

---

---

# 통신용 PC 프로그래밍

Hancheol Cho

---

---

Qt ?

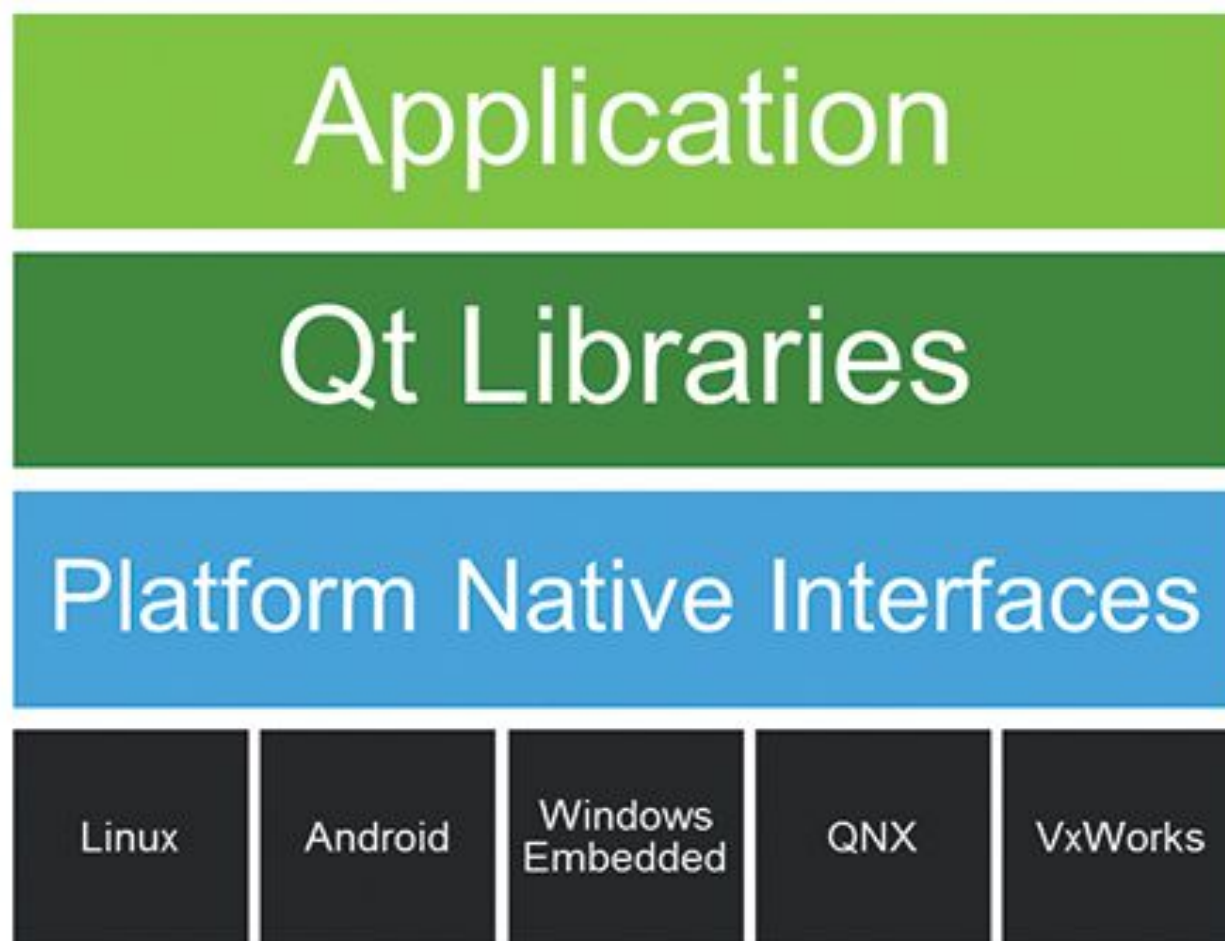
**Cross-platform**

**C++ library classes**

**Qt Creator IDE**

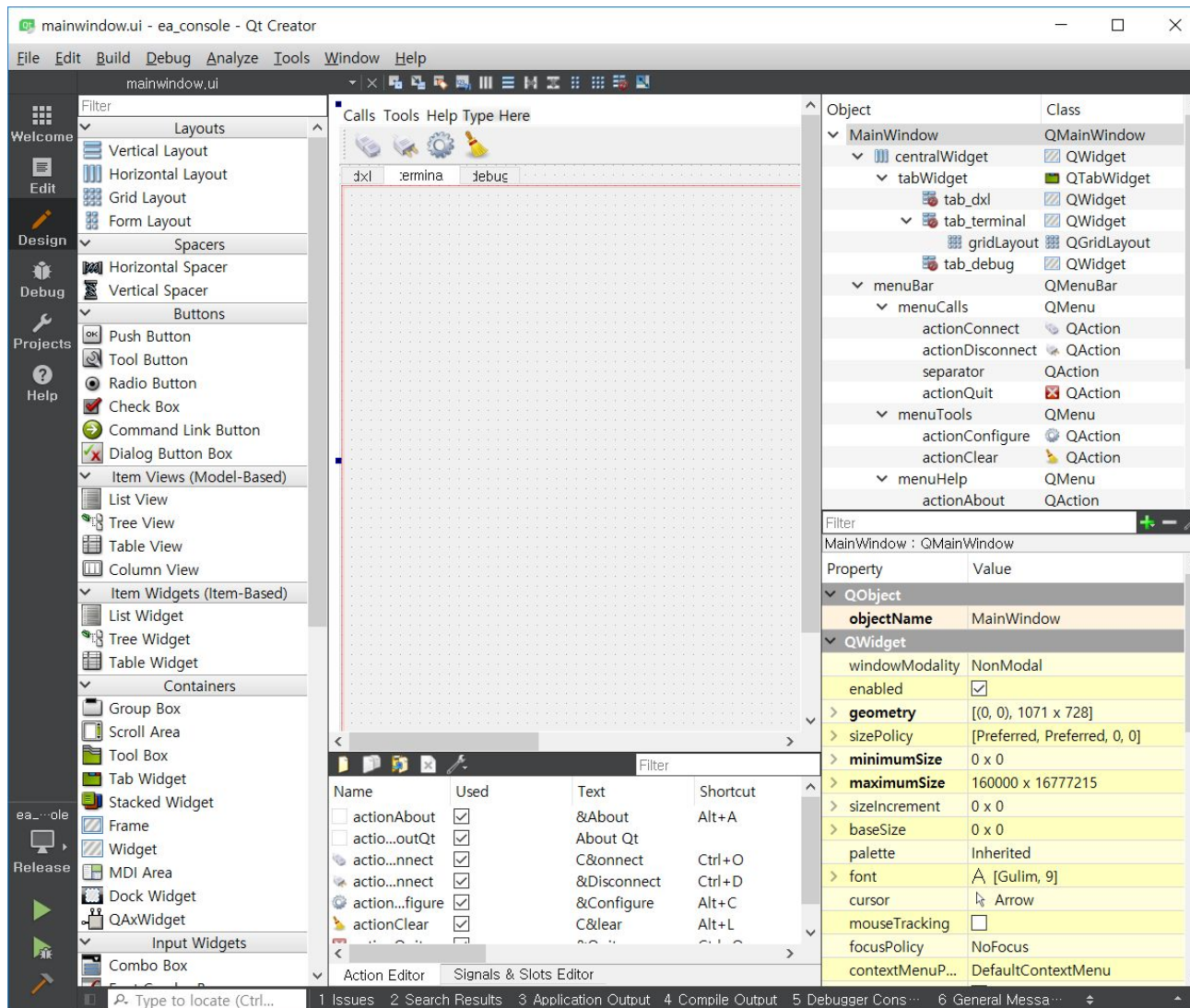
**Free to use**

Qt ?



**Why Qt ?**

# Qt creator IDE



# Qt 설치

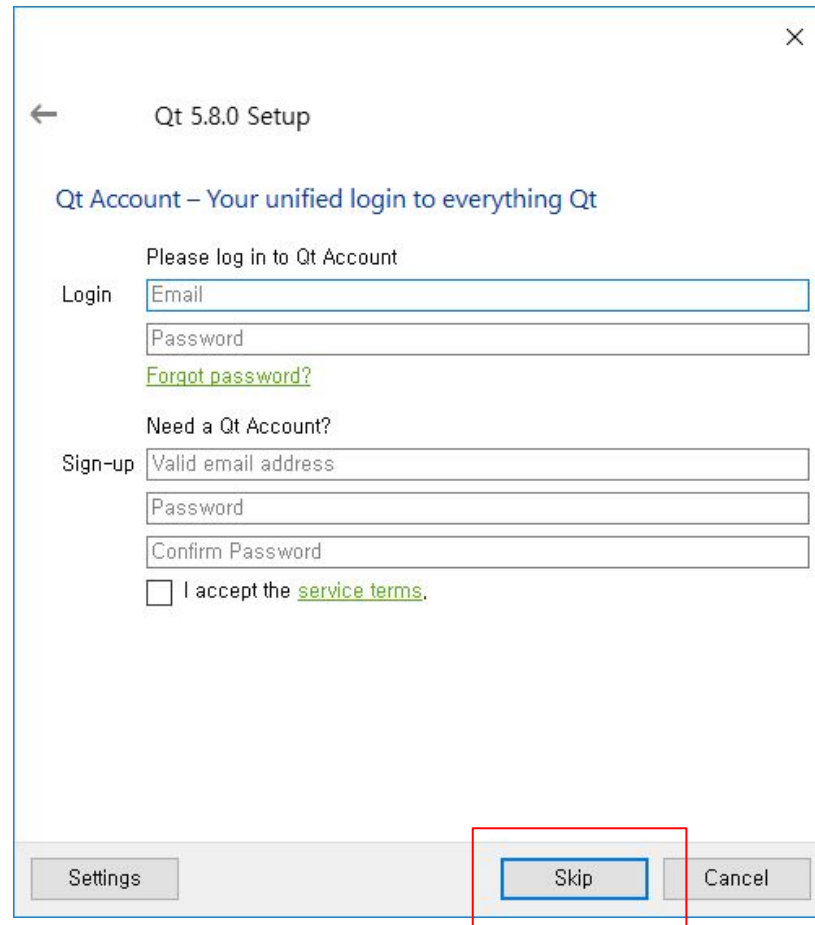
- QT 다운로드
  - <https://www.qt.io/download-open-source/#section-2> 에서 MinGW 버전으로 설치함

## Windows Host

- › Qt 5.8.0 for Windows 64-bit (VS 2015, 1.0 GB) (info)
- › Qt 5.8.0 for Windows 32-bit (VS 2015, 1.0 GB) (info)
- › Qt 5.8.0 for Windows 64-bit (VS 2013, 958 MB) (info)
- › Qt 5.8.0 for Windows 32-bit (VS 2013, 947 MB) (info)
- › Qt 5.8.0 for Windows 32-bit (MinGW 5.3.0, 1.2 GB) (info)
- › Qt 5.8.0 for Android (Windows 32-bit, 1.3 GB) (info)
- › Qt 5.8.0 for WinRT 32-bit (VS 2013, 1.2 GB) (info)
- › Qt 5.8.0 for WinRT 32-bit (VS 2015, 1.2 GB) (info)

# Qt 설치

- 계정 입력 없이 Skip하고 이후에는 기본값으로 설치함



Qt 5.8.0 Setup

Qt Account – Your unified login to everything Qt

Please log in to Qt Account

Login

Email

Password

[Forgot password?](#)

Need a Qt Account?

Sign-up

Valid email address

Password

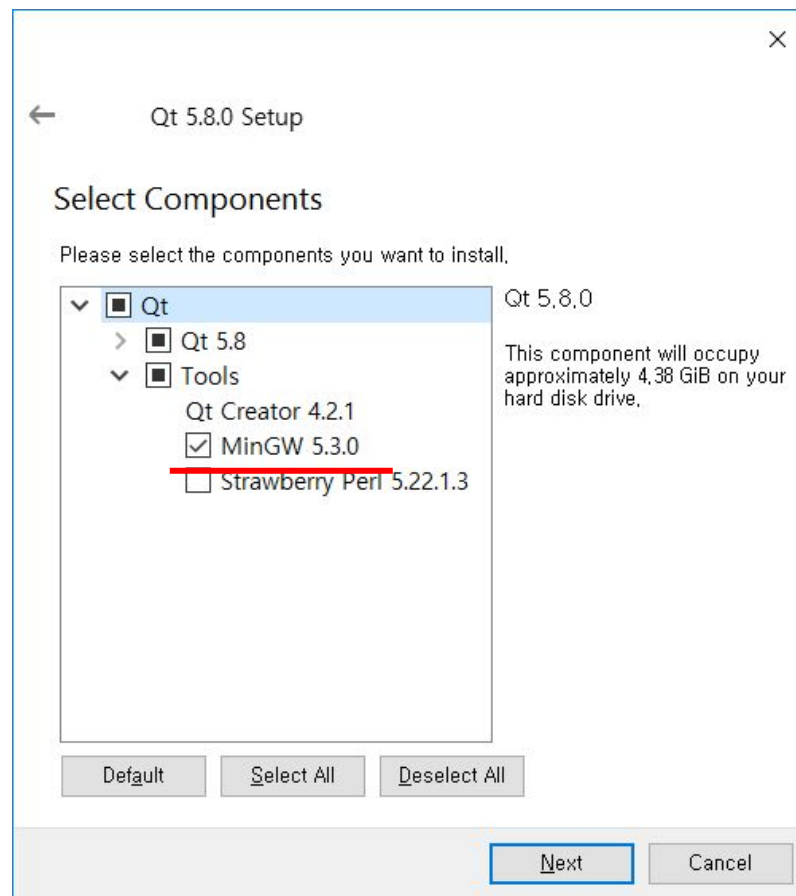
Confirm Password

☐ I accept the [service terms](#).

Settings Skip Cancel

# Qt 설치

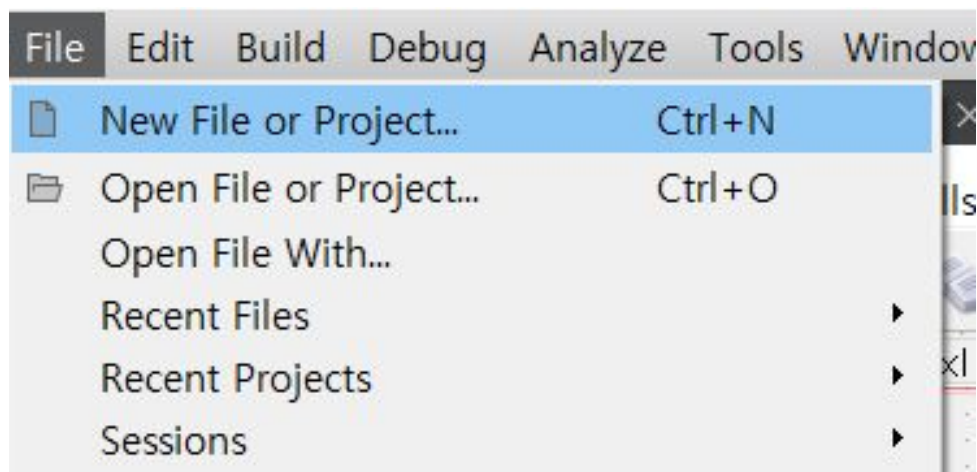
- Tools->MinGW 옵션 활성화





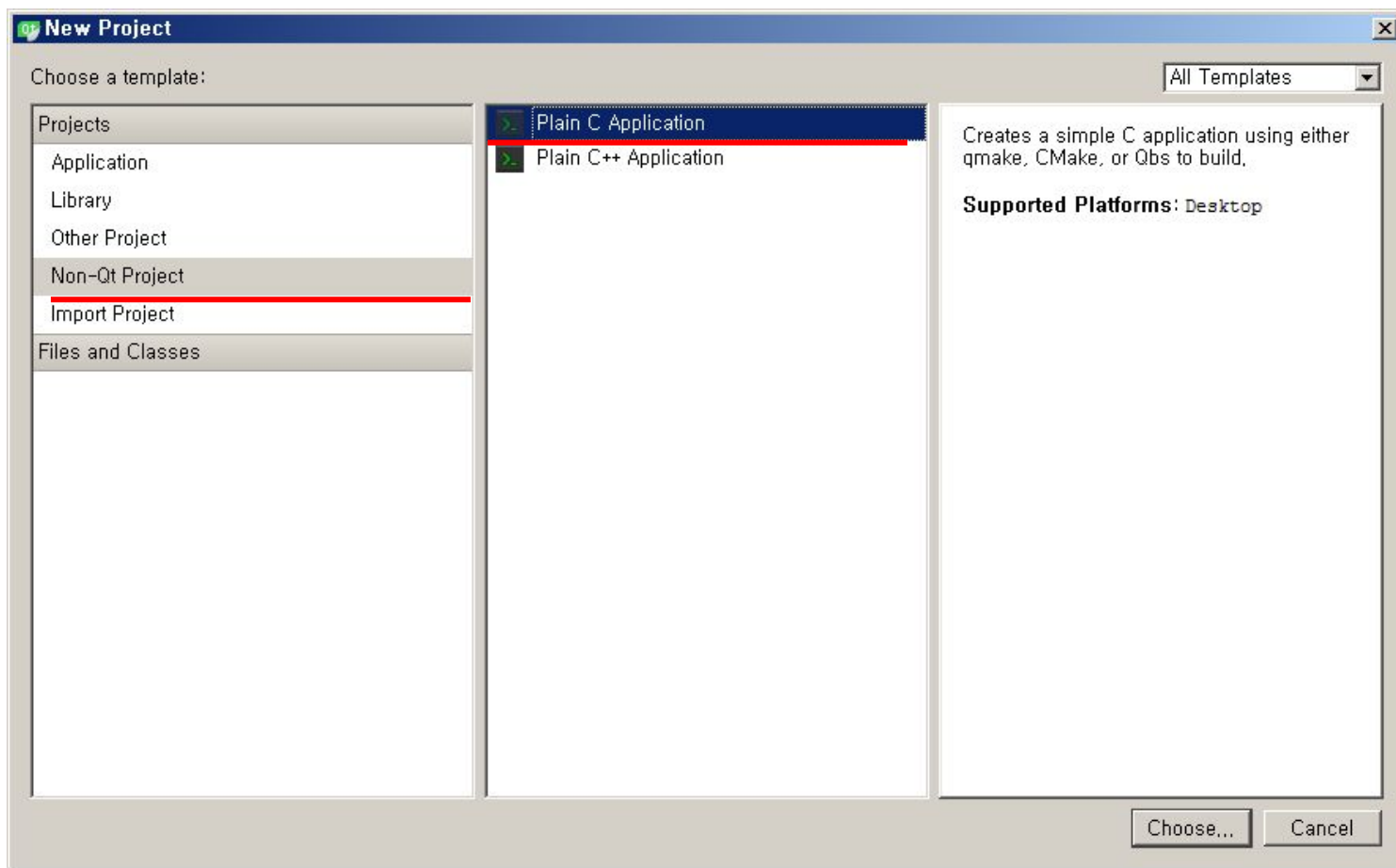
# Qt 프로젝트 생성

- File->New File or Project 선택

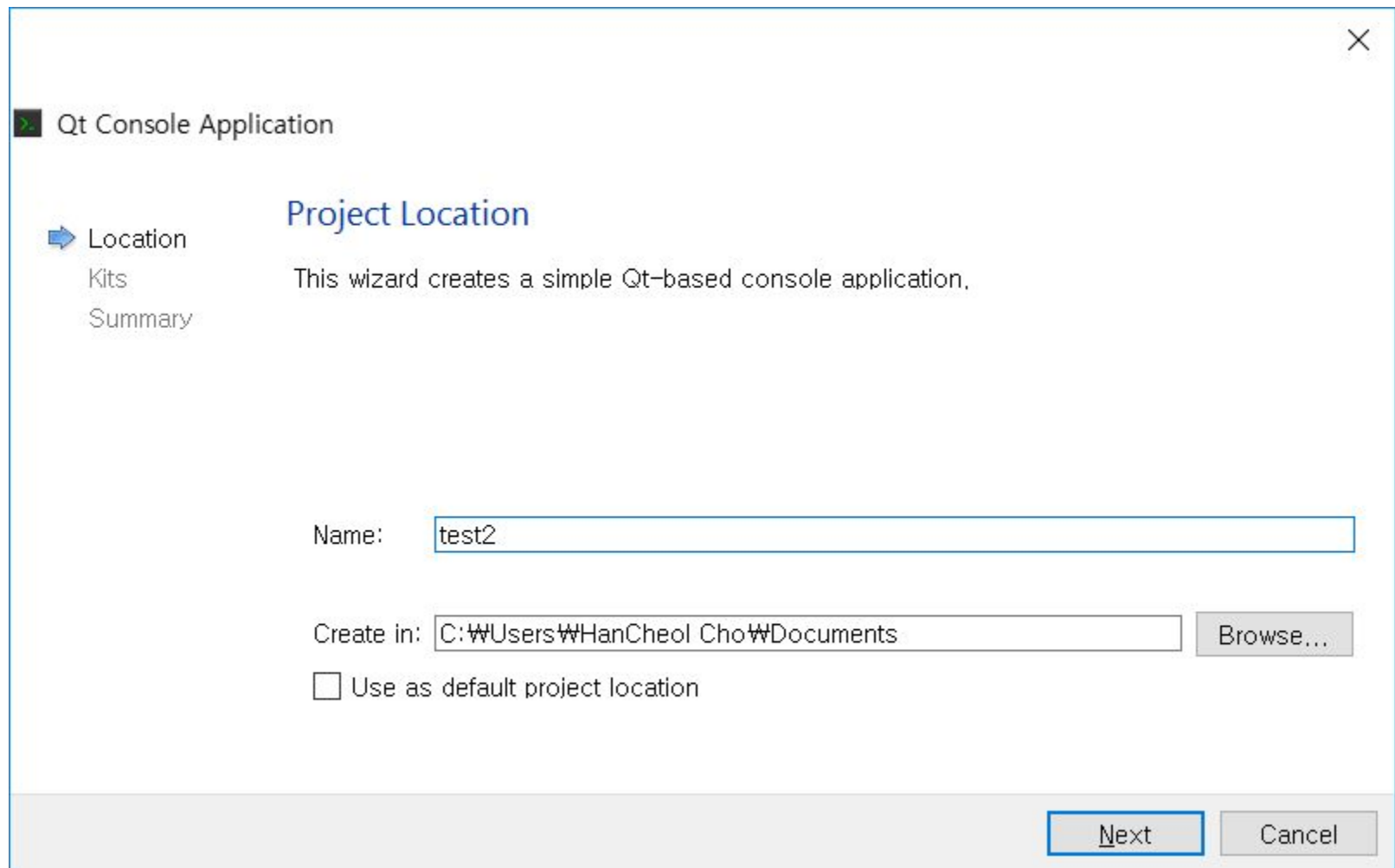


# Qt 프로젝트 생성

- Projects -> Non-Qt Project 선택
- Plain C Application 선택



# Qt 프로젝트 생성



Qt Console Application

Location  
Kits  
Summary

## Project Location

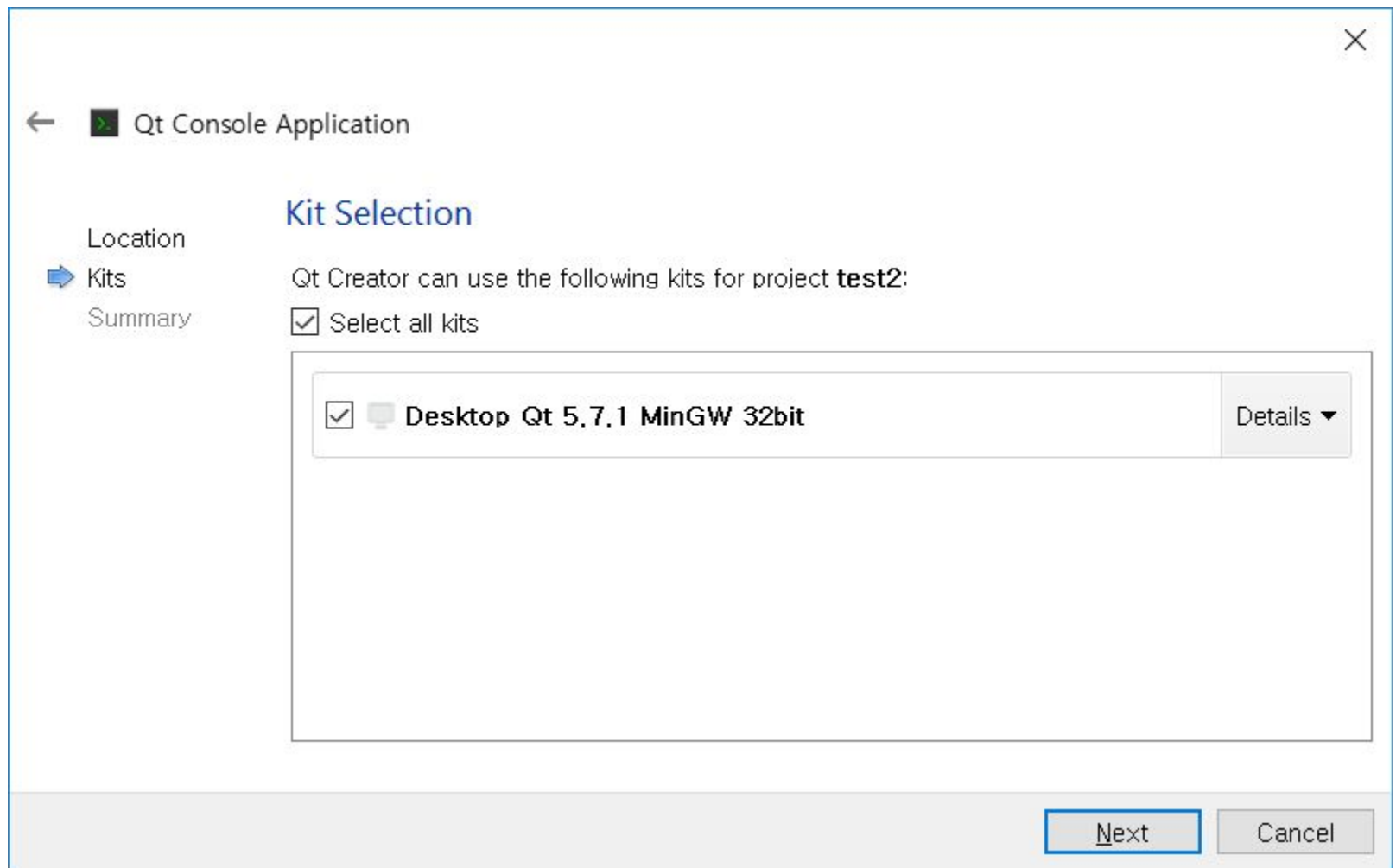
This wizard creates a simple Qt-based console application.

Name:

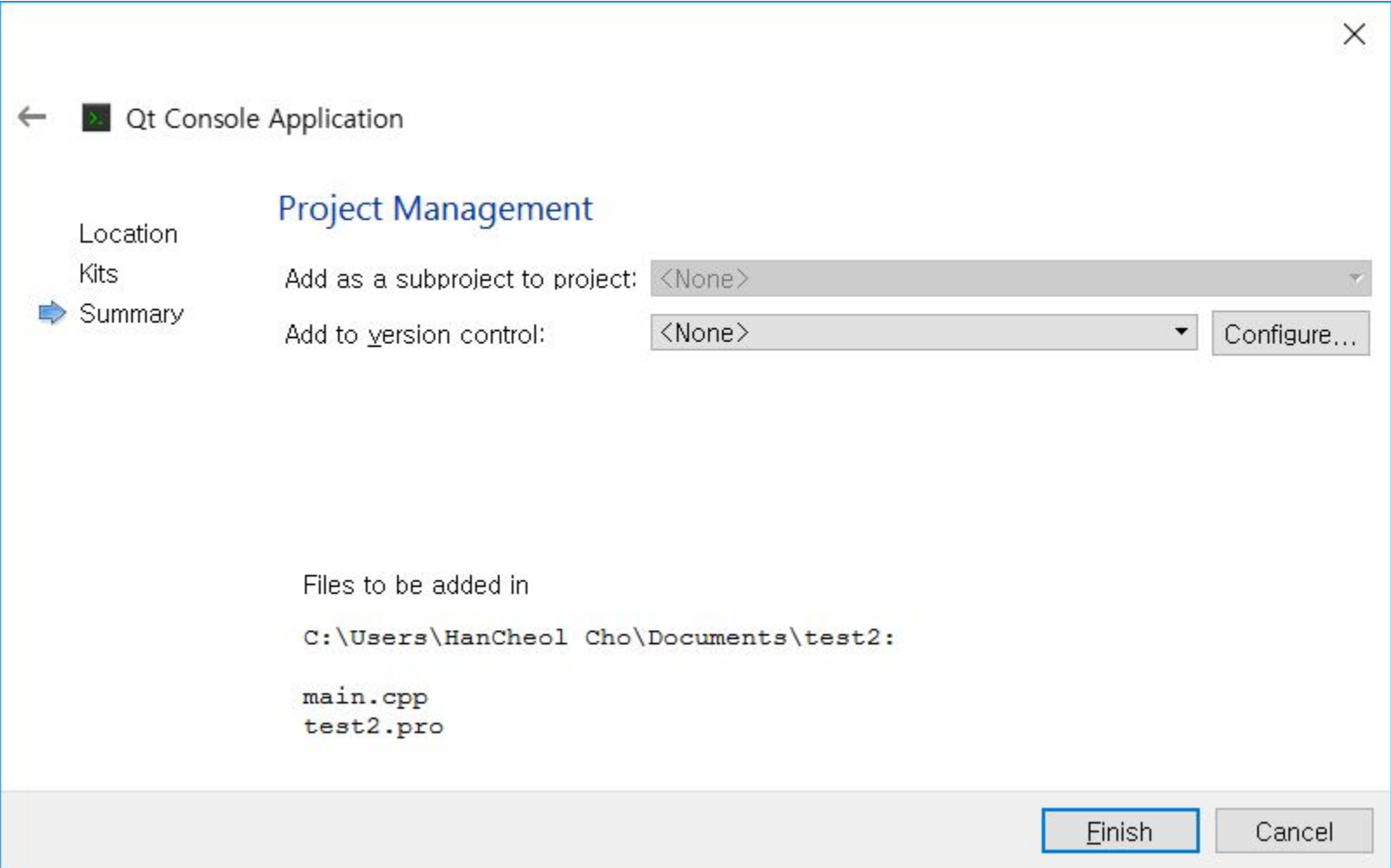
Create in:

☐ Use as default project location

# Qt 프로젝트 생성



# Qt 프로젝트 생성



Qt Console Application

Location  
Kits  
➔ Summary

### Project Management

Add as a subproject to project: <None>

Add to version control: <None> [Configure...](#)

Files to be added in

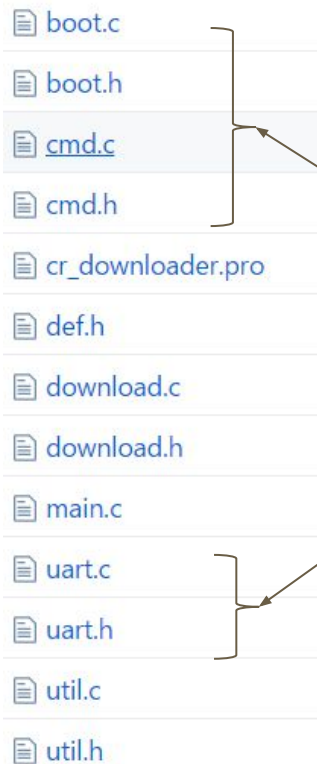
C:\Users\HanCheol Cho\Documents\test2:

main.cpp  
test2.pro

[Finish](#) [Cancel](#)

# Downloader

- 펌웨어에서 작성한 코드를 재사용 함



펌웨어에서 작성한 코드 일부 수정

# Downloader

- 펌웨어의 함수 인터페이스를 유지함
  - 코드의 일관성을 유지할 수 있음

```
bool uartInit(void);
```

```
uint32_t    uartOpen(uint8_t channel, uint32_t baud);  
uint32_t    uartAvailable(uint8_t channel);  
void        uartWaitForEnable(uint8_t channel, uint32_t timeout);  
void        uartPutch(uint8_t channel, uint8_t ch);  
uint8_t     uartGetch(uint8_t channel);  
int32_t     uartWrite(uint8_t channel, uint8_t *p_data, uint32_t length);  
uint8_t     uartRead(uint8_t channel);  
int32_t     uartPrintf(uint8_t channel, const char *fmt, ...);  
int32_t     uartPrint(uint8_t channel, uint8_t *p_str);
```

<펌웨어 코드>

```
bool uartInit(void);
```

```
uint32_t    uartOpen(uint8_t channel, char *port_name, uint32_t baud);  
uint32_t    uartClose(uint8_t channel);  
uint32_t    uartAvailable(uint8_t channel);  
void        uartWaitForEnable(uint8_t channel, uint32_t timeout);  
void        uartPutch(uint8_t channel, uint8_t ch);  
uint8_t     uartGetch(uint8_t channel);  
int32_t     uartWrite(uint8_t channel, uint8_t *p_data, uint32_t length);  
uint8_t     uartRead(uint8_t channel);  
int32_t     uartPrintf(uint8_t channel, const char *fmt, ...);  
int32_t     uartPrint(uint8_t channel, uint8_t *p_str);
```

<Downloader 코드>

# Downloader

- 부트로더와 통신 라이브러리는 그대로 사용 가능
  - 하드웨어와 의존성을 줄이면 코드 재사용성을 높일 수 있음

```
void cmdInit(cmd_t *p_cmd);
```

```
void cmdBegin(cmd_t *p_cmd, uint8_t ch, uint32_t baud);
```

```
bool cmdReceivePacket(cmd_t *p_cmd);
```

```
void cmdSendCmd(cmd_t *p_cmd, uint8_t cmd, uint8_t *p_data, uint32_t length);
```

```
void cmdSendResp(cmd_t *p_cmd, uint8_t err_code, uint8_t *p_data, uint32_t length);
```



## Mission #1

**“test.exe com1 115200 1” 을  
입력하면 아래 화면이 나오도록 작성**

Port : com1

Baud : 115200 bps

Mode : 1

## Mission #2

**“test.exe com1 115200 파일명 1”  
을 입력하면 아래 화면이 나오도록  
작성**

File : test.bin

File Size : 10 KB

Port : com1

Baud : 115200 bps

Mode : 1