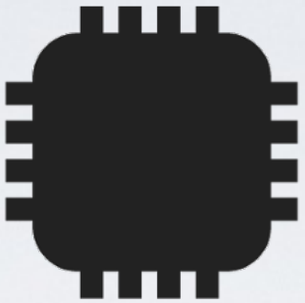
Hes·so VALAIS WALLIS

Kart Programming

Michael Clausen

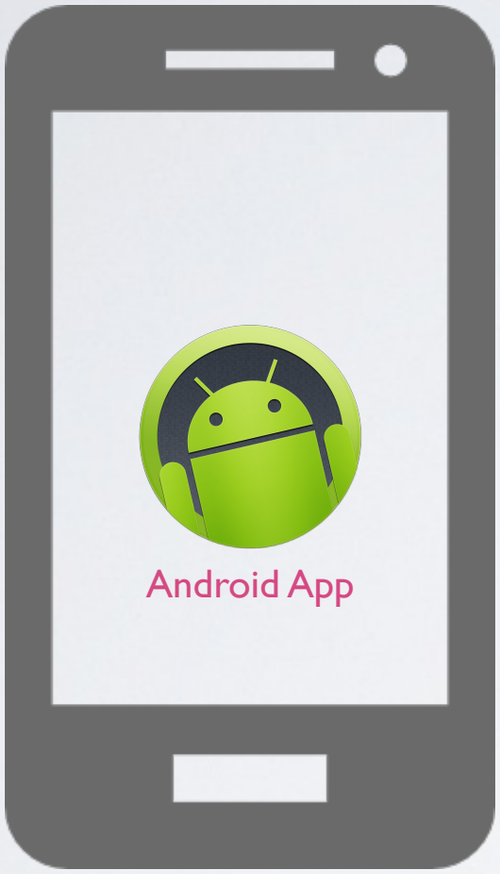
Goals



Use basic elements seen in ELN course.

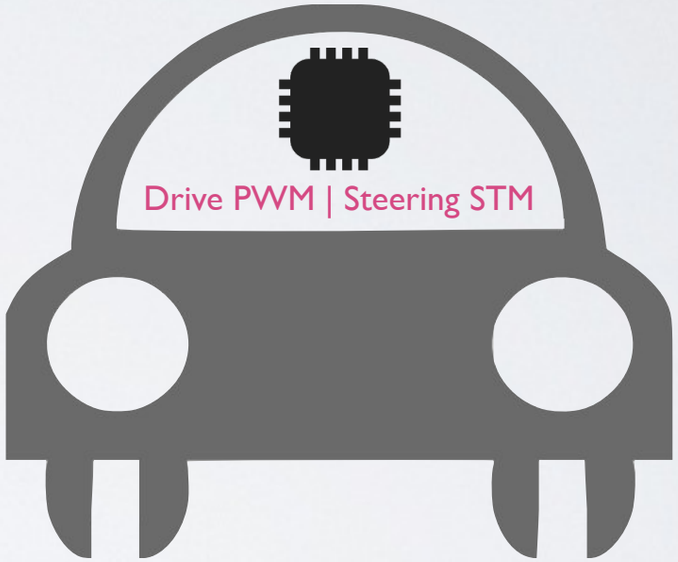
Use basic elements seen in INF I course.

Goals



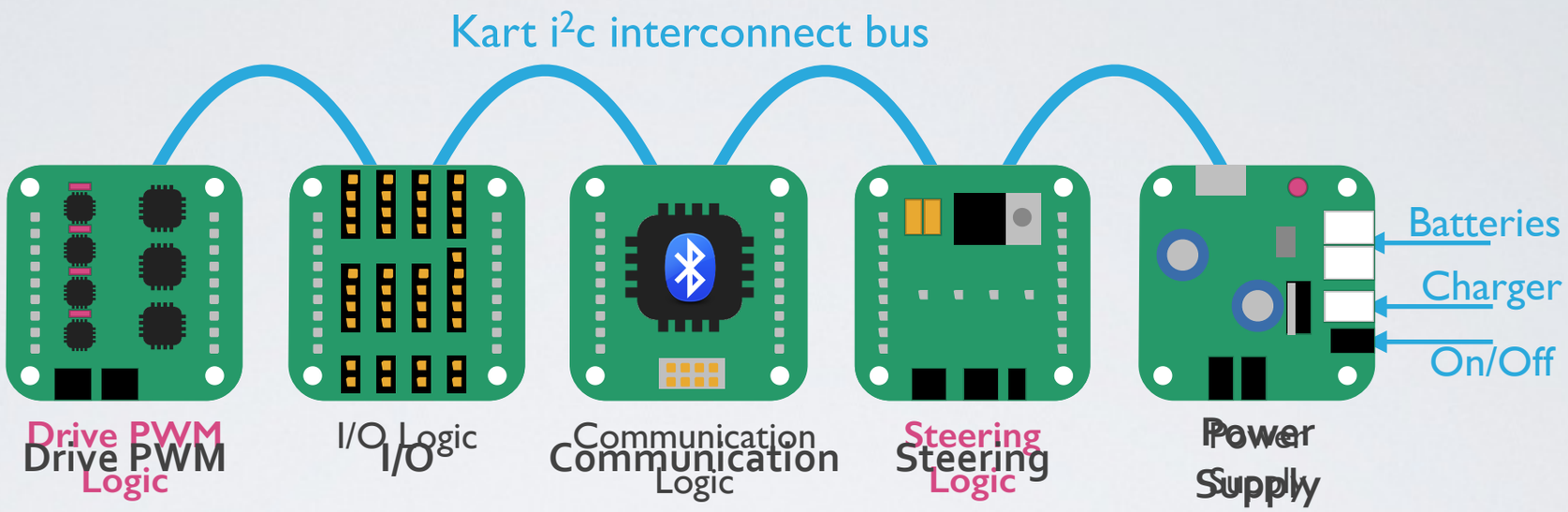
Galaxy Nexus

Bluetooth



Kart

Modular concept



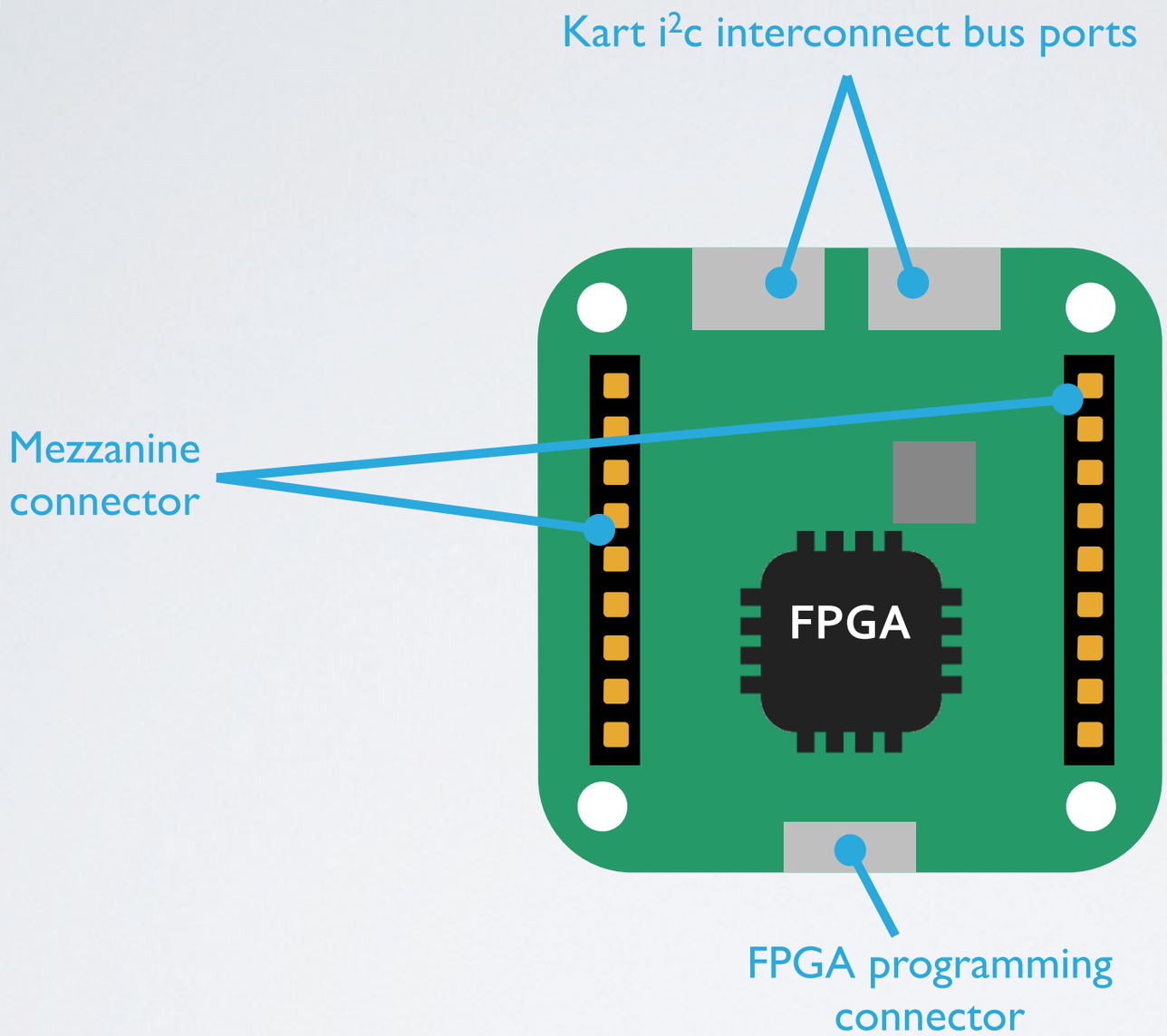
Drive PWM
Analog Frontend

IO Breakout
Board

Bluetooth
Communication

Steering STM
Analog Frontend

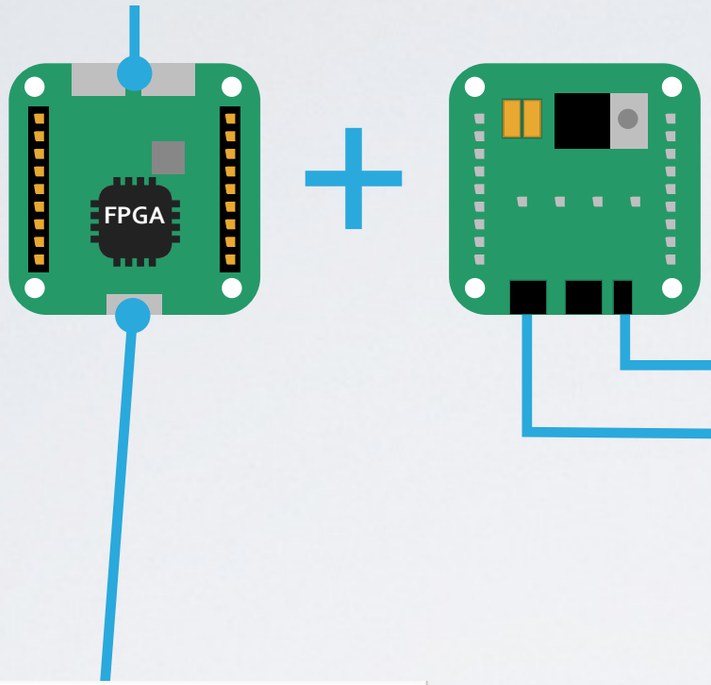
Generic FPGA Board



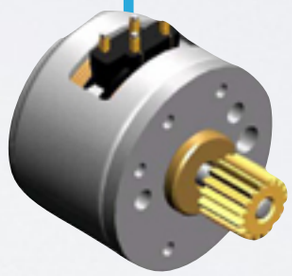
Stepper Motor

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

Kart i²c interconnect bus



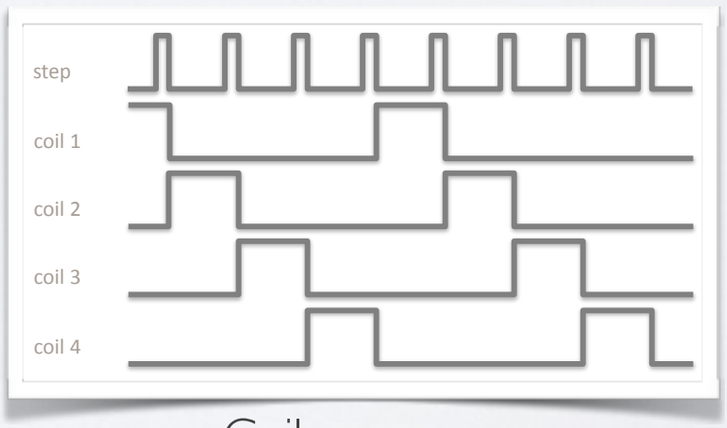
Limit switch



Stepper 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
```

Stepper motor control

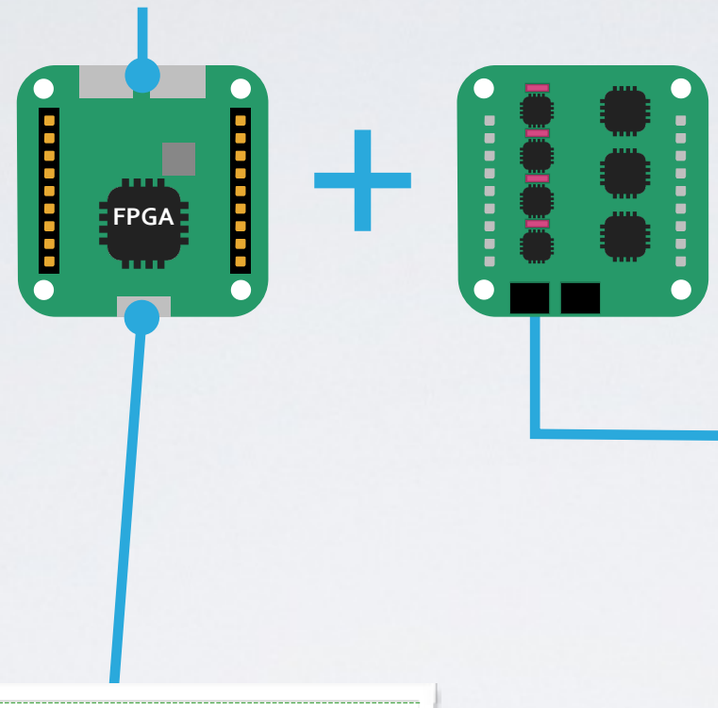


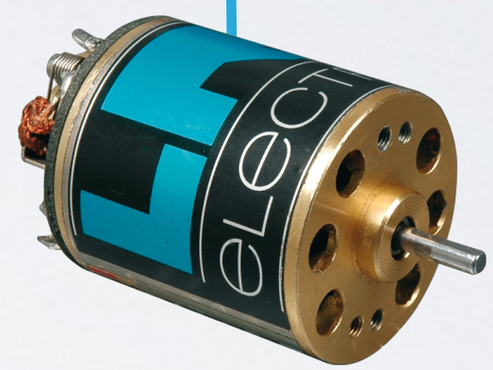
Coil sequence

Drive Motor

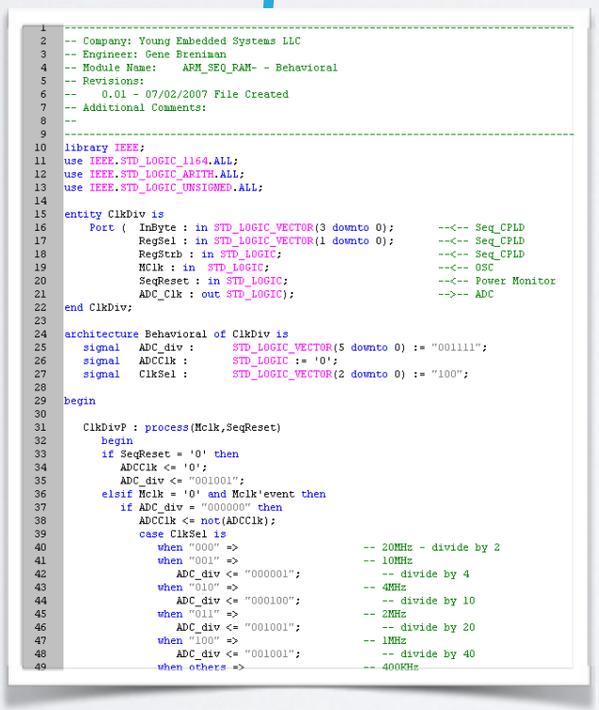
Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

Kart i²c interconnect bus





Drive motor



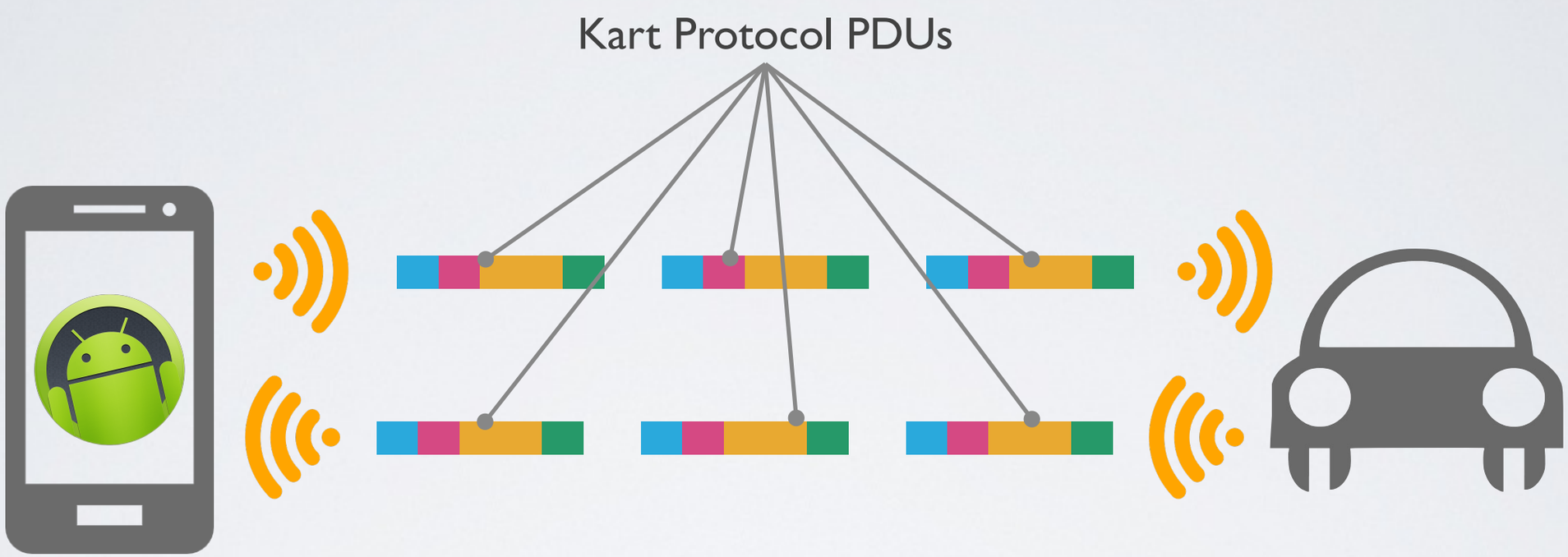
```
--------------------------------------------------------------------------------
-- Company: Young Embedded Systems LLC
-- Engineer: Gene Breniman
-- Module Name:    ARM_SEQ_RAM- - Behavioral
-- Revisions:
--    0.01 - 07/02/2007 File Created
-- Additional Comments:
--
--------------------------------------------------------------------------------
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

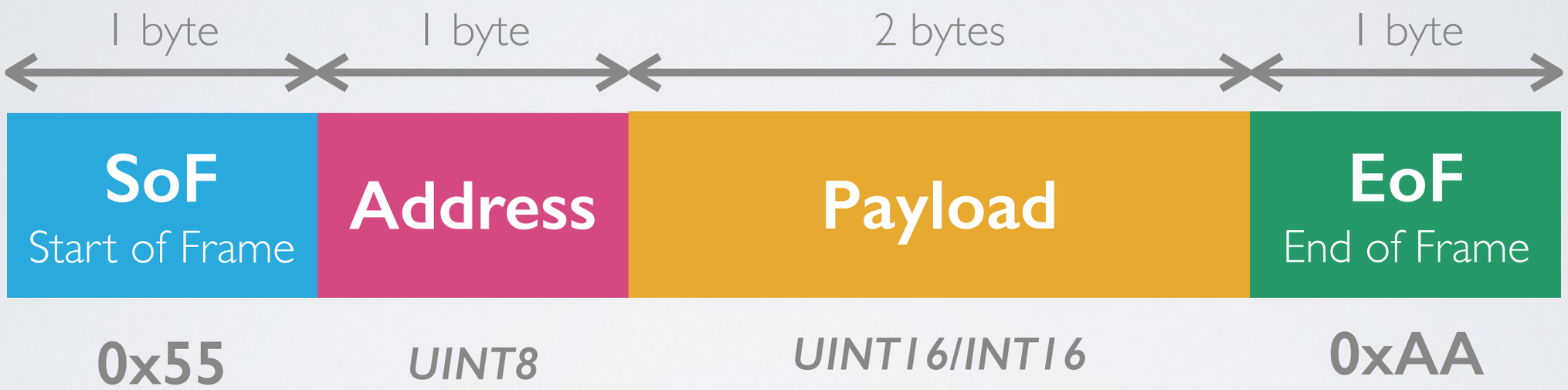
Hardware Goals

- **Control block for DC motor**
  - Pulse Width Modulation (PWM) generator
- **Control block for stepper motor**
  - 4 Coil forward/backward sequence generator
- **Speed sensor**
- **Anti-collision emergency stop**
  - IR distance sensors

Remote Control Protocol



Remote Control Protocol PDU



Remote Control Protocol Addresses

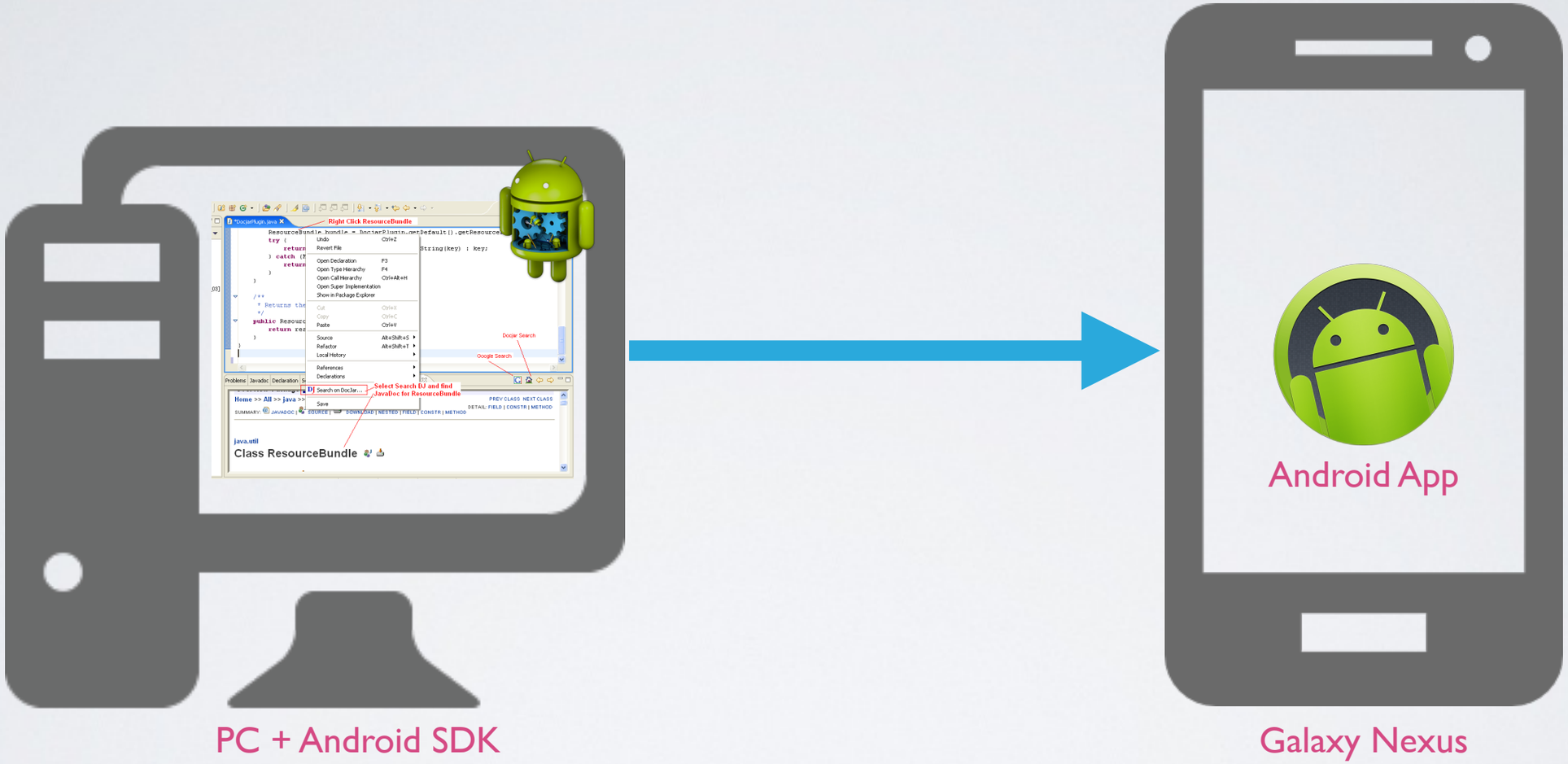
Address	Type	Description
0x00	UINT16	Drive motor PWM period.
0x01	INT5	Drive motor speed [-15..15] (negative=backwards).
0x02	UINT16	Steering motor step period (speed proportional to 1/period).
0x03	UINT16	Steering position set point.
0x04	UINT5	Steering end switch address.
0x05	UINT5	Hardware settings (Enumeration mask).
0x06	UINT4	LED control.
0x15	UINT16	Update interval (from kart to phone) in ms.

Remote Control Protocol Addresses

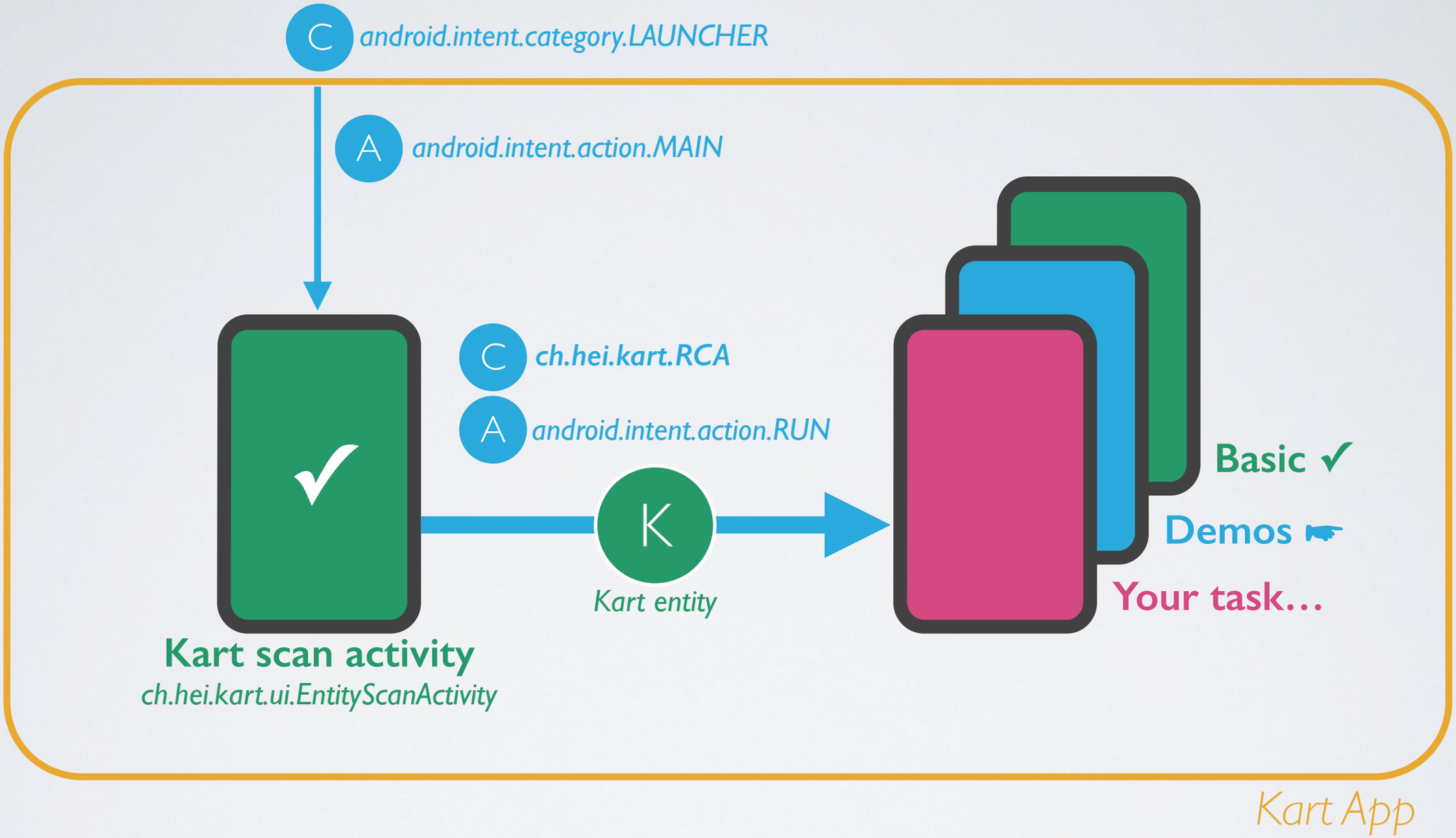
Address	Type	Description
0x00	UINT16	Hall sensor 1 speed count.
0x01	UINT16	Hall sensor 2 speed count.
0x02	UINT1	Steering angle reached (1=reached, 0=busy).
0x03	UINT16	Actual steering position.
0x04	UINT1	Steering end contact state (0=contact closed).
0x05	UINT16	ADC value of battery voltage level.
0x06	UINT16	Distance (Ultrasonic sensor)
0x08	UINT16	Proximity 1 (IR sensor)
0x09	UINT16	Proximity 2 (IR sensor)
0x0A	UINT16	Proximity 3 (IR sensor)
0x0B	UINT16	Proximity 4 (IR sensor)
0x0C	UINT16	Ambient Light 1 (IR sensor)
0x0D	UINT16	Ambient Light 2 (IR sensor)
0x0E	UINT16	Ambient Light 3 (IR sensor)
0c0F	UINT16	Ambient Light 4 (IR sensor)

Remote Control Android App

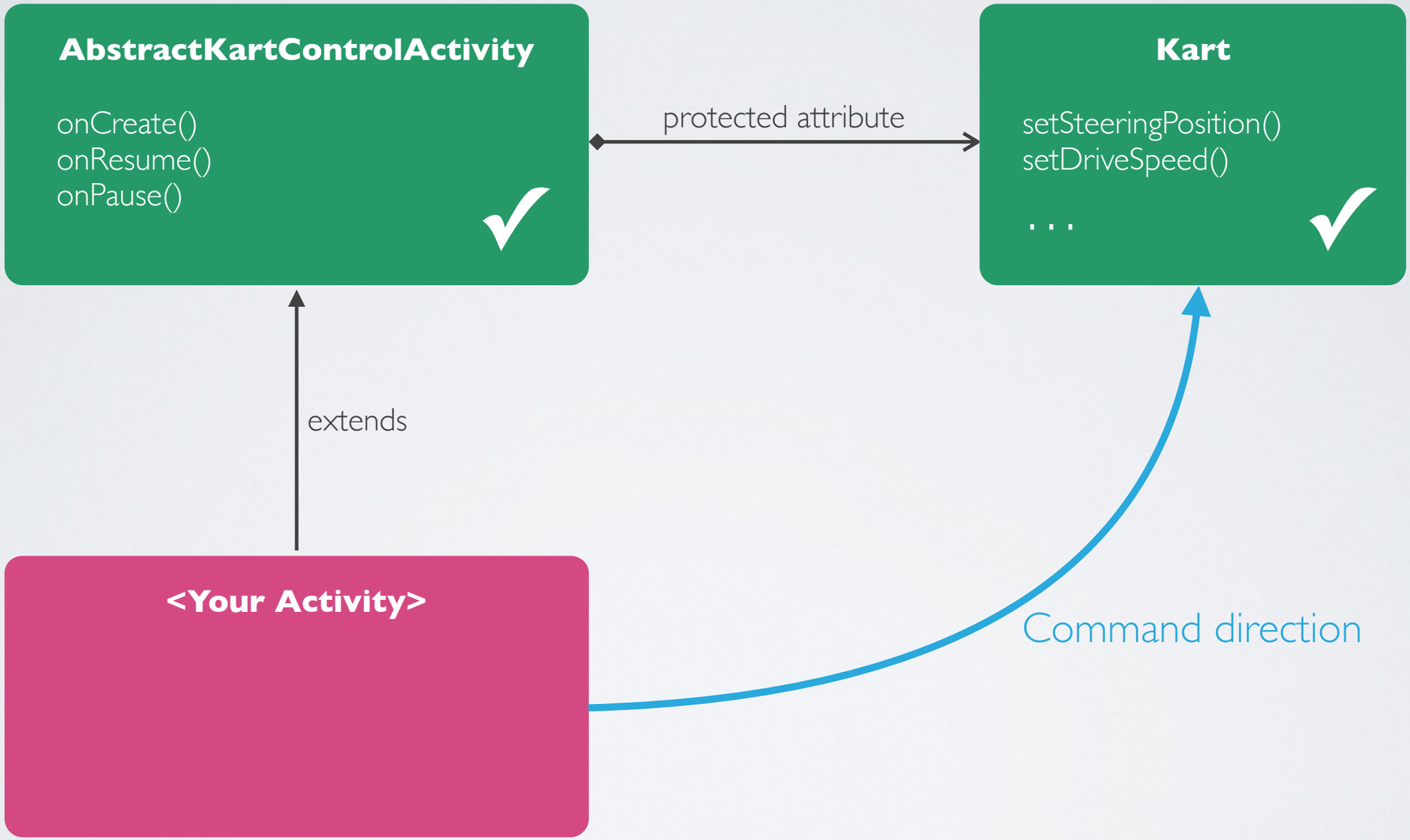
Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften



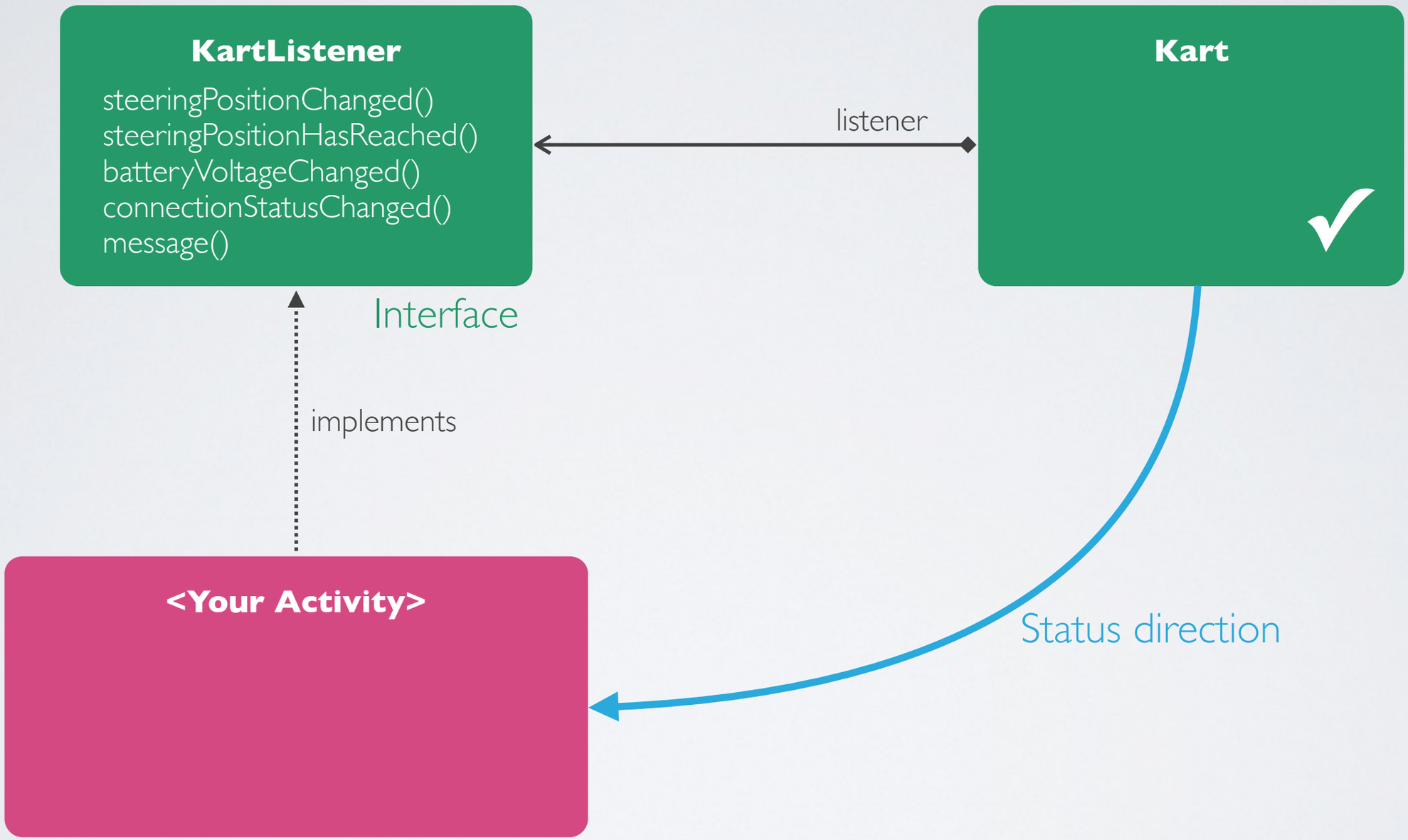
Remote Control Android App



Remote Control Android App



Remote Control Android App



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