



RC Car Programming / Electronics Control

Christopher Métrailler | Michael Clausen | Patrice Rudaz
François Corthay | Charles Praplan



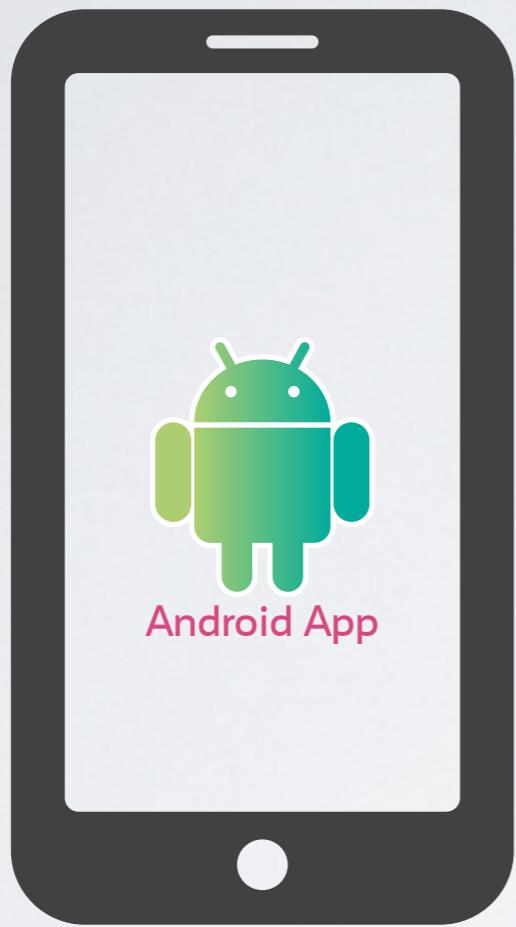
Goals



Use basic elements seen in
INF I / Eln
courses.



Goals



Smartphone



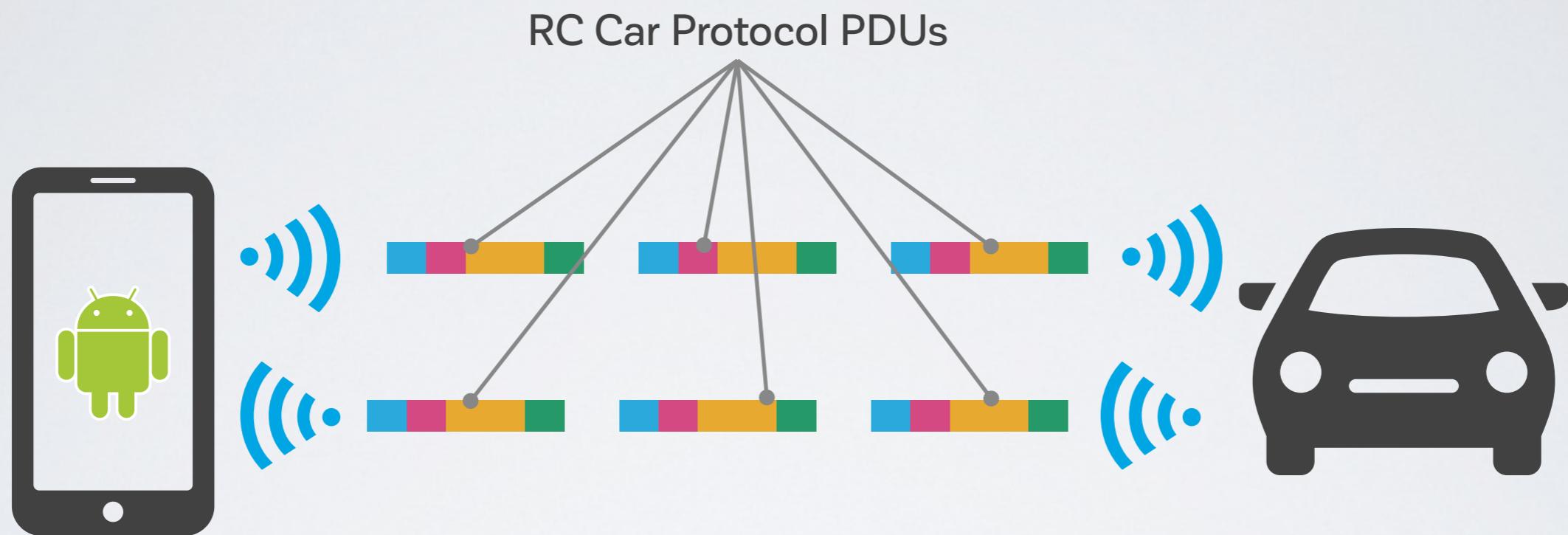
Bluetooth



RC Car



Remote Control Protocol





Hes-SO // VALAIS
WALLIS

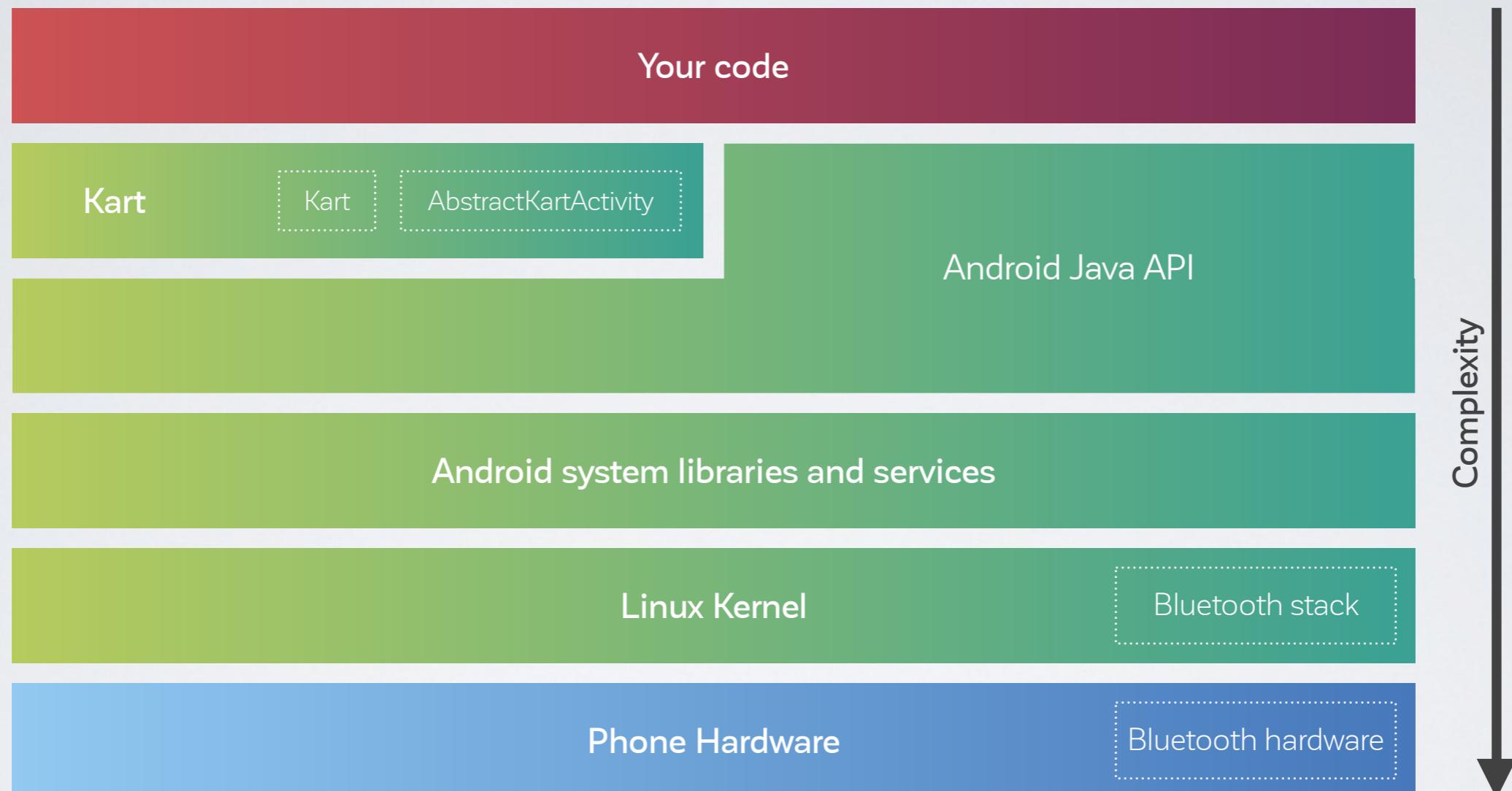


Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

1. Android Development

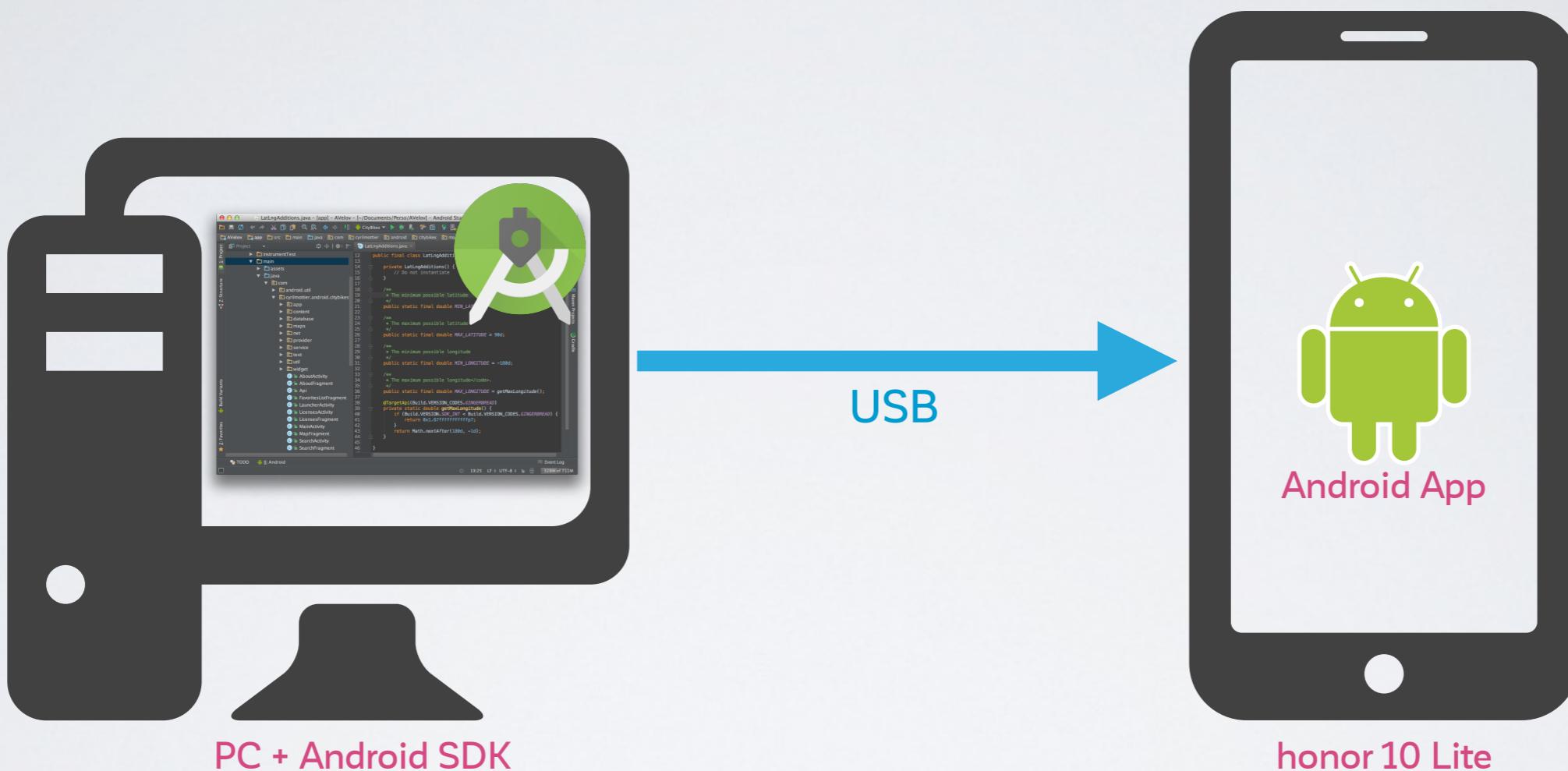


Remote Control Protocol





Remote Control Android App





Android

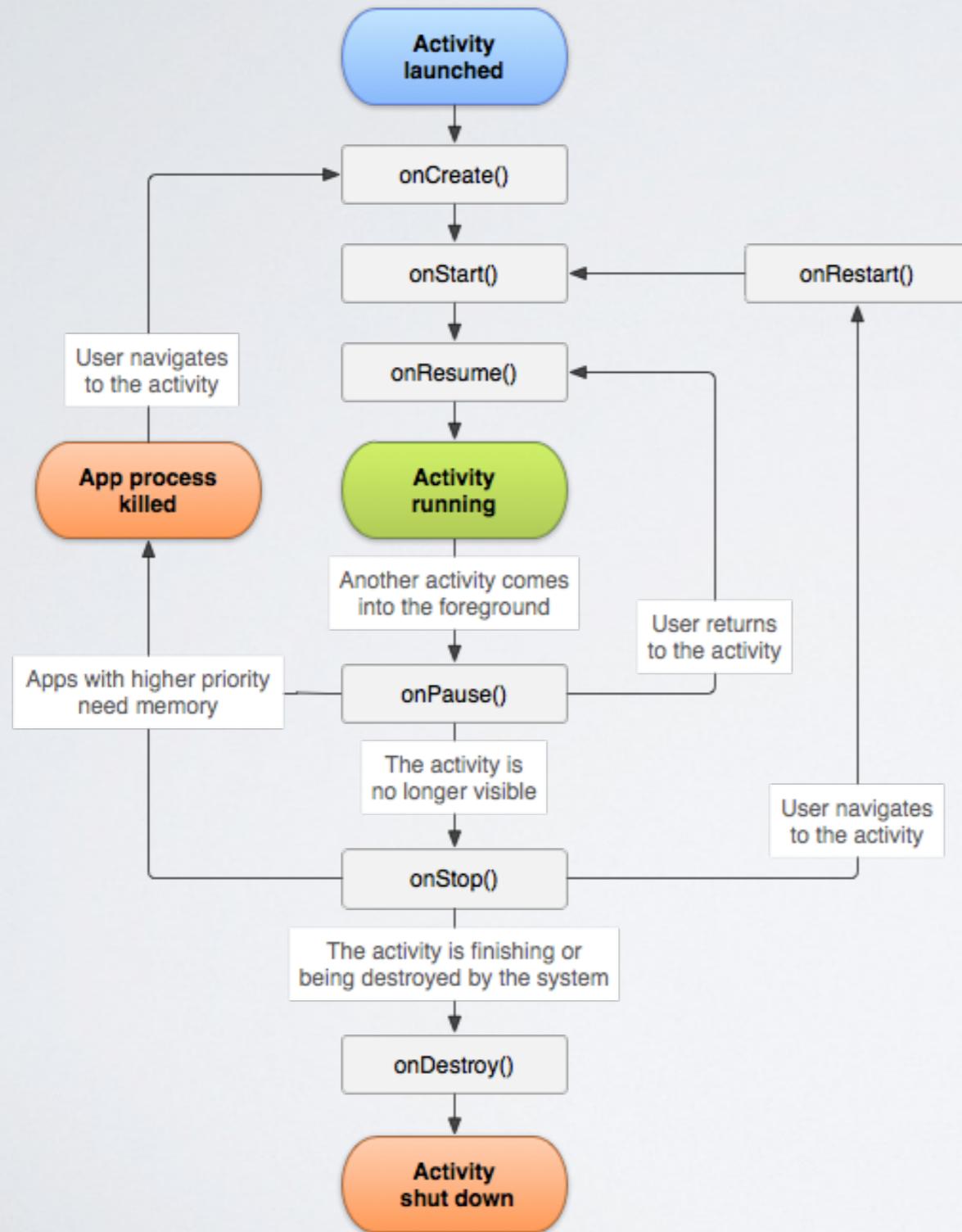
- Mobile Operating System developed by **Alphabet** (Google)
- **Abstracts hardware** from different manufacturers to a **common API**
- Applications are written in **Java or Kotlin** and run on a **Virtual Machine** (ART)
- Android is **open source**, based on **Linux**
- The **SDK & Android Studio** (based on **IntelliJ IDEA**) are free to use and allow everyone to build applications for Android



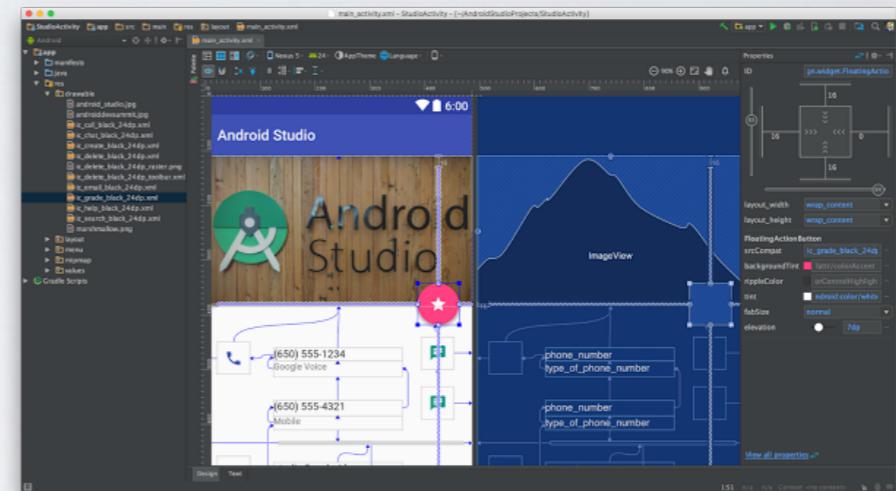
Android



Android Application lifecycle



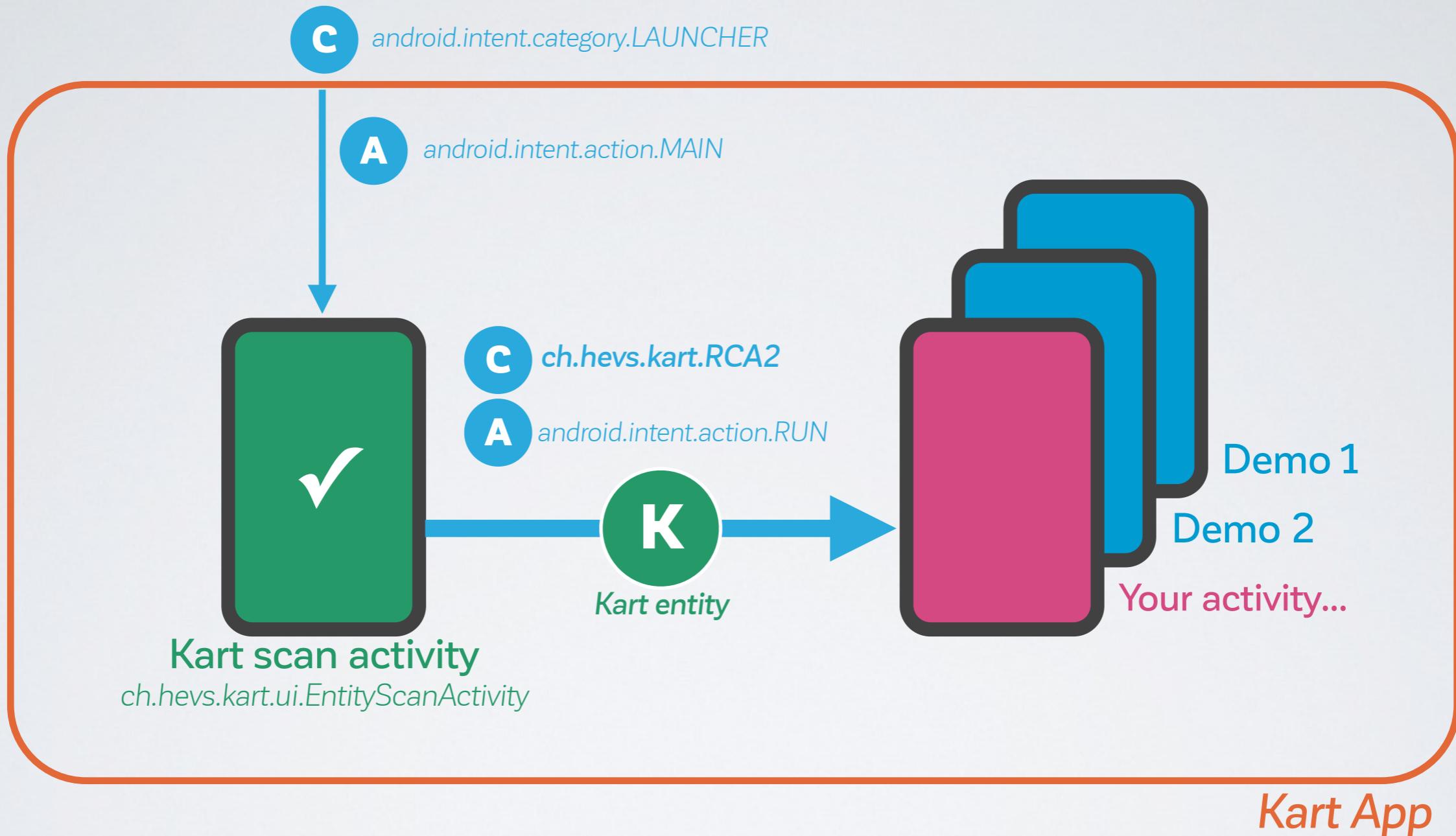
Layouts



- UI layouts can be designed using an editor integrated into Android Studio.
 - Layouts are serialized to XML files.
 - Those Layouts can be loaded in code.

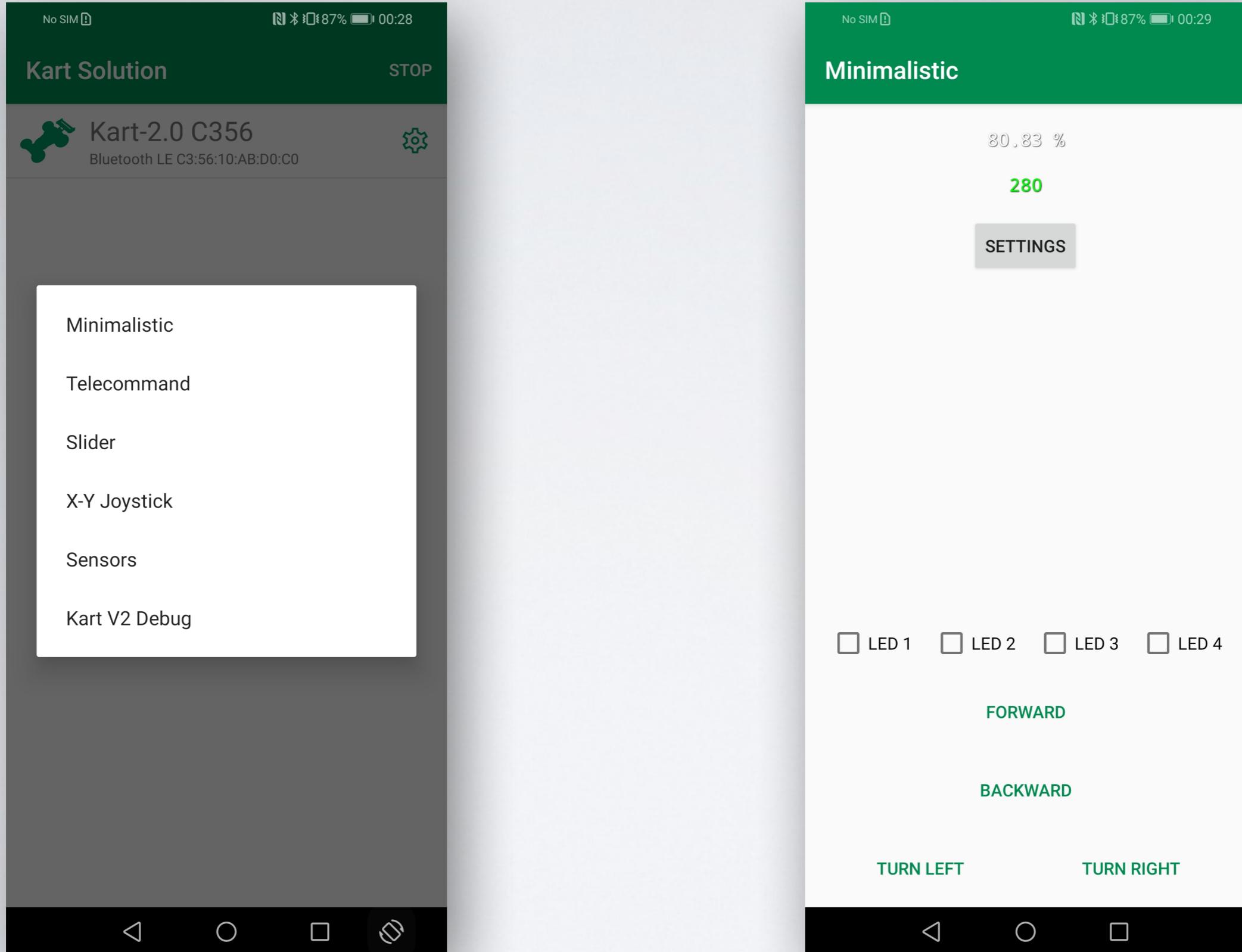


Remote Control Android App



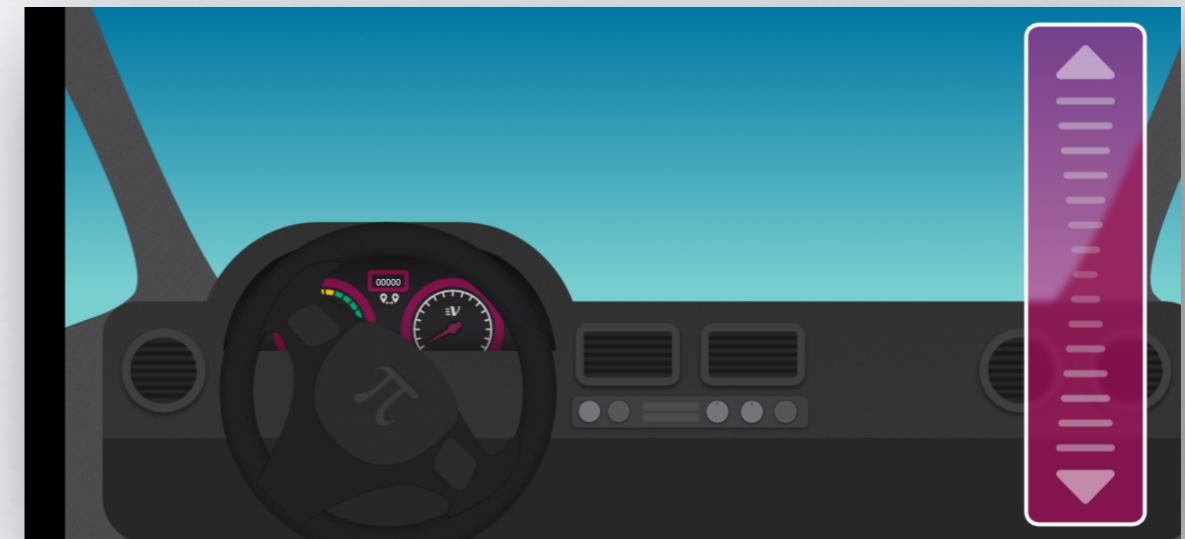
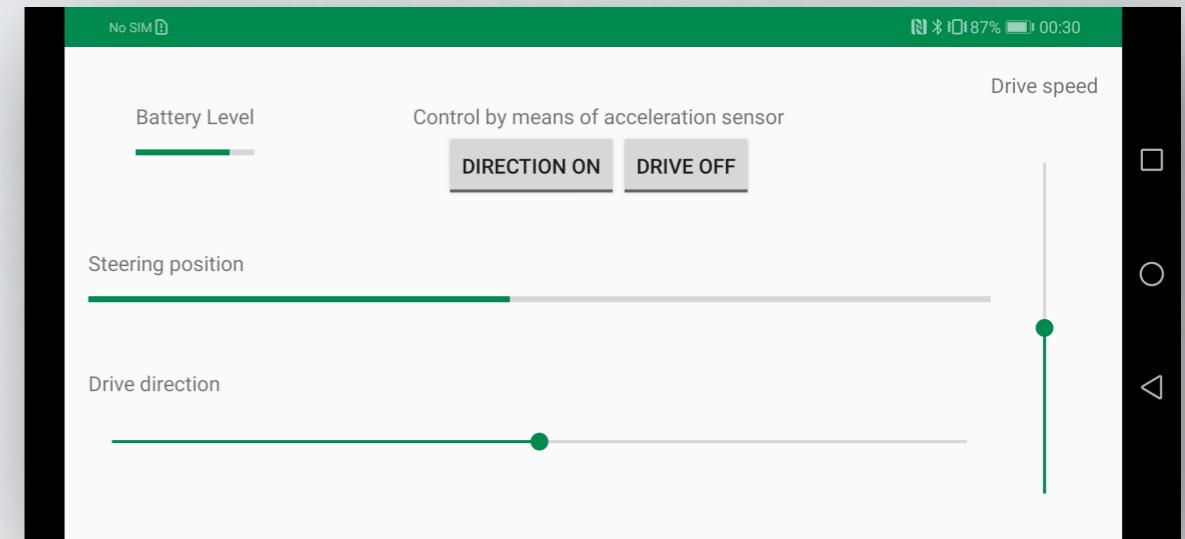


RC Android App Views



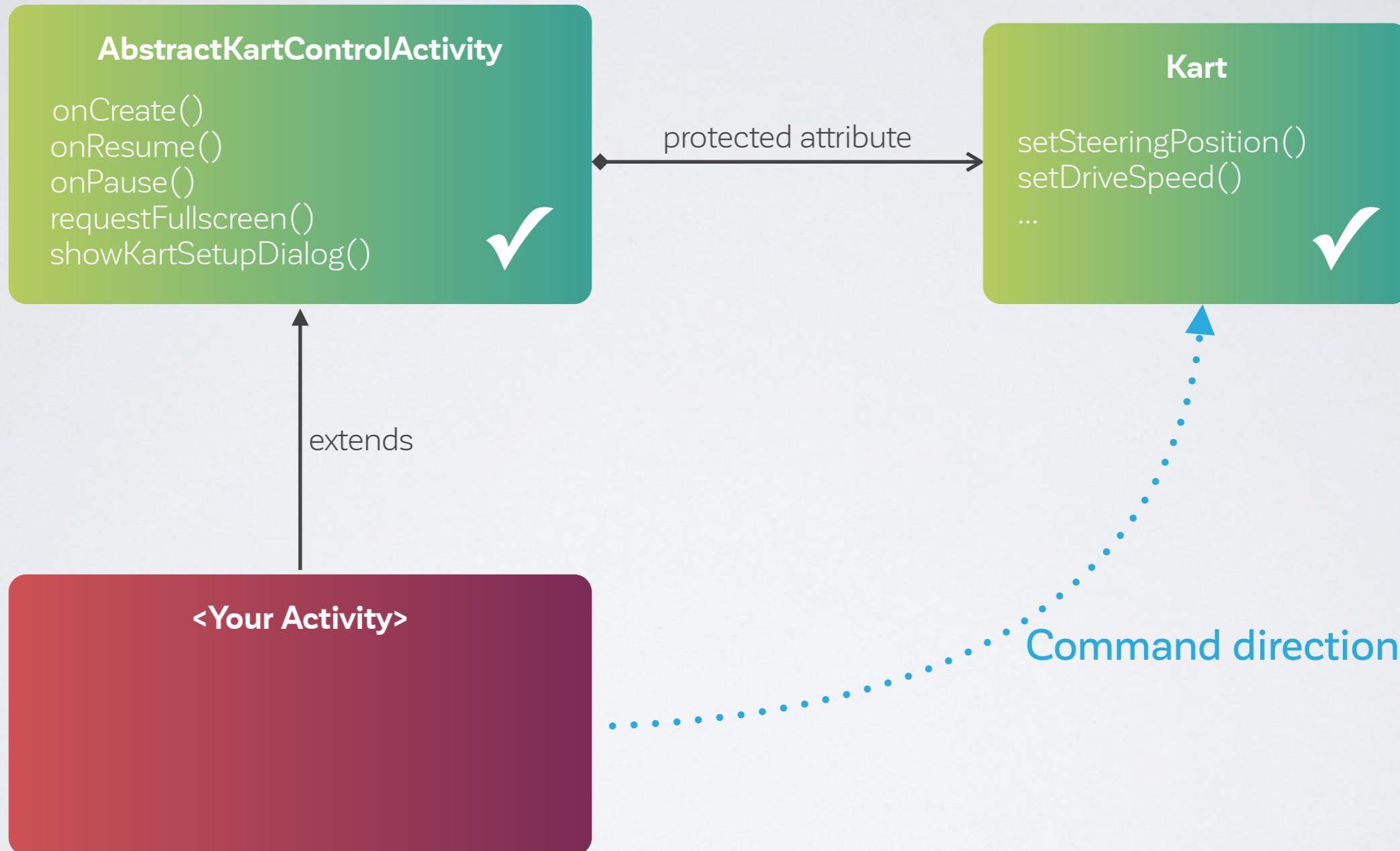


RC Android App Views



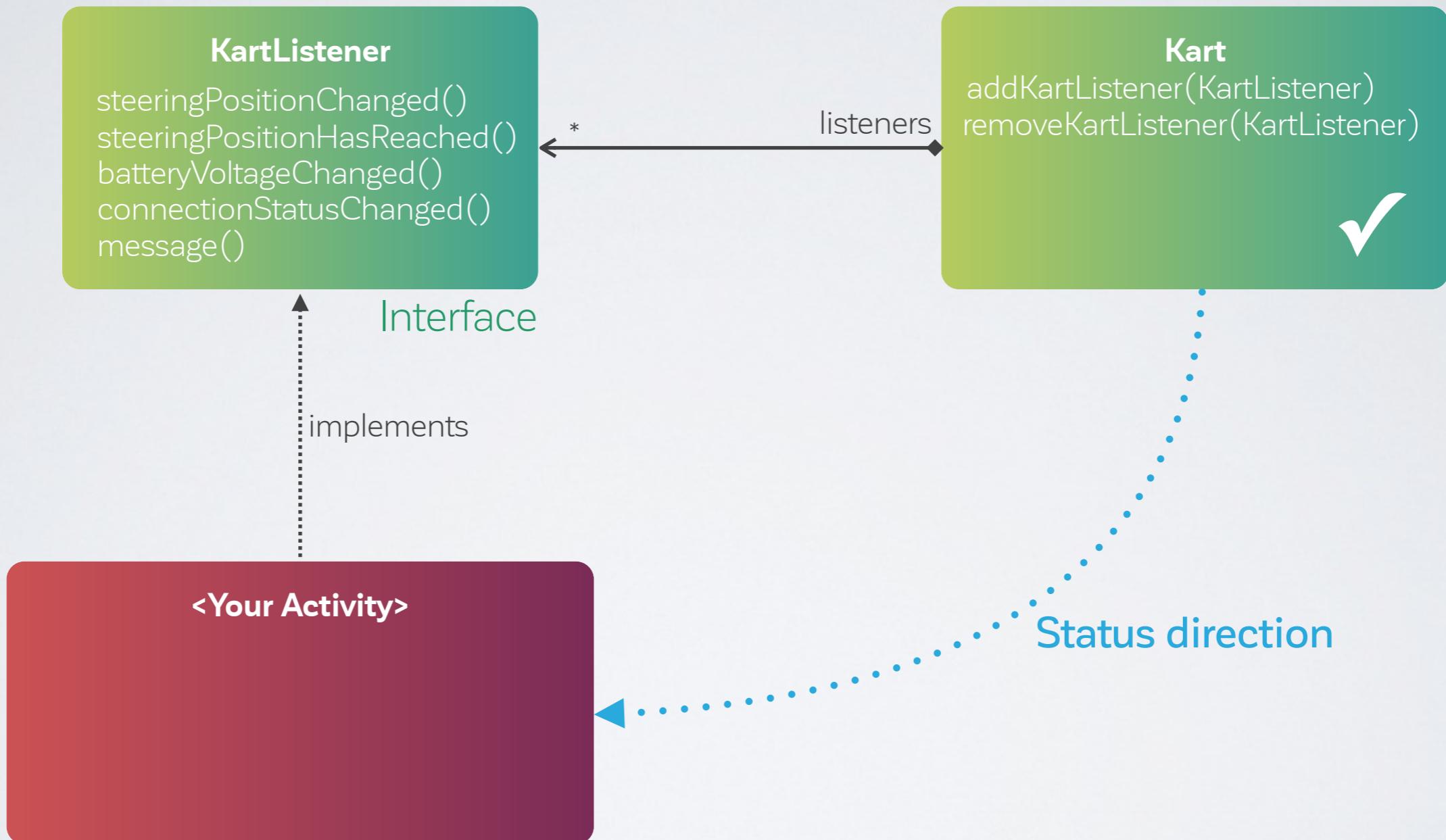


Remote Control Android App



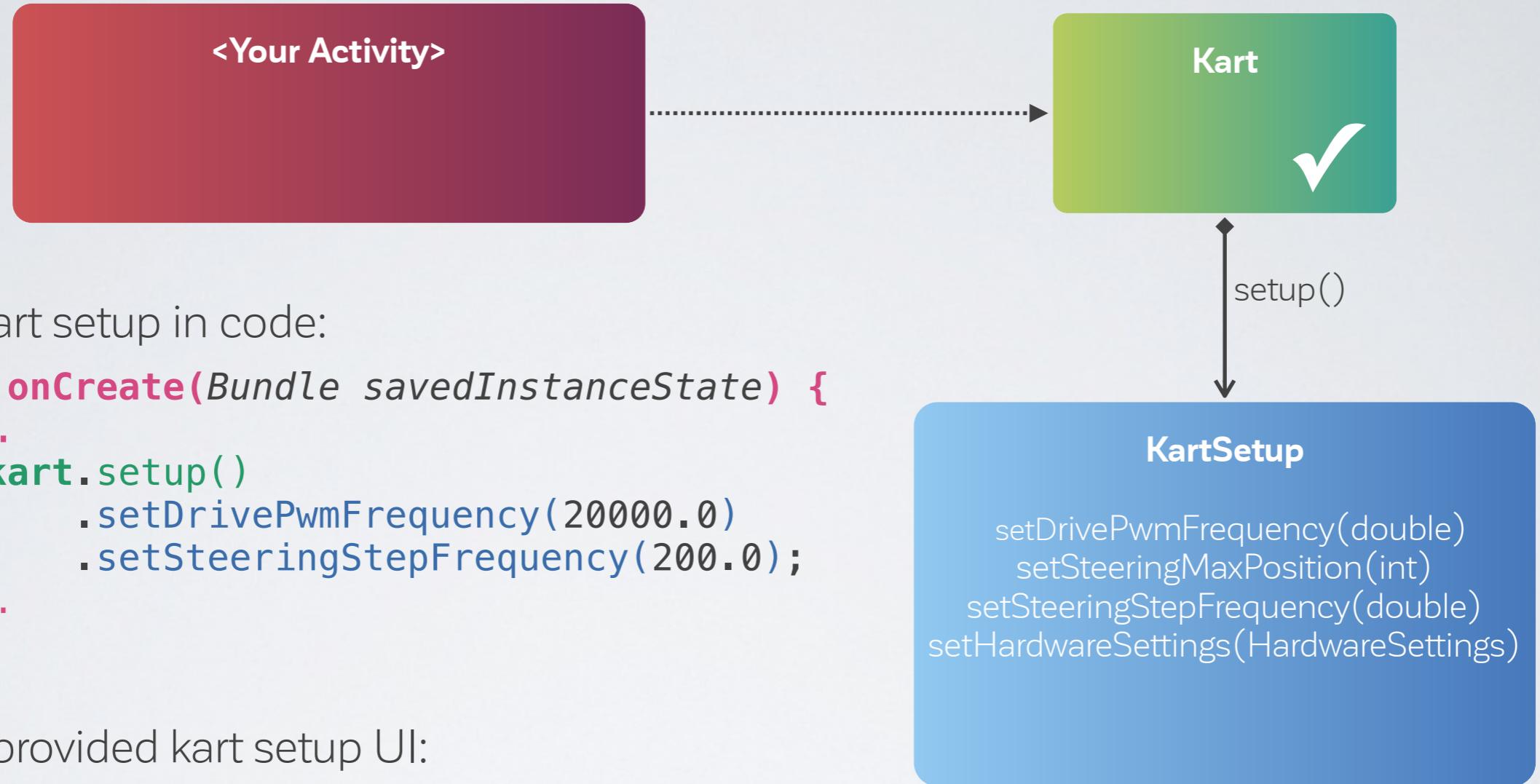


Remote Control Android App





Remote Control Android App



The setup can be configured even before connecting to the Kart by clicking on the symbol on the discovery list.



Software Goals

- **Slider control**
 - Direction
 - Speed
- **Progress Bar status**
 - Battery level
 - Steering position
- **Accelerometer (Orientation) control**
 - Button to enable orientation control
 - Device orientation controls sliders or kart



Software Grade



All mandatory features

Direction control
Speed control
Battery display
Direction display

4.0

+
0.5

=
Grade

Functional blackbox tests
during morning of the last
day

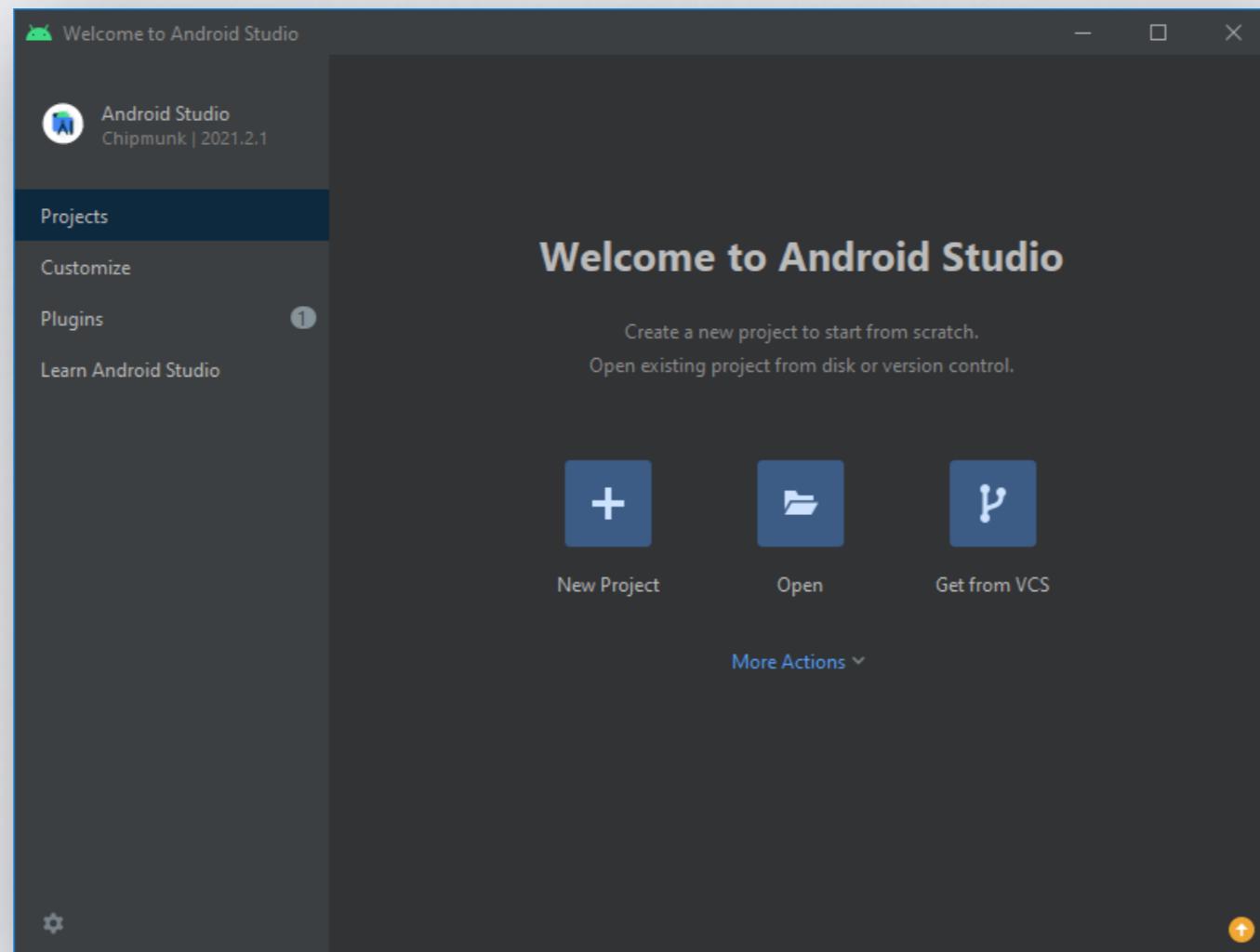
Per optional feature

LED's
Ultrasound sensor
Proximity sensor
Hall sensor
Steering assistant
Other improvements



Android Studio

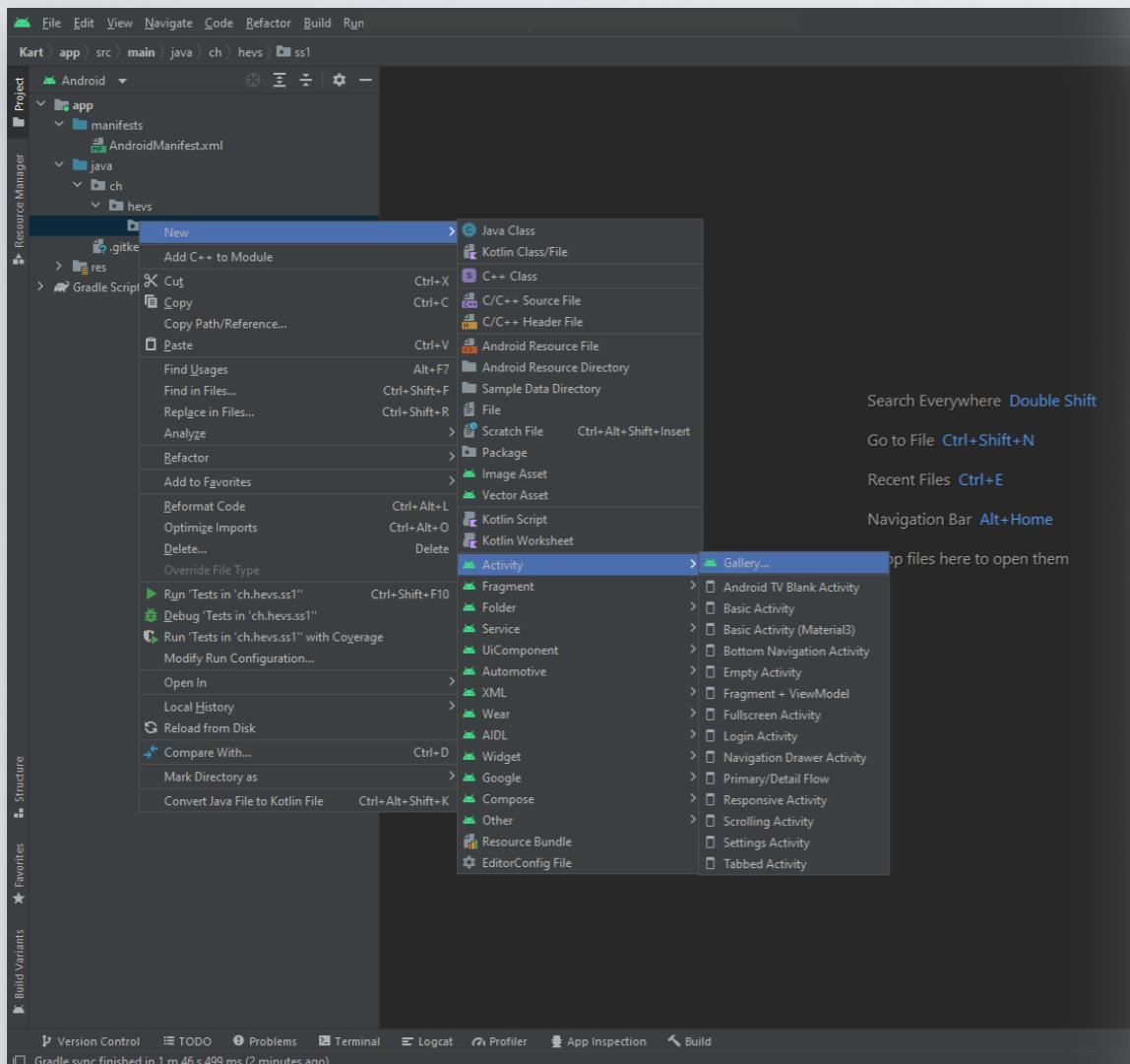
- ➊ Download the Kart project template from the wiki
- ➋ Extract the archive to the **local** disk
- ➌ Open the extracted folder in Android Studio



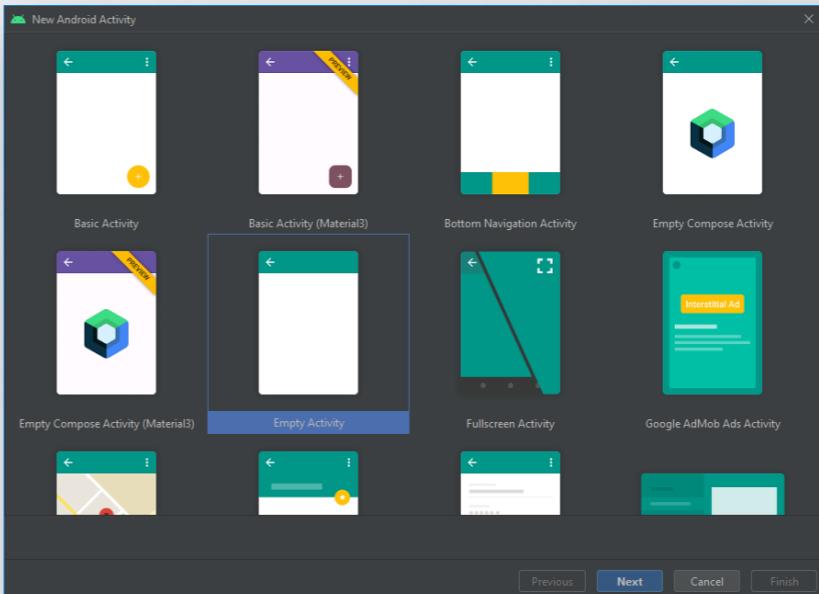


Android Studio

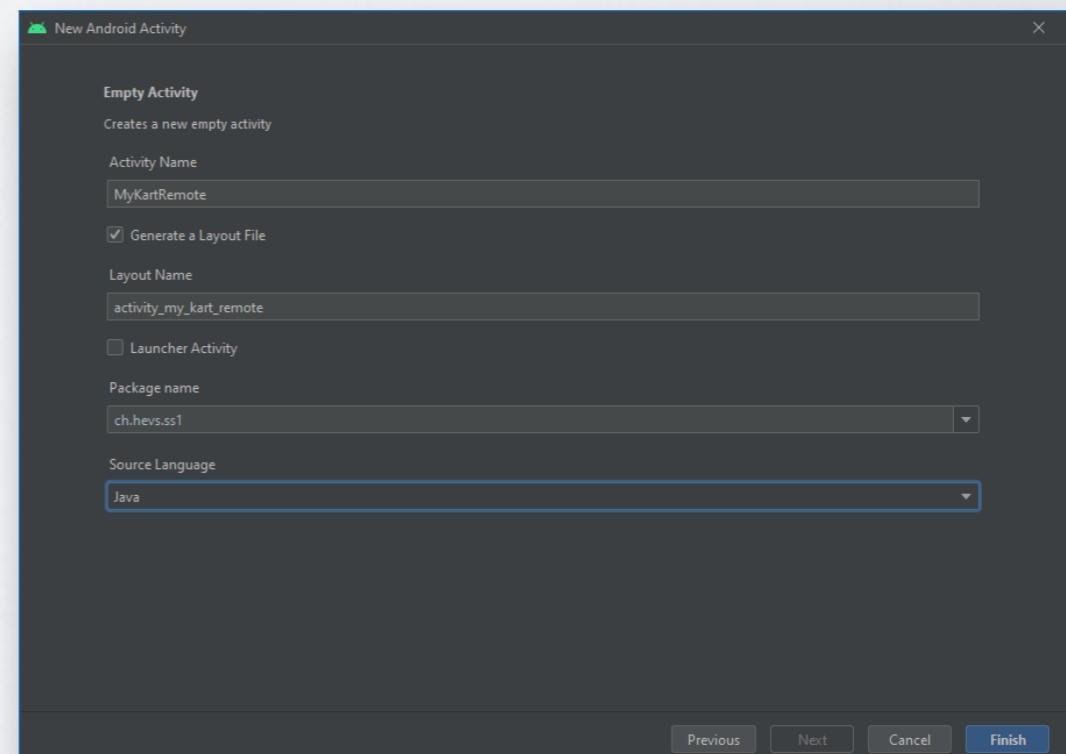
4 Select New > Activity > Gallery...



5 Select Empty Activity



6 Configure Activity





Android Studio

7 Modify Activity Java code

```
package ch.hevs.ss1;

import android.os.Bundle;

import ch.hevs.kart.AbstractKartControlActivity;
import ch.hevs.kart.Kart;

public class MyKartRemote extends AbstractKartControlActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my_kart_remote);
    }
    // Override methods to get events from kart.
}
```

8 Modify AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="ch.hevs.ss1">

    <application
        android:allowBackup="false"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:icon="@mipmap/ic_icon"
        android:theme="@style/Theme.AppCompat.DayNight.NoActionBar"
        tools:ignore="GoogleAppIndexingWarning">

        <activity android:name=".MyKartRemote">
            <intent-filter>
                <action android:name="android.intent.action.RUN" />
                <category android:name="ch.hevs.kart.RCA2" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```



honor 10 Lite

- Connect Phone to PC's USB port
- Power on the phone
- Use default configurations during setup wizard
- Enable developer mode:
Go to Settings > System > About Phone and press 7 times on „Build Number“
- Enable USB debugging:
Go to Settings > System > Developer options and check „USB debugging“
- Install and start your Android application:
 - In Android Studio, press green button and select honor 10 Lite phone from the list.
 - On the phone, answer yes to allow USB debugging in popup.
 - Now your application should be running on the phone.





Install & Run Demo

Download Kart.apk from:

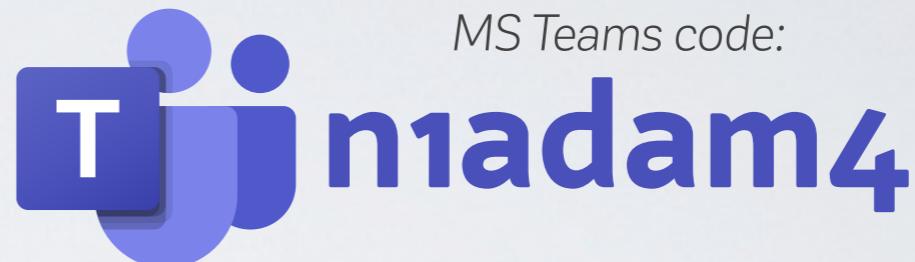
<http://kart-javadoc.hevs.ch/Kart.zip>

Install it using the **adb** command line tool:

```
c:\Users\your.account> c:\devel\Android\sdk\platform-tools\adb install Downloads\Kart.apk
3979 KB/s (937315 bytes in 0.230s)
    pkg: /data/local/tmp/Kart.apk
Success
```



Tipp #1: Read the docs



MS Teams code:

- You find all information needed here:
 - Your copy of the kart project documentation and tasks document
 - Kart wiki: <http://wiki.hevs.ch/fsi/index.php5/Kart>
 - Kart project JavaDoc: <http://kart-javadoc.hevs.ch>
 - Android: <https://developer.android.com/index.html>

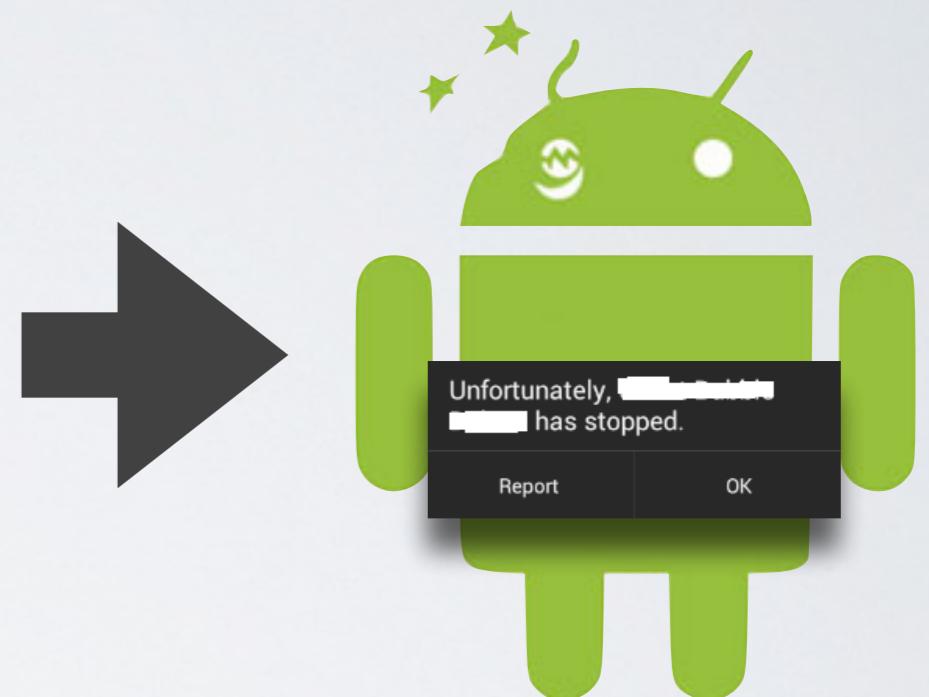
Ask us, we kindly like to help you!



Tipp #2: No infinite loops

```
// Blink a led
try {
    while (true) {
        kart.setLed(0, true);
        Thread.sleep(500);
        kart.setLed(0, false);
        Thread.sleep(500);
    }
} catch (InterruptedException e) {
    e.printStackTrace();
}
```

If you add infinite loops to the main thread, your application will crash!





Tipp #3: Periodic Timer

Inside your Activity, add the attribute:

```
private Timer ledBlinker = new Timer() {  
    @Override  
    public void onTimeout() {  
        kart.toggleLed(0);  
    }  
};
```

You can start the timer using:

```
ledBlinker.schedulePeriodically(500);
```

You can stop the timer with:

```
ledBlinker.stop();
```

Documentation:

<http://kart-javadoc.hevs.ch/ch/hevs/kart/utils/Timer.html>



Tipp #4: Doing something later

```
Timer doLater = new Timer() {  
    @Override  
    public void onTimeout() {  
        kart.setLed(0, true);  
    }  
};  
doLater.scheduleOnce(5000);
```

This code snippet will turn the LED 0 on
after 5 seconds.

Documentation:

<http://kart-javadoc.hevs.ch/ch/hevs/kart/utils/Timer.html>



Tipp #5: Animations

```
Animation animation = Animation.Builder(kart)
    .ledOn(0).ledOff(1).wait(100)
    .ledOff(0).ledOn(1).wait(100)
    .build();
animation.loop();
```

The animation will turn LED 0 on and LED 1 off, then wait for 0.1s. Next it will turn LED 0 off and LED 1 on and then wait again for 0.1s. The animation is looped until the method cancel() is called...

Documentation:

<http://kart-javadoc.hevs.ch/ch/hevs/kart/utils/Animation.html>
<http://kart-javadoc.hevs.ch/ch/hevs/kart/utils/Animation.Builder.html>



Hes-SO // VALAIS
WALLIS

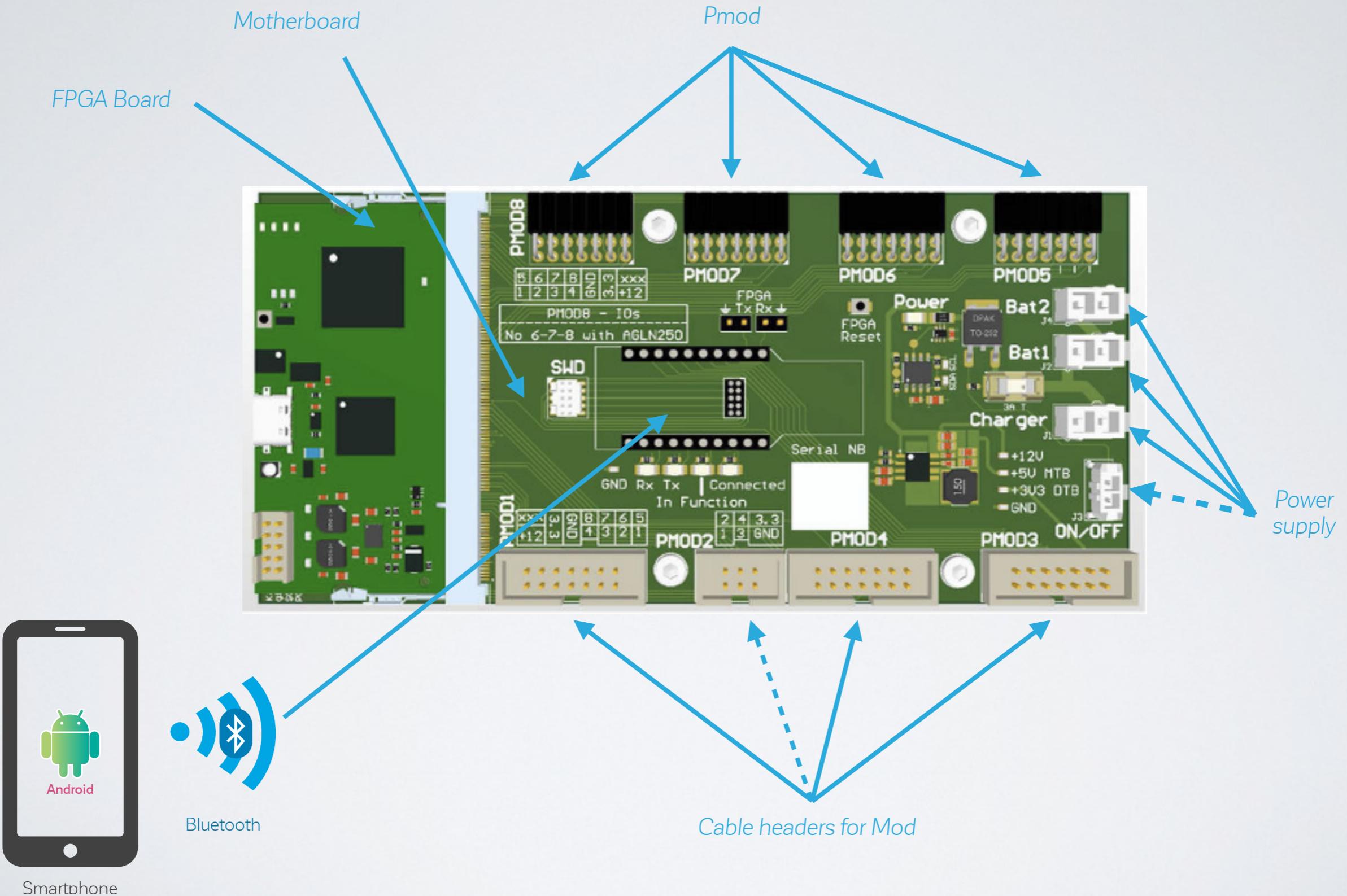


Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

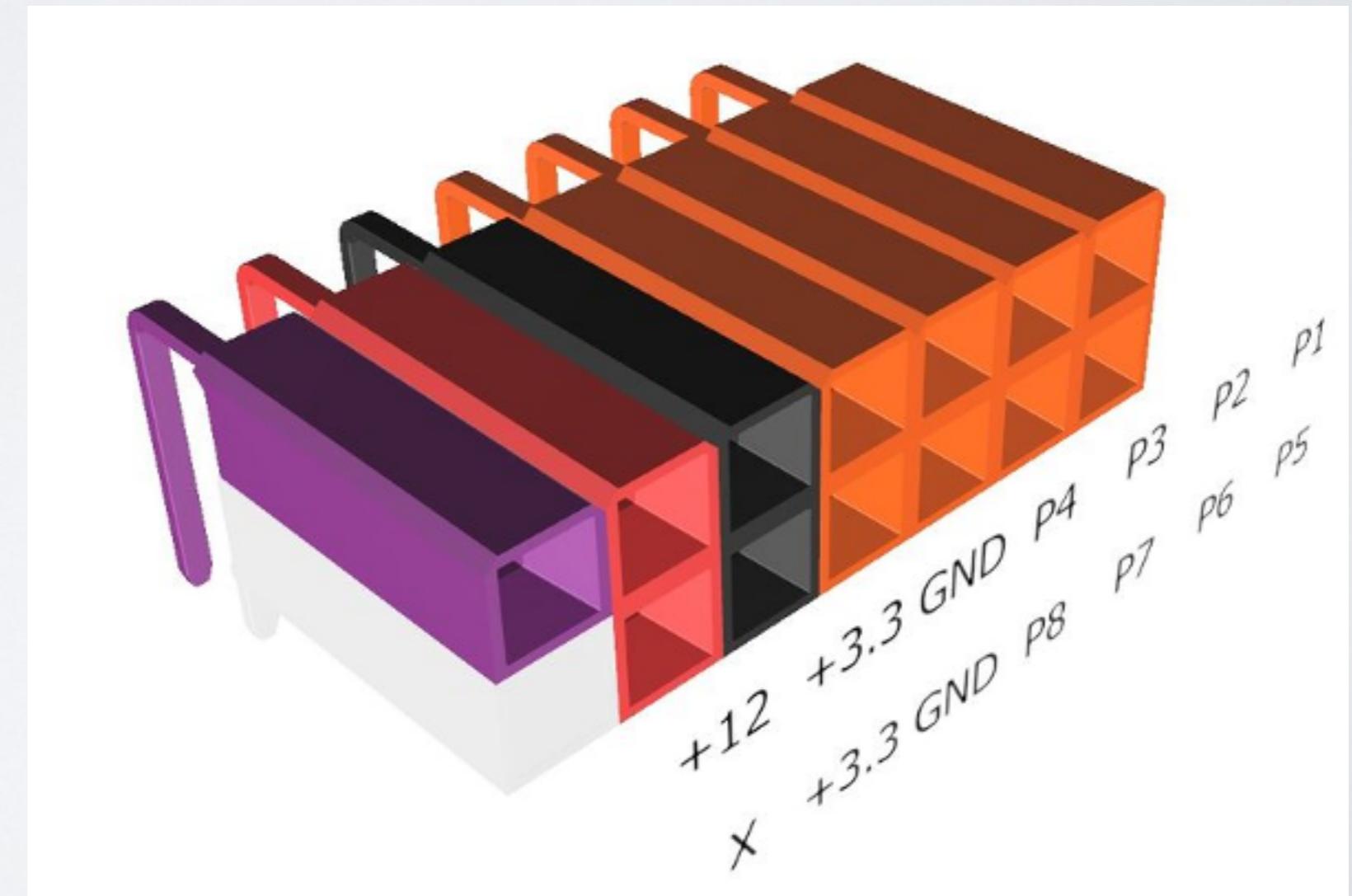
2. Control Electronics



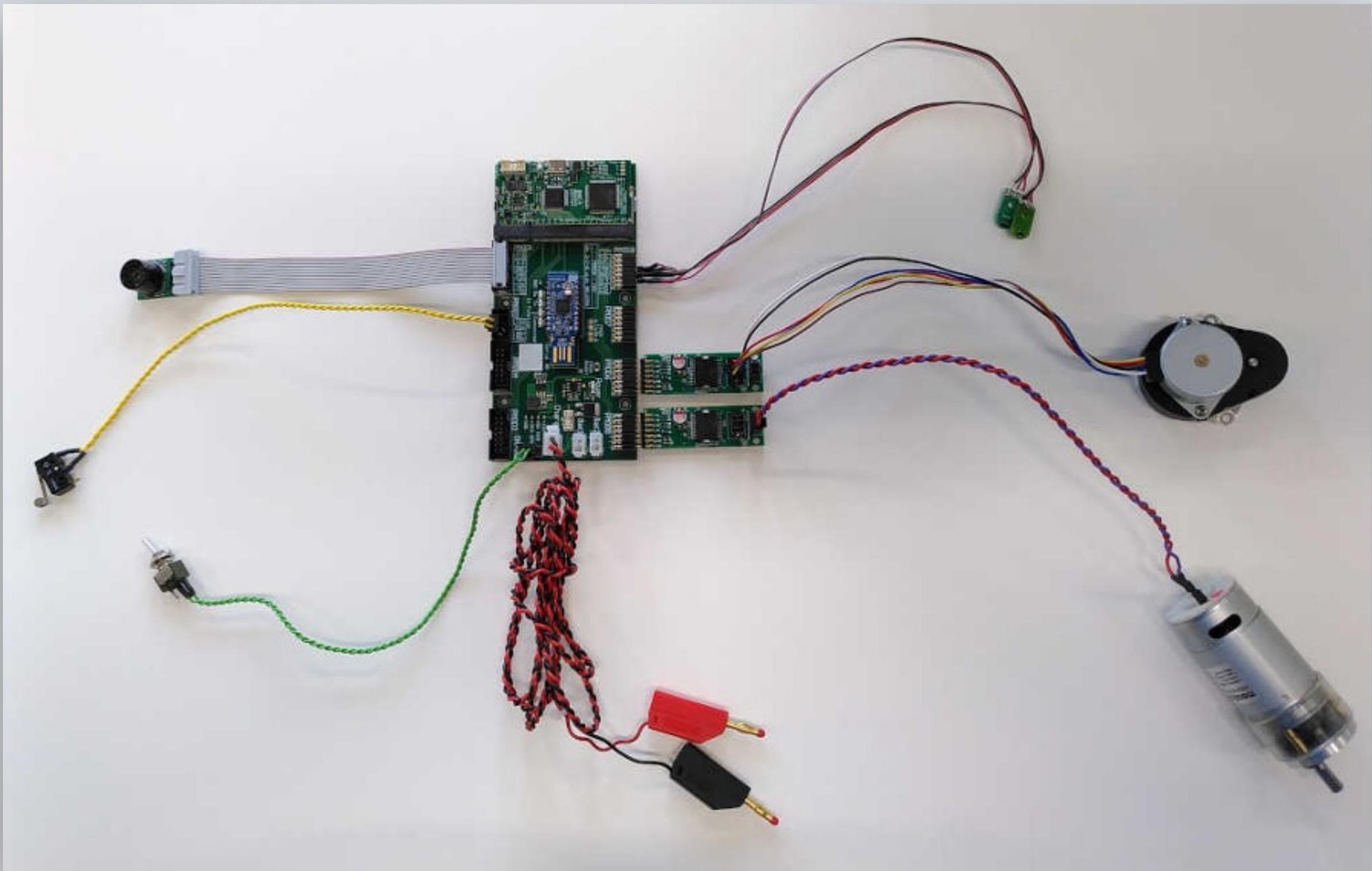
Electronics



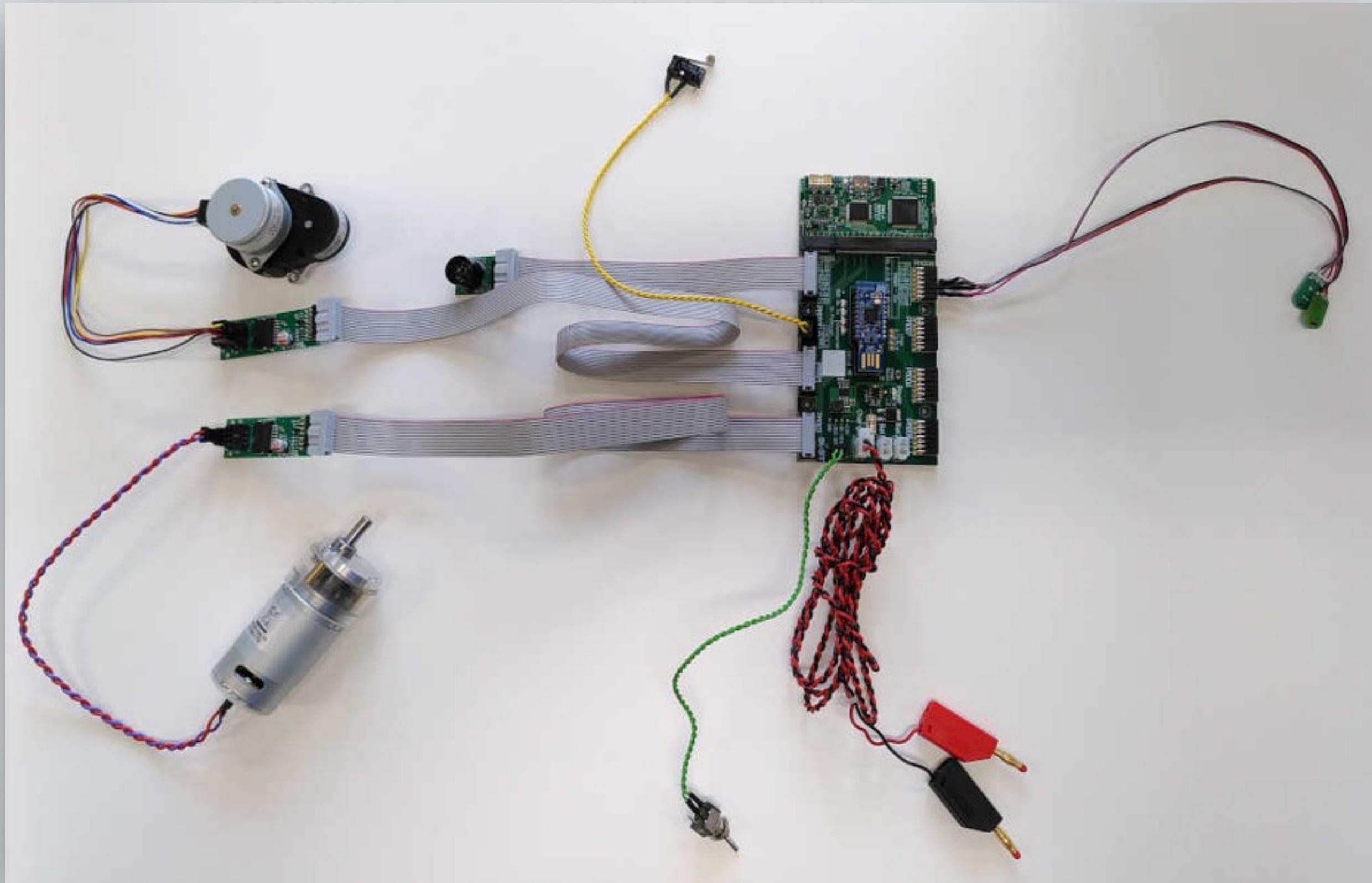
- Standard by Diligent
<<https://digilent.com/reference/pmod/specification>>
- $\frac{1}{2}$ lines with 4 I/Os
- RC Car Variant: V_{batt}



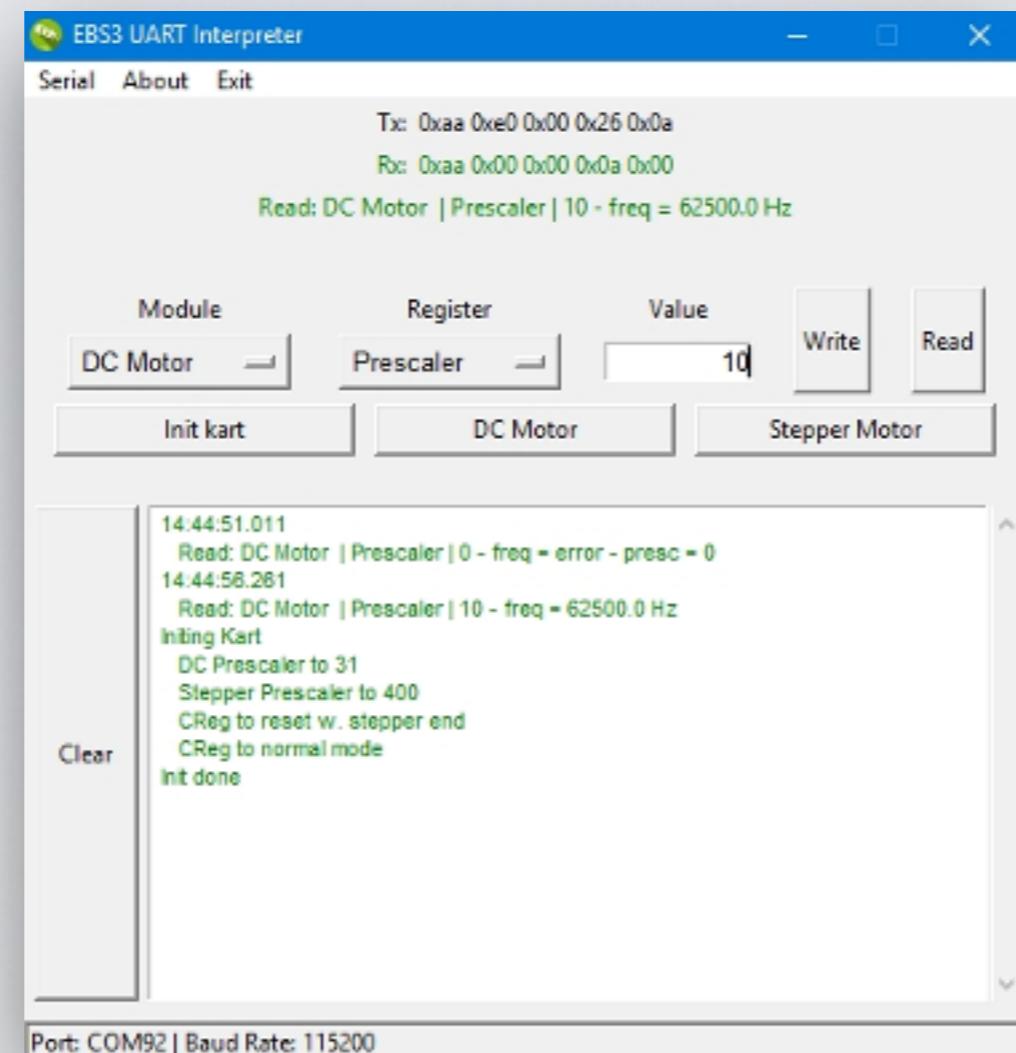
Wiring



Wiring



First tests





Battery management

- For the first tests, one can use a 12 V power supply
 - Skip to the provided batteries later
- Batteries can be recharged using a charger or from a power supply
 - Ask how to do this
- Recharge your batteries before the race !

