Hes·so VALAIS WALLIS

Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

Kart Programming

François Corthay | Charles Praplan

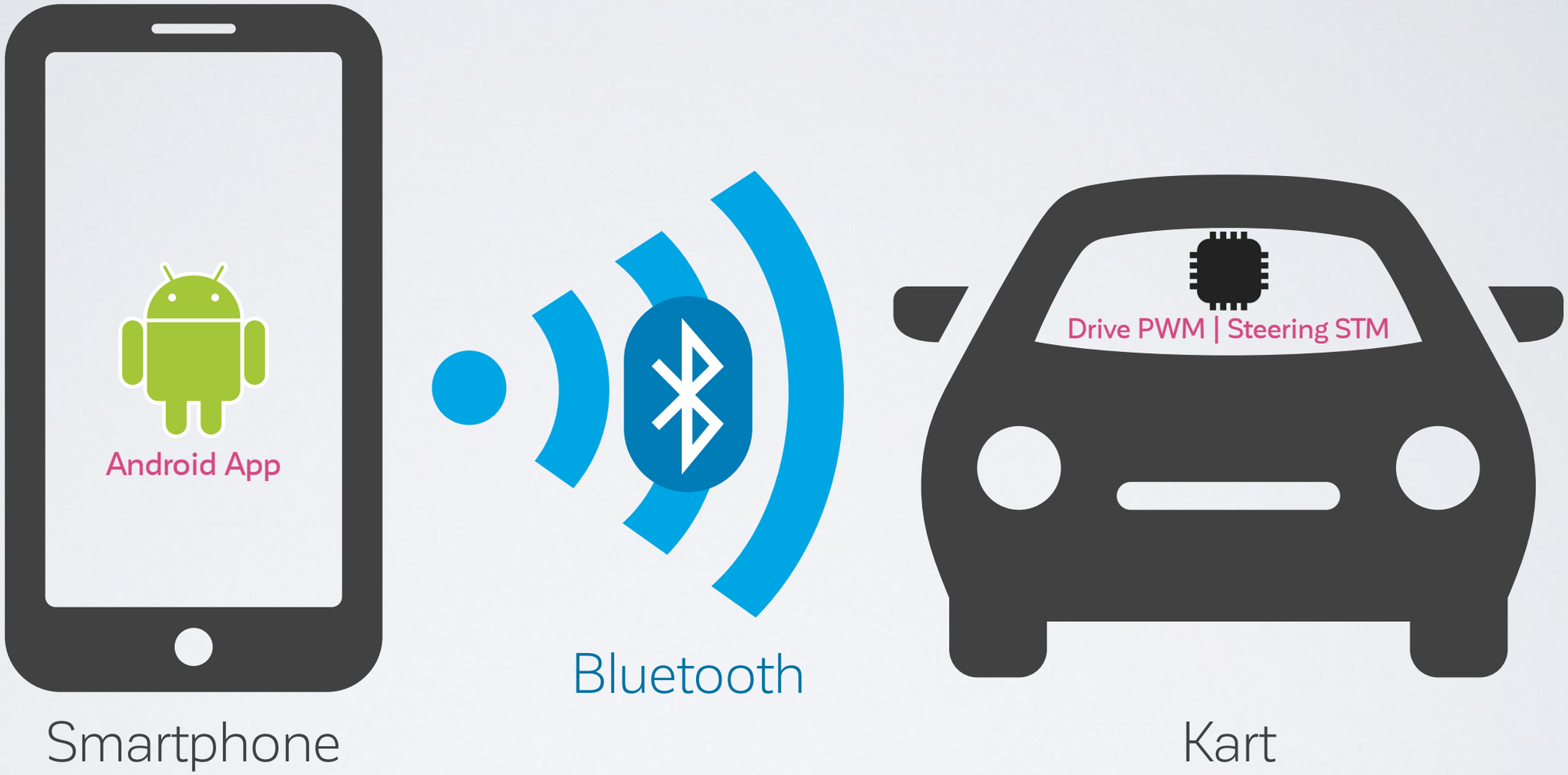
Christopher Metrailler | Patrice Rudaz | Michael Clausen | Yoan Rossier

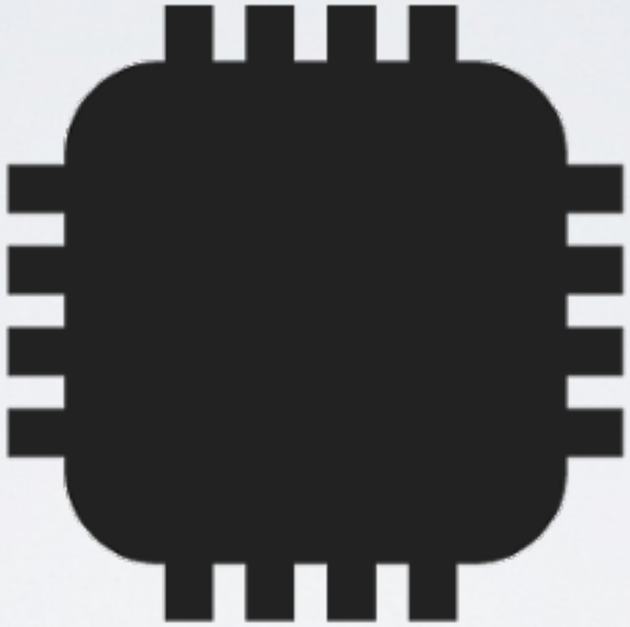
Goals

Use basic elements seen in **ELN** course.

Use basic elements seen in **INF I** course.

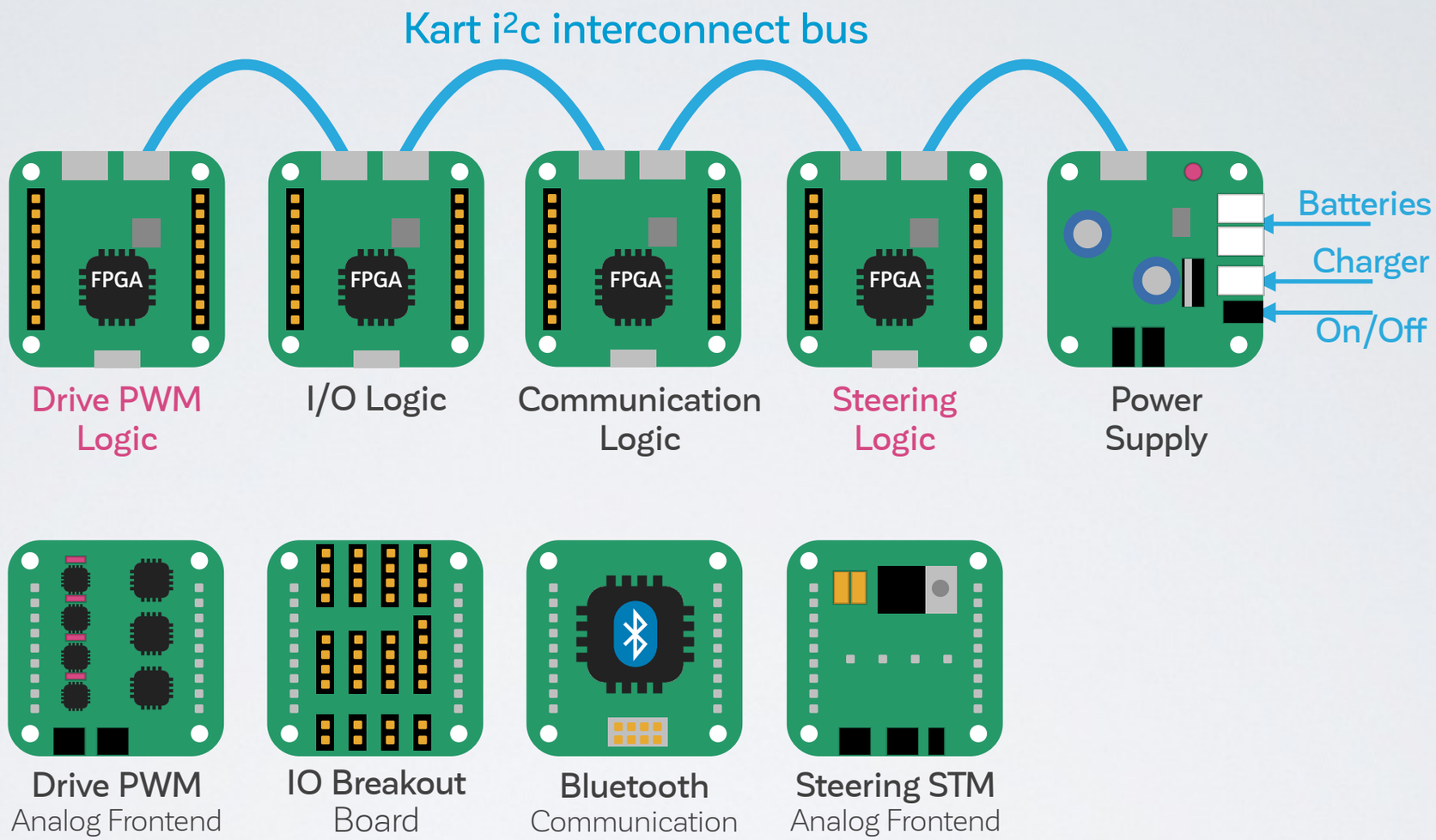
Goals


Android App
Smartphone
Bluetooth
Drive PWM | Steering STM
Kart

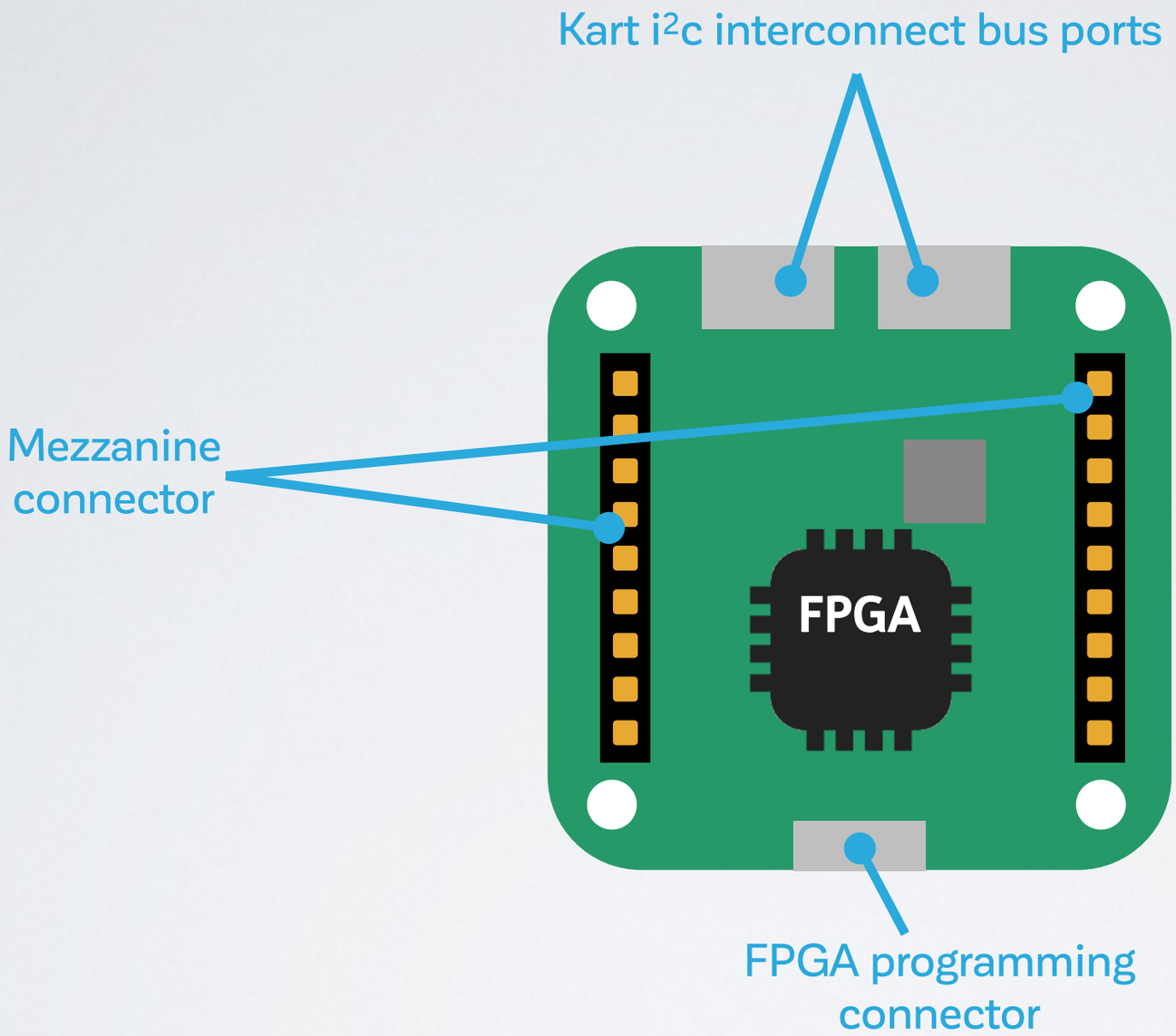


ELN part

Modular concept



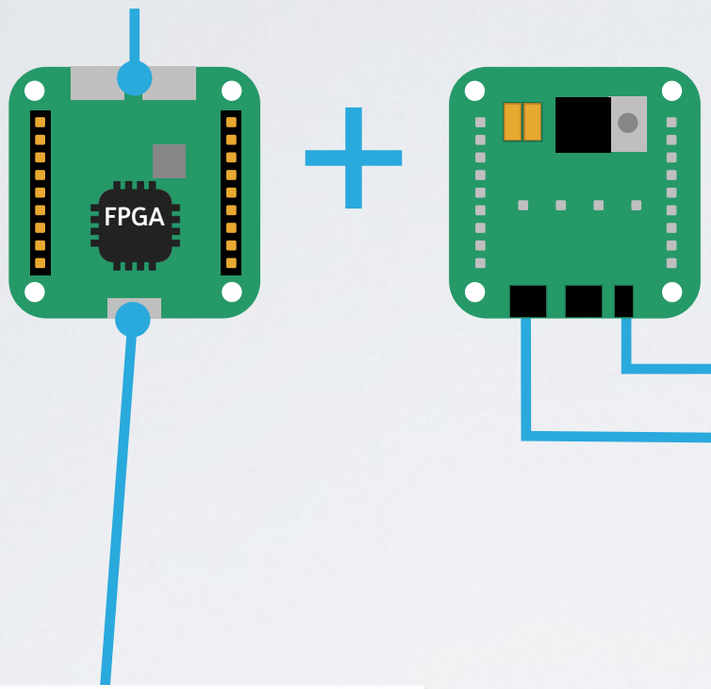
Generic FPGA Board

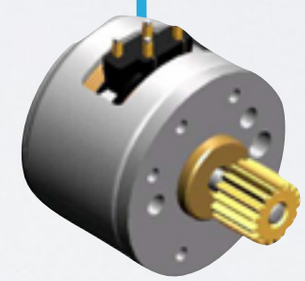


Stepper Motor

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

Kart i²c interconnect bus



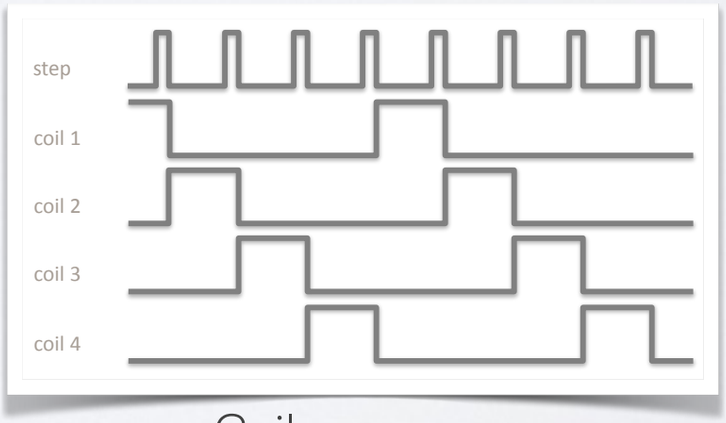


Stepper motor

Limit switch

```
--------------------------------------------------------------------------------
-- Company: Young Embedded Systems LLC
-- Engineer: Gene Breniman
-- Module Name:    ARM_SEQ_RAM- - Behavioral
-- Revisions:
--    0.01 - 07/02/2007 File Created
-- Additional Comments:
--
--------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ClkDiv is
    Port (  InByte : in STD_LOGIC_VECTOR(3 downto 0);       --<-- Seq_CPLD
            RegSel : in STD_LOGIC_VECTOR(1 downto 0);       --<-- Seq_CPLD
            RegStrb : in STD_LOGIC;                         --<-- Seq_CPLD
            MClk : in  STD_LOGIC;                           --<-- OSC
            SeqReset : in STD_LOGIC;                        --<-- Power Monitor
            ADC_Clk : out STD_LOGIC);                       -->-- ADC
end ClkDiv;

architecture Behavioral of ClkDiv is
   signal   ADC_div :      STD_LOGIC_VECTOR(5 downto 0) := "001111";
   signal   ADCClk :       STD_LOGIC := '0';
   signal   ClkSel :       STD_LOGIC_VECTOR(2 downto 0) := "100";

begin

   ClkDivP : process(Mclk,SeqReset)
      begin
      if SeqReset = '0' then
         ADCClk <= '0';
         ADC_div <= "001001";
      elsif Mclk = '0' and Mclk'event then
         if ADC_div = "000000" then
            ADCClk <= not(ADCClk);
            case ClkSel is
               when "000" =>                    -- 20MHz - divide by 2
               when "001" =>                    -- 10MHz
                  ADC_div <= "000001";             -- divide by 4
               when "010" =>                    -- 4MHz
                  ADC_div <= "000100";             -- divide by 10
               when "011" =>                    -- 2MHz
                  ADC_div <= "001001";             -- divide by 20
               when "100" =>                    -- 1MHz
                  ADC_div <= "001001";             -- divide by 40
               when others =>                   -- 400KHz
```

Stepper motor control

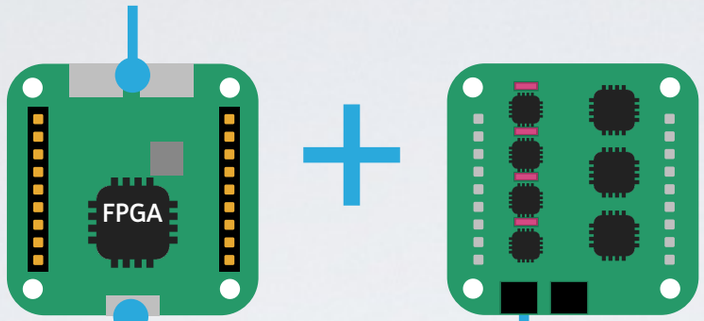


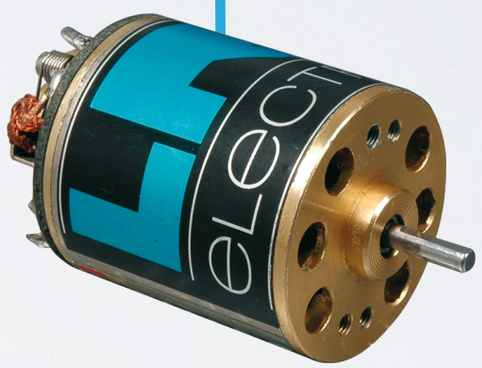
Coil sequence

Drive Motor

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

Kart i²c interconnect bus



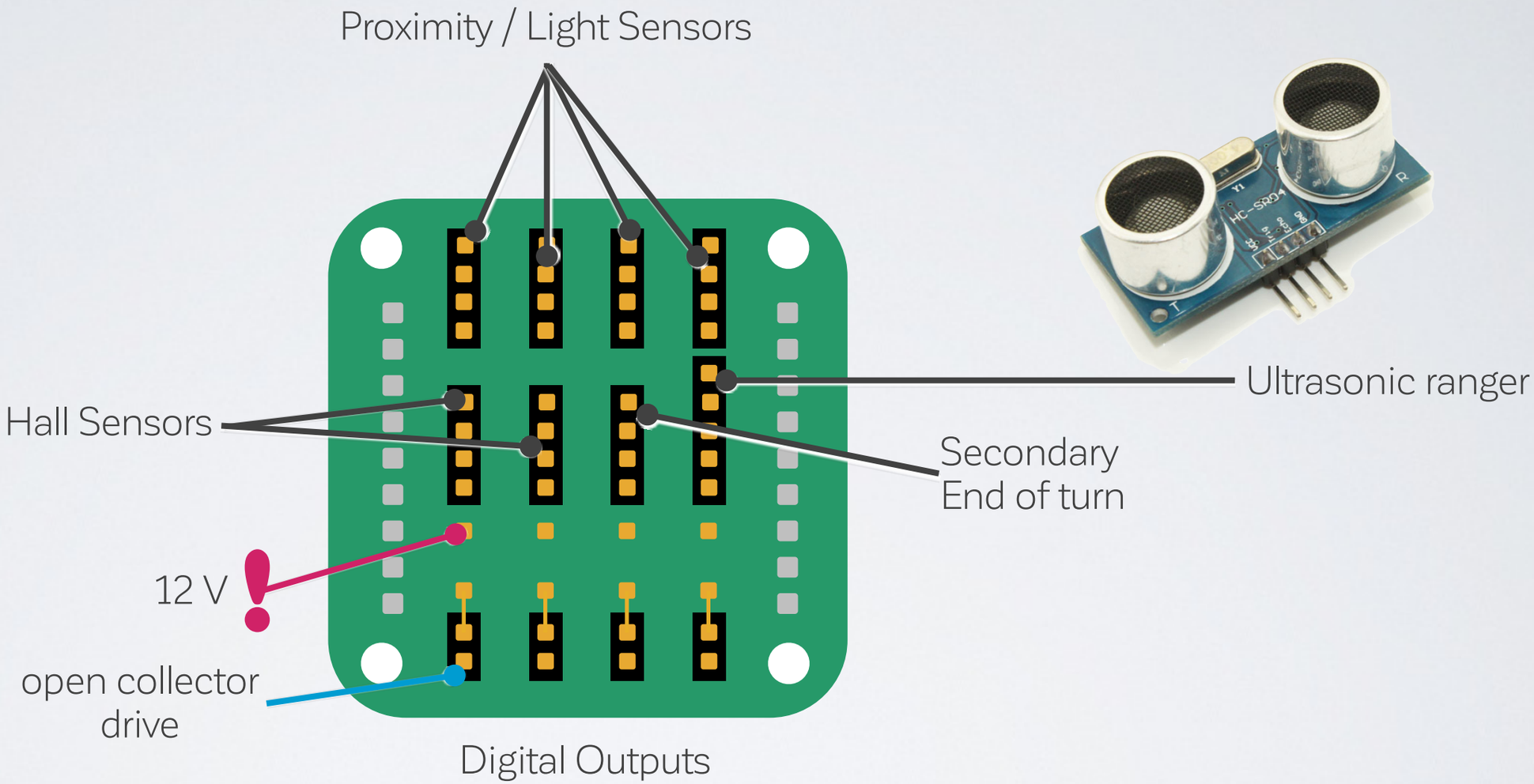


Drive motor

```
--------------------------------------------------------------------------------
-- Company: Young Embedded Systems LLC
-- Engineer: Gene Breniman
-- Module Name:    ARM_SEQ_RAM- - Behavioral
-- Revisions:
--    0.01 - 07/02/2007 File Created
-- Additional Comments:
--
--------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ClkDiv is
    Port (  InByte : in STD_LOGIC_VECTOR(3 downto 0);       --<-- Seq_CPLD
            RegSel : in STD_LOGIC_VECTOR(1 downto 0);       --<-- Seq_CPLD
            RegStrb : in STD_LOGIC;                         --<-- Seq_CPLD
            Mclk : in  STD_LOGIC;                           --<-- OSC
            SeqReset : in STD_LOGIC;                        --<-- Power Monitor
            ADC_Clk : out STD_LOGIC);                       -->-- ADC
end ClkDiv;

architecture Behavioral of ClkDiv is
   signal   ADC_div :      STD_LOGIC_VECTOR(5 downto 0) := "001111";
   signal   ADCClk :       STD_LOGIC := '0';
   signal   ClkSel :       STD_LOGIC_VECTOR(2 downto 0) := "100";

begin

   ClkDivP : process(Mclk,SeqReset)
      begin
      if SeqReset = '0' then
         ADCClk <= '0';
         ADC_div <= "001001";
      elsif Mclk = '0' and Mclk'event then
         if ADC_div = "000000" then
            ADCClk <= not(ADCClk);
            case ClkSel is
               when "000" =>                    -- 20MHz - divide by 2
               when "001" =>                    -- 10MHz
                  ADC_div <= "000001";             -- divide by 4
               when "010" =>                    -- 4MHz
                  ADC_div <= "000100";             -- divide by 10
               when "011" =>                    -- 2MHz
                  ADC_div <= "001001";             -- divide by 20
               when "100" =>                    -- 1MHz
                  ADC_div <= "001001";             -- divide by 40
               when others =>                   -- 400KHz
```

Drive PWM control

IO Board


Proximity / Light Sensors
Ultrasonic ranger
Hall Sensors
Secondary
End of turn
12 V
open collector
drive
Digital Outputs

Avoid Hardware Damages

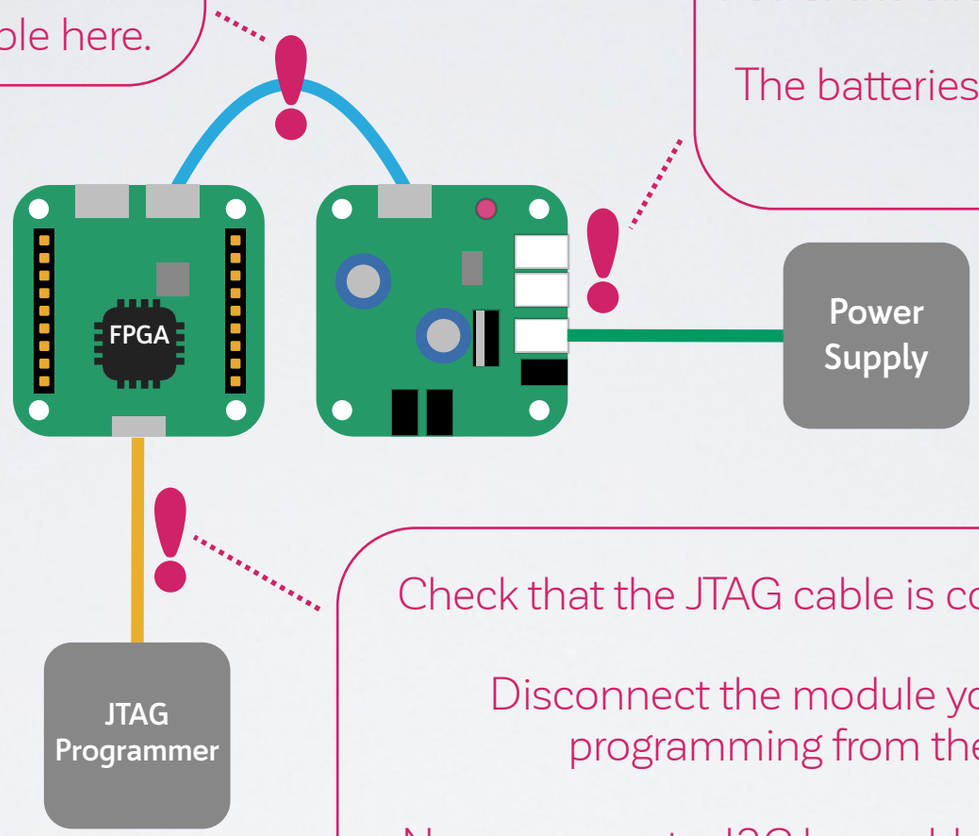
Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

Double-check that the cable/connector are connected correctly.

Never connect JTAG connector cable here.

Power the circuit using a laboratory power supply.

The batteries should never be connected during development!



Check that the JTAG cable is connected correctly!

Disconnect the module you are actually programming from the I2C bus!

Never connect a I2C bus cable to this connector!

If you connect something wrong, the FPGA might be **damaged**.

The costs to change a FPGA are about **50 SFr.**

You will be charged for the reparation if you did not follow this guidelines!

Hardware Goals

- Control block for DC motor
  - Pulse Width Modulation (PWM) generator
- Control block for stepper motor
  - 4 Coil forward/backward sequence generator
- Hall-Sensor Counter
- Various additional sensors and actuators
  - Personal ideas are welcome

Hardware Grade

Presentation of blocks and simulation results during morning of the last day

All mandatory features
Direction Stepper control
Speed PWM control
Hall sensor counter

→ **4.0**

+

1 ← **Per optional feature**
Ultrasound sensor
Emergency Stop (Proximity sensor)
Other improvements

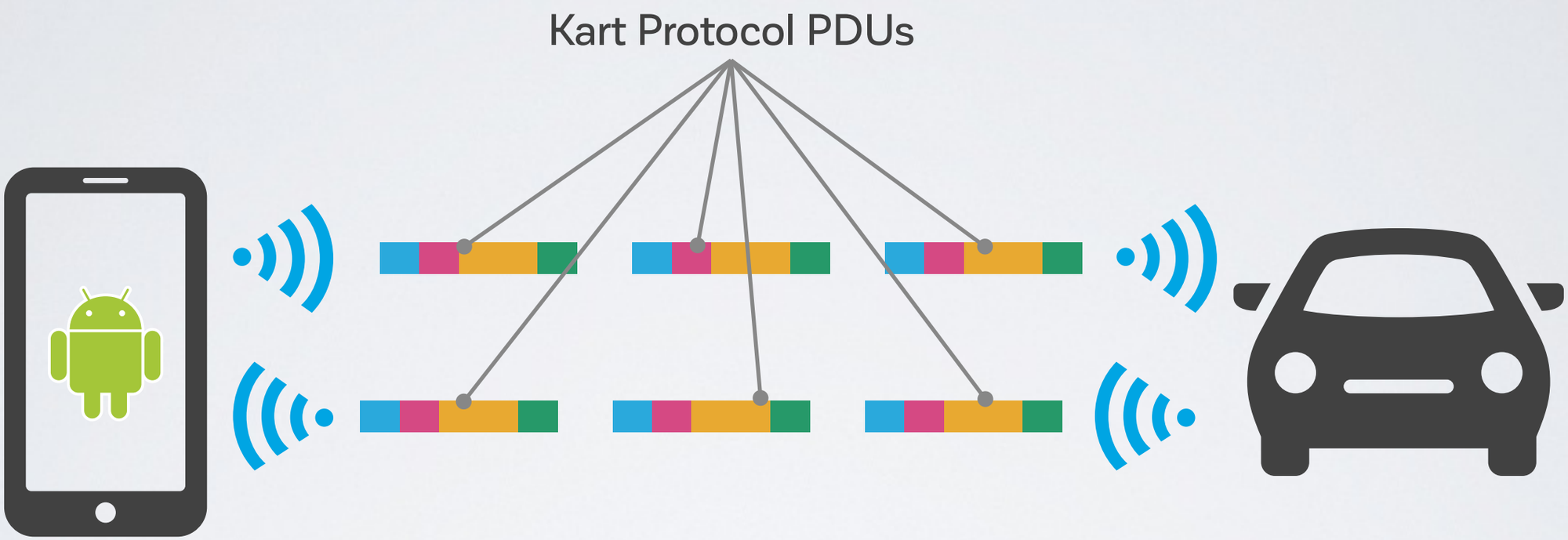
=

Grade

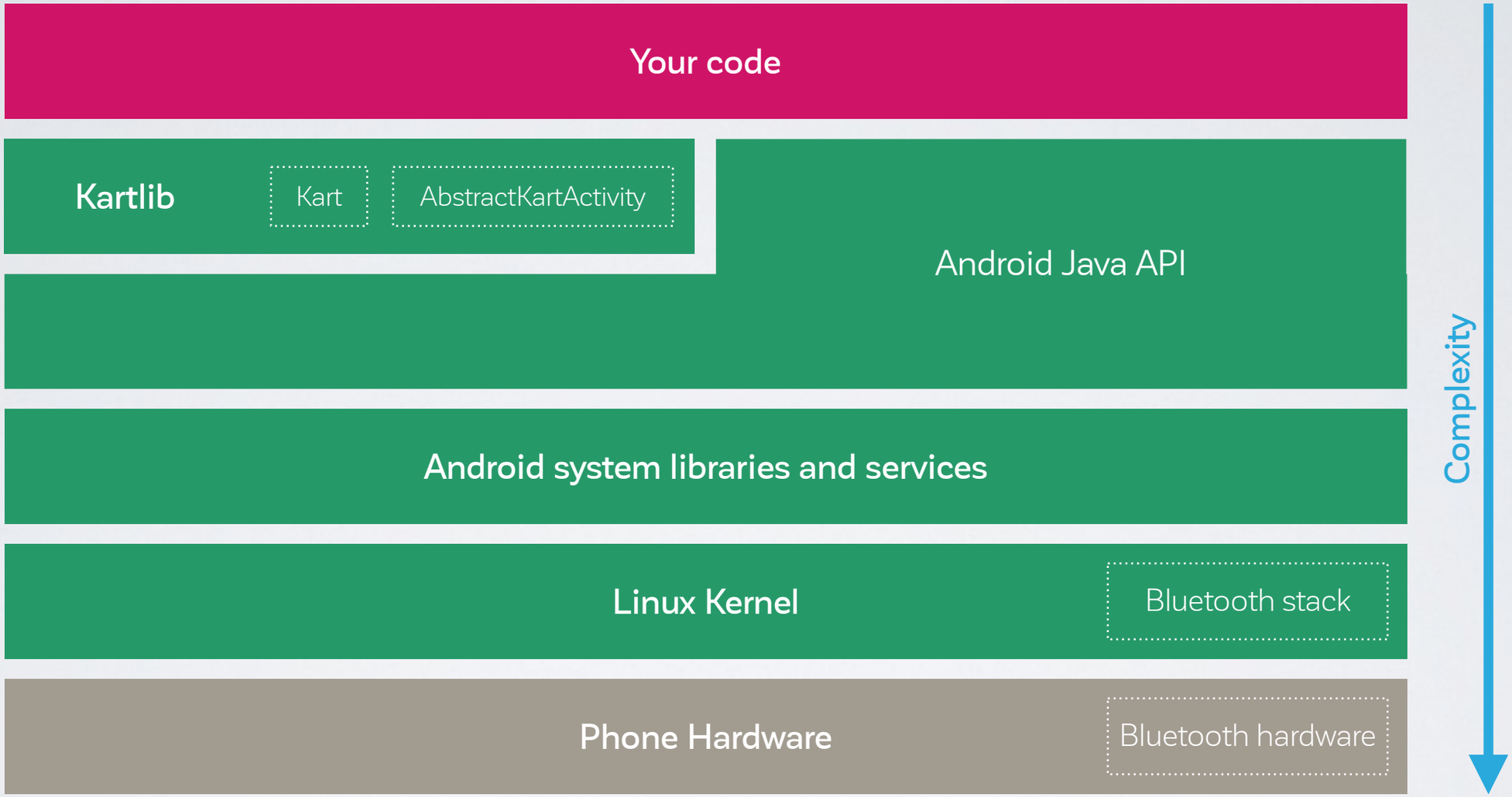
INF part

Remote Control Protocol

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



Remote Control Protocol



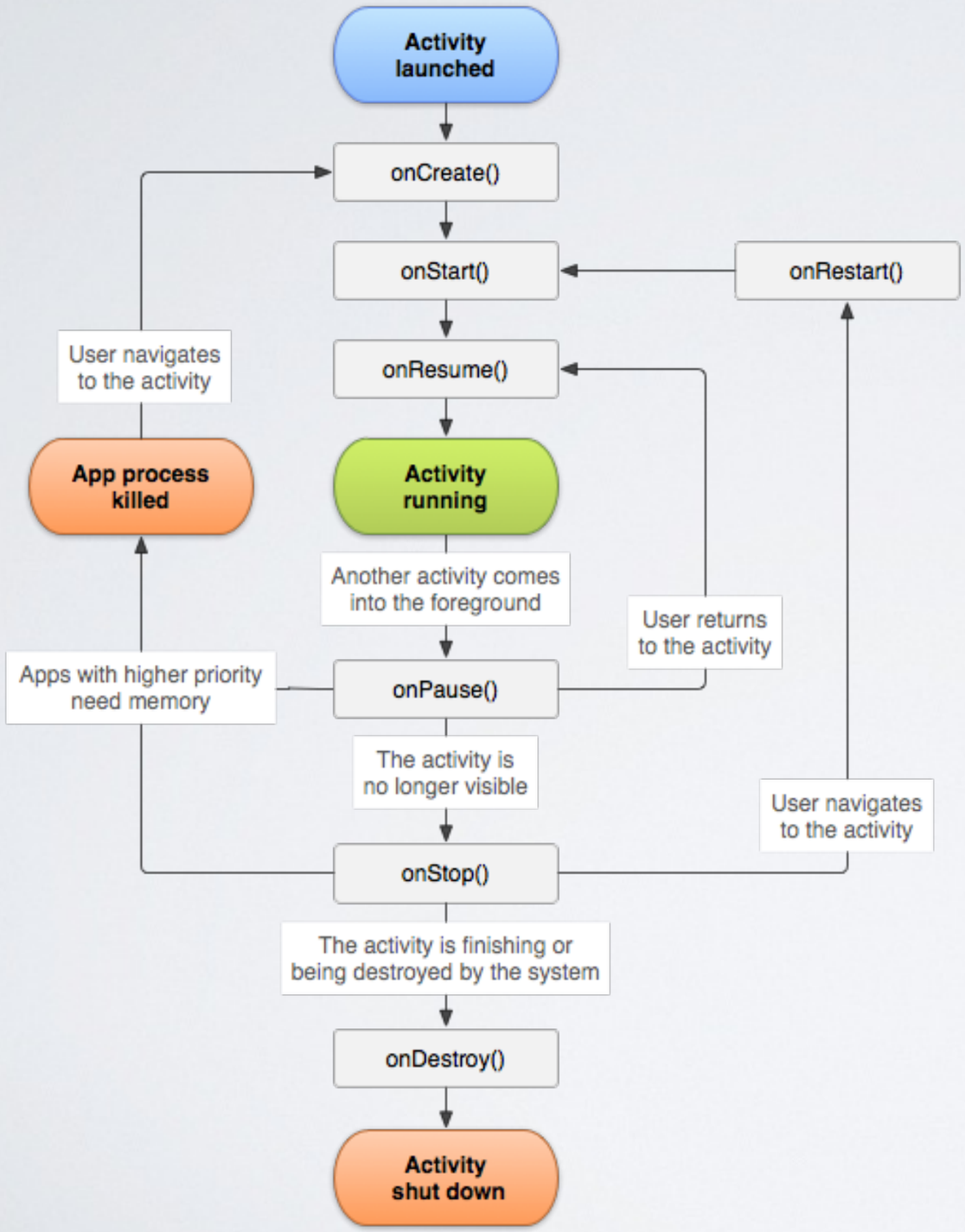
Remote Control Android App

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

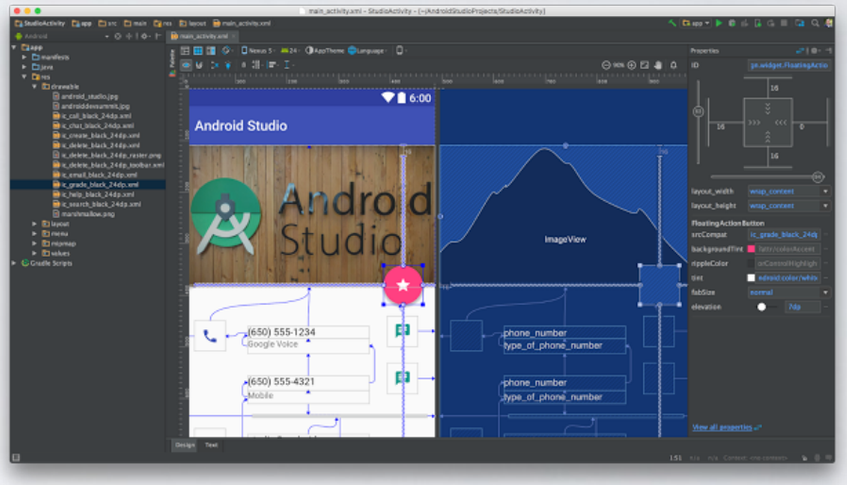
Android

- 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** (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 Application lifecycle



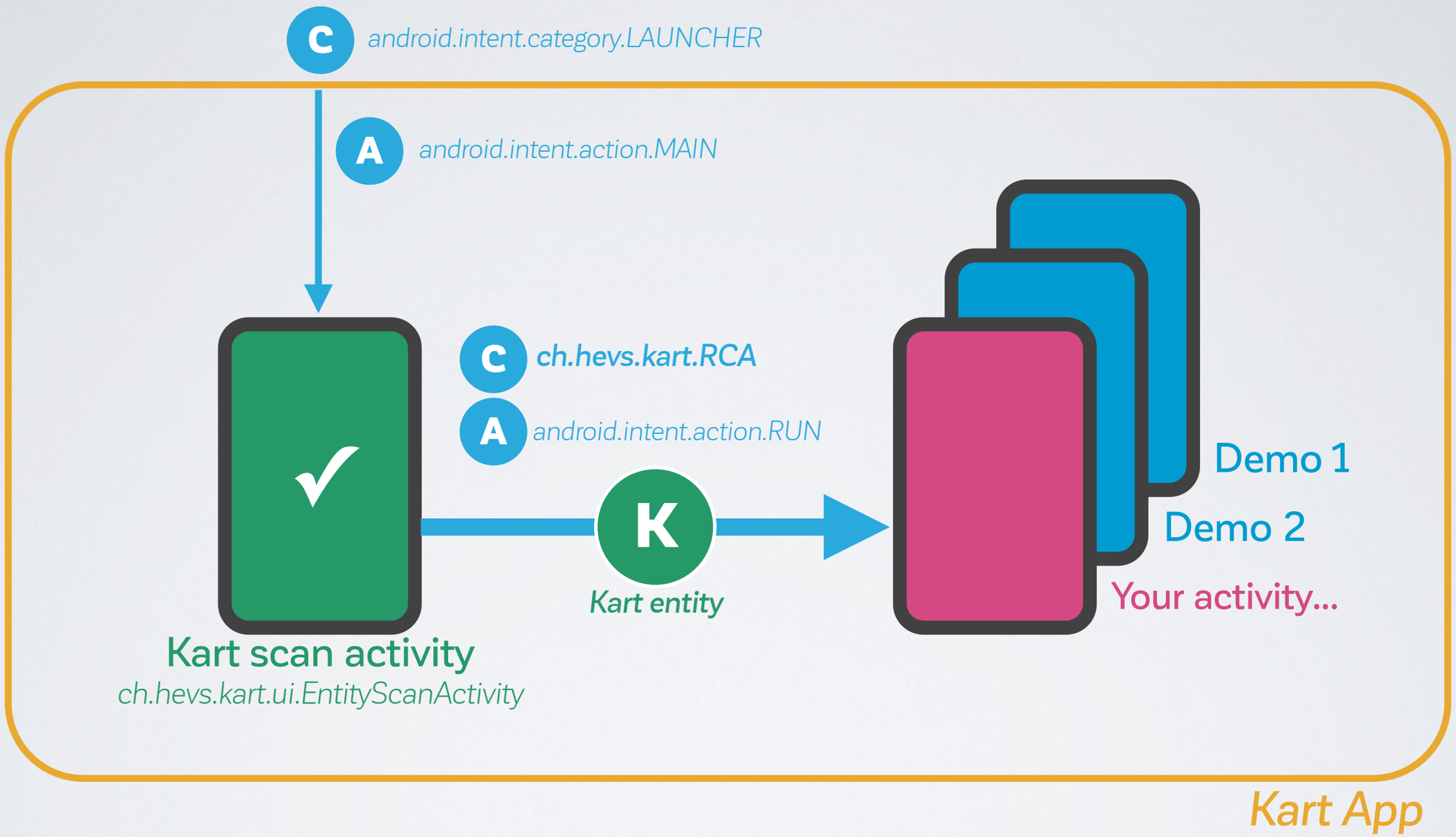
Layouts



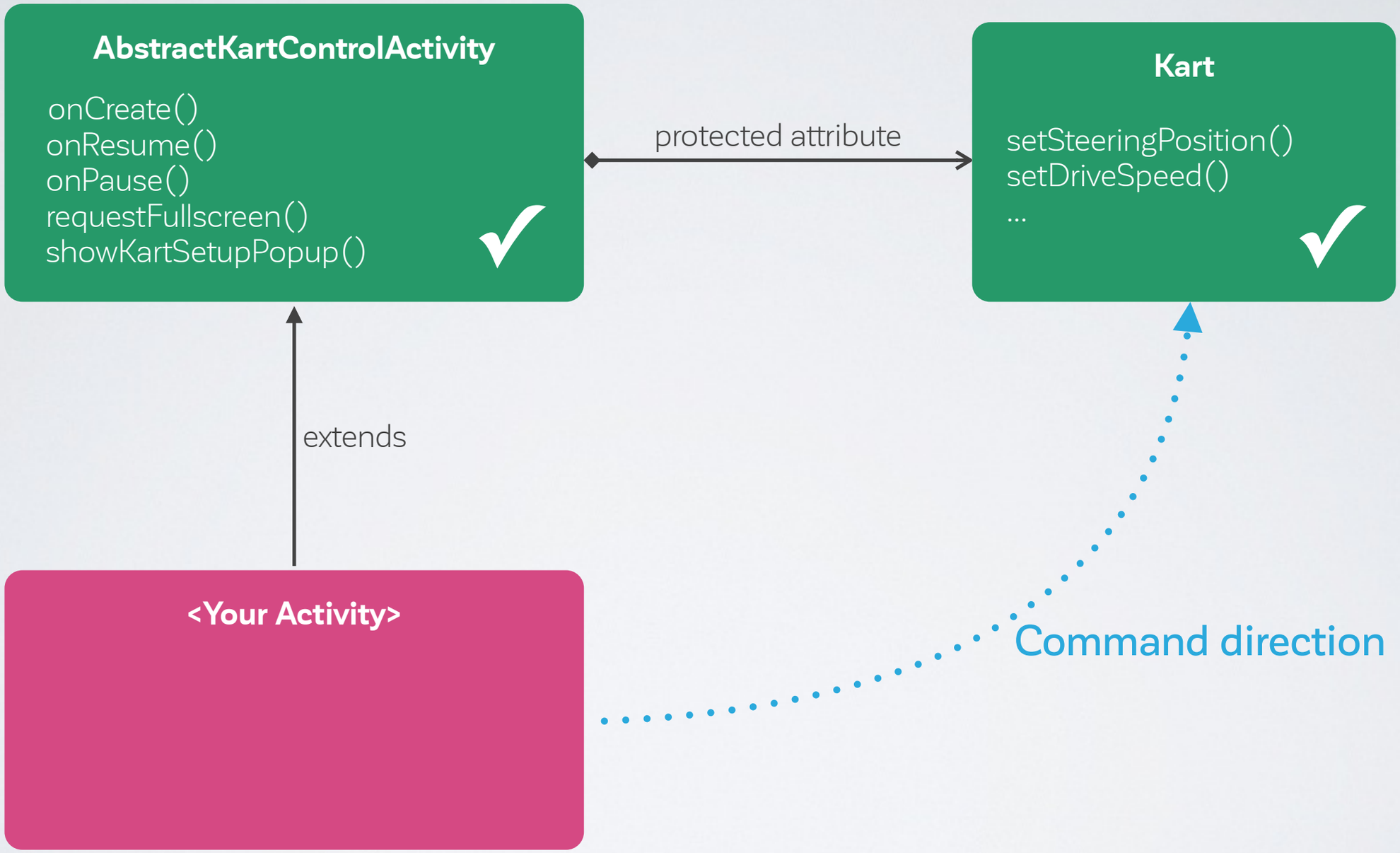
- UI layouts can be designed using an editor integrated into Android Studio.
  - Layouts are serialized to XML files.
  - Those Layouts can be loaded in code.

Remote Control Android App

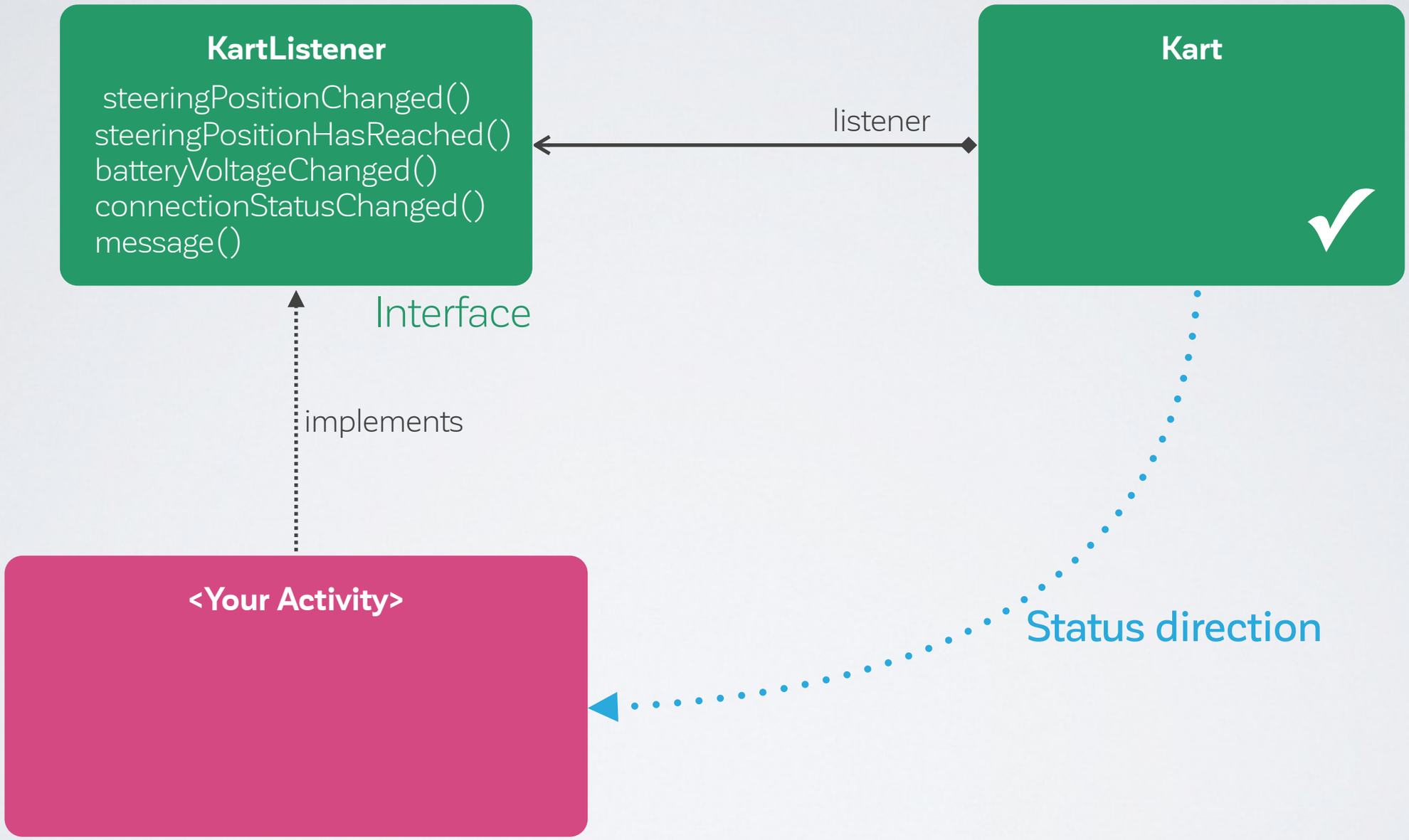
Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



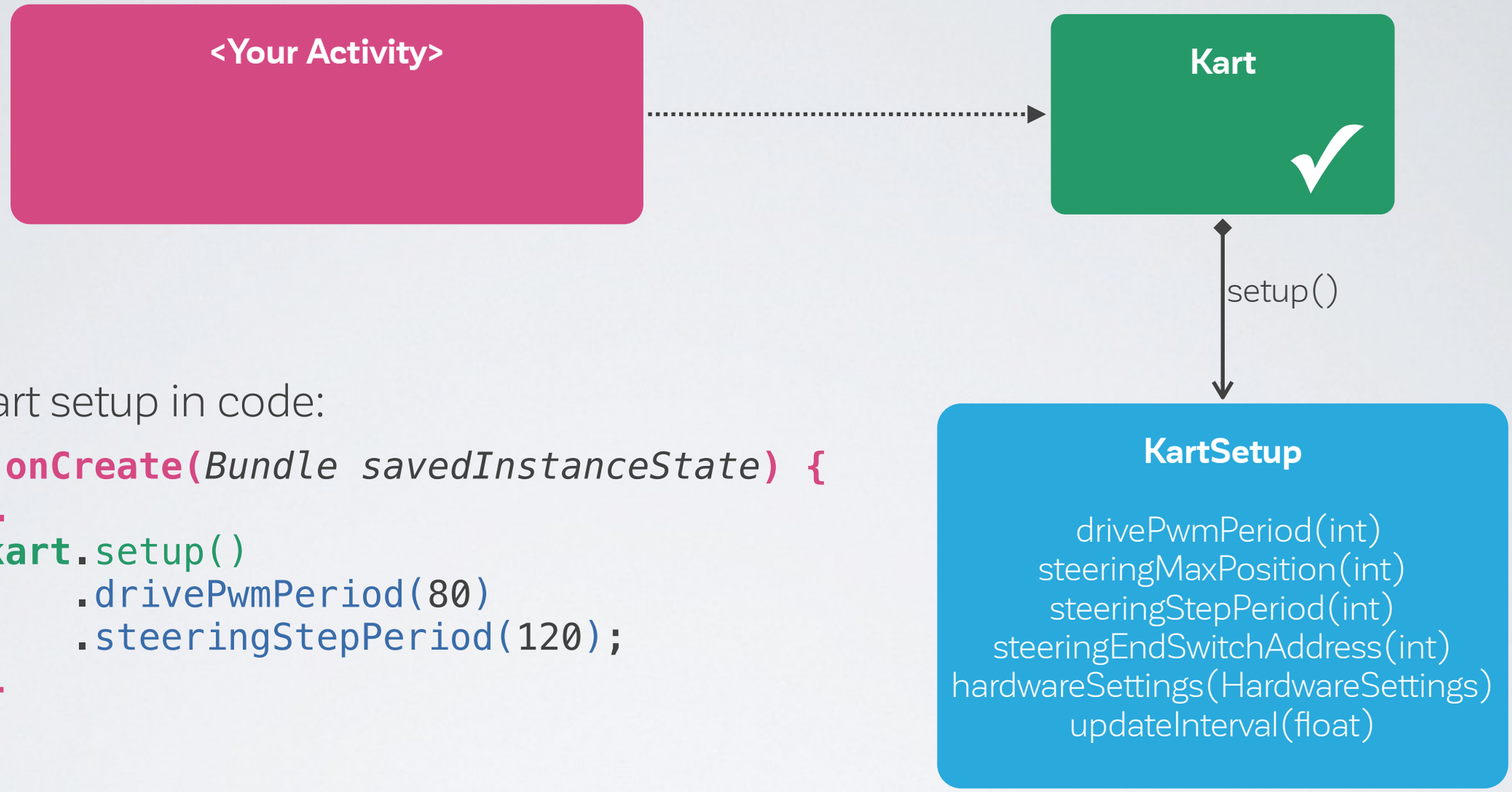
Remote Control Android App



Remote Control Android App



Remote Control Android App



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

Functional blackbox tests during morning of the last day

All mandatory features

Direction control
Speed control
Battery display
Direction display

→ **4.0**

+

0.5 ←

Per optional feature

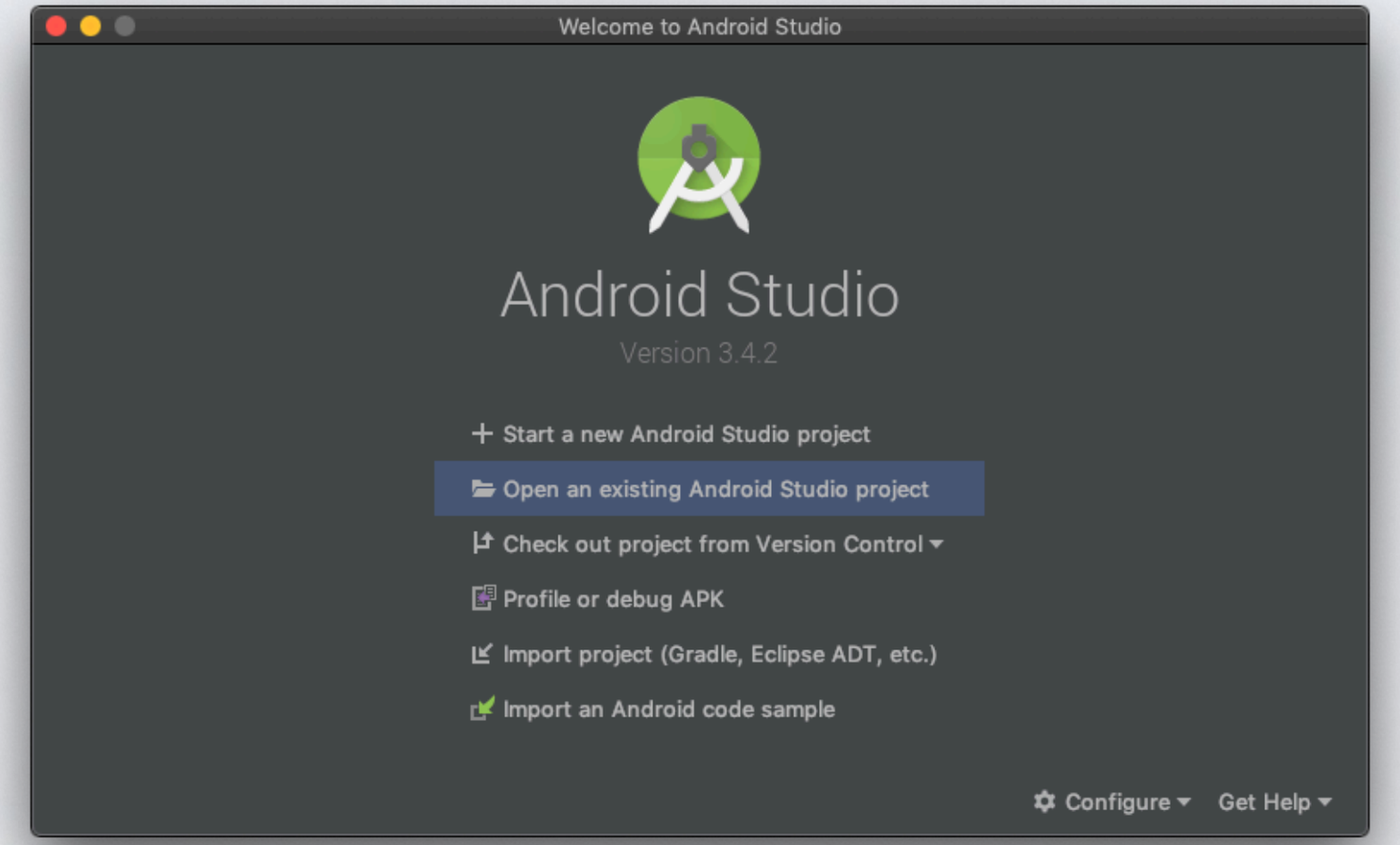
LED's
Ultrasound sensor
Proximity sensor
Hall sensor
Steering assistant
Other improvements

=

Grade

Android Studio

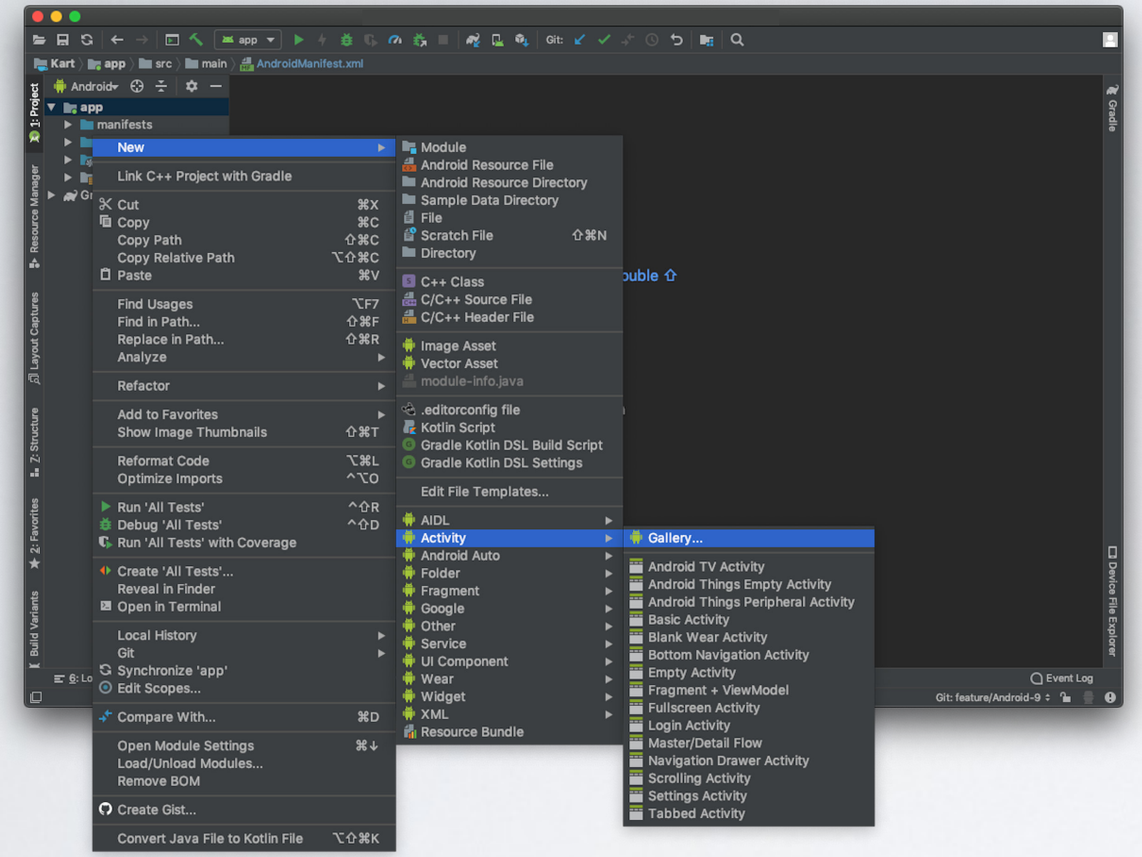
1. **Download** the Kart project template from the wiki
2. **Extract** the archive to the **local** disk
3. **Open** the extracted folder in Android Studio



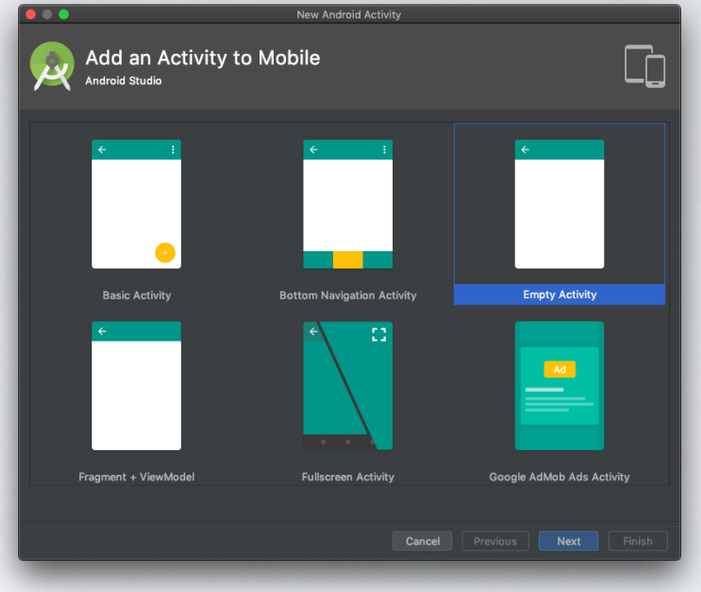
Android Studio

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

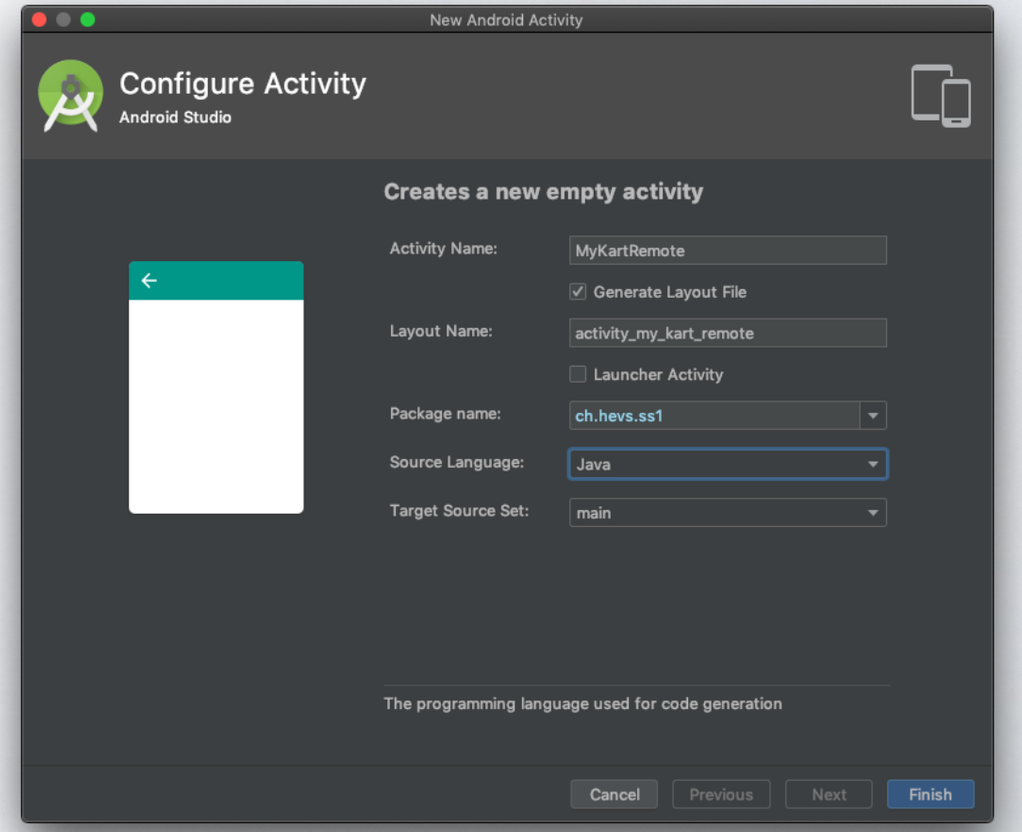
4 Select **New > Activity > Gallery...**



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>
```

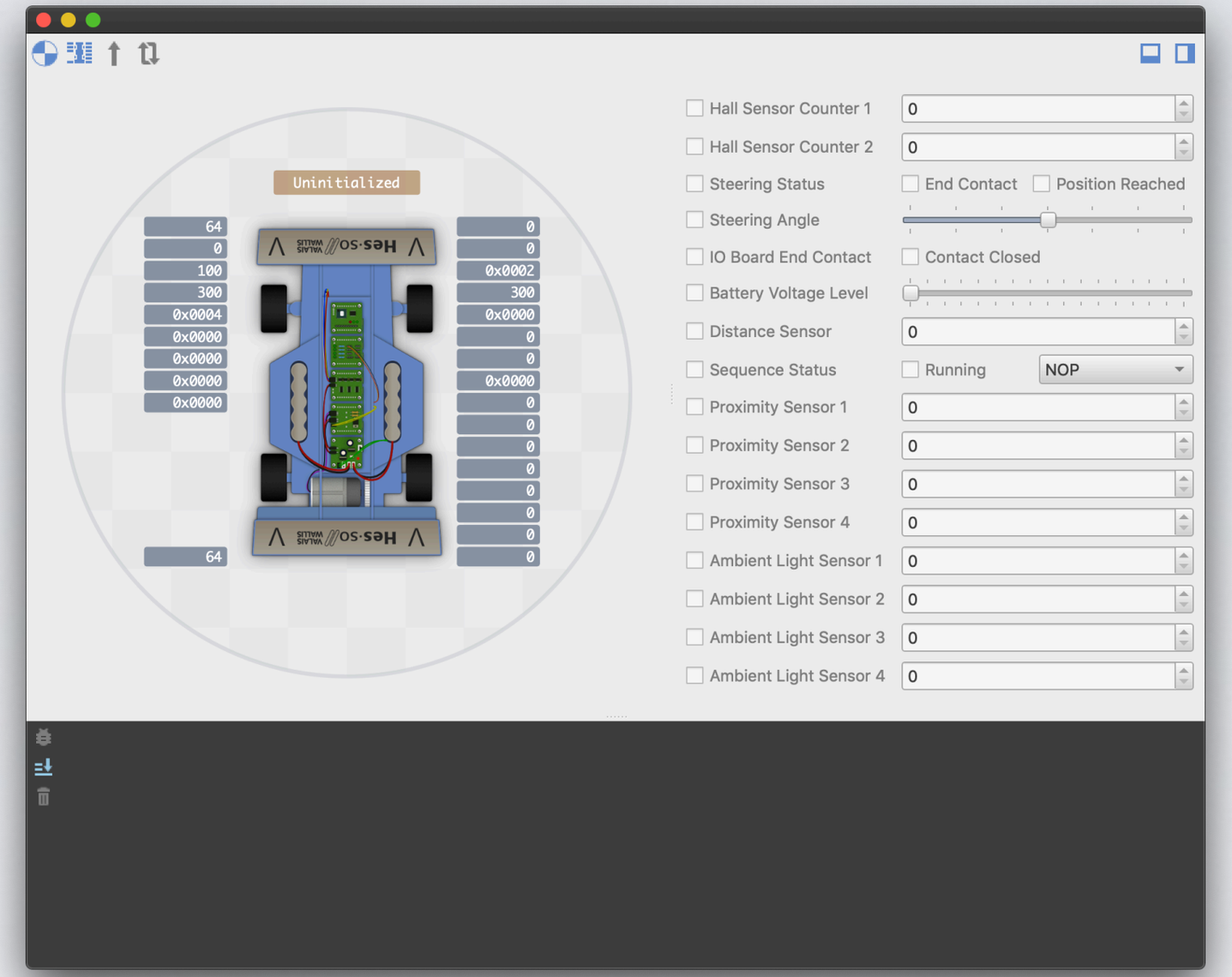
honor 10 Lite

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

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
  - 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

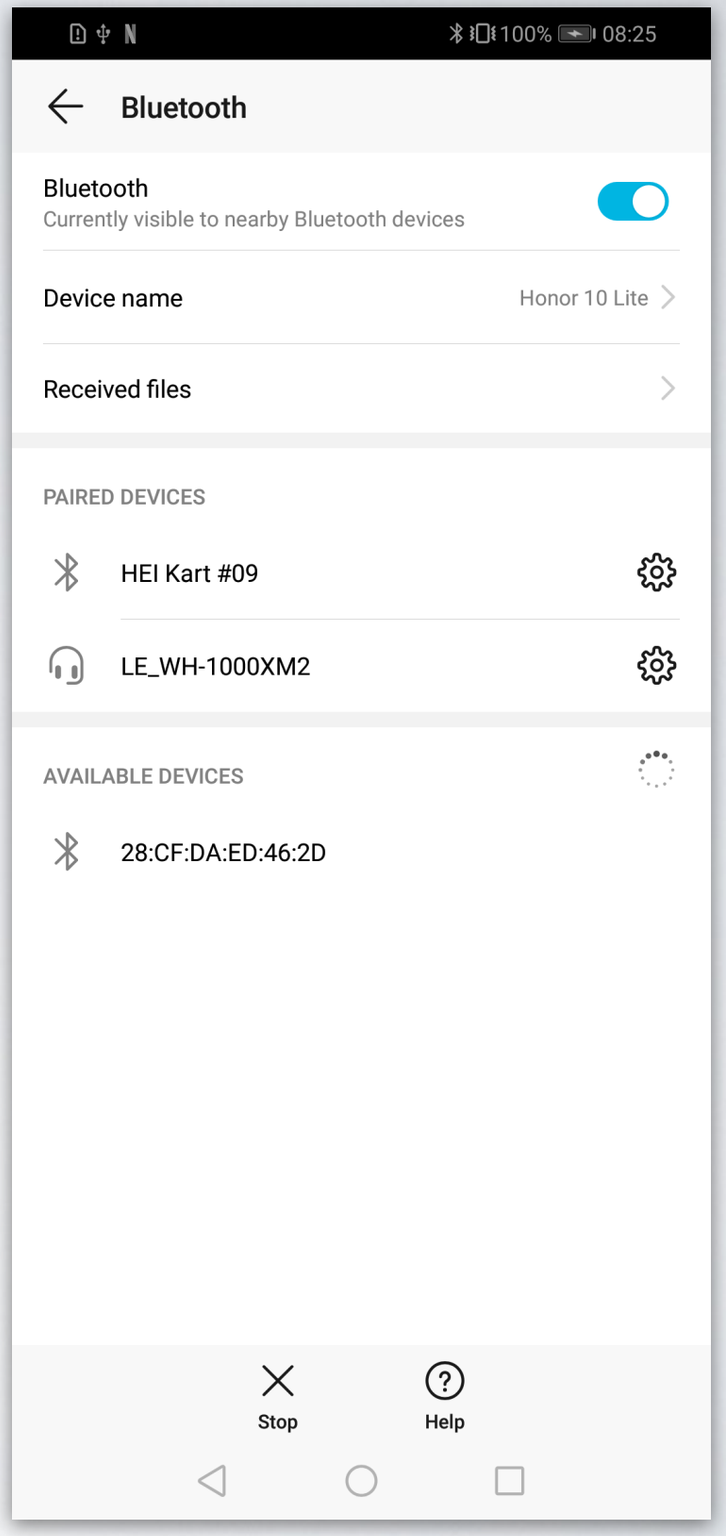
Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

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

Open "Settings"

Go to "Device connectivity" > "Bluetooth"

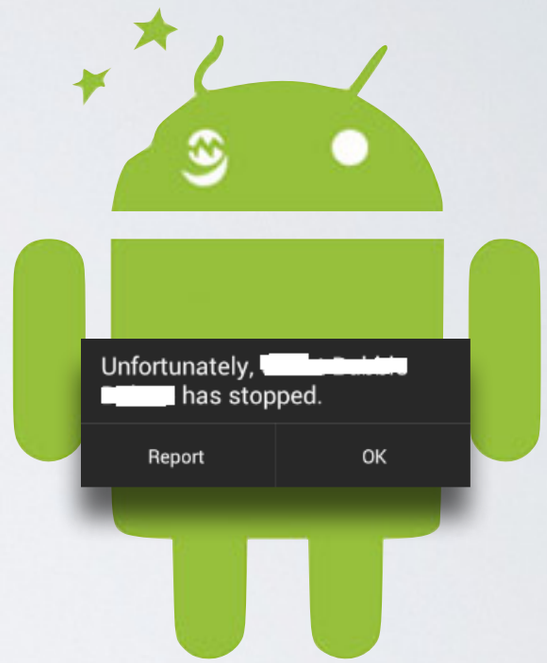
Select your kart in "Available Devices"



Tipp #3: No infinite loops


```
// 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

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


```
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