

# Tinker board

IoT를 위한 최선의 엣지 단말기



MAKER SPACE  
**G·CAMP**

## wiringPi Bulid & Push

# Contents

- Ubuntu 안드로이드 스튜디오 & ADB 설치
- WringTinker & NDK 다운로드 및 설정
- 빌드파일 실행
- ADB를 이용하여 빌드된 파일 push

# wiringPi TinkerBoard란?

- wiringPi라는 라즈베리파이 Linux에서 GPIO를 SW적으로 제어하게 해주는 C/C++ 라이브러리를 TinkerBoard로 포팅.
- wiringPi TinkerBoard에 NDK를 이용하면 Android에서 GPIO 사용 가능
- 현재 Tinker Board 및 Tinker Board 2에 포팅 완료
- Tinker Edge R은 Tinker Board 2와 같은 AP로 wiringPi가 정상 작동하는 줄 알았으나 현재 라이브러리 빌드 후 System Halt 문제가 있으며 해결 중

# Ubuntu에서 안드로이드 스튜디오 설치

- install Java SDK

- `$ sudo apt install openjdk-11-jdk`
- when installed, `$ java --version`

- install android Studio

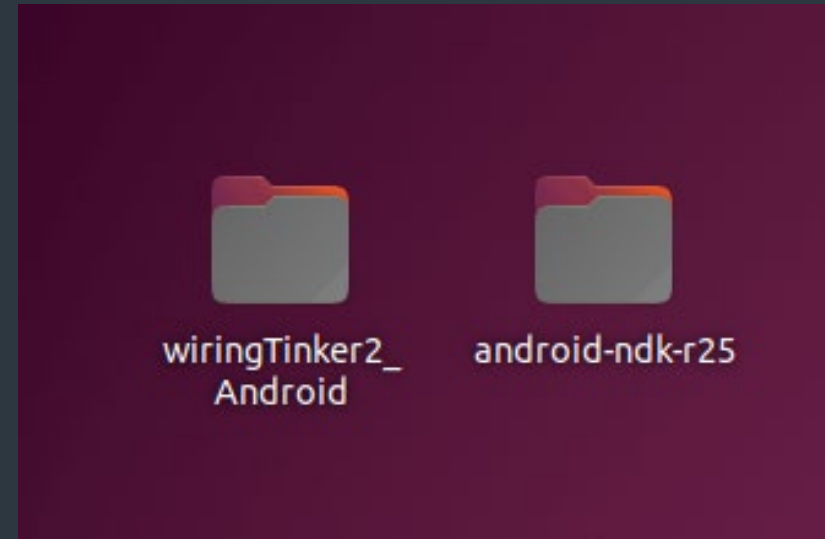
- `$sudo snap install android-studio --classic`

# Ubuntu에서 ADB 설치

- 터미널에서 root user로 변경
  - `$ sudo su`
- `install adb`
  - `$ apt-get install android-tools-adb`

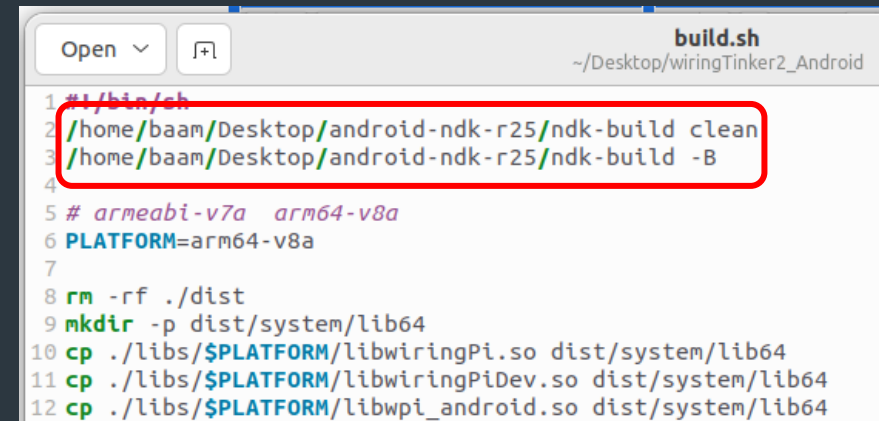
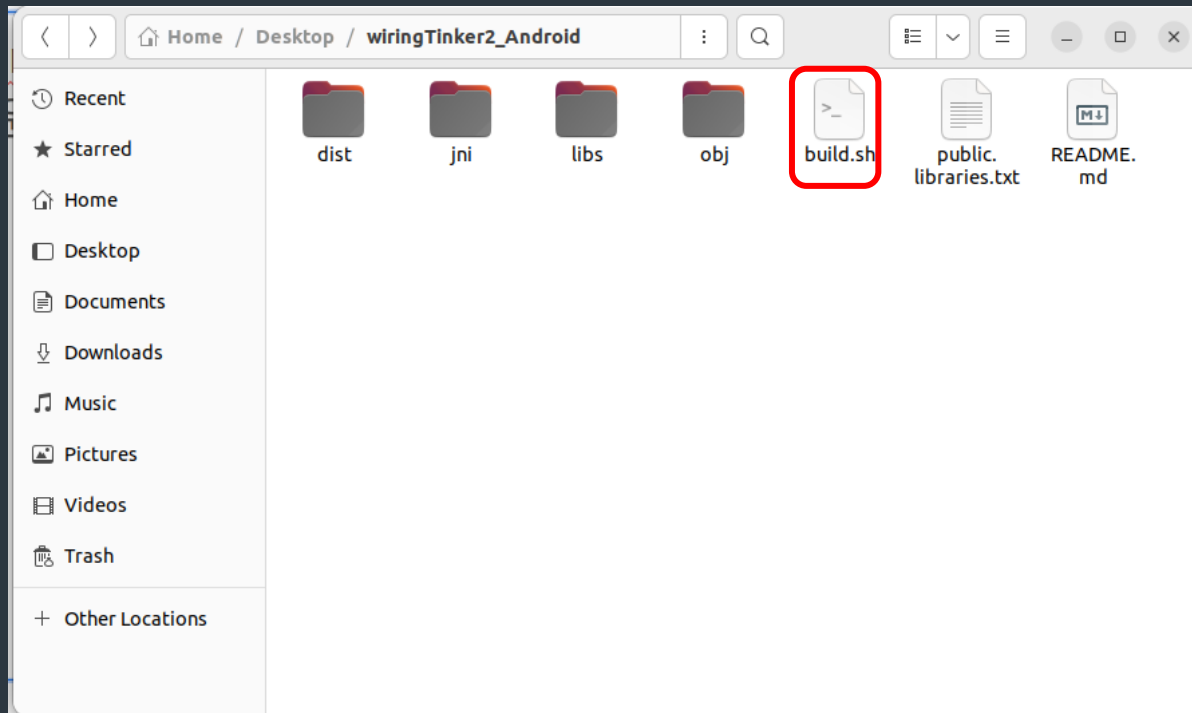
# WiringTinker2와 NDK 다운로드

- wiringTinker2\_Android 다운로드 ([link](#))
- NDK Linux version 다운로드 ([link](#))



# build.sh 파일 수정

- wiringTinker2\_Android 폴더 내의 build.sh 파일 열기
- 2번째 line과 3번째 line을 다운로드 받은 android-ndk 디렉토리 위치로 수정





```
baam@baam-Latitude-5290-2-in-1: ~/Desktop/wiringTinker2_...
baam@baam-Latitude-5290-2-in-1:~/Desktop/wiringTinker2_Android$ ./build.sh
[arm64-v8a] Clean      : gpio [arm64-v8a]
[arm64-v8a] Clean      : led [arm64-v8a]
[arm64-v8a] Clean      : spiSpeed [arm64-v8a]
[arm64-v8a] Clean      : tinker_gpio_init [arm64-v8a]
[arm64-v8a] Clean      : tinker_i2c [arm64-v8a]
[arm64-v8a] Clean      : tinker_uart_echo [arm64-v8a]
[arm64-v8a] Clean      : wiringPi [arm64-v8a]
[arm64-v8a] Clean      : wiringPiDev [arm64-v8a]
[arm64-v8a] Clean      : wpi_android [arm64-v8a]
[arm64-v8a] Compile    : gpio <= gpio.c
[arm64-v8a] Compile    : gpio <= readall.c
[arm64-v8a] Compile    : gpio <= pins.c
[arm64-v8a] Compile    : wiringPi <= wiringSerial.c
[arm64-v8a] Compile    : wiringPi <= wiringPi.c
In file included from jni/wiringPi/wiringPi.c:76:
In file included from jni/wiringPi/wiringPi.h:28:
In file included from jni/wiringPi/wiringTB.h:4:
jni/wiringPi/RKIO.h:8:9: warning: 'DEBUG' macro redefined [-Wmacro-redefined]
#define DEBUG(args...)
      ^
<command line>:4:9: note: previous definition is here
#define DEBUG 1
      ^
```

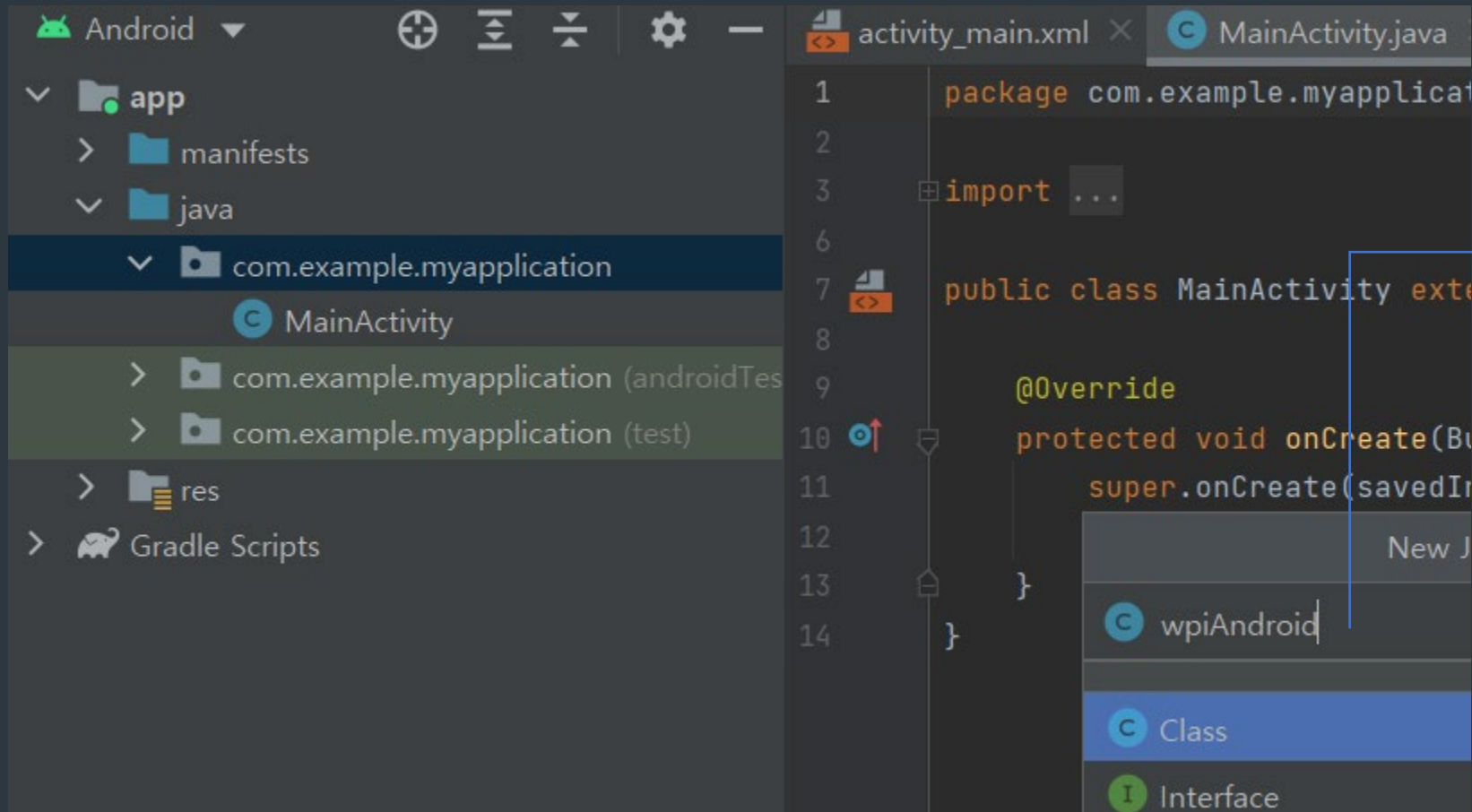


# ADB를 이용해 파일 빌드된 파일 push

- 터미널에서 wiringTinker2\_Android 디렉토리 위치로 가기
- 팅커보드를 PC와 연결
- “\$ adb devices” 커맨드를 입력하여 연결상태 확인
- 아래 커맨드를 터미널에 입력
  - \$ adb root
  - \$ adb remount
  - \$ adb push dist/system/lib64/libwiringPi.so /vendor/lib64/libwiringPi.so
  - \$ adb push dist/system/lib64/libwiringPiDev.so /vendor/lib64/libwiringPiDev.so
  - \$ adb push dist/system/lib64/libwpi\_android.so /vendor/lib64/libwpi\_android.so
  - \$ adb push dist/system/bin/gpio /vendor/bin/gpio
  - \$ adb push dist/system/etc/public.libraries.txt /system/etc/public.libraries.txt

# GPIO 사용

- Android에서 libwpi\_android.so을 이용하여 Tinker Board의 GPIO를 제어 가능케 함.



- wpiAndroid.java 파일 생성

# GPIO 사용

- wpiAndroid.java

```
public final class wpiAndroid {  
    static {  
        System.loadLibrary( libname: "wpi_android");  
    }  
    static public native int wiringPiSetup();  
    static public native int wiringPiSetupSys();  
    static public native int digitalRead(int port);  
    static public native void digitalWrite(int port, int onoff);  
    static public native void pullUpDnControl(int port, int pud);  
    static public native void pinMode(int port, int mode);  
    static public native int softPwmCreate(int port, int value, int range);  
    static public native void softPwmWrite (int port, int value) ;  
    static public native void softPwmStop (int port) ;  
}
```

- Class의 각 GPIO 기능 정의

# GPIO 사용

- MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;  
import static com.jw.wiringpi.wpiAndroid.digitalRead;  
import static com.jw.wiringpi.wpiAndroid.digitalWrite;  
import static com.jw.wiringpi.wpiAndroid.pinMode;  
import static com.jw.wiringpi.wpiAndroid.pullUpDnControl;  
import static com.jw.wiringpi.wpiAndroid.wiringPiSetup;
```

```
if (wiringPiSetup() != 0) {  
    Log.e(TAG, "Error setting up wiringPi!");  
}
```

- wpiAndroid.java에서 정의한 각 GPIO 기능 라이브러리 import
- GPIO 기능 사용 에러시 에러메세지 표시

# GPIO 사용

- MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    public static final int INPUT = 0;  
    public static final int OUTPUT = 1;  
    public static final int PWM_OUTPUT = 2;  
    public static final int GPIO_CLOCK = 3;  
    public static final int SOFT_PWM_OUTPUT = 4;  
    public static final int SOFT_TONE_OUTPUT = 5;  
    public static final int PWM_TONE_OUTPUT = 6;  
  
    public static final int PUD_OFF = 0;  
    public static final int PUD_DOWN = 1;  
    public static final int PUD_UP = 2;  
  
    public static final int INT_EDGE_SETUP = 0;  
    public static final int INT_EDGE_FALLING = 1;  
    public static final int INT_EDGE_RISING = 2;  
    public static final int INT_EDGE_BOTH = 3;
```

각 GPIO Pin 초기화

# GPIO 사용

- MainActivity.java

```
pinMode(24, INPUT);  
pullUpDnControl(24, PUD_DOWN);  
int in24 = digitalRead(24);
```

- GPIO 실제 사용

# GPIO 사용

- build.gradle

```
sourceSets {  
    main {  
        // let gradle pack the shared library into apk  
        jniLibs.srcDirs = ['path/to/wiringPi/libs']  
    }  
}
```

- 컴파일에 사용할 NDK 라이브러리 build.gradle에 sourceSets으로 정의