩 찍혔을까요??

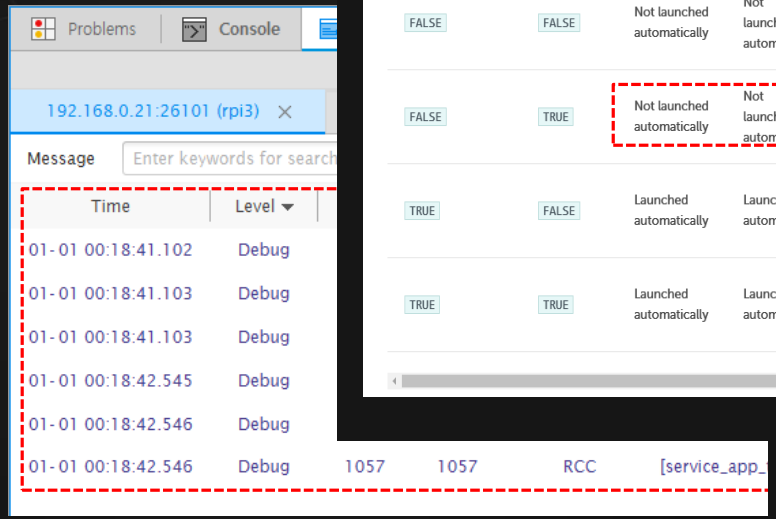
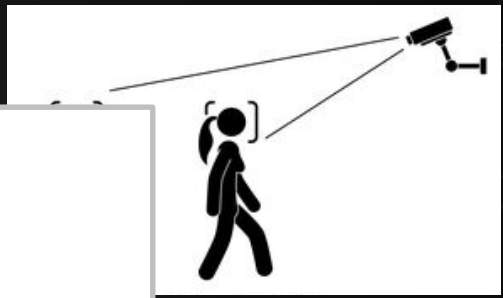


Table: Attribute combinations

auto-restart	on-boot	After normal termination	On forced close	On Reboot	After package installation	After package update
FALSE	FALSE	Not launched automatically	Not launched automatically	Not launched after reboot	Not launched	Not launched automatically
FALSE	TRUE	Not launched automatically	Not launched automatically	Launched automatically after reboot	Launched	Launched automatically
TRUE	FALSE	Launched automatically	Launched automatically	Not launched after reboot	Not launched	Launched automatically
TRUE	TRUE	Launched automatically	Launched automatically	Launched automatically after reboot	Launched	Launched automatically

Tizen Project 유용한 개발 Tip

Tizen Project 유용한 개발 Tip



정기적인, 주기적인, 지속적인, 반복적인, ...

Tizen Project 유용한 개발 Tip

Timer 를 알아보자

Ecore_Timer *	ecore_timer_add (double in, Ecore_Task_Cb func, const void *data) Creates a timer to call the given function in the given period of time.
Ecore_Timer *	ecore_timer_loop_add (double in, Ecore_Task_Cb func, const void *data) Creates a timer to call the given function in the given period of time.
void *	ecore_timer_del (Ecore_Timer *timer) Deletes the specified timer from the timer list.
void	ecore_timer_interval_set (Ecore_Timer *timer, double in) Change the interval the timer ticks off.
double	ecore_timer_interval_get (Ecore_Timer *timer) Get the interval the timer ticks on.
void	ecore_timer_freeze (Ecore_Timer *timer) Pauses a running timer.

void	ecore_timer_thaw (Ecore_Timer *timer) Resumes a frozen (paused) timer.
void	ecore_timer_delay (Ecore_Timer *timer, double add) Add some delay for the next occurrence of a timer.
void	ecore_timer_reset (Ecore_Timer *timer) Reset a timer to its full interval. This effectively makes the timer start ticking off from zero now.
double	ecore_timer_pending_get (Ecore_Timer *timer) Get the pending time regarding a timer.
double	ecore_timer_precision_get (void) Retrieves the current precision used by timer infrastructure.
void	ecore_timer_precision_set (double precision) Sets the precision to be used by timer infrastructure.
char *	ecore_timer_dump (void) Dump the all timers.

Tizen Project 유용한 개발 Tip

Timer를 알아보자

```
Ecore_Timer* ecore_timer_add(double in, Ecore_Task_Cb func, const void *data)
```

```
Ecore_Timer *timer = NULL;

timer = ecore_timer_add(5.0, __my_func, NULL);
```

넘겨 받은 timer 로 주기(얼마만에 한번), 멈추기, 늦추기 등등의 다른 API 들을 사용할 수 있게 됩니다.

Tizen Project 유용한 개발 Tip

Timer를 알아보자

void * ecore_timer_del(Ecore_Timer *timer)

** 반드시 유의해 주세요 **
func 안에서 ECORE_CALLBACK_CANCEL 시 Timer 가 더 이상 func 를 호출하지 않게 할 뿐만 아니라, Timer 인스턴스를 제거해줍니다.

이 경우 제거 된 timer에 NULL을 넣어 ecore_timer_del 로 중복 제거 하지 않도록 조심해주세요

```
Ecore_Timer *timer;
Eina_Bool __my_func(void *data)
{
    timer = NULL;
    return ECORE_CALLBACK_CANCEL;
}
```

```
void main()
{
    if(timer)
        ecore_timer_del(timer);
}
```

Tizen Project 유용한 개발 Tip

- **Timer는 언제 불릴까요?**

Timer 는
정교하고 엄격한 시간을 보장하지 않는다 !!

CPU 및 프로세스 에 따른 지연 호출 발생
아시겠지만, Tizen 만 그런 것이 아닙니다

Tizen Project 유용한 개발 Tip

Timer는 언제 불릴까요?

Main Loop

중요한 건 뭐다??!!

main loop 안에서 너무 많은 일들을 하게 하면 안 된다!!

즉, timer 의 시간 정확도는 depend on you

명심하자 ! 모든 건 내 탓일지도?!

Tizen Project 유용한 개발 Tip

Timer를 알아보자

실제로 timer를 돌려봅시다

- Timer 변수를 선언합니다
- service_app_control 함수 안에서 Timer 를 추가해줘요
- Interval 의 단위는 초 입니다!

```
#include "resource.h"
#include "controller.h"

typedef struct app_data_s {
} app_data;

Ecore_Timer *timer;
```

```
static void service_app_control(app_control_h app_control, void *data)
{
    // Todo: add your code here.

    timer = ecore_timer_add(3.0, __my_timer_func, NULL);
}
```

Tizen Project 유용한 개발 Tip

Timer를 알아보자

실제로 timer를 돌려봅시다

- Timer 변수를 선언합니다
- service_app_control 함수 안에서 Timer 를 추가해줘요
- Interval 의 단위는 초 입니다!
- 여기서 중요한 건 두 개 !!

```
#include "resource.h"
#include "controller.h"

typedef struct app_data_s {
} app_data;

Ecore_Timer *timer;
```

```
static void service_app_control(app_control_h app_control, void *data)
{
    // Todo: add your code here.

    timer = ecore_timer_add(3.0, __my_timer_func, NULL);
}
```

Tizen Project 유용한 개발 Tip

Timer를 알아보자

실제로 timer를 돌려봅시다

- 계속 계속 호출되도록 합니다
- App 이 종료 될 때는 삭제해 줍니다
- Ecore_Timer 는 Ecore.h 에 있어요 !!

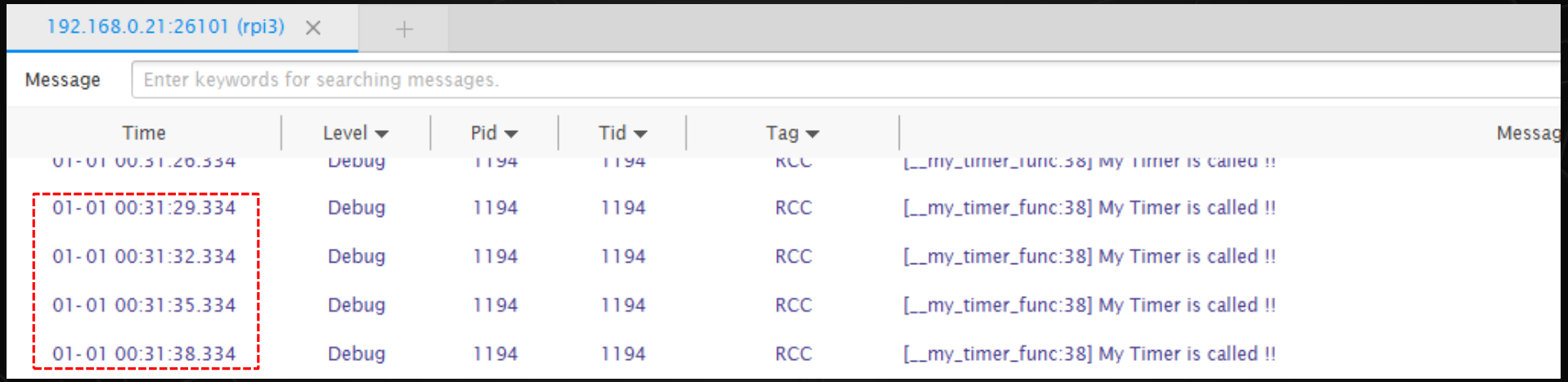
```
Eina_Bool __my_timer_func(void *data)
{
    _D("My Timer is called !!");

    return ECORE_CALLBACK_RENEW;
}
```

```
static void service_app_terminate(void *data)
{
    app_data *ad = (app_data *)data;

    ecore_timer_del(timer);

    free(ad);
}
```

```
#include <tizen.h>
#include <stdlib.h>
#include <service_app.h>
#include <Ecore.h>
```

Tizen Project 유용한 개발 Tip

- Timer를 알아보자



Tizen Project 유용한 개발 Tip

- IoT 의 핵심은 바로 이곳입니다

```
Eina_Bool __my_timer_func(void *data)
{
    _D("My Timer is called !!");

    바로 여기 ! Here !

    return ECORE_CALLBACK_RENEW;
}
```

확인 하고

가져오고

보내고

Tizen Project 유용한 개발 Tip

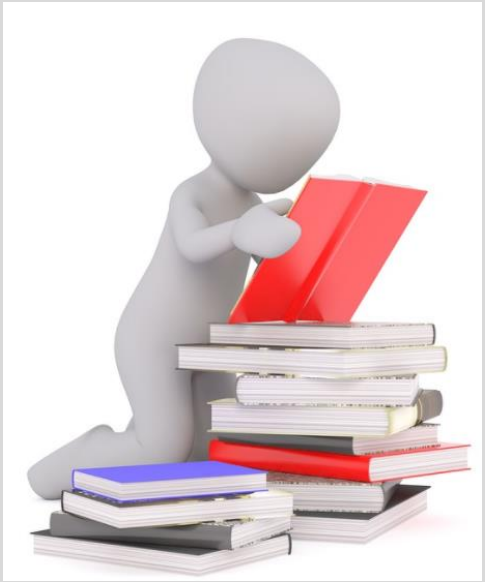
- IoT 의 어

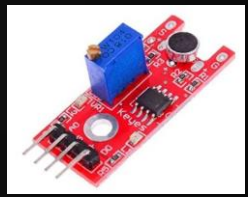
확인 하고

가져오고

다양성의 역습 !! ㅠ ㅠ

Protocol 이 너무 많아요...



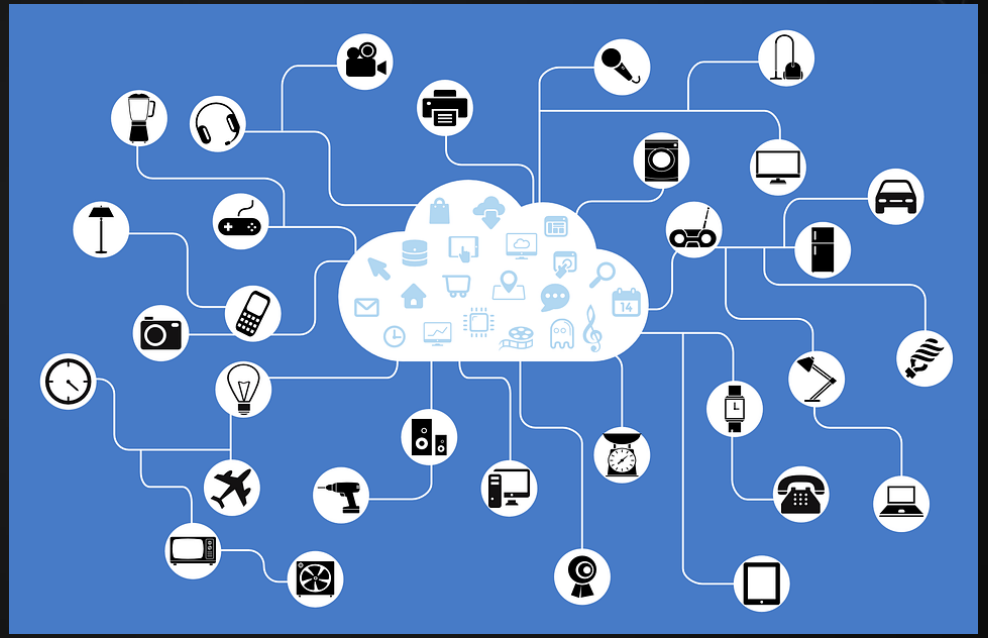


Tizen Project 유용한 개발 Tip

- IoT 의 어려운 점..

보내고

어떤 형태??
어떻게??



Tizen Project 유용한 개발 Tip

- **강력한 Reference !!**

RCC master 브랜치

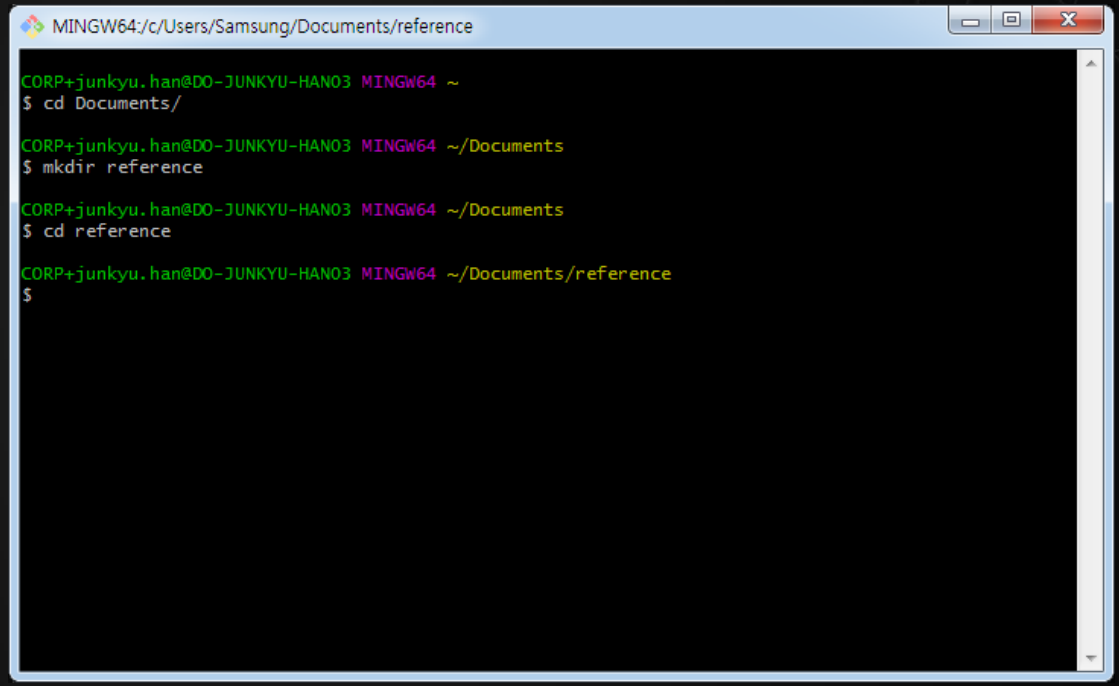
- 다양한 프로토콜에 대한 대응
- 다양한 센서들의 HW 스펙 충족
- Cloud 와의 전송 데이터 포맷 파악

Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

Git 으로부터 가져오기

- Git bash
- Tizen Project 를 가져올 폴더로 이동해줍니다
- 기존의 rcc 프로젝트가 있으면 이름이 같아서 좋지 않아요
- 다른 폴더를 하나 만들어 줍니다
- Clone 받을 프로젝트 주소 확인하러 가 볼까요?

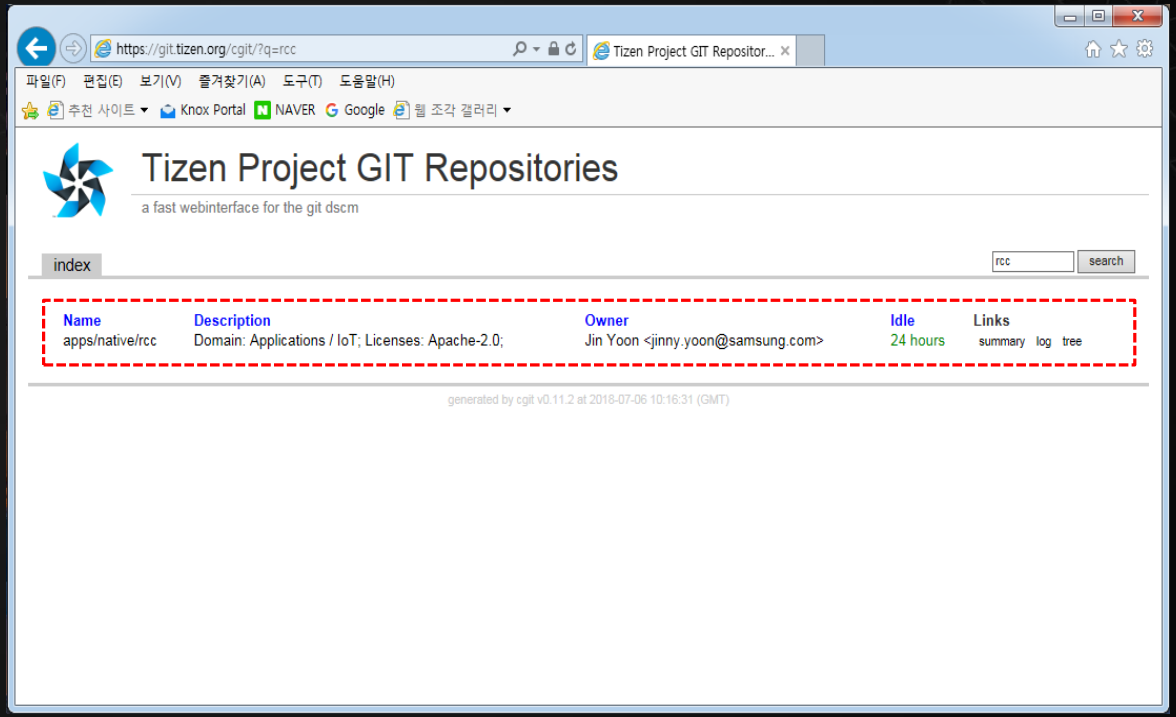


Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

Git 으로부터 가져오기

- https://git.tizen.org
- 모든 profile 의 Tizen Project 들이 보입니다
- 스마트하게 search 를 사용해요
- 'rcc' 를 찾아줍니다
- apps/native/rcc 라는 아이가 보여요
- 문제가 있으면 Owner에게....

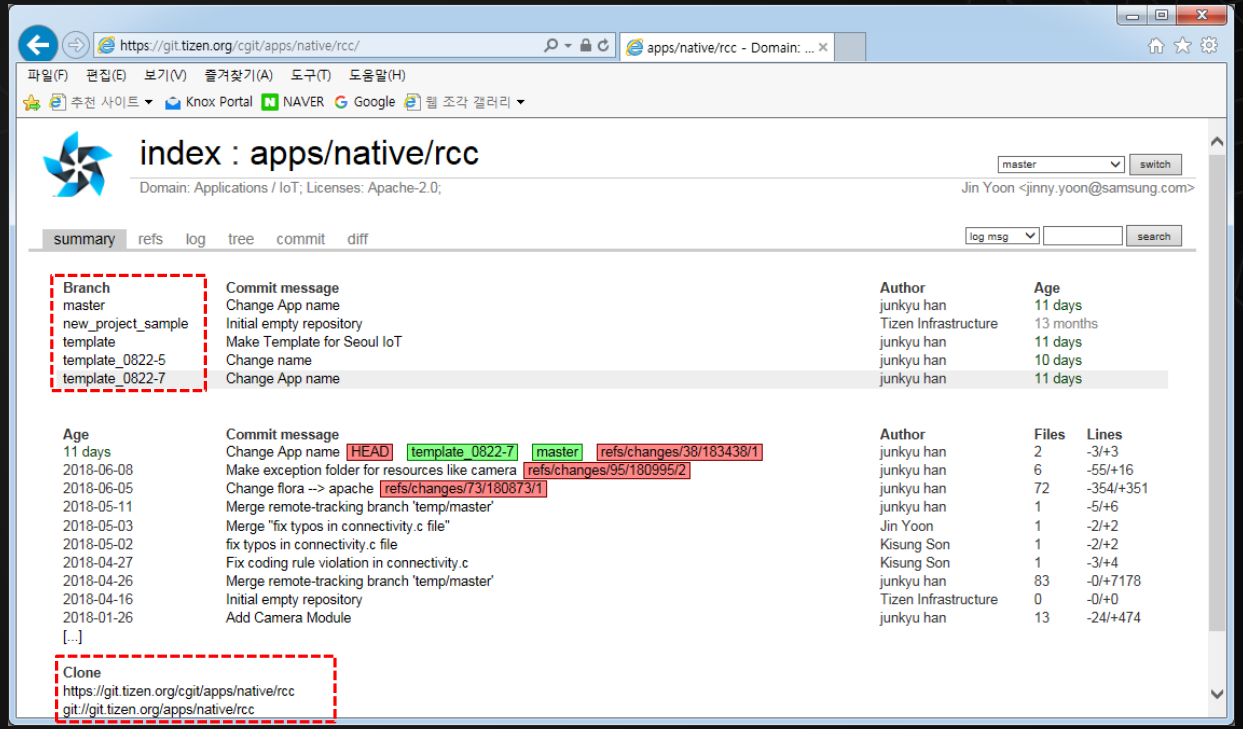


Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

Git 으로부터 가져오기

- 전체 branch 를 가지고 올 겁니다
- Clone 에서는 https 를 사용할 게요

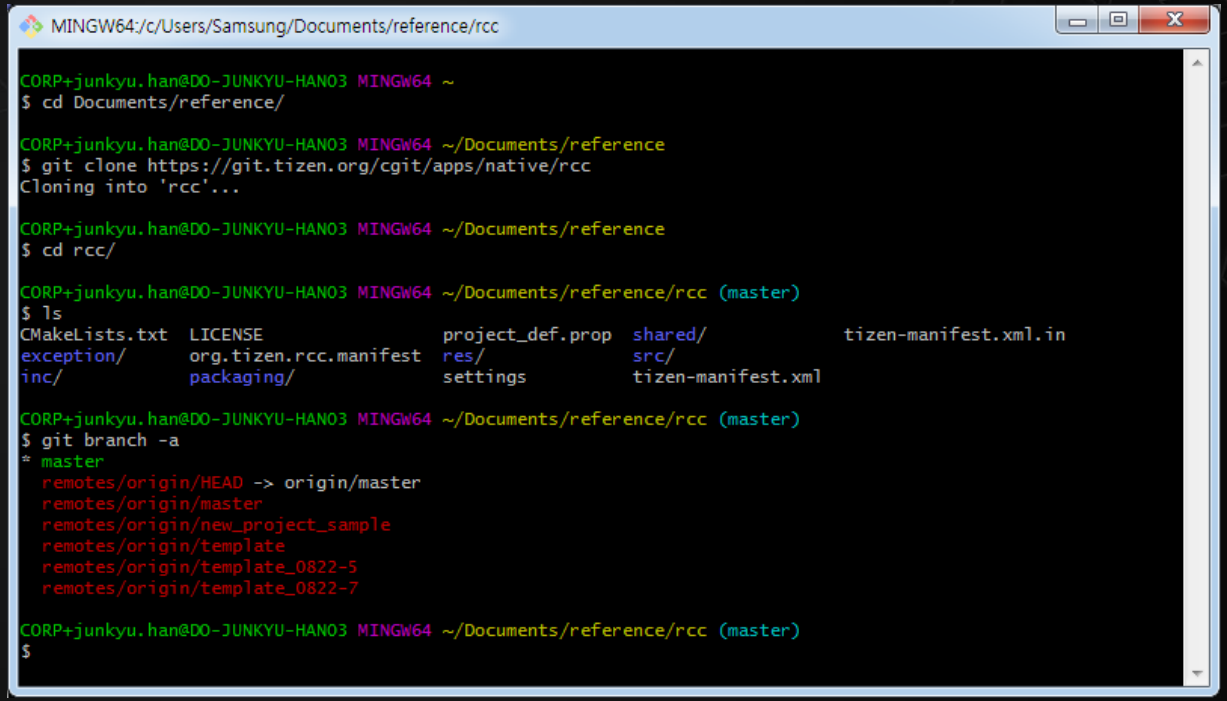


Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

Git 으로부터 가져오기

- git clone https://git.tizen.org/cgit/apps/native/rcc
- 현재 폴더에 rcc/ 폴더가 생성되었음을 확인합니다
- rcc 프로젝트로 이동해 볼게요
- git branch –a 를 통해 모든 branch 를 볼 수 있습니다

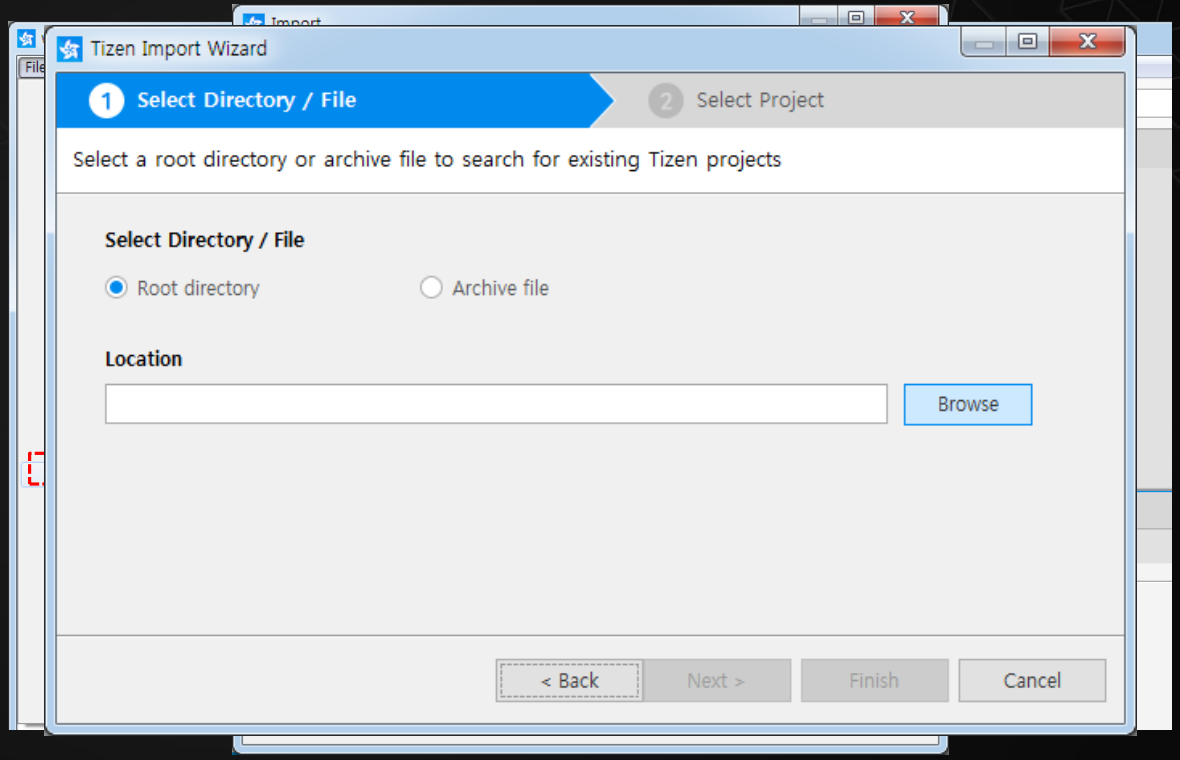


Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

rcc 프로젝트 가져오기

- File -> Import
- Tizen -> Tizen Project
- Root directory -> Browse

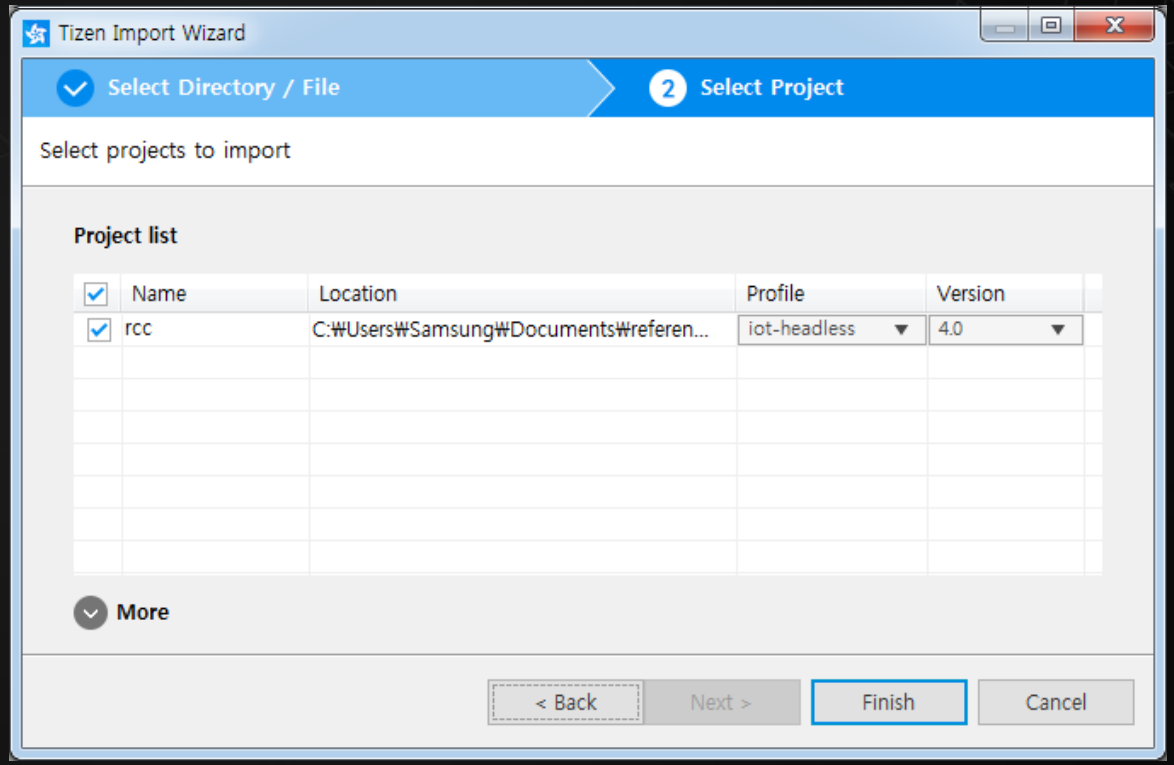


Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

rcc 프로젝트 가져오기

- rcc 프로젝트를 저장해 둔 위치로 가서 rcc 폴더를 선택합니다
- 위치를 다시 한번 확인하고 Next 를 눌러주세요
- 해당 프로젝트가 다양한 profile 을 지원한다면 원하는 profile 을 선택할 수 있습니다

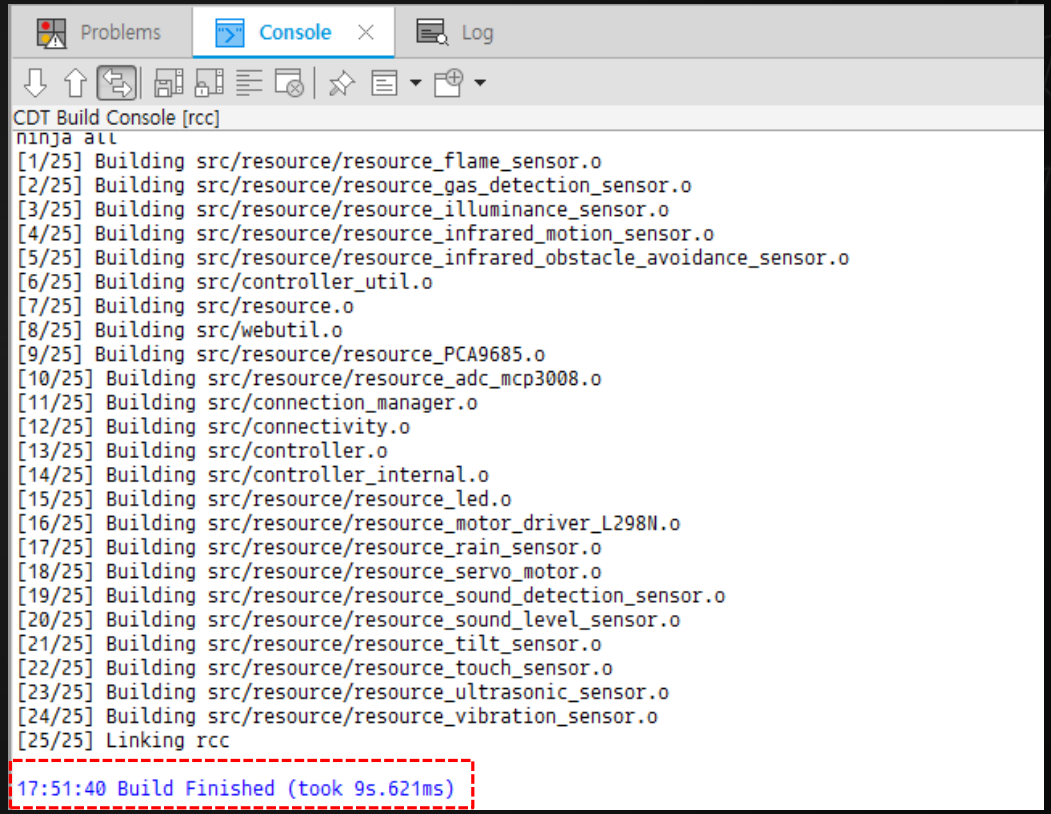


Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

빌드로 유효성 확인

- 프로젝트 이름 우 클릭 -> Build Project
- 하단 Console 창에서 빌드 프로세스 로그 확인 가능합니다
- FAILED 메시지 없이 Build Finished 가 보이면 해당 프로젝트가 성공적으로 빌드 된 것입니다



Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

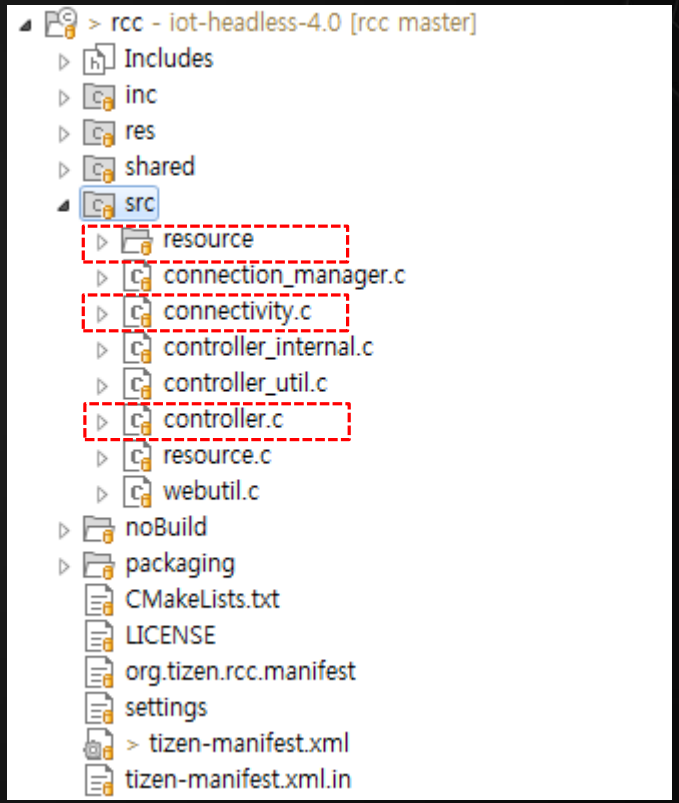
src 폴더에 집중해 주세요

- 앞서 말씀 드렸던 src, inc, res, tizen-manifest.xml 외에는 local build 를 위한 것들입니다. 무시해 주세요.
- src 에서도 3개만 봐주세요

  resource/

  connectivity.c

  controller.c



A to Z of Tizen Project

각 종 센서나 LED 와 같이 주변 환경에 대한 정보를 얻거나 그에 반응할 수 있는 모든 것들....

로컬 네트워크나 cloud 서버에 데이터를 보내는 것과 관련 된 코드

resource/

controller.c

connectivity.c

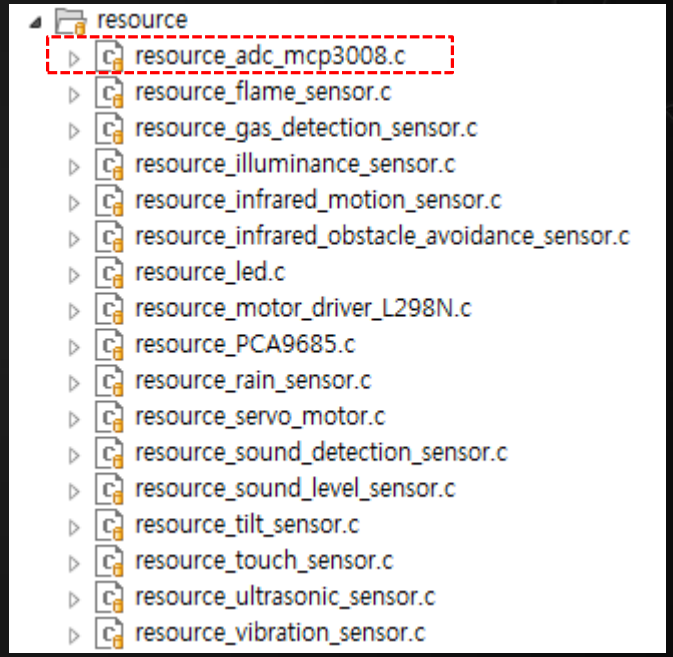
Resource 들을 connectivity 를 사용하여 연결하기

Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

resource/

- 여러가지 IoT 환경에서 사용 될 resource 들
- 모델 명 확인 필요 !!
- Protocol 확인 필요 !!

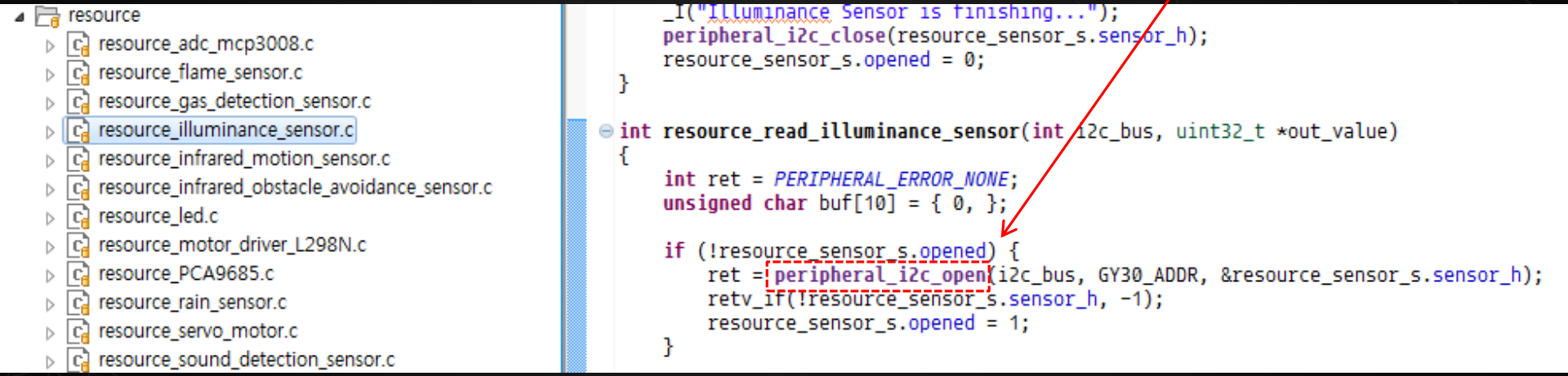


Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

Protocol 확인

I2C 를 사용한다면 OK !

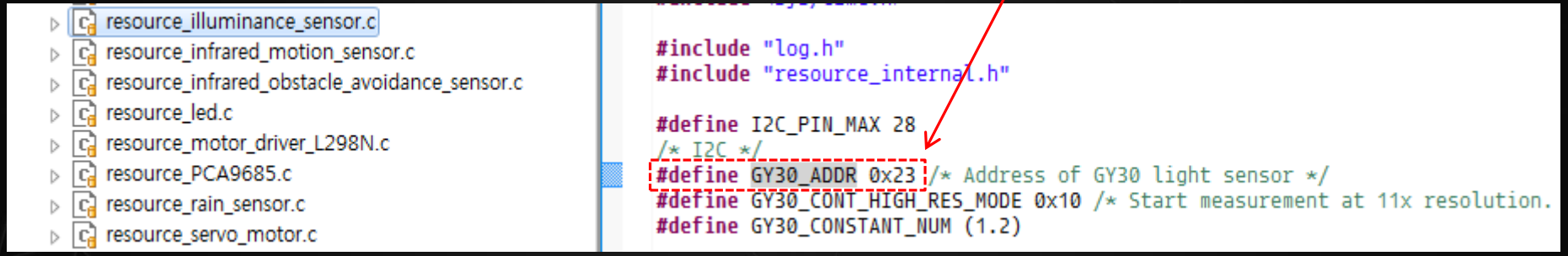


Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

꼭 체크합시다 !!

HW 스펙에서 반드시 확인해줘야 합니다



Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

직관적이고 명료한 함수

```
int resource_read_illuminance_sensor(int i2c_bus, uint32_t *out_value)

void resource_close_illuminance_sensor(void)

int resource_write_led(int pin_num, int write_value)
```

Tizen

SAMSUNG Research

강력한

내부적으로

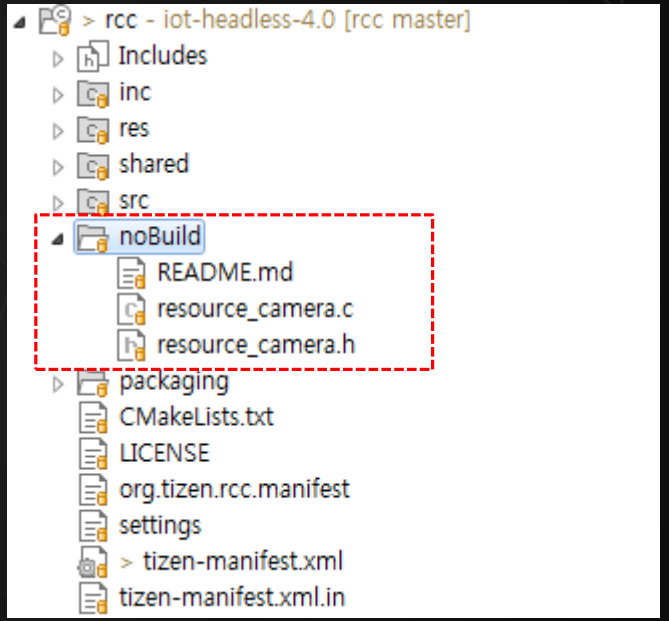
```c
int resource_read_ultrasonic_sensor(int trig_pin_num, int echo_pin_num, resource_read_cb cb, void *data)
{
    int ret = 0;

    triggered_time = 0;

    if (resource_read_info == NULL) {
        resource_read_info = calloc(1, sizeof(resource_read_s));
        retv_if(!resource_read_info, -1);
    } else {
        peripheral_gpio_unset_interrupted_cb(resource_get_info(resource_read_info->pin_num)->sensor_h);
    }
    resource_read_info->cb = cb;
    resource_read_info->data = data;
    resource_read_info->pin_num = echo_pin_num;

    if (!resource_get_info(trig_pin_num)->opened) {
        _I("Ultrasonic sensor's trig is initializing...");

        ret = peripheral_gpio_open(trig_pin_num, &resource_get_info(trig_pin_num)->sensor_h);
        retv_if(!resource_get_info(trig_pin_num)->sensor_h, -1);

        ret = peripheral_gpio_set_direction(resource_get_info(trig_pin_num)->sensor_h, PERIPHERAL_GPIO_DIRECTION_OUT_INITIALL
        retv_if(ret != 0, -1);

        resource_get_info(trig_pin_num)->opened = 1;
        resource_get_info(trig_pin_num)->close = resource_close_ultrasonic_sensor_trig;
    }

    if (!resource_get_info(echo_pin_num)->opened) {
        _I("Ultrasonic sensor's echo is initializing...");

        ret = peripheral_gpio_open(echo_pin_num, &resource_get_info(echo_pin_num)->sensor_h);
        retv_if(!resource_get_info(echo_pin_num)->sensor_h, -1);

        ret = peripheral_gpio_set_direction(resource_get_info(echo_pin_num)->sensor_h, PERIPHERAL_GPIO_DIRECTION_IN);
        retv_if(ret != 0, -1);

        ret = peripheral_gpio_set_edge_mode(resource_get_info(echo_pin_num)->sensor_h, PERIPHERAL_GPIO_EDGE_BOTH);
        retv_if(ret != 0, -1);

        resource_get_info(echo_pin_num)->opened = 1;
        resource_get_info(echo_pin_num)->close = resource_close_ultrasonic_sensor_echo;
    }

    if (resource_get_info(echo_pin_num)->sensor_h) {
        ret = peripheral_gpio_set_interrupted_cb(resource_get_info(echo_pin_num)->sensor_h, _resource_read_ultrasonic_sensor_
        retv_if(ret != 0, -1);
    }
```

Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

custom 이미지용 resource

- 현재 IoT preview 이미지에서는 빌드 되지 않는 resource 들이 있습니다
- 사용하고자 하는 resource 들에 맞춰 custom 한 이미지를 만들어줘야 빌드가 가능합니다
- Build 되지 않도록 noBuild 란 폴더를 만들어서 관리하고 있습니다

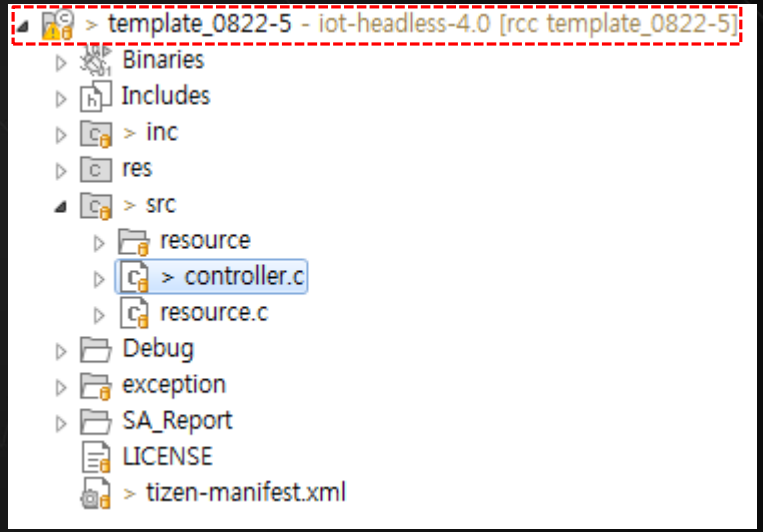


Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

사용 시나리오

적외선 동작 감지



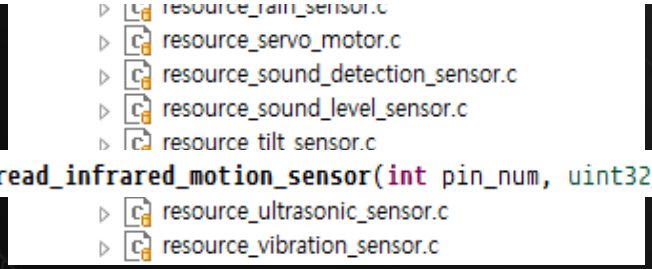
Tizen Project 유용한 개발 Tip


```c
int resource_read_infrared_motion_sensor(int pin_num, uint32_t *out_value)
{
    int ret = PERIPHERAL_ERROR_NONE;

    if (!resource_get_info(pin_num)->opened) {
        ret = peripheral_gpio_open(pin_num, &resource_get_info(pin_num)->sensor_h);
        retv_if(!resource_get_info(pin_num)->sensor_h, -1);

        ret = peripheral_gpio_set_direction(resource_get_info(pin_num)->sensor_h, PERIPHERAL_GPIO_DIRECTION_IN);
        retv_if(ret != 0, -1);

        resource_get_info(pin_num)->opened = 1;
        resource_get_info(pin_num)->close = resource_close_infrared_motion_sensor;
    }

    ret = peripheral_gpio_read(resource_get_info(pin_num)->sensor_h, out_value);
    retv_if(ret < 0, -1);

    return 0;
}
```

지 되었는지

요??

_sensor.c



너무나 직관적인 이름의 함수가 보입니다

Tizen Project 유용한 개발 Tip

✪ 강력한 Reference !!

사용 시나리오

센서와 연결할 GPIO 번호

```
int resource_read_infrared_motion_sensor(int pin_num, uint32_t *out_value)
{
    int ret = PERIPHERAL_ERROR_NONE;

    if (!resource_get_info(pin_num)->opened) {
        ret = peripheral_gpio_open(pin_num, &resource_get_info(pin_num)->sensor_h);
        retv_if(!resource_get_info(pin_num)->sensor_h, -1);

        ret = peripheral_gpio_set_direction(resource_get_info(pin_num)->sensor_h, PERIPHERAL_GPIO_DIRECTION_IN);
        retv_if(ret != 0, -1);

        resource_get_info(pin_num)->opened = 1;
        resource_get_info(pin_num)->close = resource_close_infrared_motion_sensor;
    }

    ret = peripheral_gpio_read(resource_get_info(pin_num)->sensor_h, out_value);
    retv_if(ret < 0, -1);

    return 0;
}
```

이 변수에 동작이 있는지 없는지 1 또는 0의 값을 씁니다

Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

사용 시나리오

```
Eina_Bool __my_timer_func(void *data)
{
    _D("My Timer is called !!");




    return ECORE_CALLBACK_RENEW;
}
```

```
static Eina_Bool __my_timer_func(void *data)
{
    _D("My Timer is called !!");
    int ret = 0;
    int value = 0;

    ret = resource_read_infrared_motion_sensor(11, &value);
    if (ret < 0)
        return ECORE_CALLBACK_CANCEL;

    if (value == 1)
        _D("Someone is there");
    else
        _D("No one is there");

    return ECORE_CALLBACK_RENEW;
}
```

0 or 1

Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

사용 시나리오

- 3초에 한번씩 !!
- 동작 센서를 읽어와 라고 명령을 내리고 !!
- 명령 하달에 실패 시 __my_timer_func 가 더 이상 불리지 않도록 합니다
- 읽어 온 값에 따라 원하는 동작을 수행합니다

```c
static void service_app_control(app_control_h app_control, void *data)
{
    // Todo: add your code here.

    timer = ecore_timer_add(3.0, __my_timer_func, NULL);
}
```

```c
static Eina_Bool __my_timer_func(void *data)
{
    _D("My Timer is called !!");
    int ret = 0;
    int value = 0;

    ret = resource_read_infrared_motion_sensor(11, &value);
    if (ret < 0)
        return ECORE_CALLBACK_CANCEL;

    if (value == 1)
        _D("Someone is there");
    else
        _D("No one is there");

    return ECORE_CALLBACK_RENEW;
}
```

Tizen Project 유용한 개발 Tip

- 강력한 Reference !!

사용 시나리오

```
int resource_read_infrared_motion_sensor(int pin_num, uint32_t *out_value)
{
    int ret = PERIPHERAL_ERROR_NONE;

    if (!resource_get_info(pin_num)->opened) {
        ret = peripheral_gpio_open(pin_num, &resource_get_info(pin_num)->sensor_h);
        retv_if(!resource_get_info(pin_num)->sensor_h, -1);

        ret = peripheral_gpio_set_direction(resource_get_info(pin_num)->sensor_h, PERIPHERAL_GPIO_DIRECTION_IN);
        retv_if(ret != 0, -1);

        resource_get_info(pin_num)->opened = 1;
        resource_get_info(pin_num)->close = resource_close_infrared_motion_sensor;
    }

    ret = peripheral_gpio_read(resource_get_info(pin_num)->sensor_h, out_value);
    retv_if(ret < 0, -1);

    return 0;
}
```

```
int resource_read_infrared_motion_sensor(int pin_num, uint32_t *out_value)
```

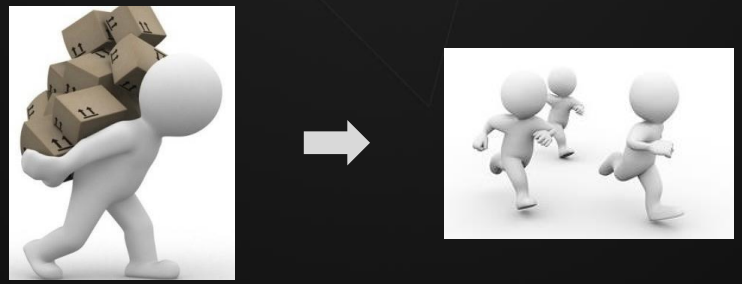
Tizen Project 유용한 개발 Tip

Callback 과 Thread

자주 쓰이는 callback...

Asycncronous

```
int resource_read_ultrasonic_sensor(int trig_pin_num, int echo_pin_num, resource_read_cb cb, void *data);

int resource_capture_camera(capture_completed_cb capture_completed, void *user_data);

int st_things_register_request_cb(st_things_get_request_cb get_cb, st_things_set_request_cb set_cb);
```



Tizen Project 유용한 개발 Tip

Callback 과 Thread

체크 포인트 !!

- Asynchronous
- Timing
- Collision
- Mutex

SAMSUNG Research

Thank you