5LIA0 – Embedded Visual Control

What is this course about?

- Embedded
  - Implementing your algorithms on embedded compute platforms.
- Visual
  - Using computer vision to perform a given task.
- Control
  - Controlling the actuators of your robot.

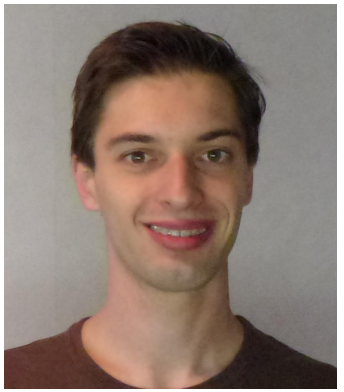
Who are involved?

Henk Corporaal
(Embedded)

Egor Bondarev
(Visual)

Dip Goswami
(Control)

Luc Waeijen

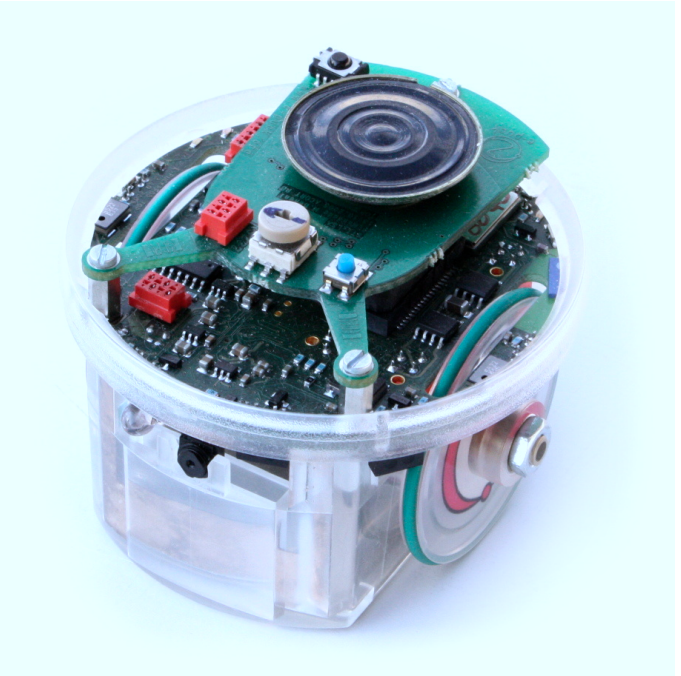


Mark Wijtvliet

Roel Jordans

Previous years (2010-2011)

- Non-holonomic robots



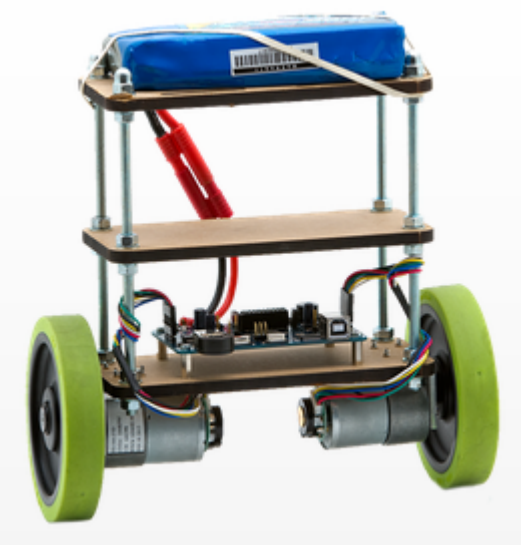
- Robot arm

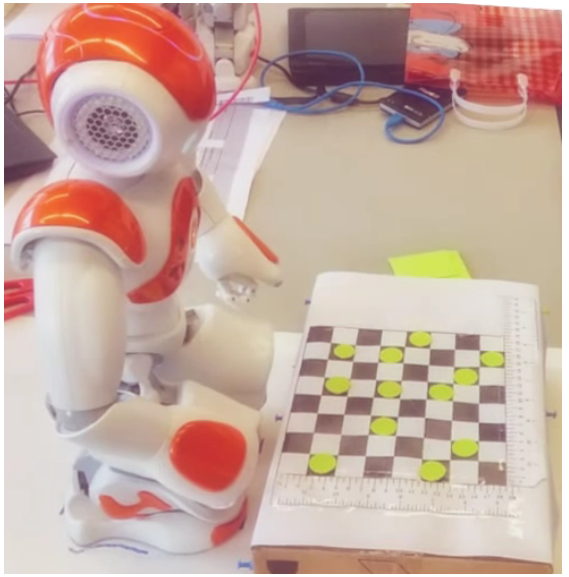
Previous years (2012-2015)

- Quadcopters

Previous years (2015)

- Quadcopters
- Sail boats
- Solar cars
- Balancing robots
- Checkers playing robot





Previous years

- 2014: 19 students
- 2015: 32 students
- 2016: 77 students*

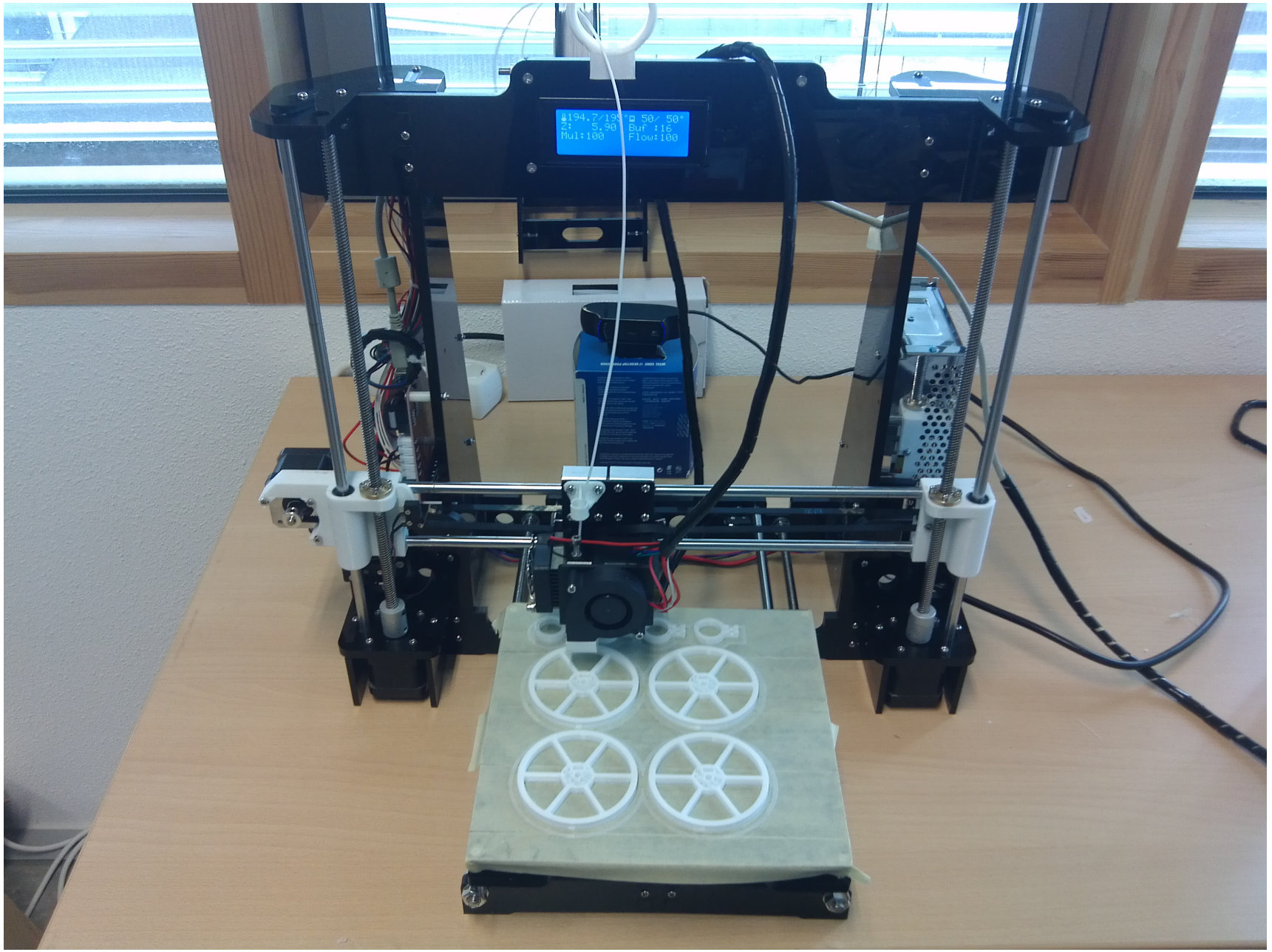
- Previous years: 1 semester, this year 1 quartile.

* Registered in OASE

This year

- All groups will build the same thing, but you can extend it with your own ideas.
- We pre-ordered most parts.
- For your own extensions we have...

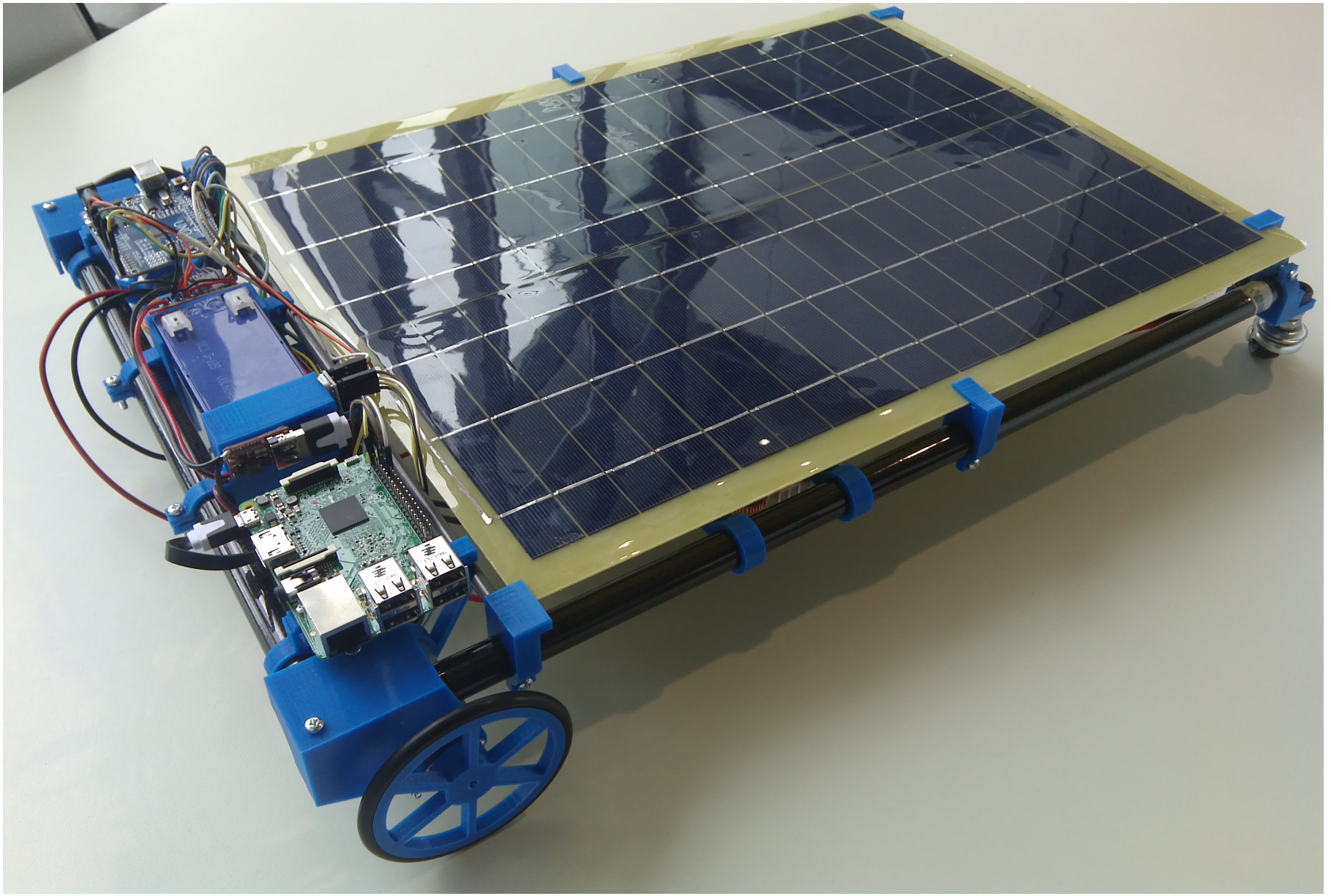
This year



This year

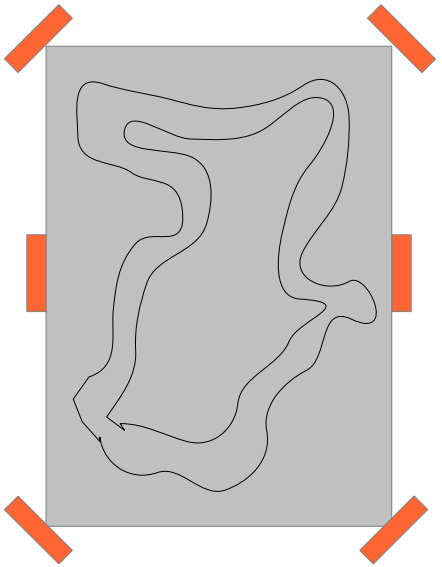
- So, what are we going to build?

This year



This year

- Two part assignment:
  - Follow a route based on visual markers:
    - Detect the sides of the 'road'
      - will probably be marked with 2-lines of black tape on a light gray surface
    - 6 Global markers can be used to estimate your position
    - On your route you will find traffic signs indicating what to do.
      - Turn left/right at a junction or fork.
      - Stop for 5 seconds.
      - Turn 180 degrees.



This year

- Each group defines their own plan. This could be (but not limited to):
  - Add a gripper to your robot and move an object to a certain destination on the route.
  - Add collision avoidance (avoid moving objects on the route).
  - Add pan/tilt to your camera so you can look around.
  - Try to avoid shadows.
  - Implement platooning (forming a formation of solar cars).
  - …

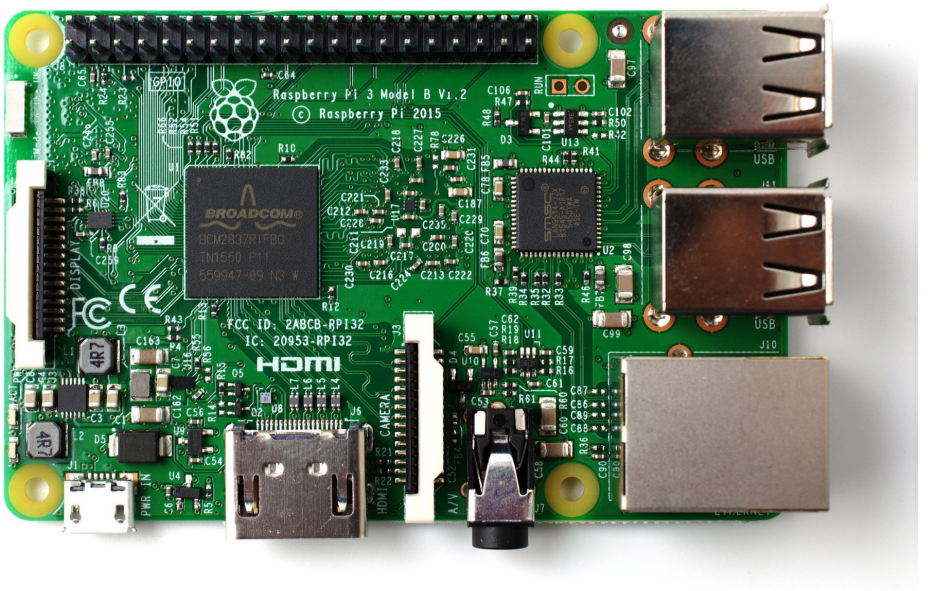
- Any ideas already?

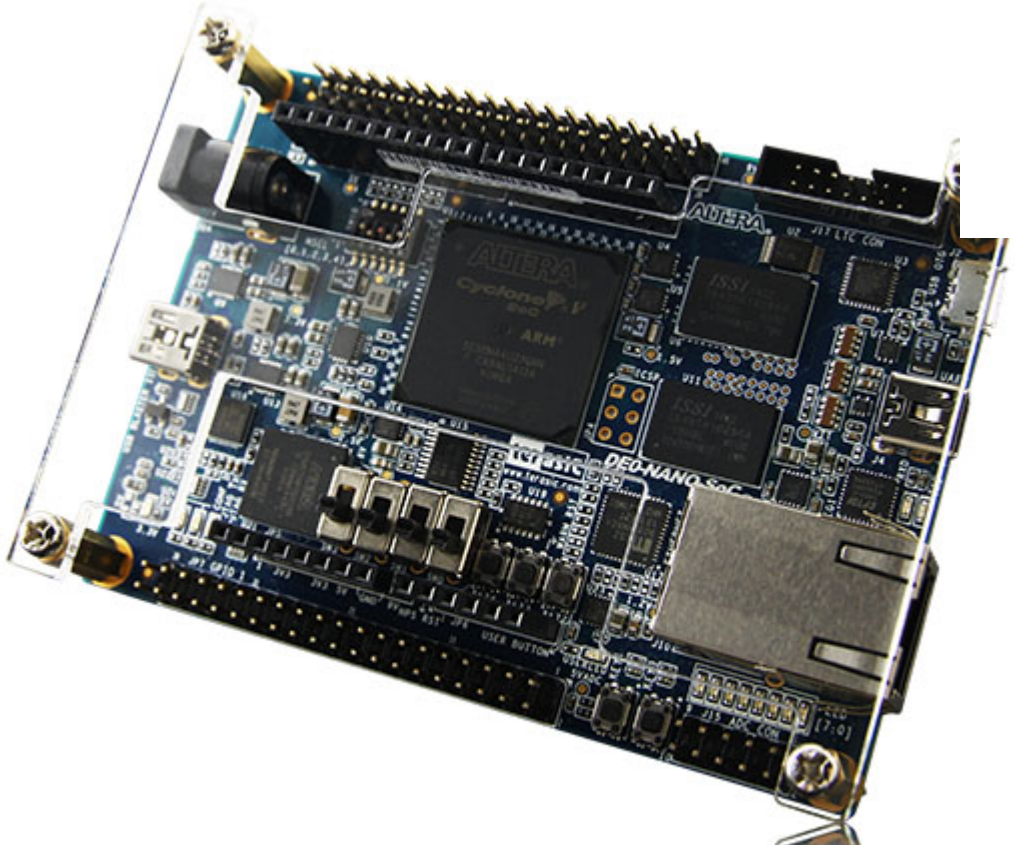
This year

- Competition
  - The solar car includes two current sensors:
    - One that measures charge current from the solar panel.
    - One that measures current of the embedded system (Raspberry PI, Arduino, Motors, …).
  - The group with the lowest ***energy*** consumption wins.
    - while still completing the course before a time limit.

Solar car

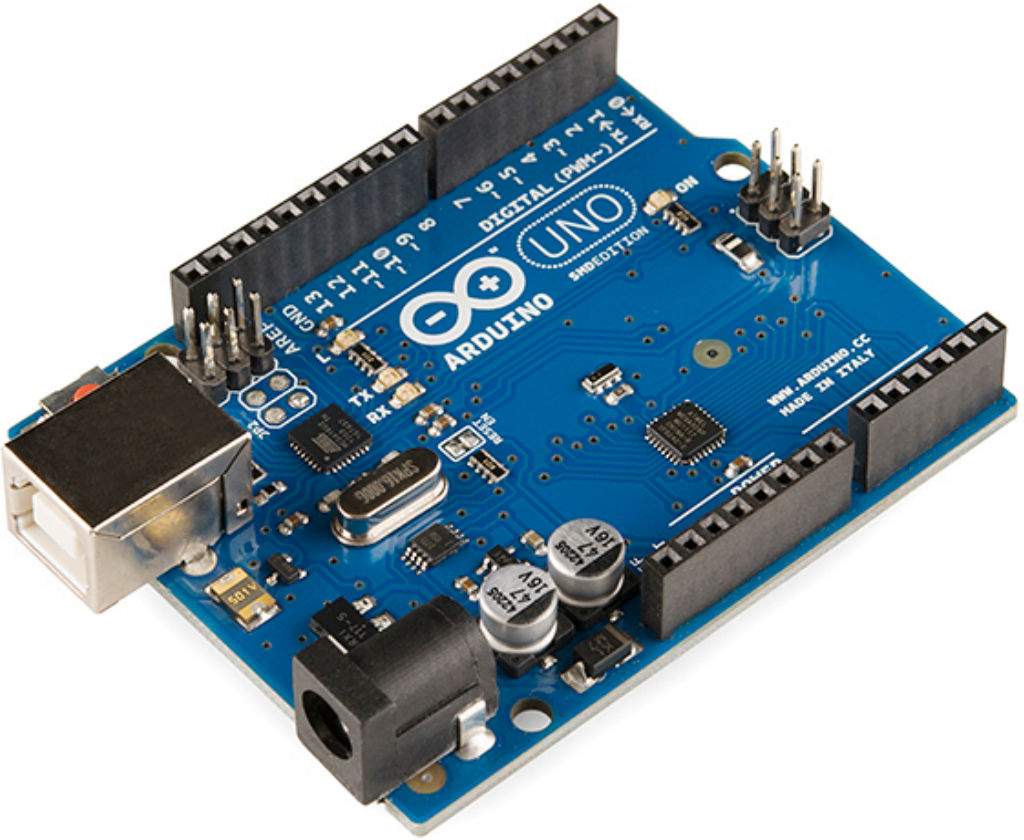
- Embedded & Visual
  - Raspberry Pi 3
  - DE0-Nano-SoC





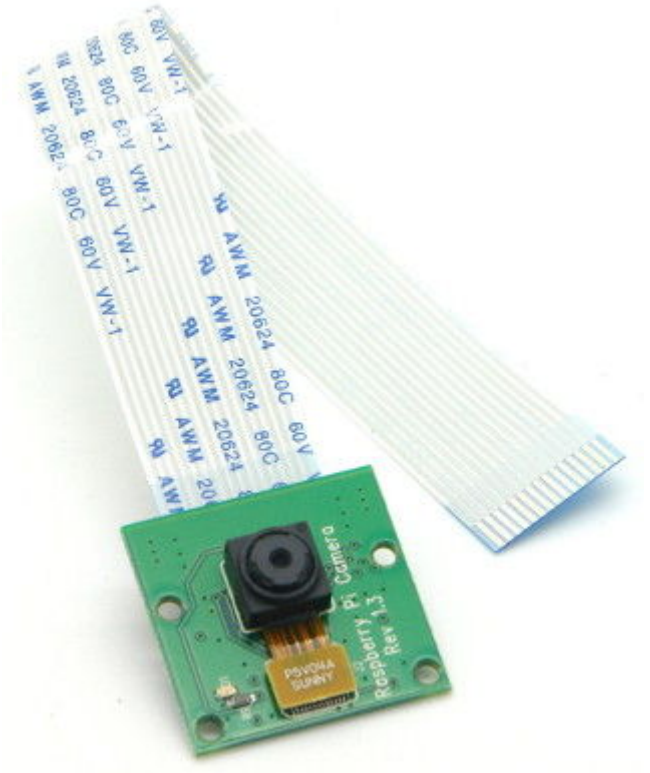
Solar car

- Embedded & Control
  - Arduino UNO



Solar car

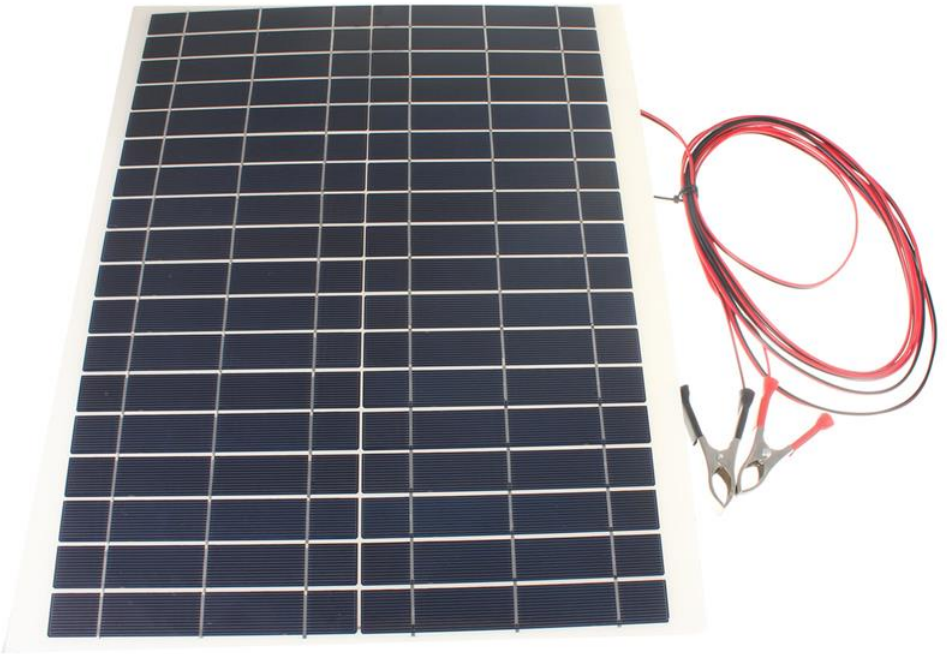
- Visual
  - OpenCV computer vision library
  - Works directly on the Raspberry Pi
  - Requires algorithm porting and camera interface construction on the FPGA.
- Raspberry Pi camera



Solar car

- Motors
  - Encoder

- Solar panel
  - 20 Watt
  - 18 Volt

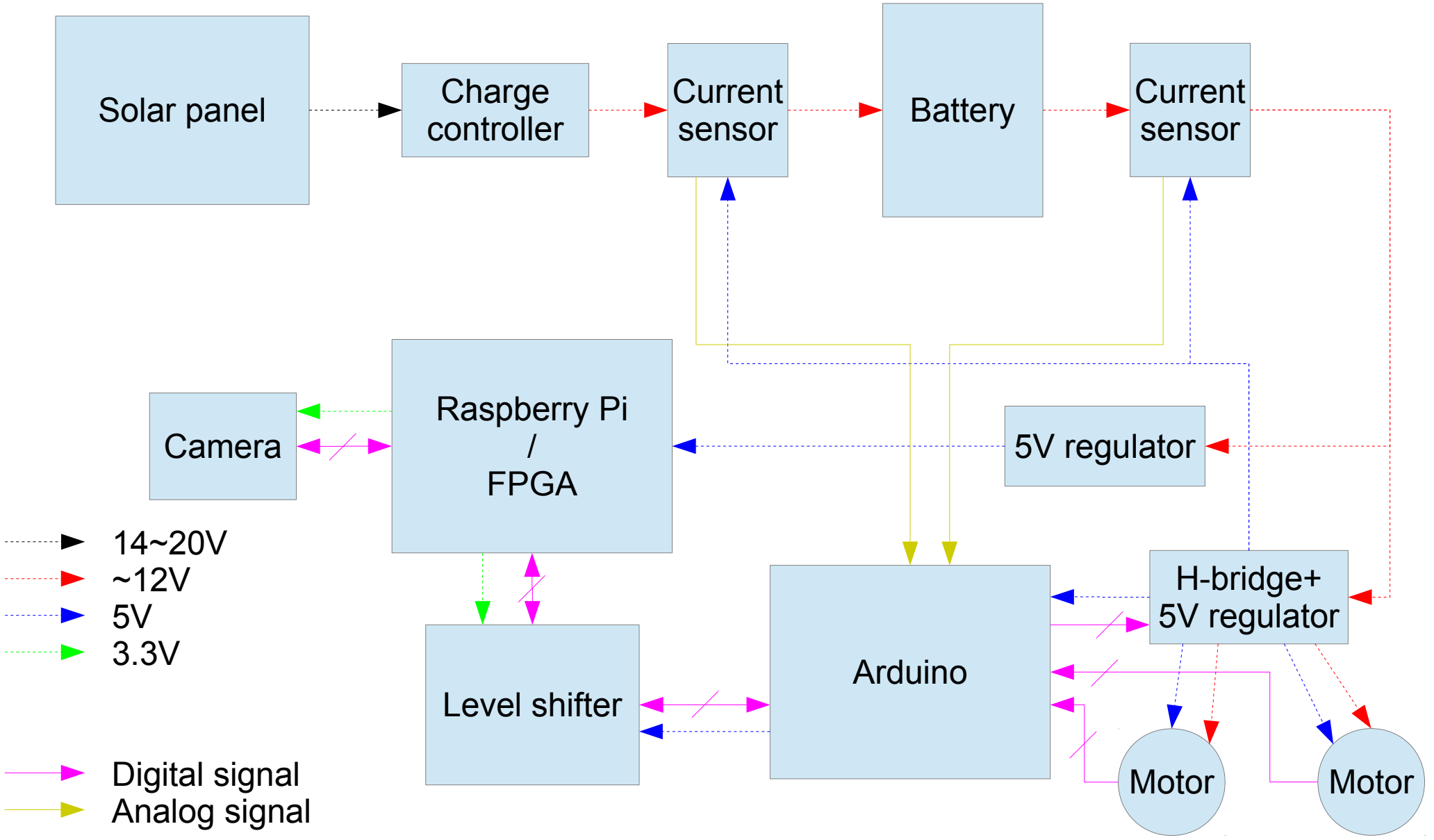


Solar car

- Battery

- Current sensors

Solar car – system overview



Groups

- If you choose the Raspberry Pi
  - 3 person group
- If you choose the FPGA board
  - 5 persons group
- Absolute maximum number of groups: 20

Groups

- Please add your group to this spreadsheet (shared, google docs)
- https://goo.gl/LMjjHG

Course organization

- 2x2 lecture hours per week.
  - Each day 10 groups will present will present their progress, you will get 3 minutes per group.
    - We might ask you some questions.
    - The remaining time will be used for lectures.
  - During the break between these lecture hours you can ask questions about the assignment.
  - 5lia0.cgra.nl contains a forum where you can ask questions (and answer them).

Course organization

- 3D printer(s)
  - The 3D printers are in a lab with limited access.
  - We need 2 or 3 volunteers to manage printing parts. You will get access to the lab.

What can you already do?

- Let us know if you use the Raspberry Pi or FPGA, we will order them as soon as possible.
- Raspberry Pi and FPGA
  - Put the linux image on the SD card
  - Install opencv
  - Try some of the openCV tutorials
- FPGA
  - Install Quartus II and do some Verilog experiments, you can already start with the FPGA ↔ CPU communication.
- Arduino
  - Start implementing the communication protocol.
  - Start implementing a initial version of the motor control.
    - Mandatory use of the low-level motor driver!