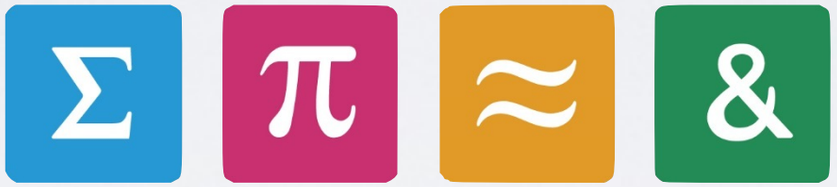
Hes·so // VALAIS WALLIS

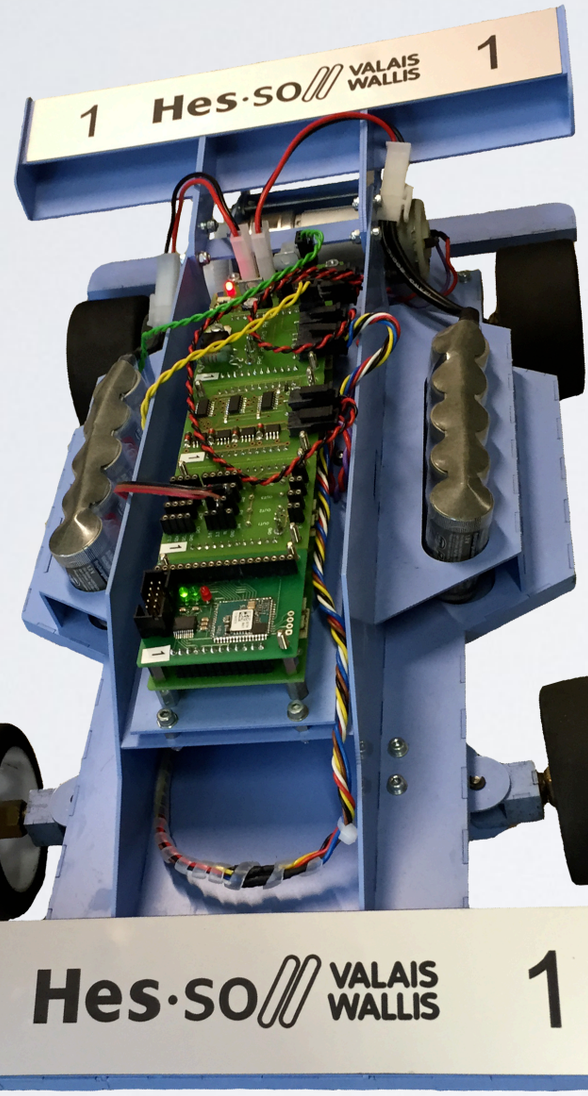


Kart Remote Programming

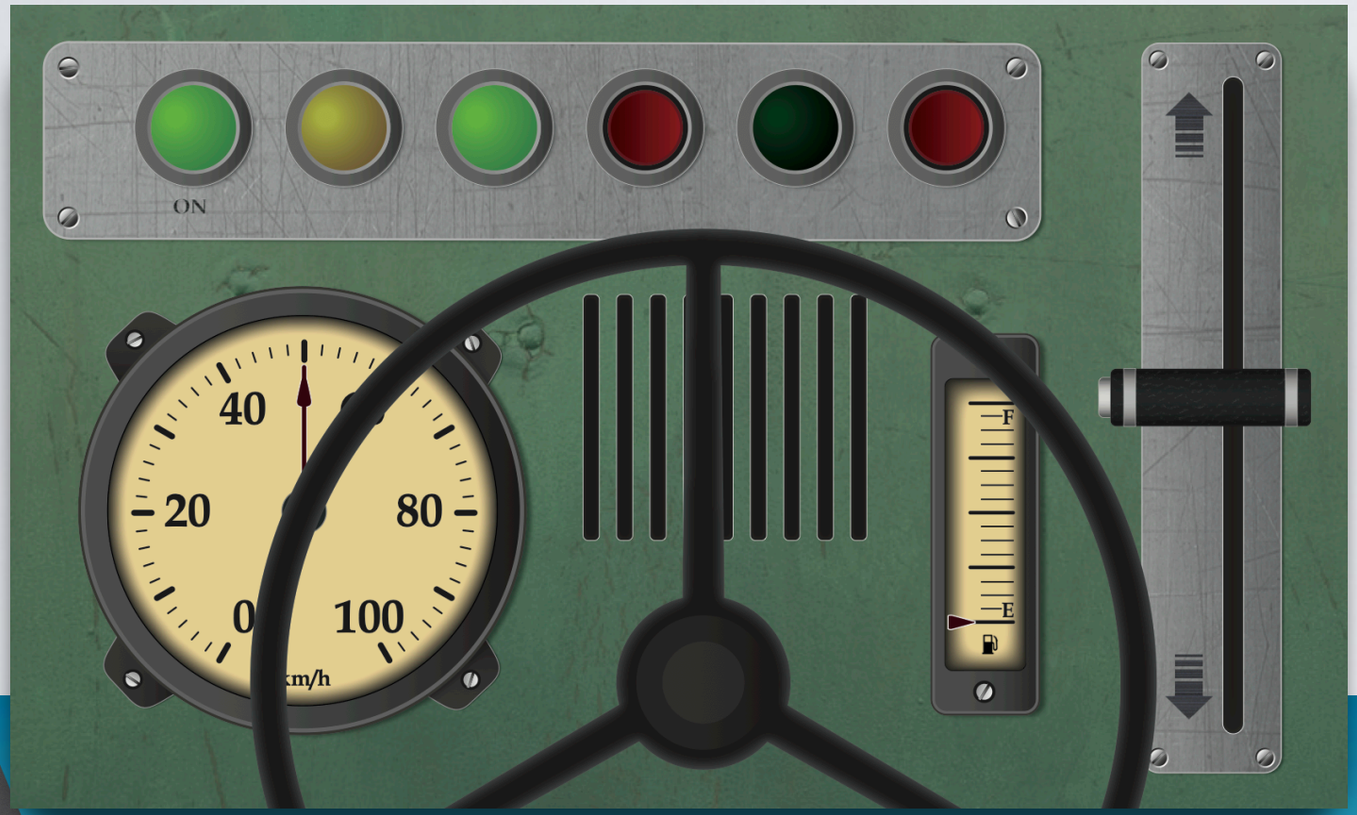
Christopher Metrailler | Patrice Rudaz | Michael Clausen

Hes·so VALAIS WALLIS

Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



Battery Level

Steering position

SETTINGS

LED 1 LED 2 LED 3 LED 4

FORWARD

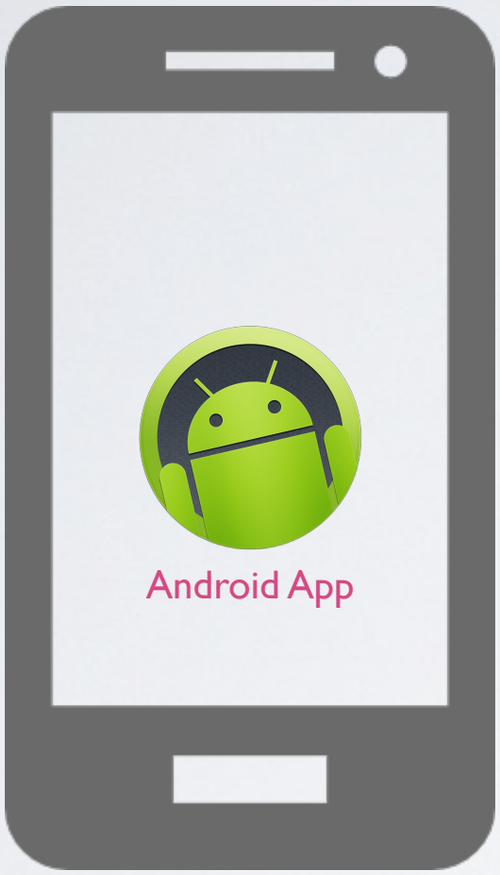
BACKWARD

TURN LEFT TURN RIGHT

Goals

Use basic elements seen in INF I course.

Goals



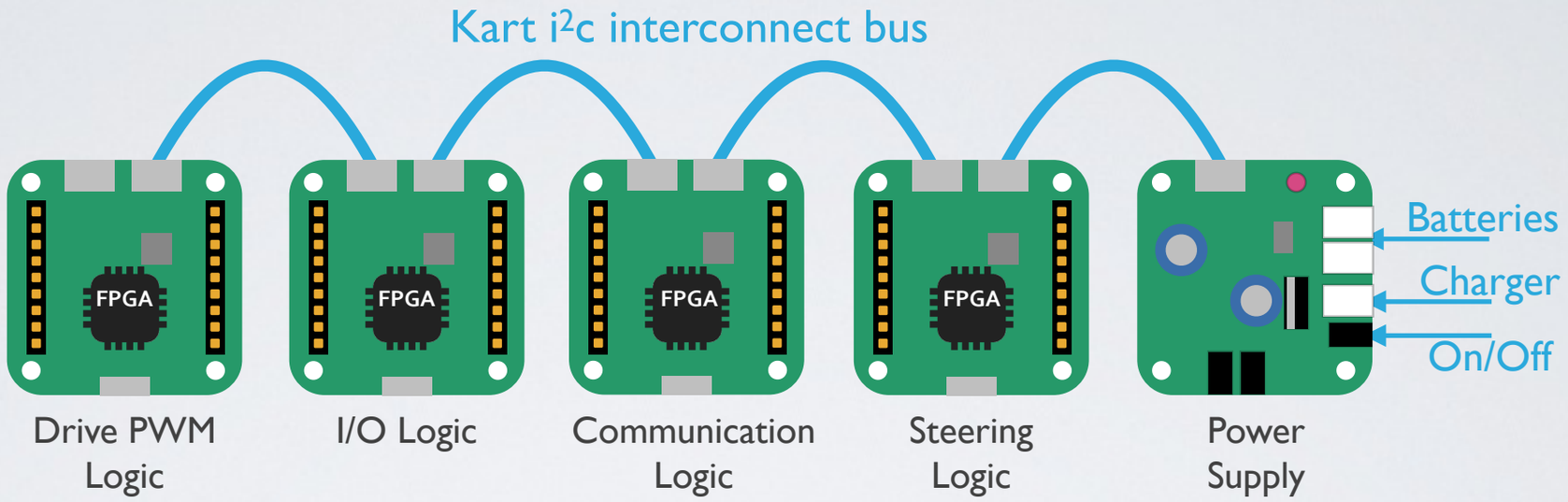
Galaxy Nexus

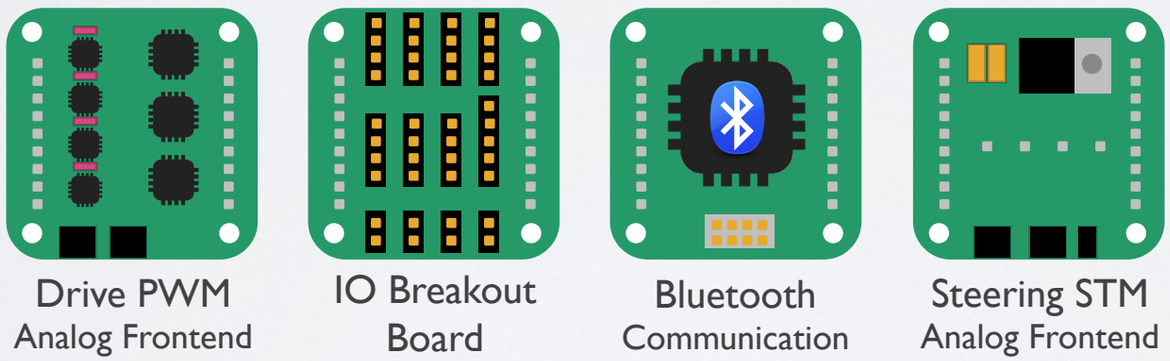
Bluetooth

Kart

Hardware

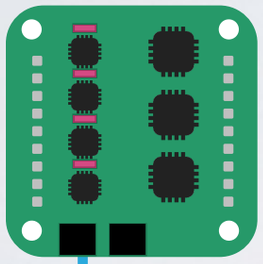
Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

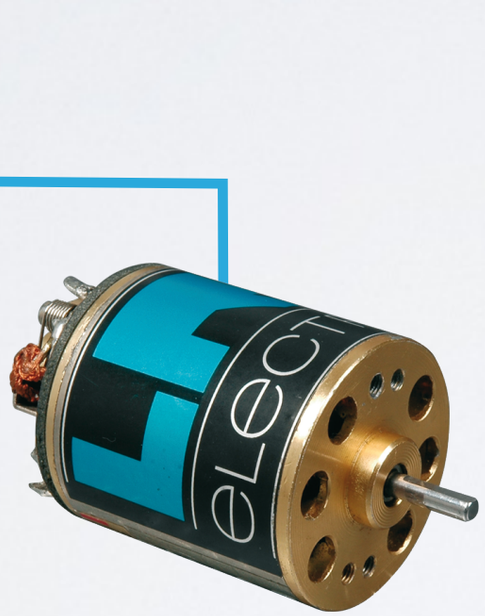




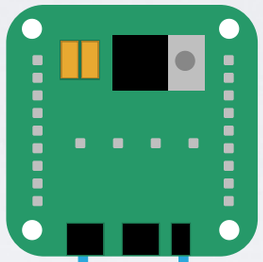
Hardware (2)

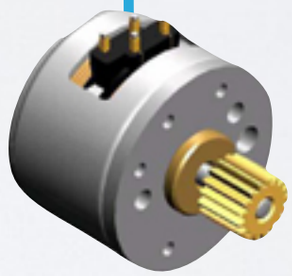
Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften





Drive motor



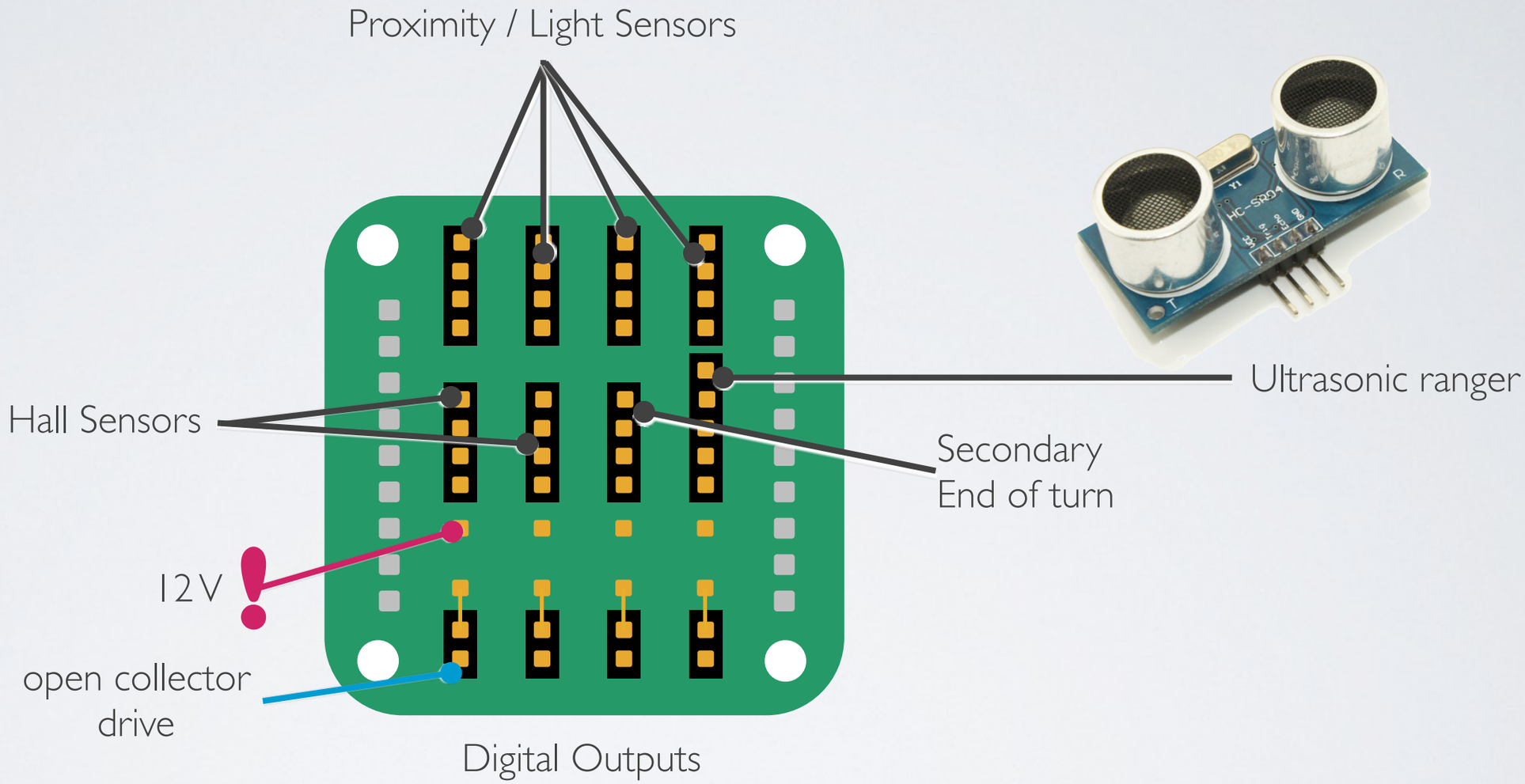


Stepper motor

Limit switch

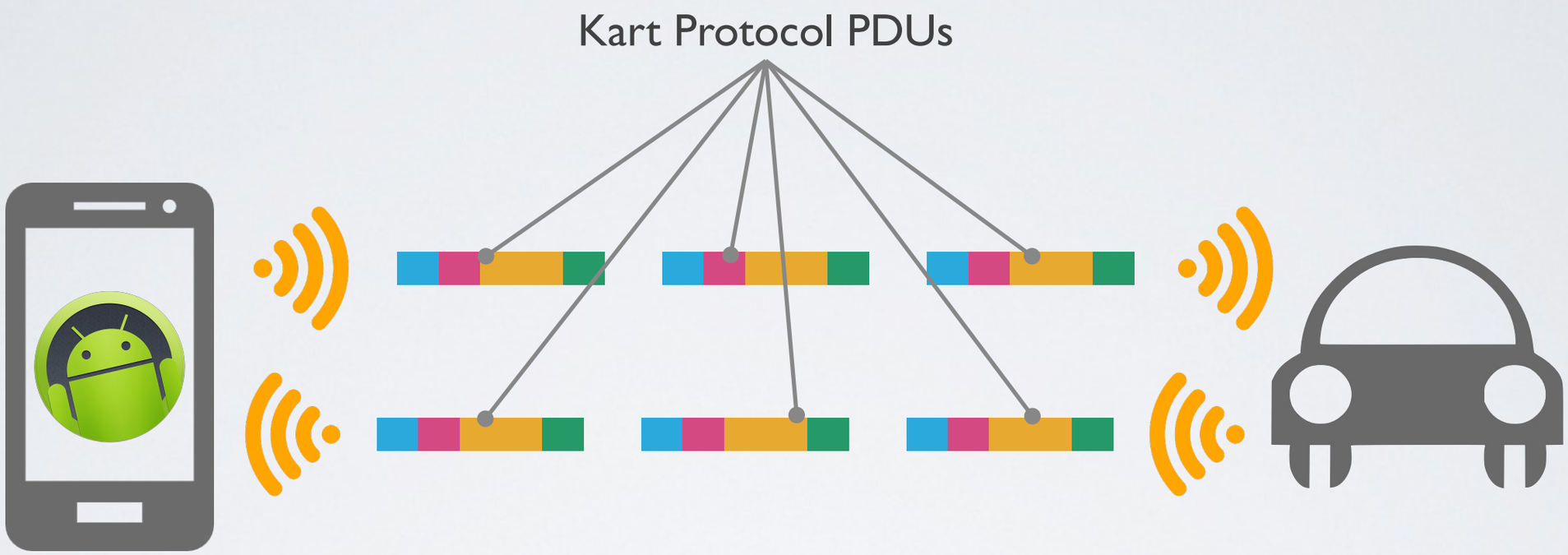
Hardware (3)

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



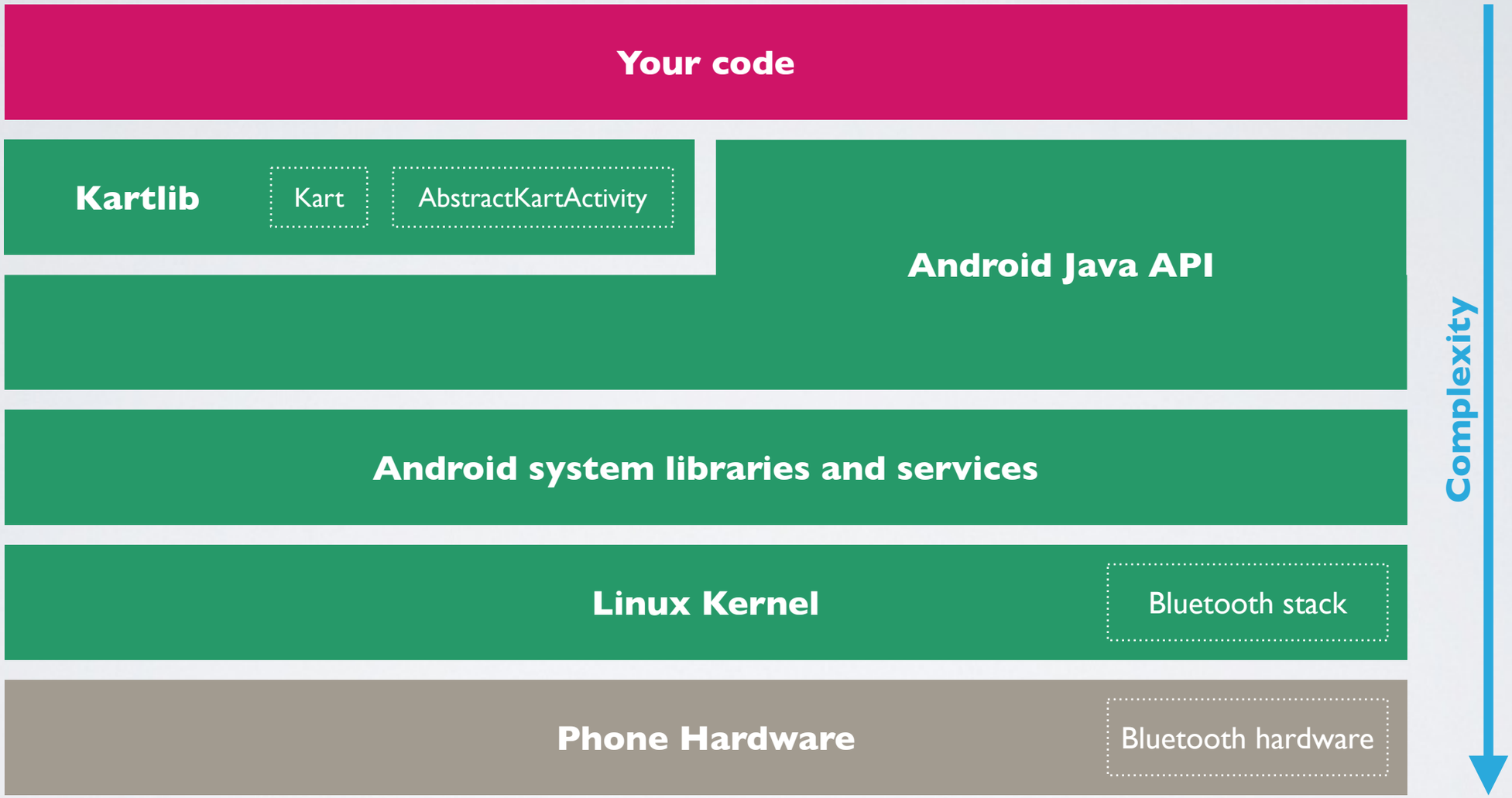
Remote Control Protocol

Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



Remote Control Protocol

Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



Remote Control Android App

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

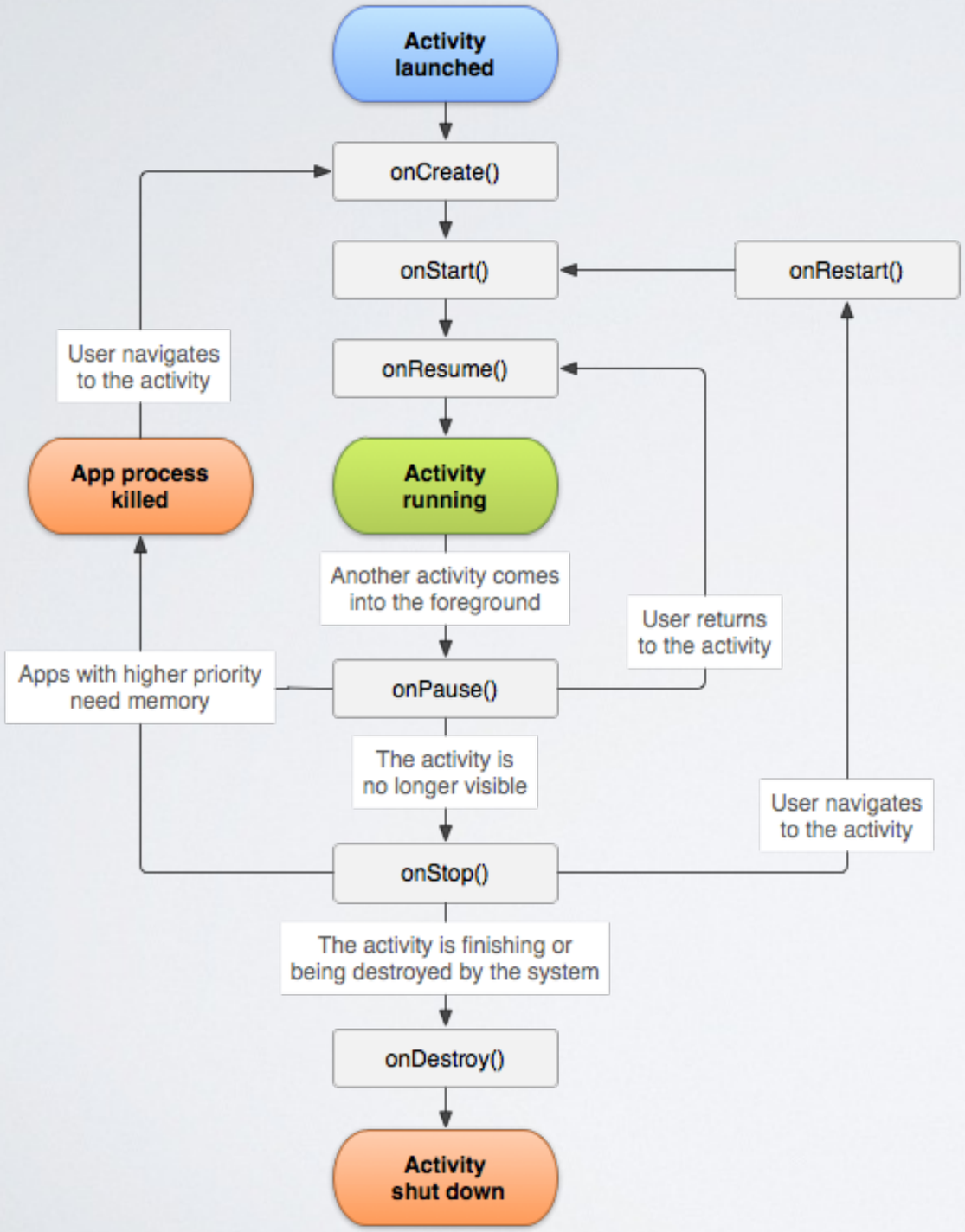
Android

- Mobile **O**perating **S**ystem 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** (Dalvik or 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.

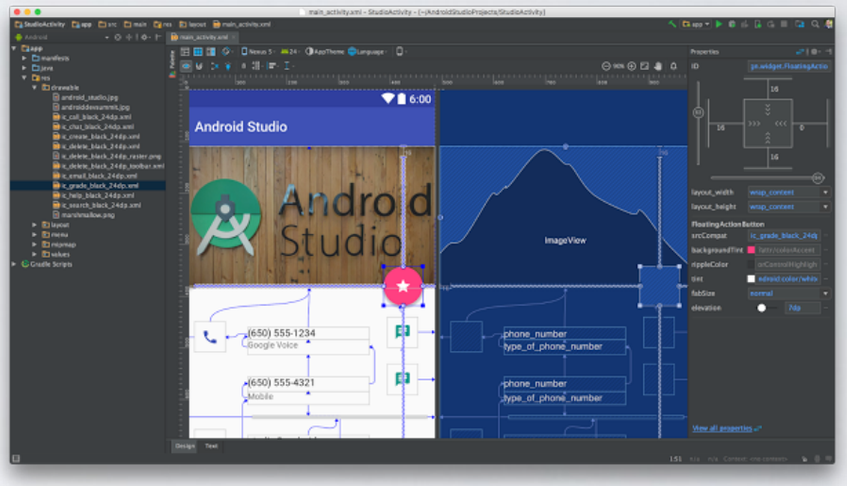
Android

Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

Android Application lifecycle



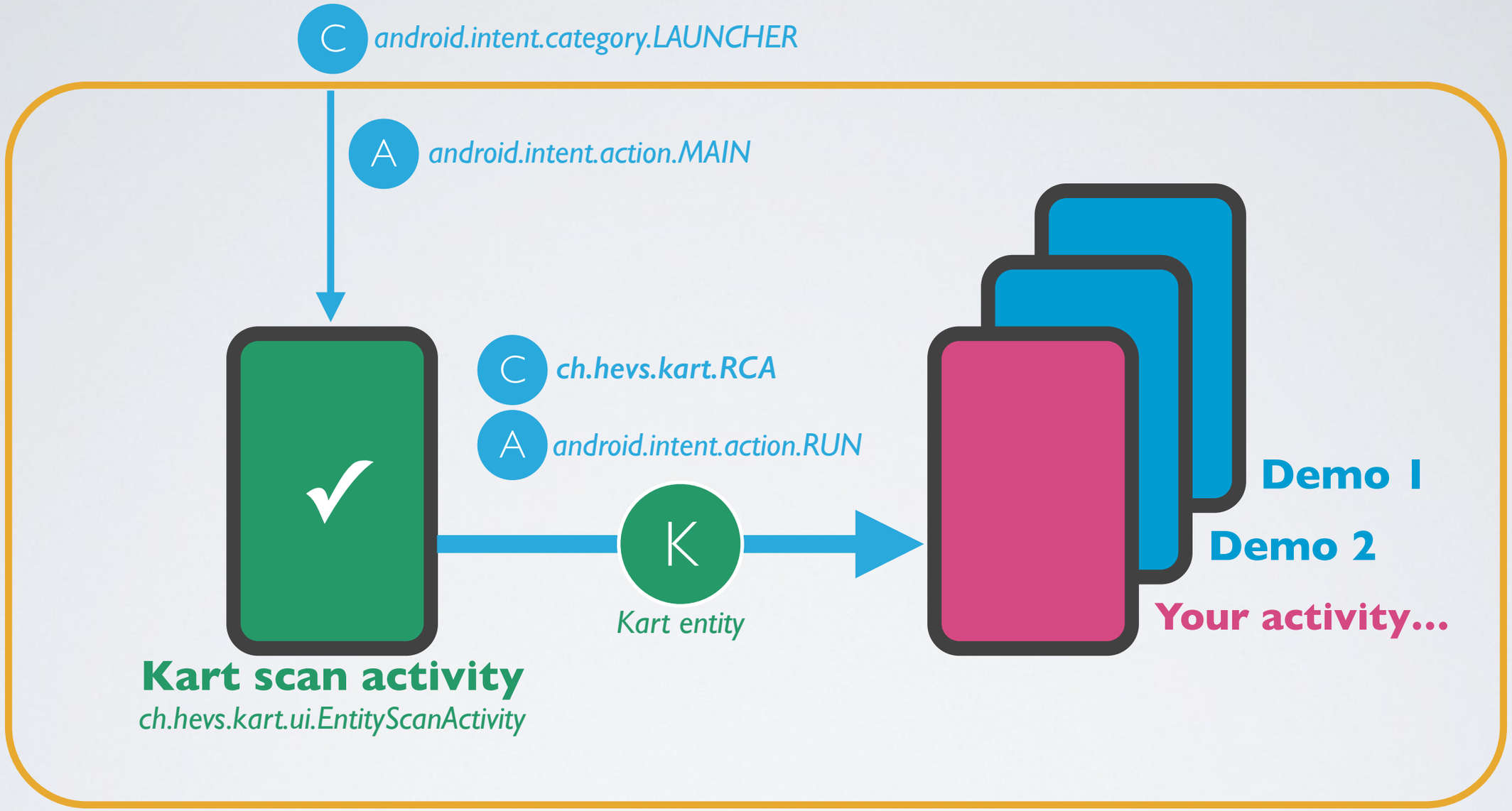
Layouts



- 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.

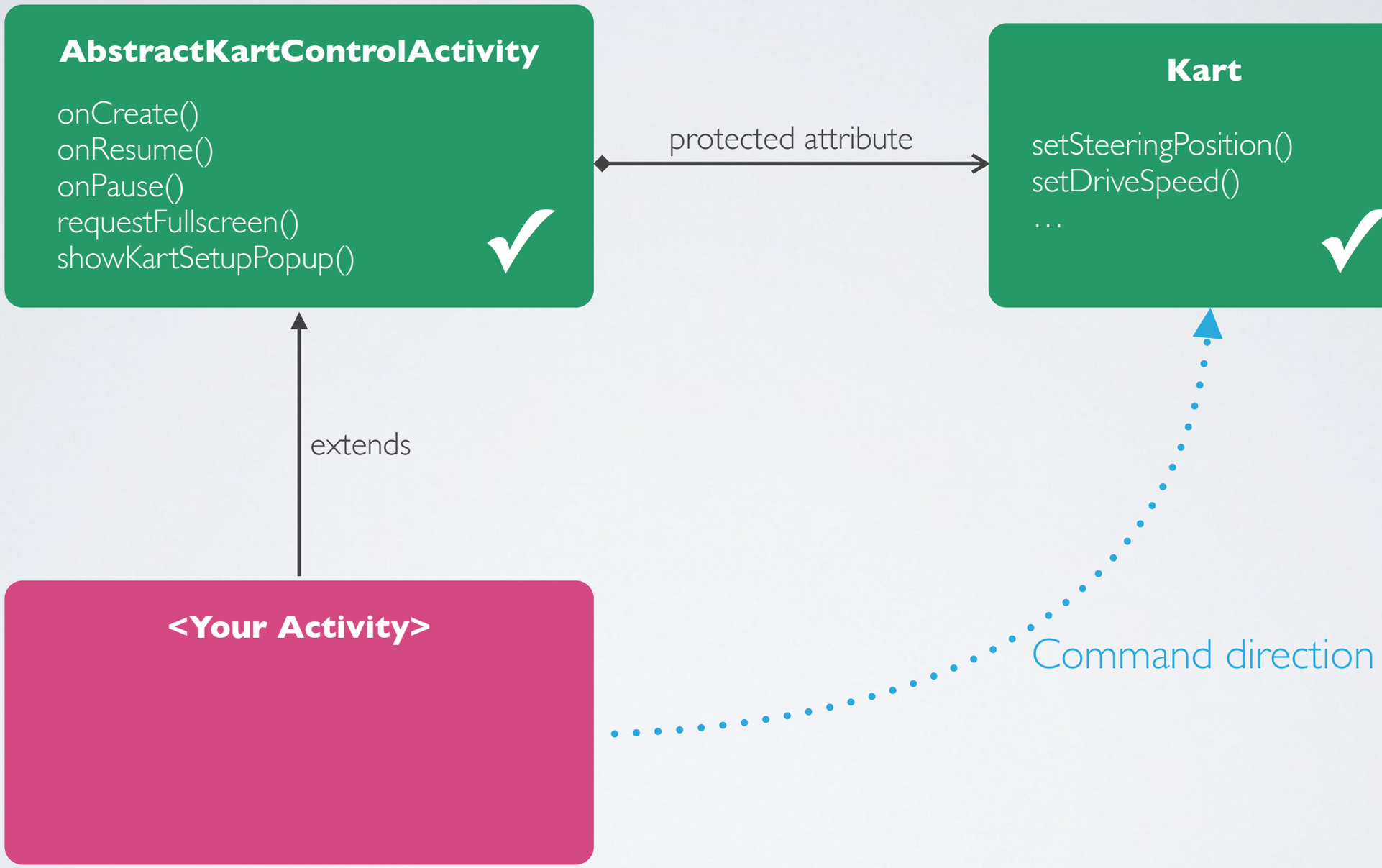
Remote Control Android App

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



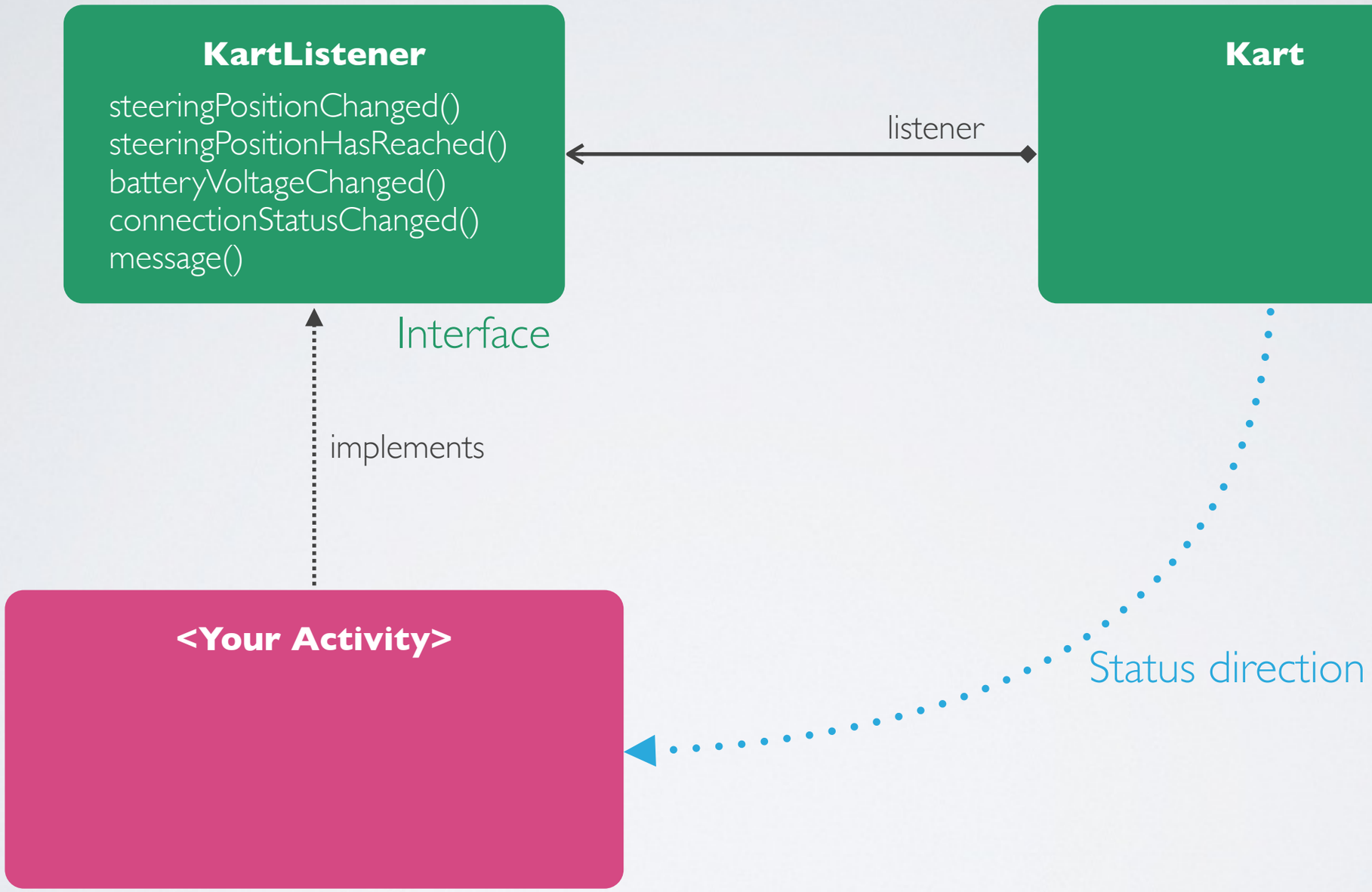
Remote Control Android App

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



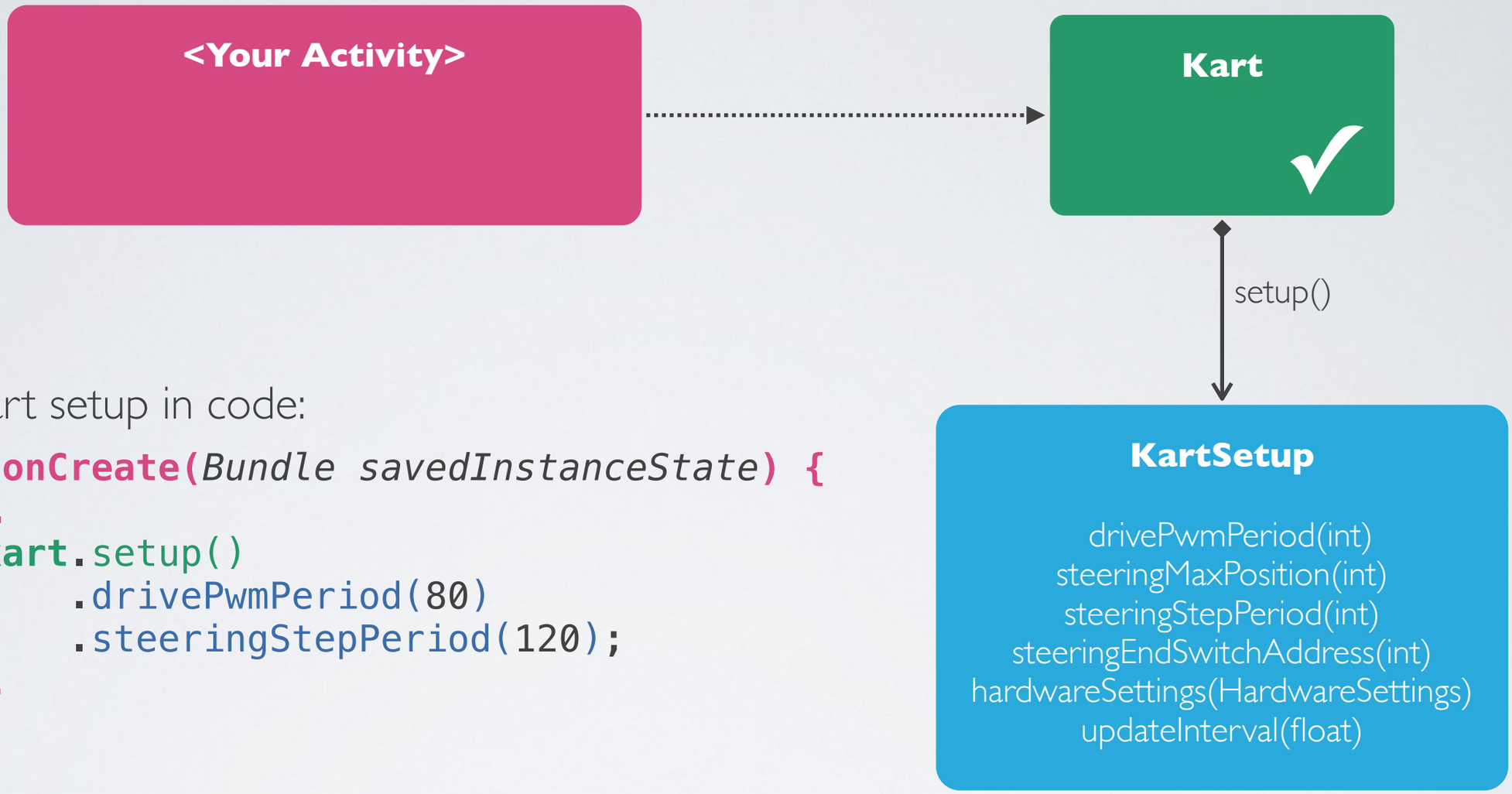
Remote Control Android App

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



Remote Control Android App

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



Manual kart setup in code:

```
void onCreate(Bundle savedInstanceState) {
    …
    kart.setup()
        .drivePwmPeriod(80)
        .steeringStepPeriod(120);
    …
}
```

Using the provided kart setup UI:

```
void onCreate(Bundle savedInstanceState) {
    …
    showKartSetupPopup();
    …
}
```

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

Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

Functional blackbox tests during morning of the last day

All mandatory features

Direction control
Speed control
Battery display
Direction display

→ 5.0

\+

0.25 ← **Per optional feature**

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

=

Grade

Android Studio

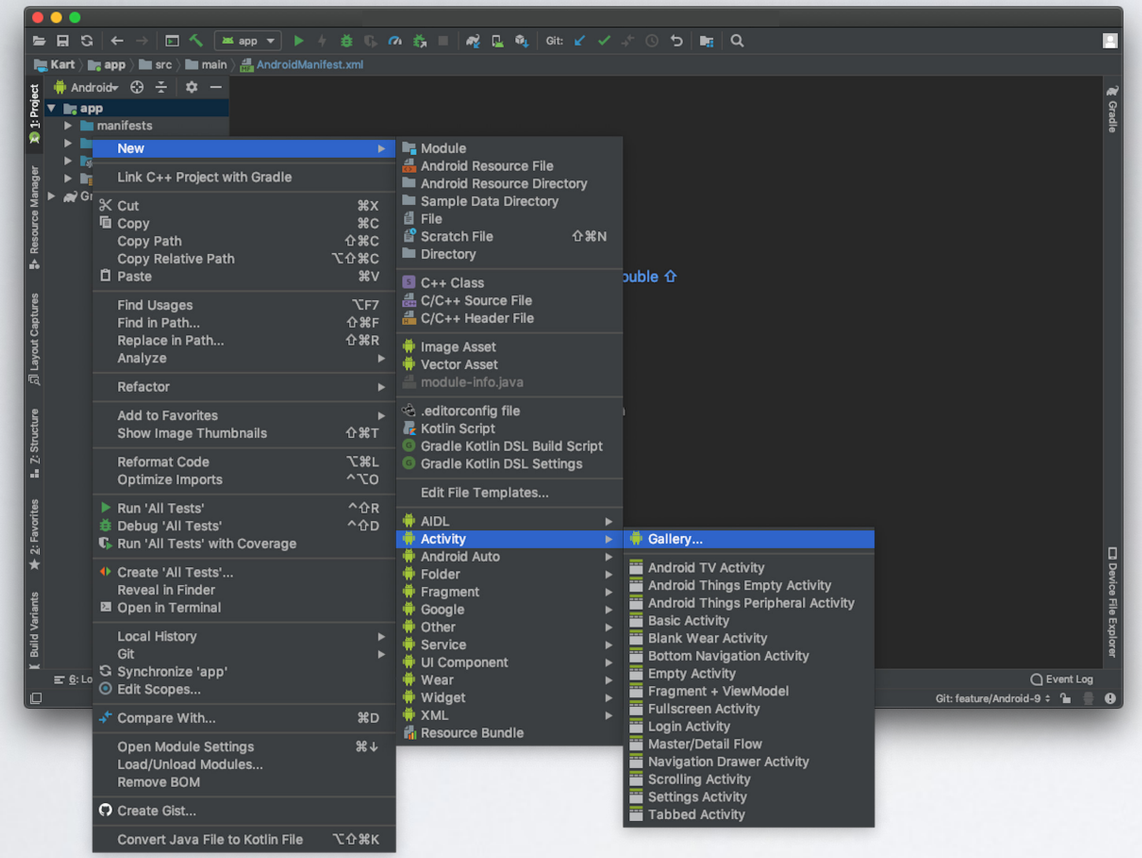
Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

1. **Download** the Kart project template from the wiki
2. **Extract** the archive to the **local** disk
3. **Open** the extracted folder in Android Studio

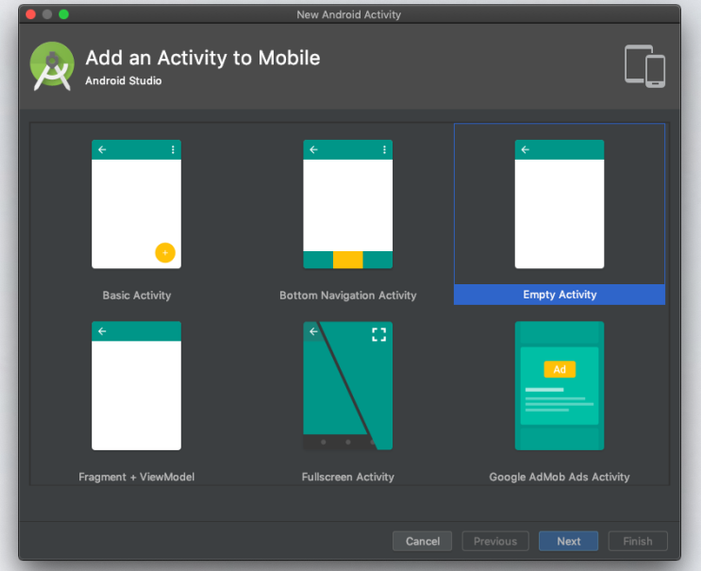
Android Studio

Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

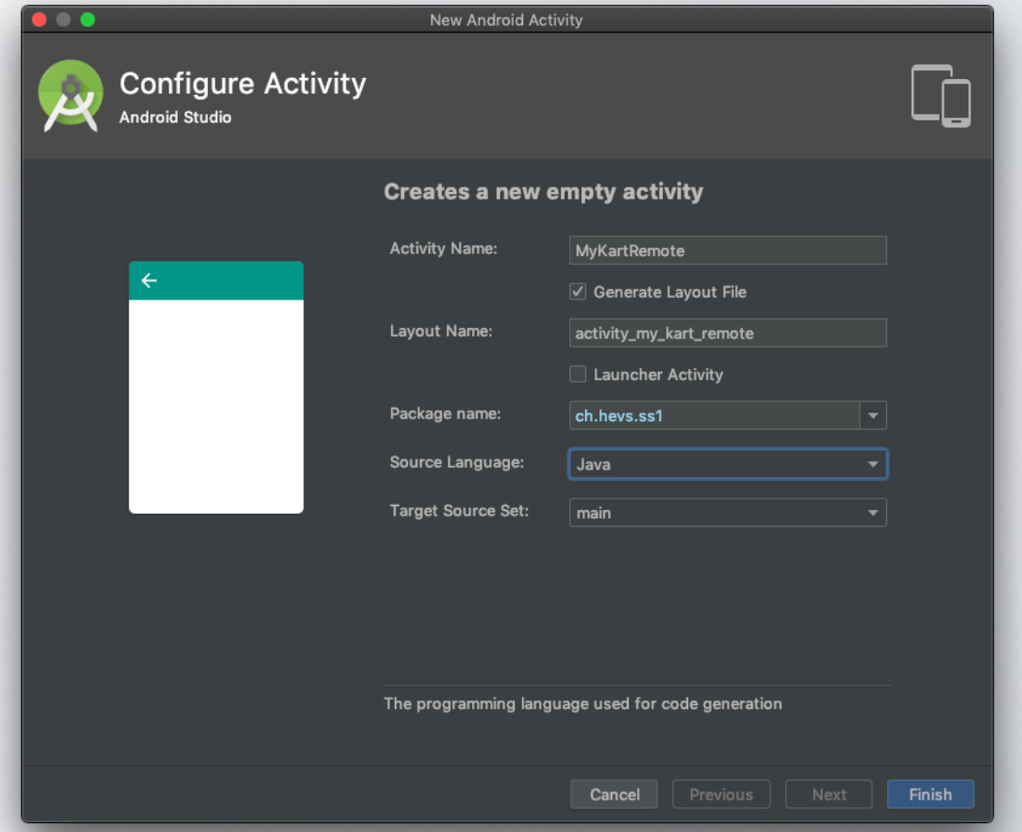
4 Select New > Activity > Gallery…



5 Select Empty Activity



6 Configure Activity



Android Studio

7 Modify **Activity Java code**

```java
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
    public void steeringPositionChanged(Kart kart, int i, float v) { }

    @Override
    public void steeringPositionReachedChanged(Kart kart, boolean b) { }

    @Override
    public void batteryVoltageChanged(Kart kart, float v) { }

    @Override
    public void sequenceCompleted(Kart kart) { }

    @Override
    public void connectionStatusChanged(Kart kart, boolean b) { }

    @Override
    public void message(Kart kart, String s) { }
}
```

8 Modify **AndroidManifest.xml**

```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"
        tools:ignore="GoogleAppIndexingWarning">

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

    </application>

</manifest>
```

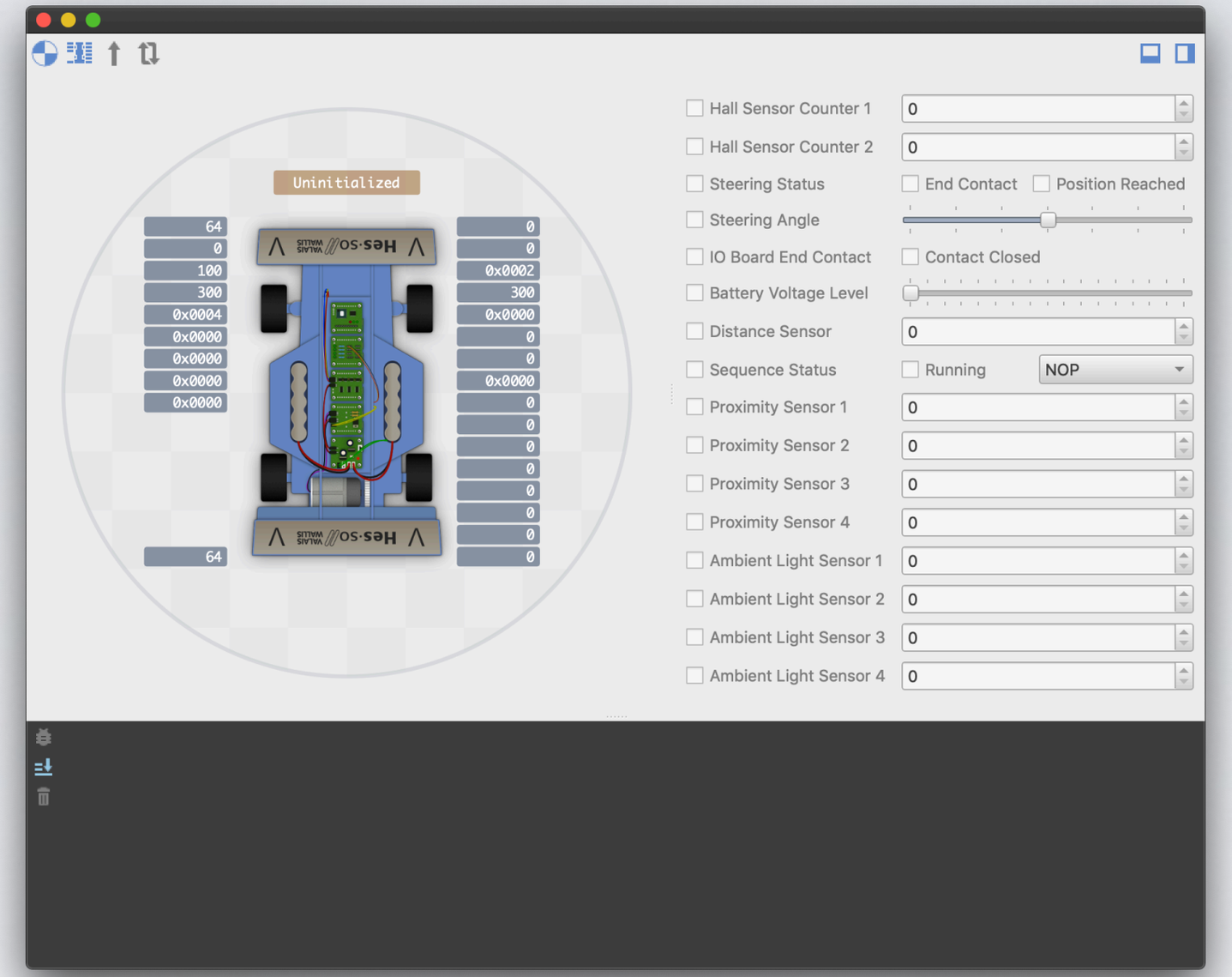
honor 10 Lite

Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

- 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.

Virtual Kart

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



Install & Run Demo

Download **Kart.apk** from:
http://wiki.hevs.ch/fsi/index.php5/Kart_2020

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

```
c:\Users\your.account> c:\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

- **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_2020
  - 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: Pair with your kart

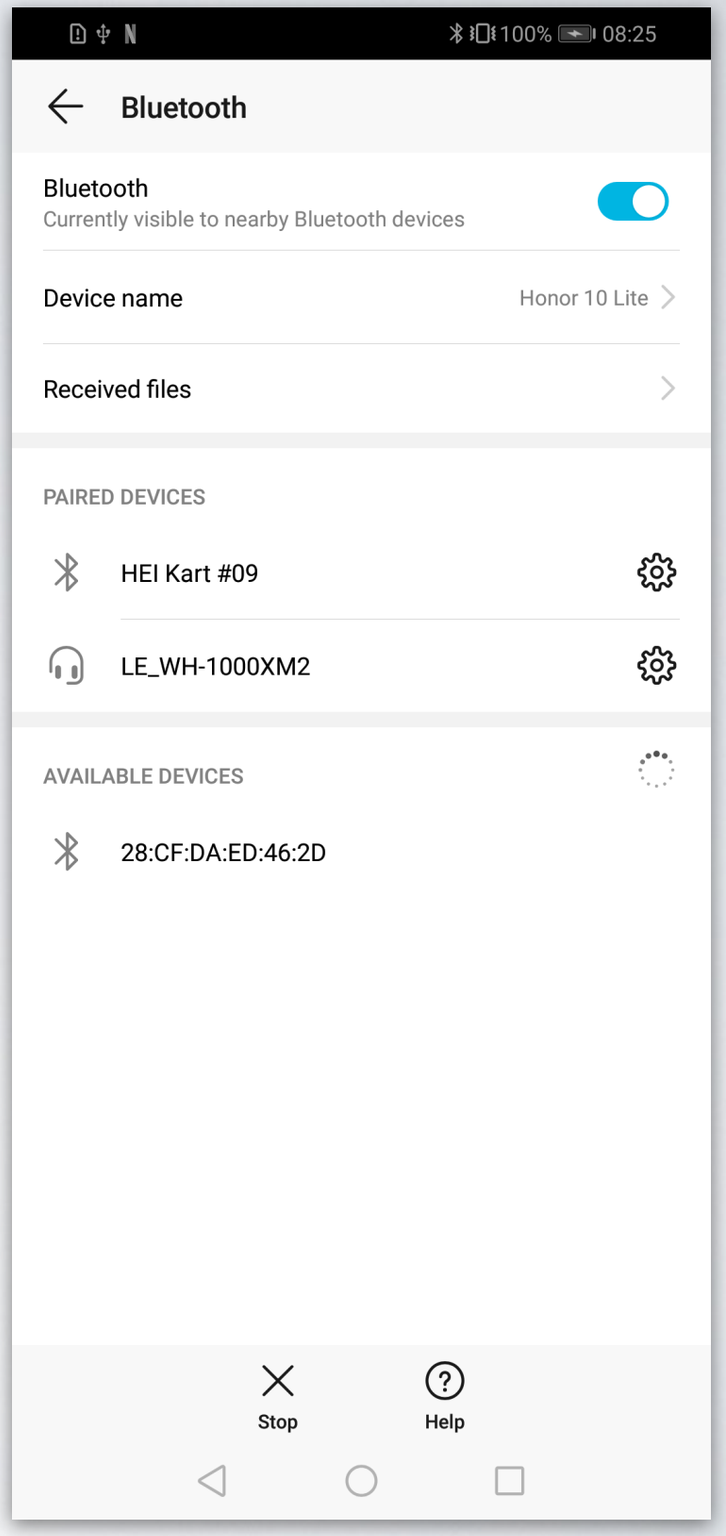
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

You need to pair with your kart before it will be listed by the Kart app

Open "Settings"

Go to "Device connectivity" > "Bluetooth"

Select your kart in "Available Devices"

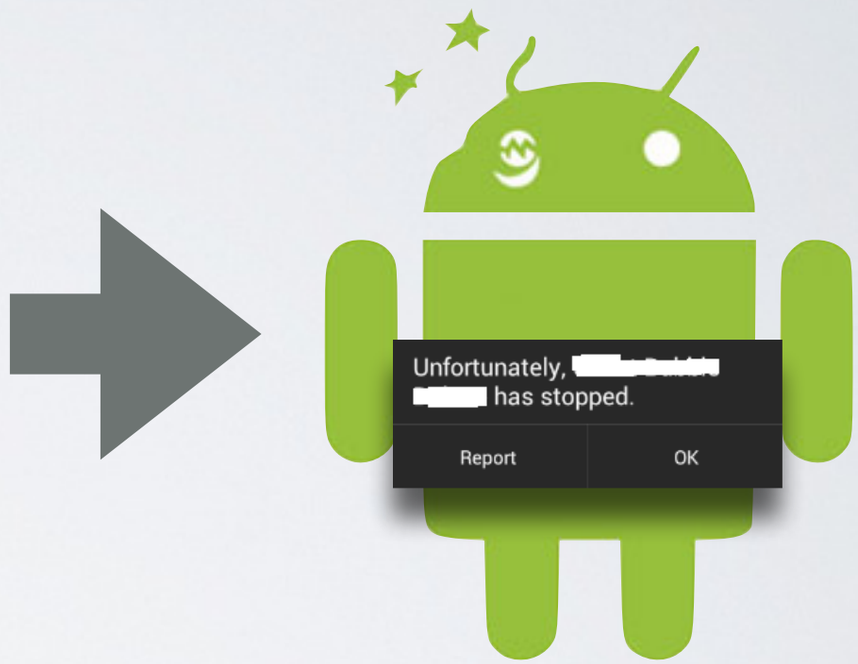


Tipp #3: No infinite loops

Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

```
// 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 #4: Periodic Timer

Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

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 #5: 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 #6: Animations…

Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

```
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

Tipp #7: Enumerations

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

You may not have seen Java Enumerations during your first year informatics course…

```java
public enum DayOfWeek {
    MONDAY,
    TUESDAY,
    WEDNESDAY,
    THURSDAY,
    FRIDAY,
    SATURDAY,
    SUNDAY
}
```

DayofWeek.java

```
Worktime
Today is TUESDAY
MONDAY
TUESDAY
WEDNESDAY
THURSDAY
FRIDAY
SATURDAY
SUNDAY
```

Program output

```java
DayOfWeek today = DayOfWeek.TUESDAY;

if (today == DayOfWeek.MONDAY) {
    System.out.println("Today sucks!");
}

switch (today) {
    case SATURDAY:
    case SUNDAY:
        System.out.println("Freetime");

    default:
        System.out.println("Worktime");
}

System.out.println("Today is " + today);

for (DayOfWeek dow: DayOfWeek.values()) {
    System.out.println(dow);
}
```

Main.java

Tipp #7: Enumerations

Hes·so // VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

```java
public enum KartControlRegister {
    DrivePwmPeriod(0),
    DriveSpeed(1),
    SteeringStepPeriod(2),
    SteeringPosition(3),
    SteeringEndSwitchAddress(4),
    HardwareSettings(5),
    LEDs(6),
    SequenceControlRegister(7),
    SequenceFifoRegister(8),
    UpdateInterval(15);

    private final int address;

    KartControlRegister(final int address) {
        this.address = address;
    }

    public int getAddress() {
        return address;
    }

    static final int COUNT = 16;
}
```

KartControlRegister.java

```
KartControlRegister has 16 registers.
LEDs register has address 6
```

Program output

```java
System.out.println("KartControlRegister has " + KartControlRegister.COUNT + " registers. ");

KartControlRegister register = KartControlRegister.LEDs;
System.out.println("" + register + " register has address " + register.getAddress());
```

Main.java