Tizen 실습 예제: Remote Key Framework

시스템소프트웨어특론 (2014년 2학기)

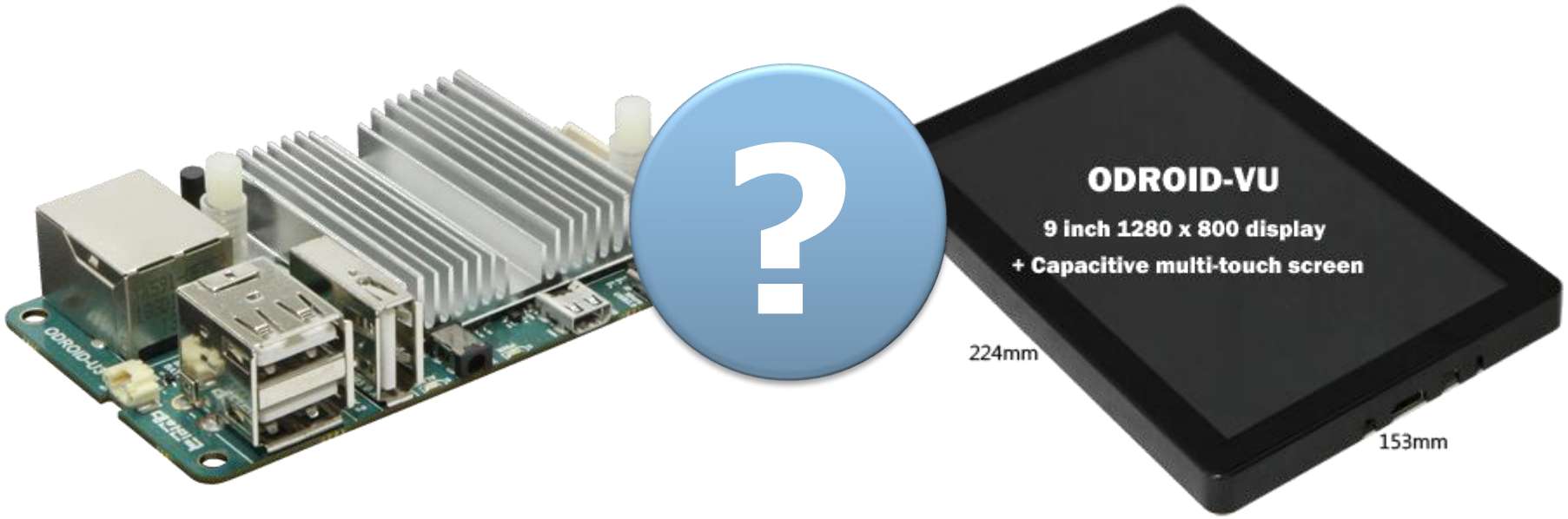
Sungkyunkwan University

Contents

- **Motivation and Concept**
- **Requirements**
- **Design**
- **Implementation**
  - Virtual Input Device Driver 제작
  - Tizen Service 개발 절차
  - Github에 공개하기

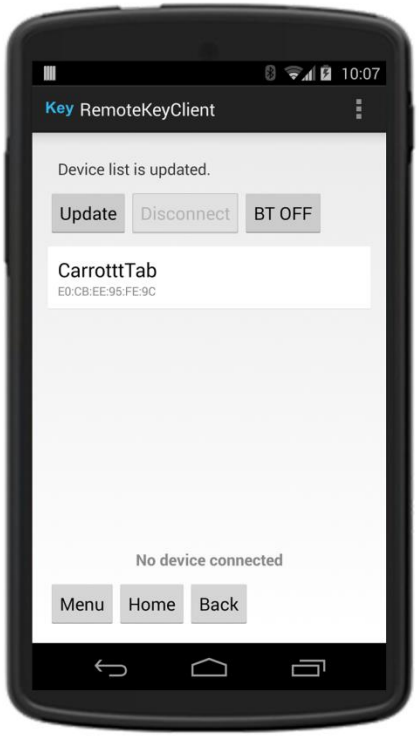
Motivation

- **ODROID-U3와 ODROID-VU에는 home key, back key, menu key로 사용할 수 있는 버튼이 없어서 기기를 조작하기 매우 불편함**



Concept

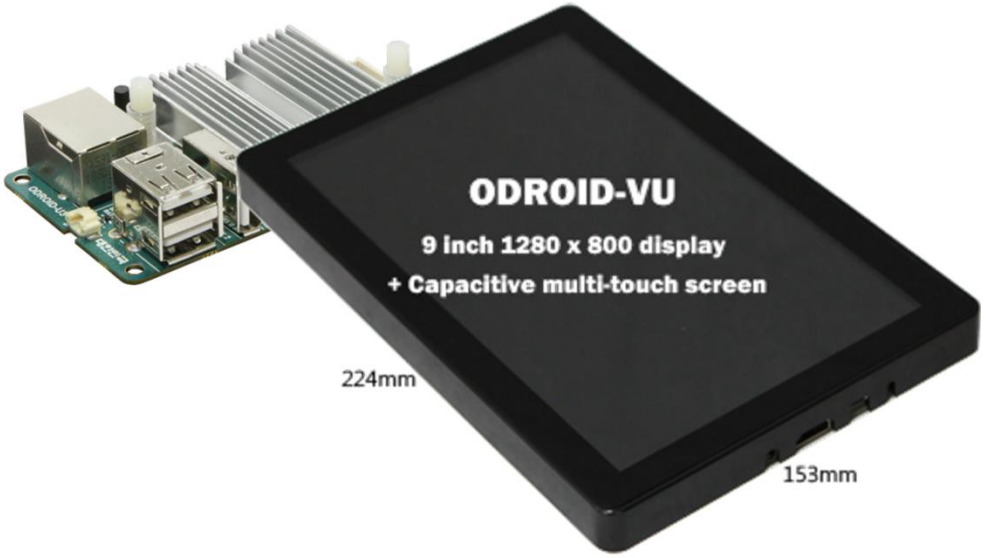
- **Android 스마트 폰으로 원격으로 Tizen 장치를 조종하는 프레임워크 제작**



Android Phone

Wireless

Message



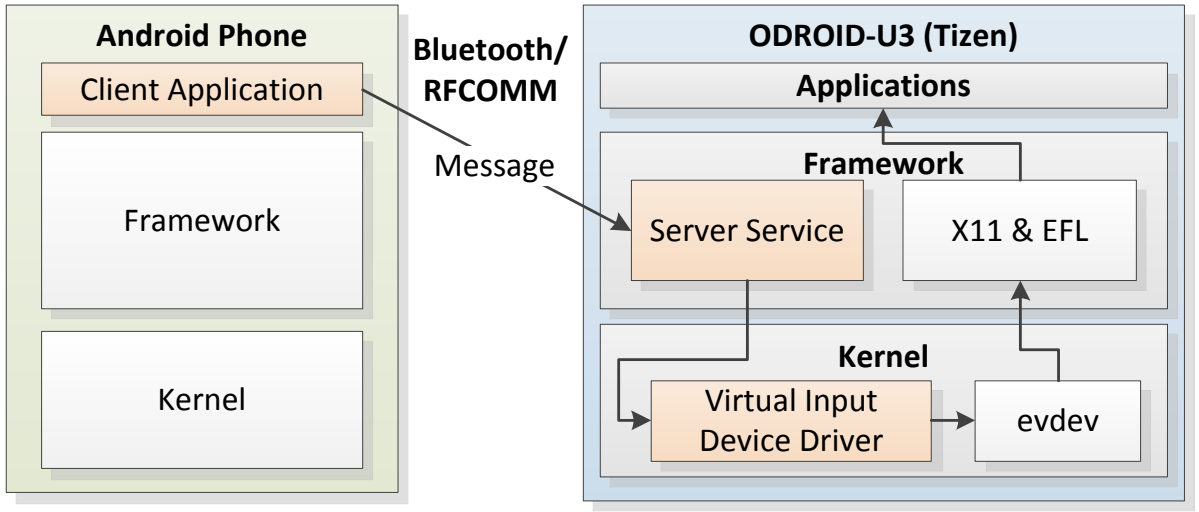
ODROID-U3 (Tizen)

Requirements

- **Android 장치와 Tizen 장치를 무선으로 연결**
  - Bluetooth, RFCOMM 프로토콜
  - Android과 Tizen의 Bluetooth framework 사용
- **Tizen 장치에 키 입력 생성**
  - input device driver 제작
- **Tizen Service로 제작**
  - Systemd 설정 변경
- **손쉬운 Debugging**
  - Dlog 사용
- **Pacakge로 손쉽게 배포 가능해야 함**
  - RPM package로 묶기

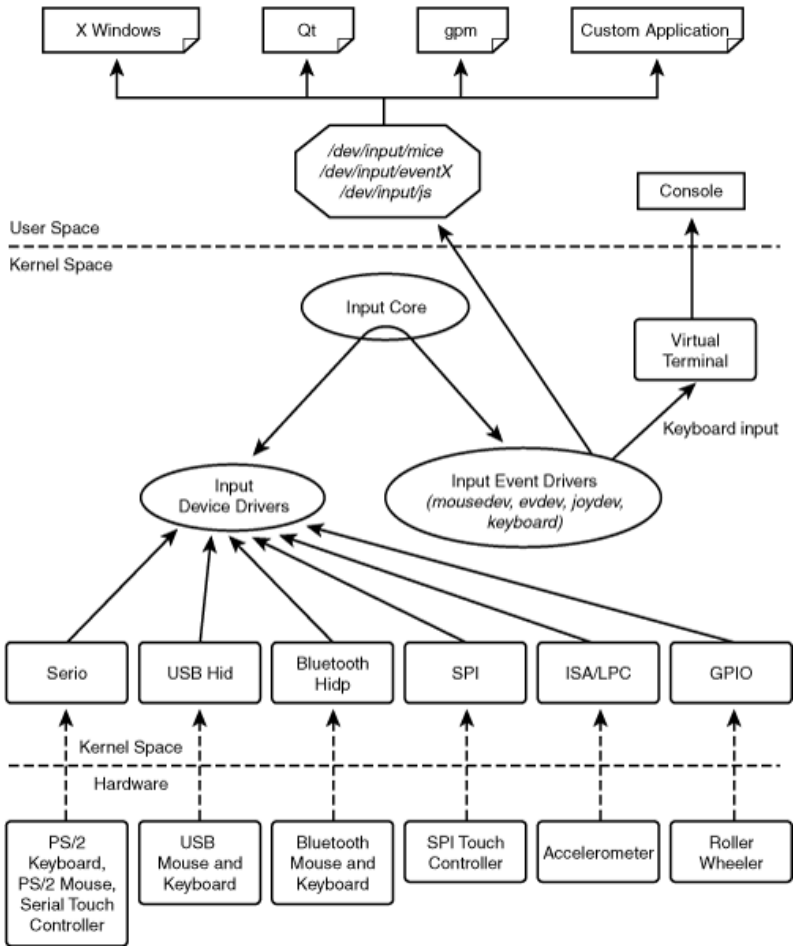
Design

- **Client (Android Application)**
  - UI for end users, Tizen 장치와 Bluetooth 연결
- **Server Service (Tizen Framework Service)**
  - Android 장치와 Bluetooth 연결, virtual input D/D에 명령
- **Virtual Input Device Driver (Kernel Module)**
  - Server Service로부터 명령이 들어오면 key code 생성



Virtual Input Device Driver 제작

- **Input Device Driver**
  - platform device로 host-level device driver 제작
  - host driver가 evdev (event device driver)와 연결
  - evdev가 udev에 event 전송
  - udev에서 xinput으로 device event 전송
  - xinput에서 확인 가능



Host Device Driver


```
static int __init homekey_init(void){

        int error;

        homekey = platform_device_register_simple("homekey", -1, NULL, 0);
        if(IS_ERR(homekey)){
                PTR_ERR(homekey);
                printk("homekey_init : error during platfrom device register\n");
        }

        sysfs_create_group(&homekey->dev.kobj, &homekey_attr_group);

        input = input_allocate_device();
        if(!input){
                printk("allocation failed\n");
        }

        input->evbit[0] = BIT_MASK(EV_KEY);
        input->keybit[BIT_WORD(KEY_XFER)] = BIT_MASK(KEY_XFER);

        input->name = "homekey";
        input->id.bustype = BUS_HOST;
        input->id.vendor = 0x0001;
        input->id.product = 0x0001;
        input->id.version = 0x0001;

        error = input_register_device(input);
        if(error){
                printk("input device register failed\n");
                goto err_free_dev;
        }

        return 0;

err_free_dev:
        input_free_device(input);
        return error;

}
```

Platform Device 등록

Sysfs event 등록

Input device 구조체 할당

Keymap 등록 및 bus type, name 등록

evdev에 input device 등록

Host Device Driver (cont)


```
static ssize_t homekey_click(struct device *dev, struct device_attribute *attr,
                             const char *buffer, size_t count){
        input_report_key(input, KEY_XFER, 1);
        input_report_key(input, KEY_XFER, 0);
        input_sync(input);
        return count;
}

DEVICE_ATTR(coordinates, 0644, NULL, homekey_click);

static struct attribute *homekey_attrs[] = {
        &dev_attr_coordinates.attr,
        NULL
};

static struct attribute_group homekey_attr_group = {
        .attrs = homekey_attrs,
};
```

Input device에 KEY_XFER (key 147) input event 발생시킴.
Input sync를 통해 input event sync

sysfs에 등록된 homekey device에 입력값을 전송할 수 있도록 함.

```
sh-4.1$ xinput
⎡ Virtual core pointer                    id=2    [master pointer  (3)]
⎜   ↳ Virtual core XTEST pointer          id=4    [slave  pointer  (2)]
⎜   ↳ Touchscreen                         id=7    [slave  pointer  (2)]
⎣ Virtual core keyboard                   id=3    [master keyboard (2)]
    ↳ Virtual core XTEST keyboard         id=5    [slave  keyboard (3)]
~ Gesture                                 id=6    [floating slave]
~ gpio_keys.6                             id=8    [floating slave]
~ homekey                                 id=9    [floating slave]
~ Touchscreen subdev 1                    id=10   [floating slave]
~ Touchscreen subdev 2                    id=11   [floating slave]
```

Tizen Service 개발 절차 (1/7)

1. **Git repository 만들기**
2. **각종 설정 파일 만들기**
    1. RPM package specification 파일
    2. 빌드 설정 파일(CMakelists.txt)
    3. Pkgconfig 설정 파일
    4. SMACK manifest 파일
    5. Systemd service 파일
3. **코딩**
4. **빌드**
5. **패키지 설치**

Tizen Service 개발 절차 (2/7)

1. Git repository 만들기

- Tizen 소스 코드 중, framework/system이 가장 적합한 위치라고 판단

1. $ cd ~/tizen-platform/framework/system
2. $ mkdir remote-key-framework
3. $ cd remote-key-framework
4. $ git init

```
Initialized empty Git repository in
/home/user/repository/tizen/2.2/framework/system/remote-key-framework/.git/
```

Tizen Service 개발 절차 (3/7)

2. 각종 설정 파일 만들기

– Tizen project들은 최소한 다음 파일들이 필요

- AUTHORS: 제작자 목록
- LICENSE: 라이선스 명세
- CMakeLists.txt: cmake 설정 파일(전반적인 빌드 설정)
- <project-name>.manifest: SMACK manifest 파일
- packaging
  - <project-name>.manifest: SMACK manifest 파일
  - <project-name>.spec: RPM package specification 파일
- <sub project 1>: 세부 디렉토리
  - CMakeLists.txt: cmake 설정 파일(이 디렉토리의 빌드 설정)
  - include
    - 헤더 파일들
  - src
    - 소스 코드 파일들
- <sub project 2>, <sub project 3>, …

Tizen Service 개발 절차 (4/7)

– Remote Key Service Project의 경우:

- AUTHORS: 제작자 목록
- LICENSE: 라이선스 명세
- CMakeLists.txt: cmake 설정 파일(전반적인 빌드 설정)
- remote-key-framework.manifest: SMACK manifest 파일
- packaging
  - remote-key-framework.manifest: SMACK manifest 파일
  - remote-key-framework.spec: RPM package specification 파일
  - remote-key-framework.service: Systemd service 설정 파일
- server: 세부 디렉토리
  - CMakeLists.txt: cmake 설정 파일(이 디렉토리의 빌드 설정)
  - include
    - common.h: 공통 헤더 파일
  - src
    - main.cpp: 메인 소스 코드
    - common.cpp: DLog 관련 매크로 소스 코드

Tizen Service 개발 절차 (5/7)

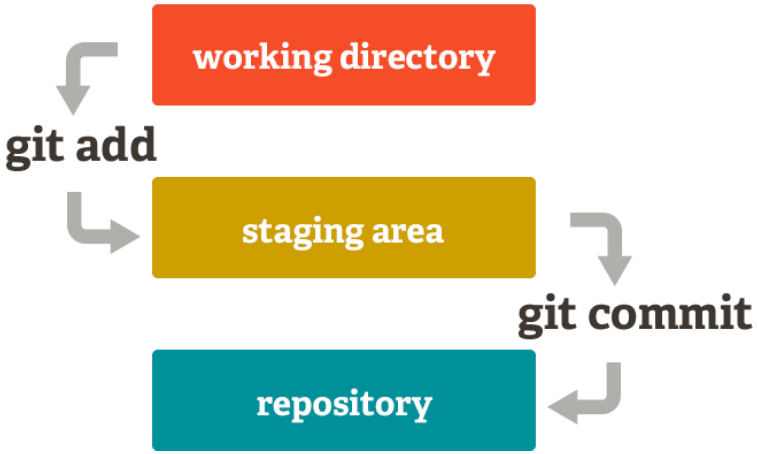
3. 코딩

- Project 디렉토리 내에 여러 sub-project 디렉토리를 만들어서 코딩
- **기존 Tizen framework의 source code를 참고**하면 많은 도움이 됨.
- Remote Key Service Framework에서 참고한 project:
  - Bluetooth Native API (framework/api/bluetooth)
    - Android와 동일하게 RFCOMM/Bluetooth 지원
    - API가 명시된 header 파일들(framework/api/bluetooth/include)
    - Test case 코드(framework/api/bluetooth/test): 실제로 API를 사용하는 예제를 보여줌
  - Bluetooth WRT API (framework/web/wrt-plugin-tizen/bluetooth)
  - Bluetooth Framework (framework/connectivity/bluetooth-frwk)

Tizen Service 개발 절차 (6/7)

4. 빌드

- Commit하지 않은 working directory를 즉시 빌드
  - $ gbs build -A armv7l --include-all
- Git repository에 commit한 소스 코드를 빌드
  - $ gbs build -A armv7l



Tizen Service 개발 절차 (7/7)

5. 패키지 설치

– 빌드 후, 다음 경로에 rpm 패키지가 나옴

- ~/GBS-ROOT/local/repos/tizen2.2/armv7l/RPMS/remote-key-framework-1.0.1.rpm (binary package)
- ~/GBS-ROOT/local/repos/tizen2.2/armv7l/SRPMS/remote-key-framework-src-1.0.1.rpm (source package)

– $ sdb root on

– $ sdb push ~/GBS-ROOT/local/repos/tizen2.2/armv7l/RPMS/remote-key-framework-1.0.1.rpm /

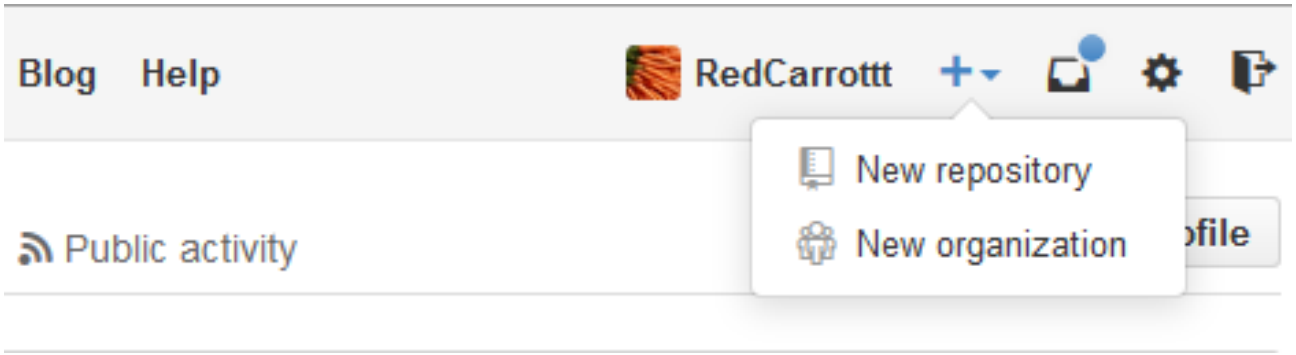
– $ sdb shell rpm -ivh --nodeps --force remote-key-framework-1.0.1.rpm

– $ sdb shell reboot -f

Github에 공개하기 (1/4)

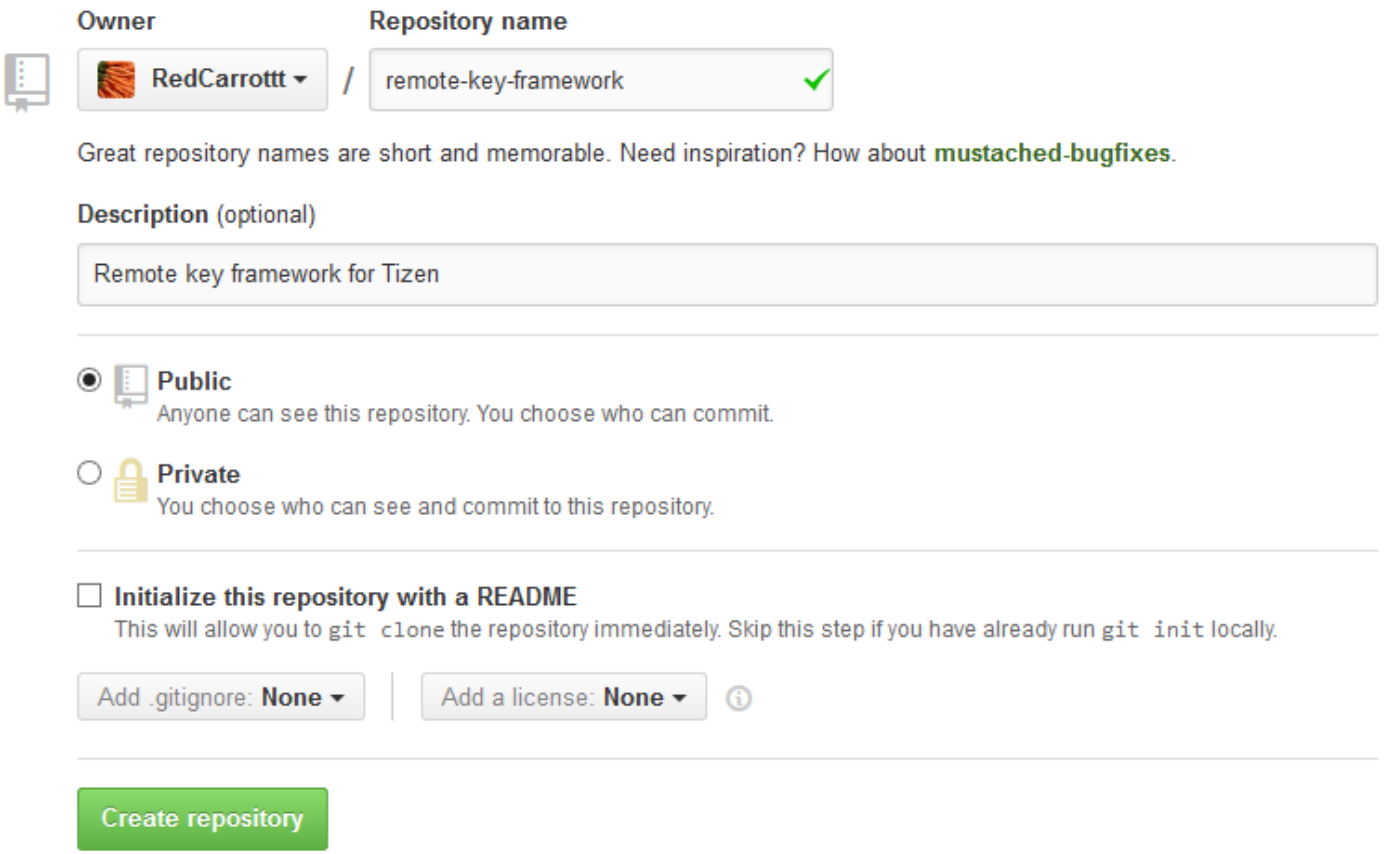
1. Github에 git repository 만들기

1. http://github.com 접속
2. "New repository" 클릭



Github에 공개하기 (2/4)

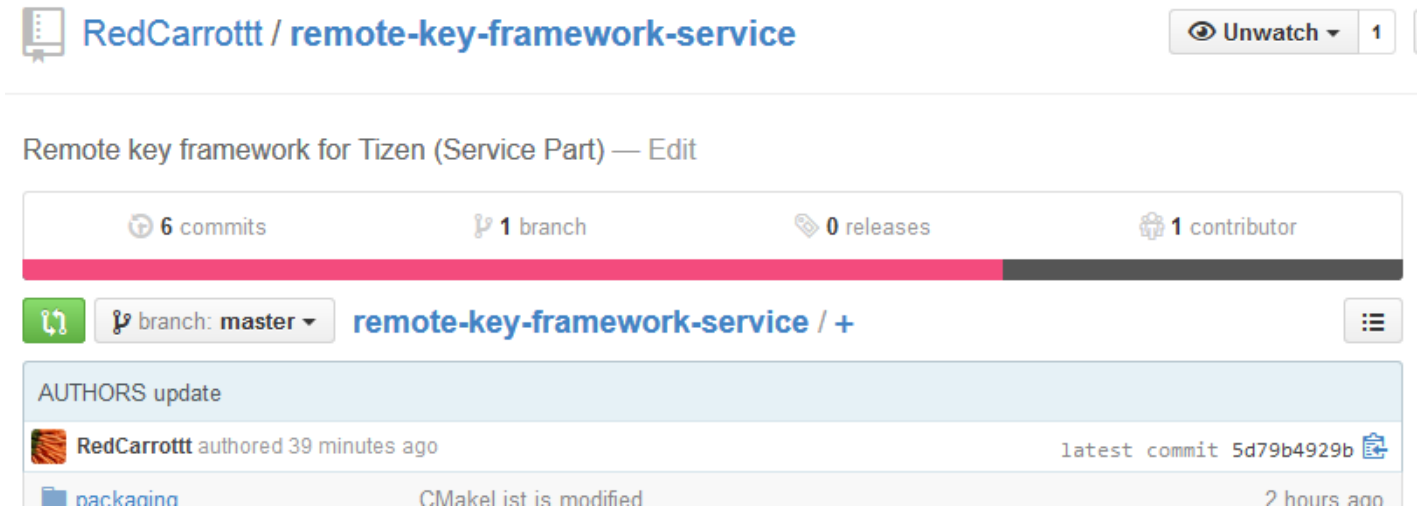
3. reposisotry 이름과 세부 설명 기입



Github에 공개하기 (3/4)

2. Git repository 등록하기

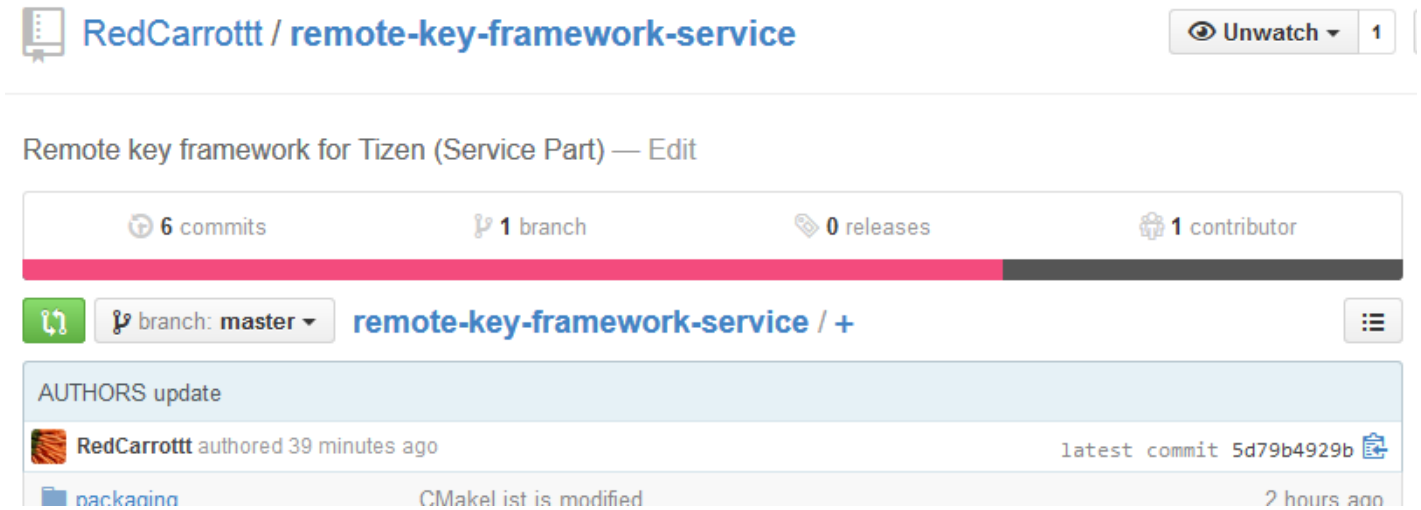
1. 내가 작업한 project 폴더로 이동
2. $ git remote add origin https://github.com/<user-name>/<project-name>.git
   - ex. $ git remote add origin https://github.com/RedCarrottt/remote-key-framework.git



Github에 공개하기 (4/4)

3. Git repository로 업로드

1. $ git push <remote name> <remote branch>
   1. ex. $ git push origin master
   - Local repository에 있는 현재 branch의 모든 내용이 remote repository의 master branch로 업로드 됨.



Remote Key FW 소스 코드 주소

- **Android Client Application**
  - https://github.com/RedCarrottt/remote-key-framework-client
- **Tizen Remote Key Server Service**
  - https://github.com/RedCarrottt/remote-key-framework-service
- **Tizen Virtual Key Device Driver**
  - https://github.com/wangmir/virtual_inputdevice