Lego Car

Midterm Presentation

Moritz Dötterl, Berkay Sümer, Abdallah Emad, Heiko Lengenfelder

30.5.2016

Agenda

- Wiring
- Nano Board
- Raspberry Pi
- OpenCV

Wiring
overview

Components:

Wiring
overview

Components:

- Sensors
  - Ultrasonic
  - Camera

Wiring
overview

Components:

- Sensors
  - Ultrasonic
  - Camera
- Actuators
  - Steering Servo
  - Drive Motor

Wiring
overview

Components:

- Sensors
  - Ultrasonic
  - Camera
- Actuators
  - Steering Servo
  - Drive Motor
- Controll
  - Nanoboard
  - Raspberry Pi

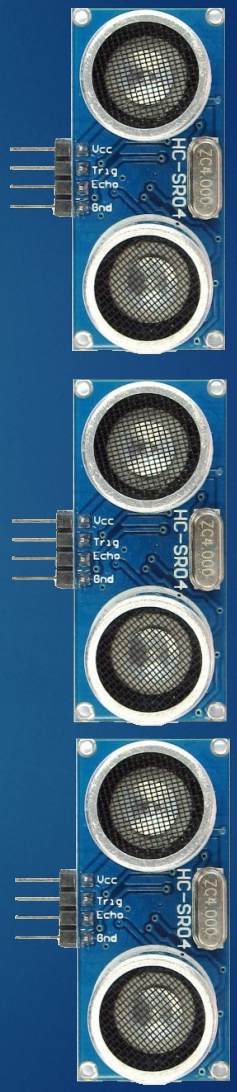
Wiring
overview

Components:

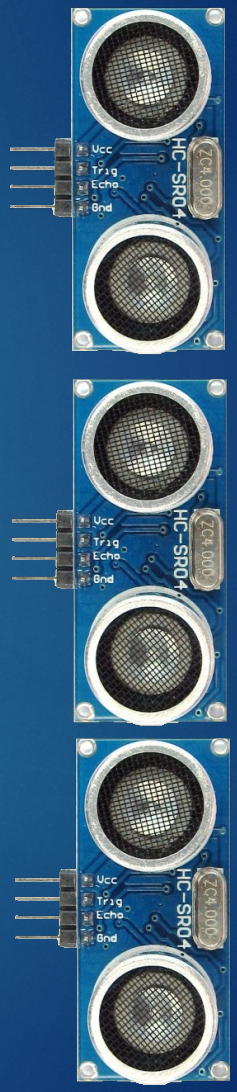
- Sensors
  - Ultrasonic
  - Camera
- Actuators
  - Steering Servo
  - Drive Motor
- Controll
  - Nanoboard
  - Raspberry Pi
- Power supply
  - Battery
  - H-Bridge

Wiring
circuit layout

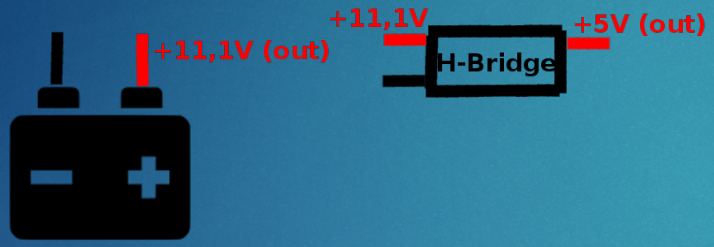
Wiring
circuit layout

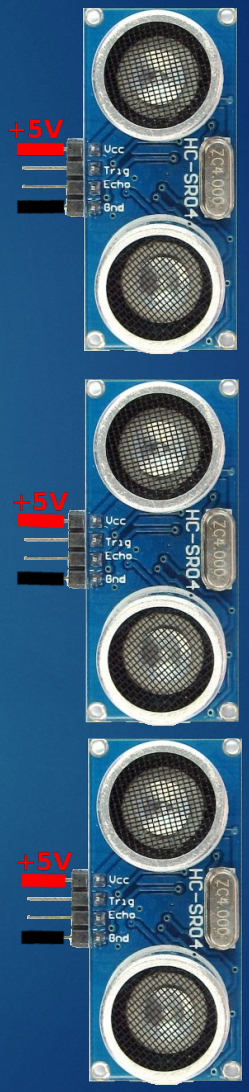


Wiring
circuit layout

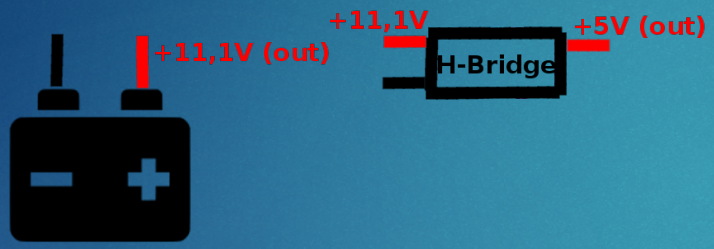


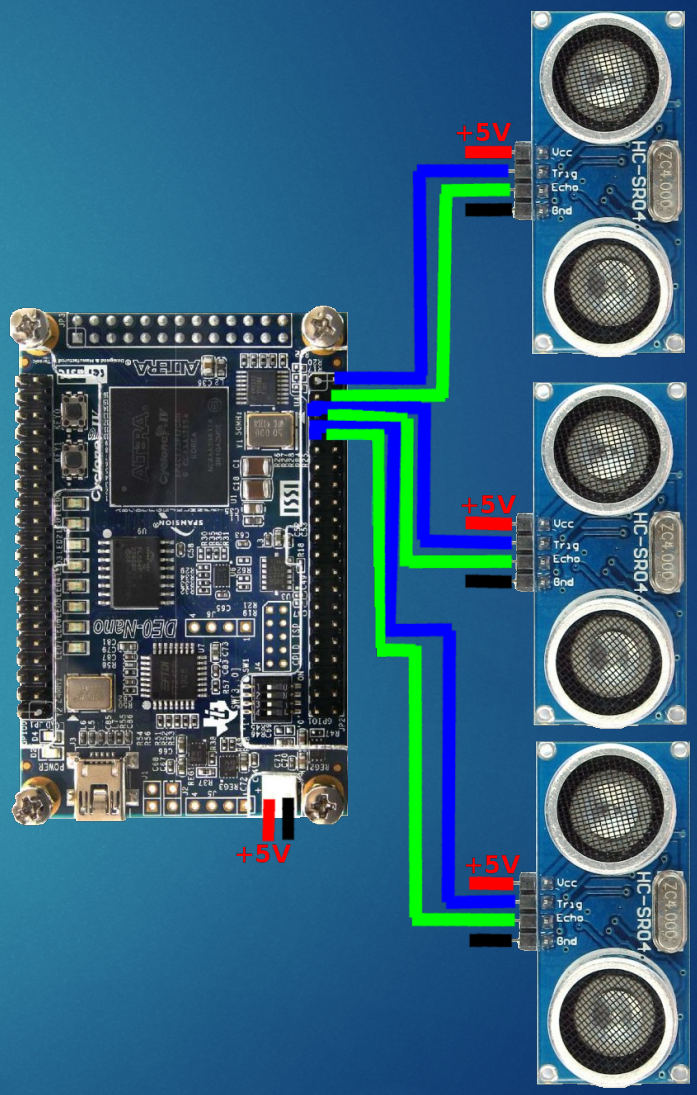
Wiring
circuit layout



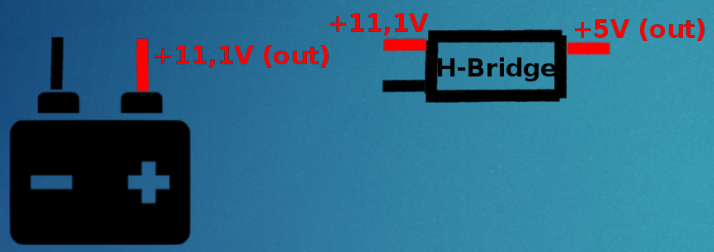


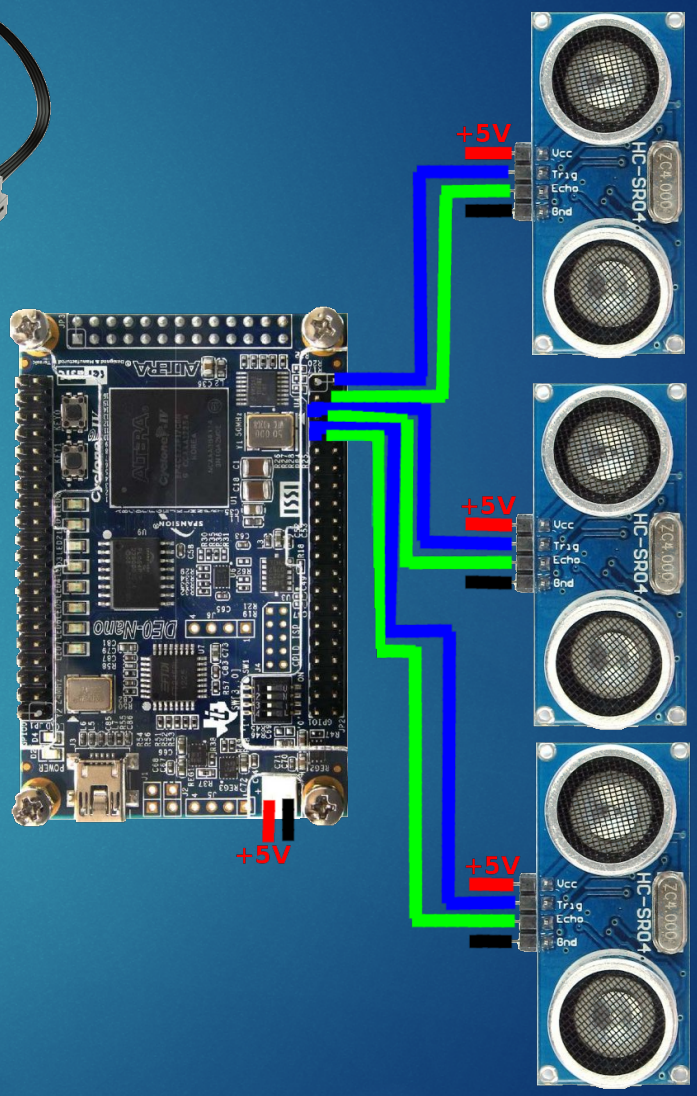
Wiring
circuit layout



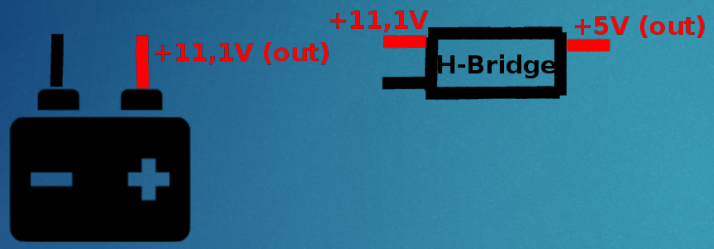


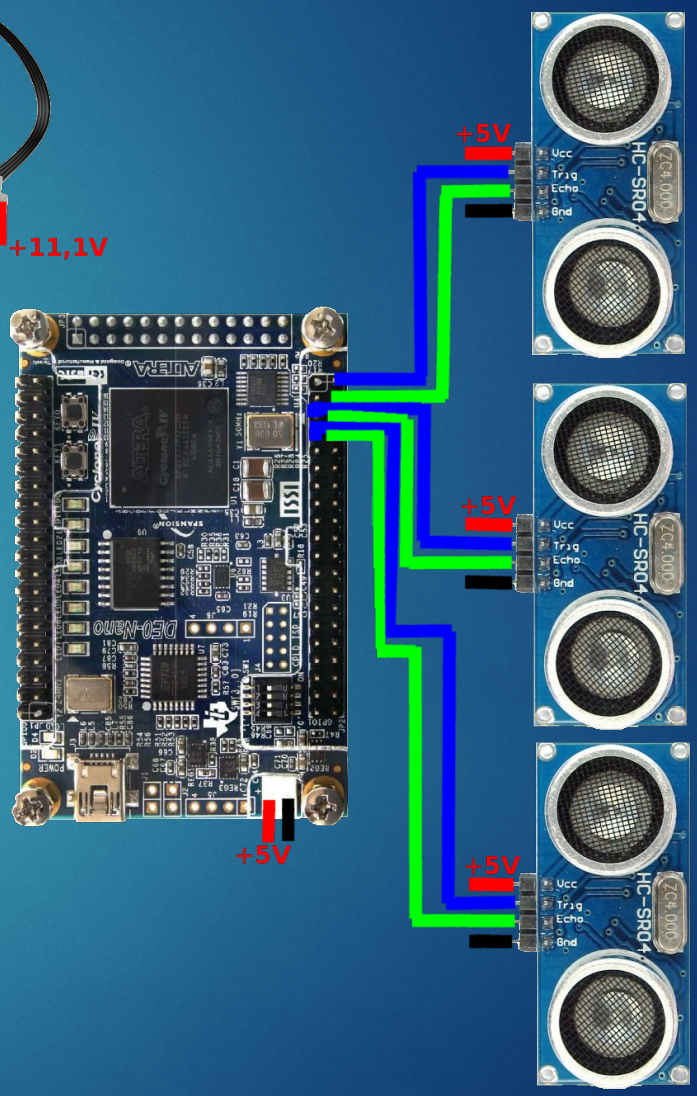
Wiring
circuit layout



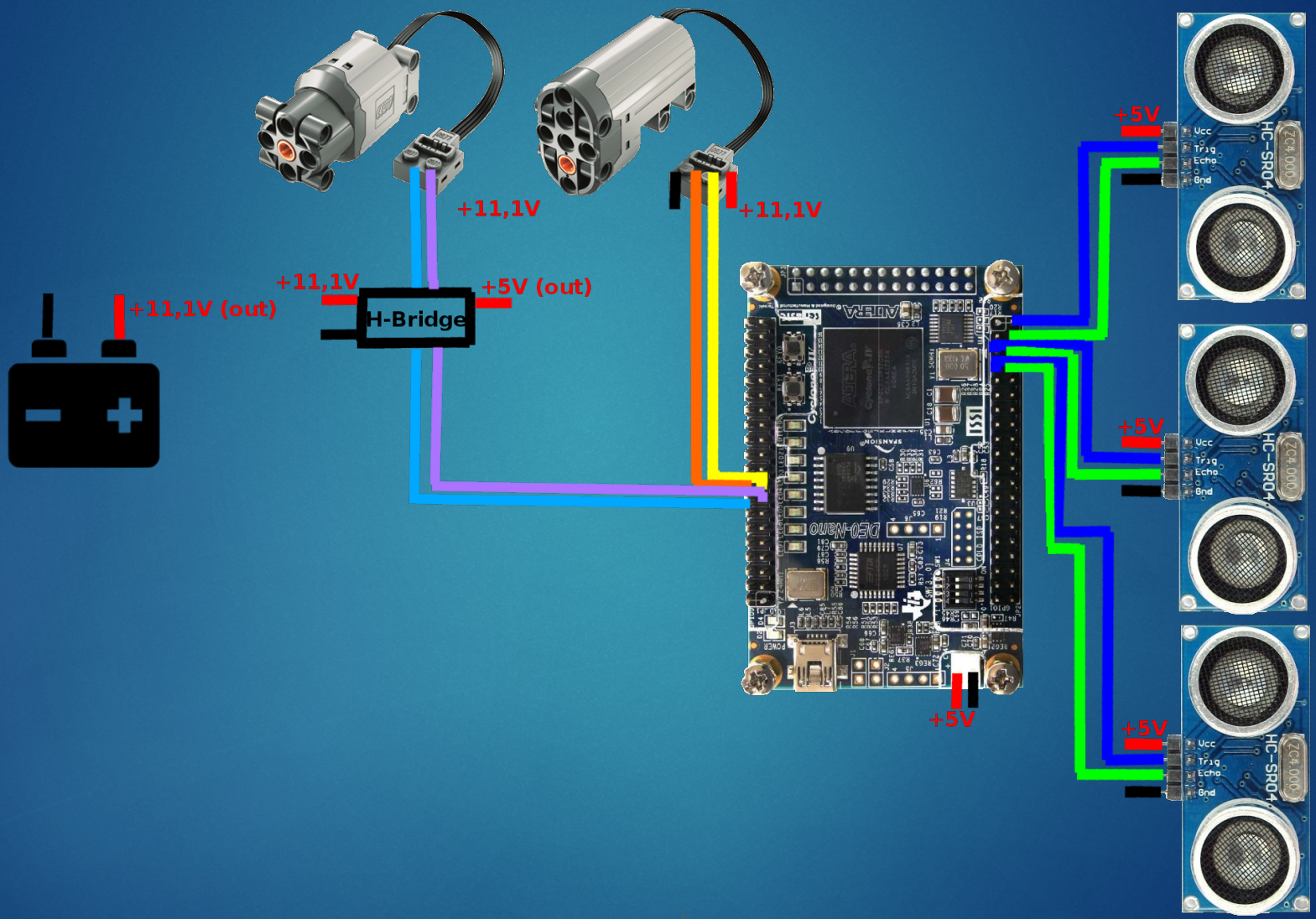


Wiring
circuit layout

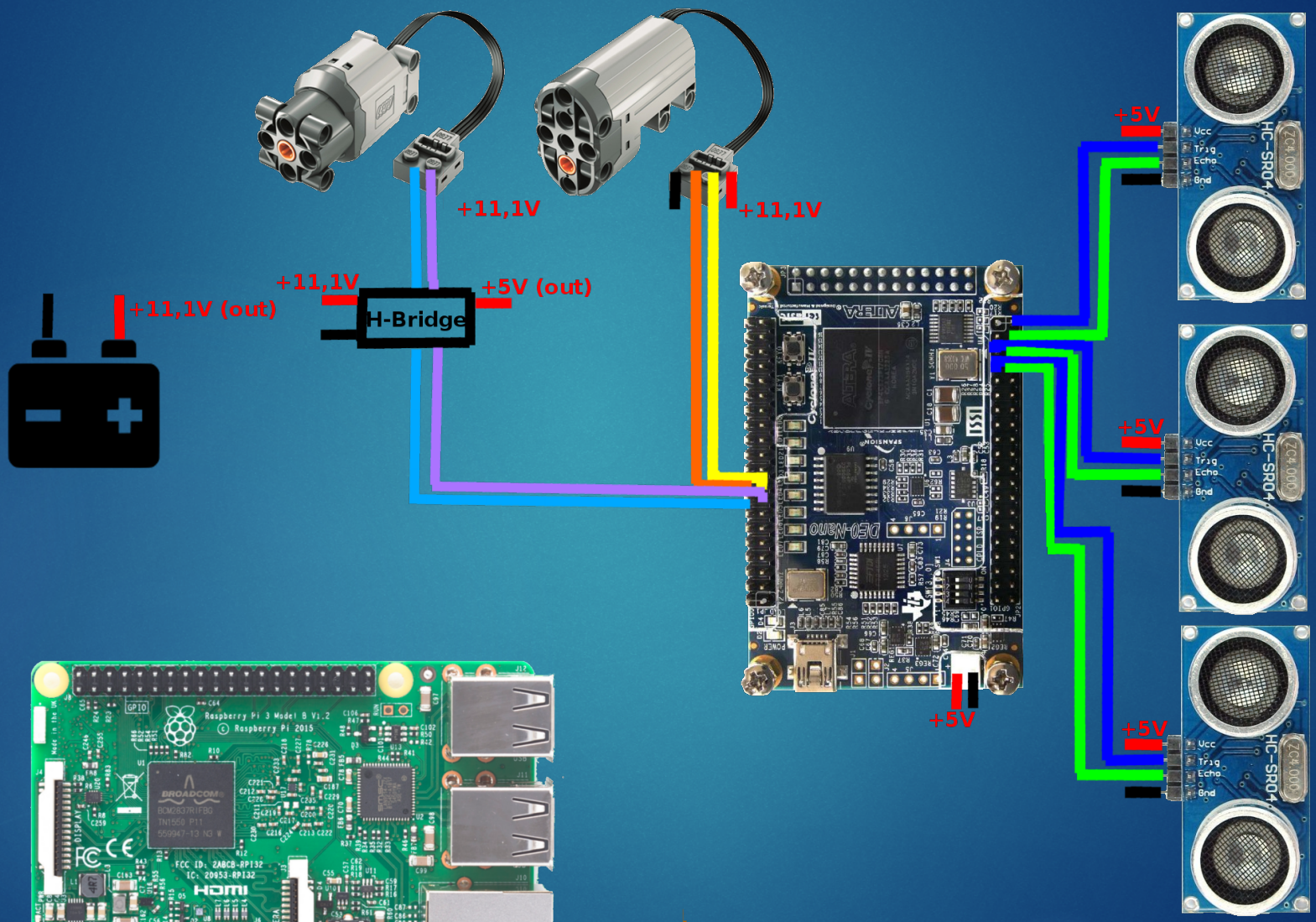




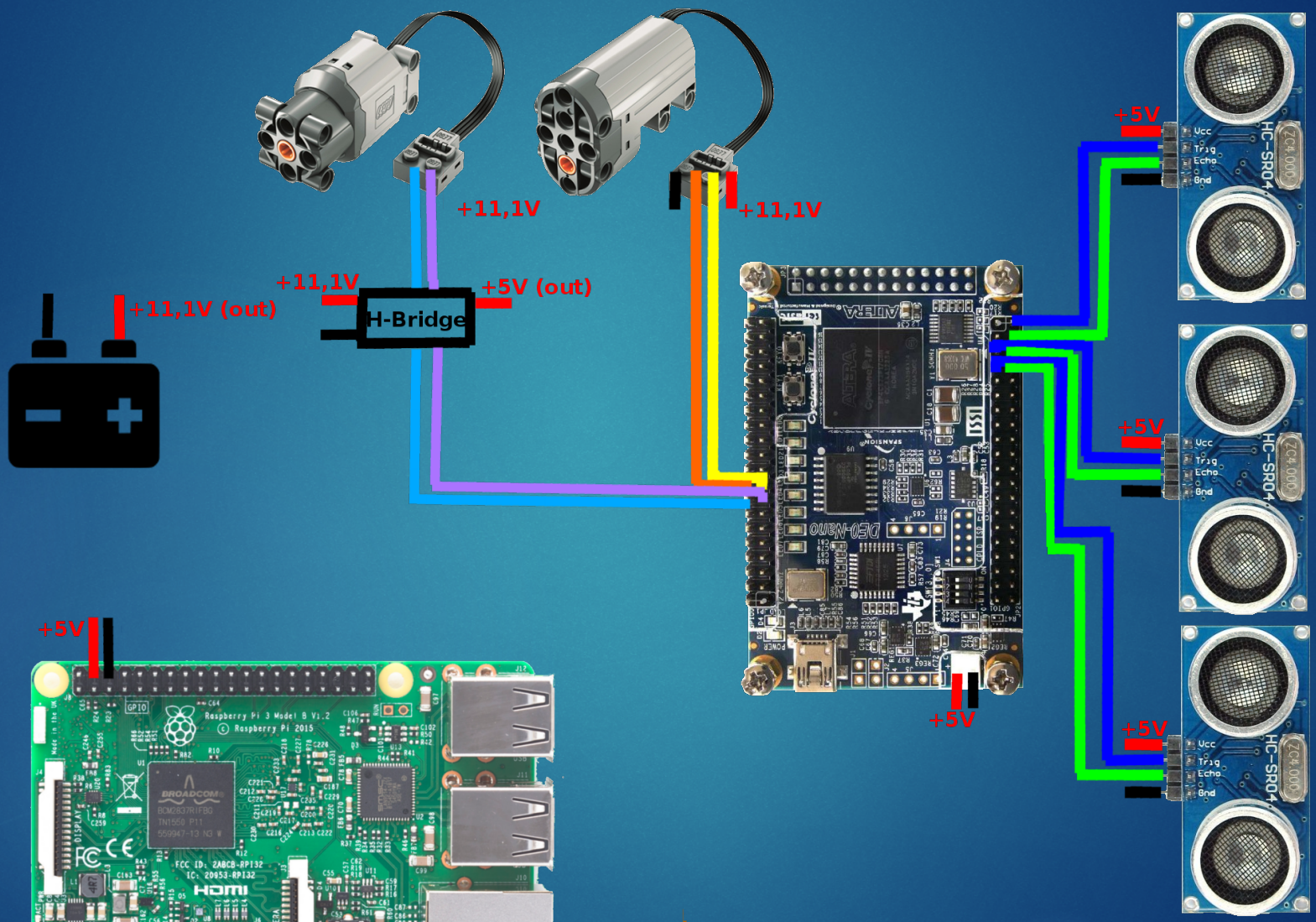
Wiring
circuit layout



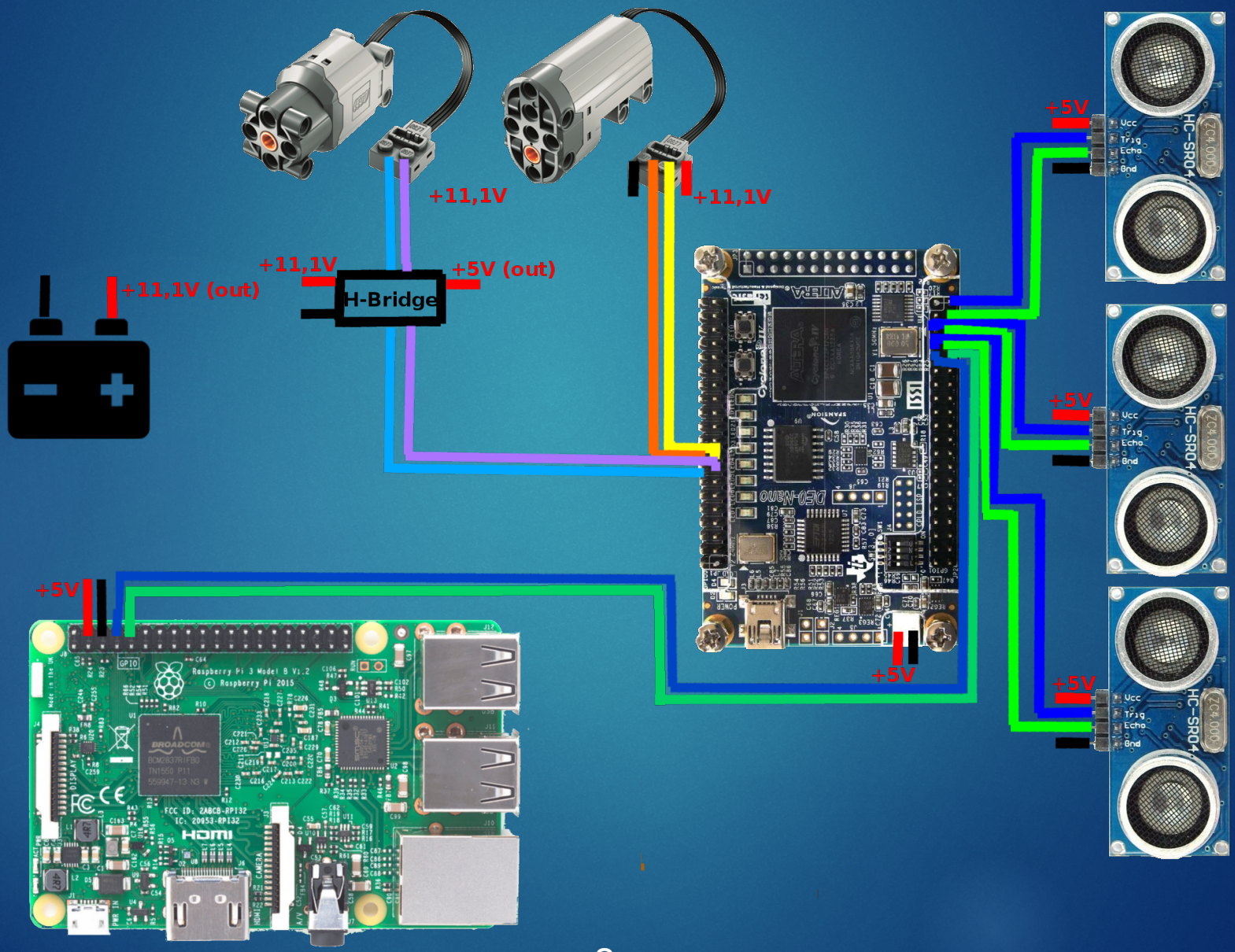
Wiring
circuit layout



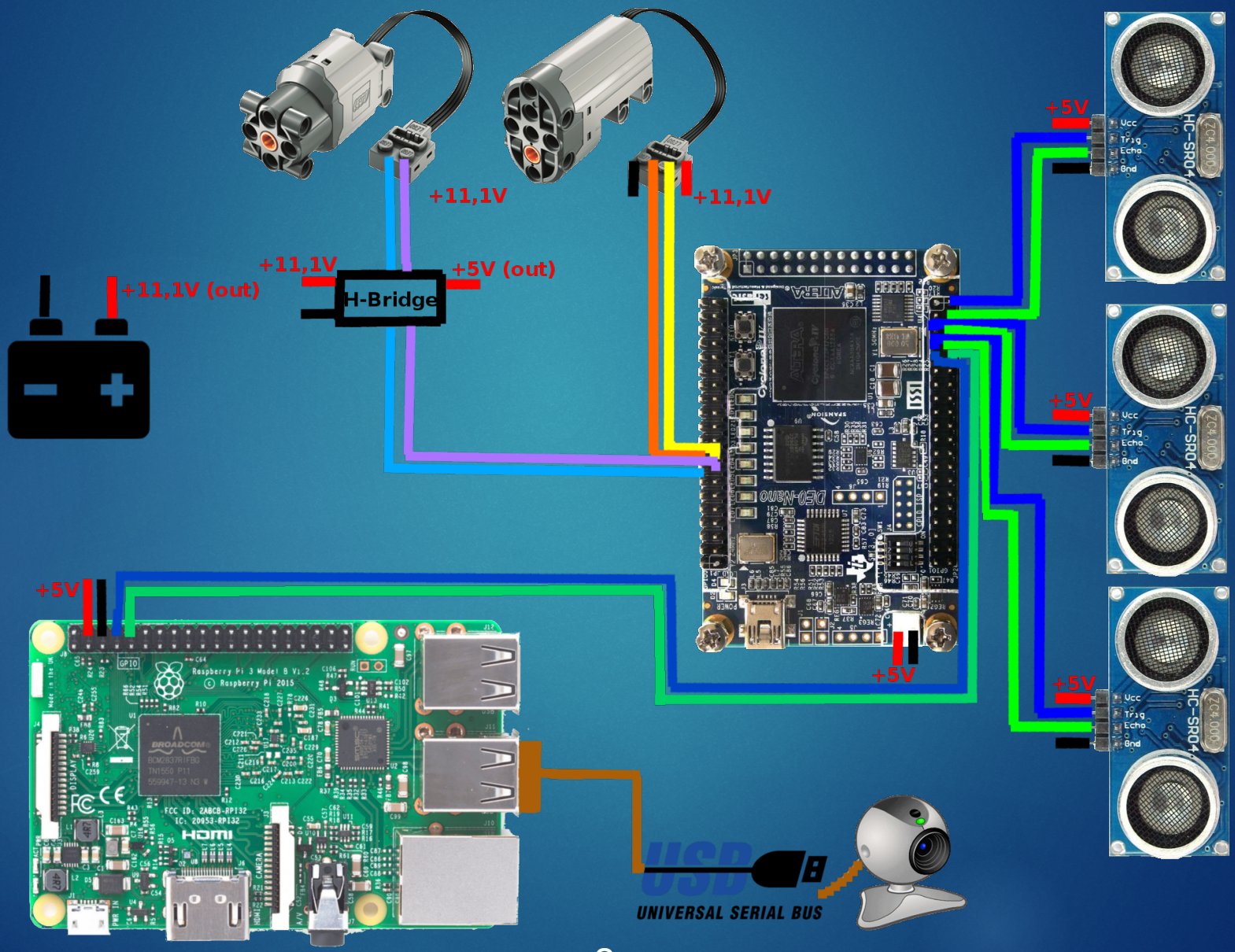
Wiring
circuit layout



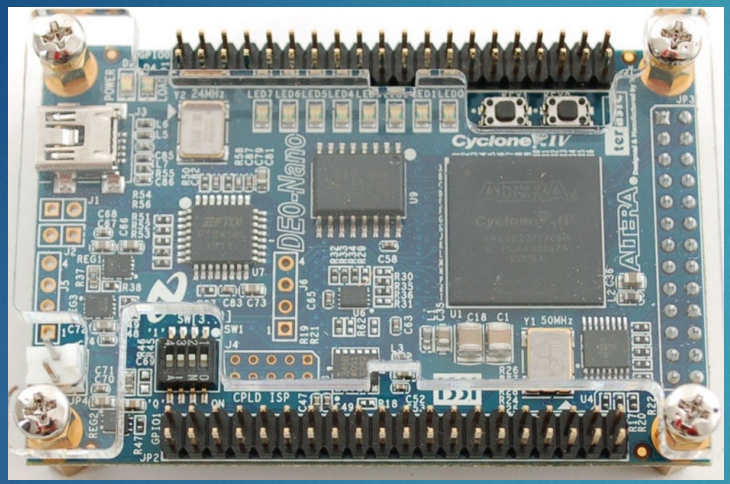
Wiring
circuit layout

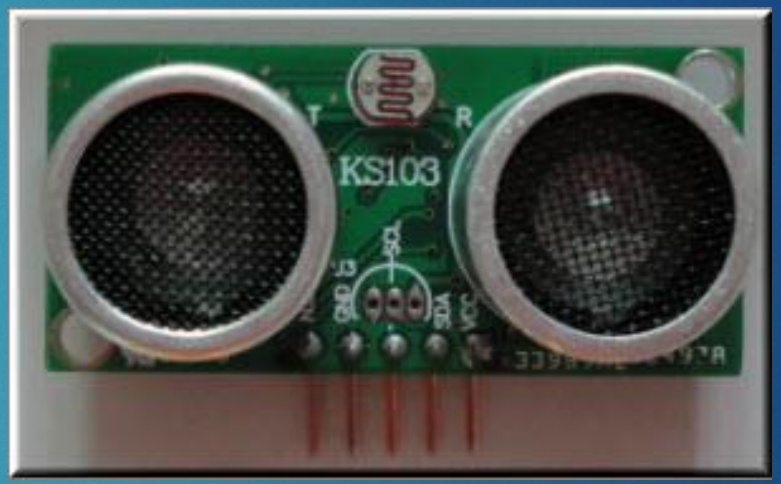


Wiring
circuit layout

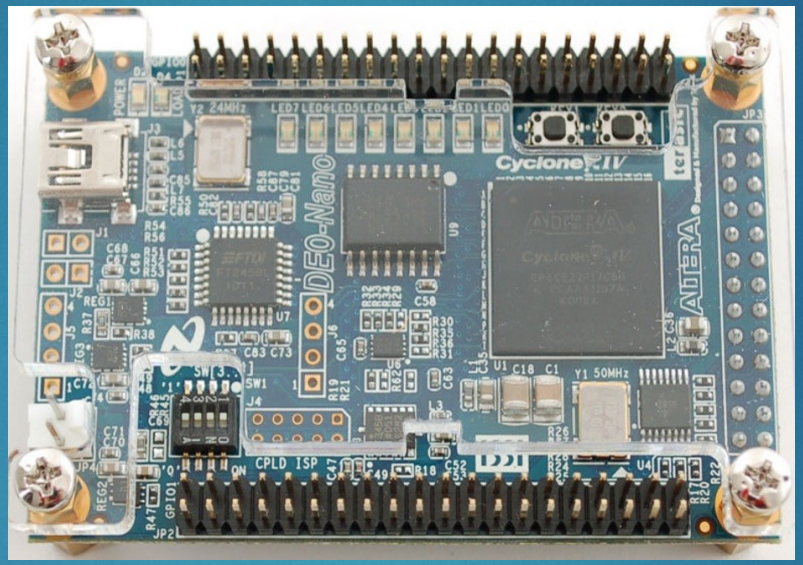


DE0-Nano Board & KS103 Ultrasound Module

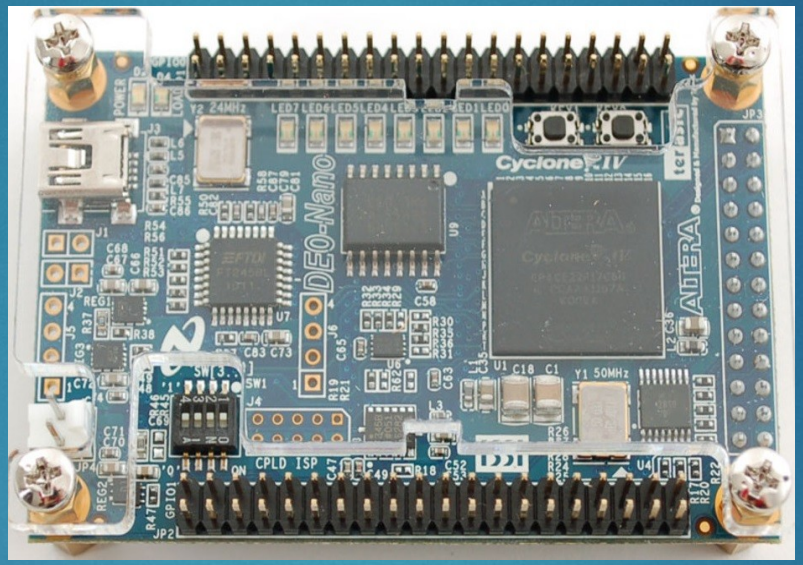




Why do we need the Nano Board?

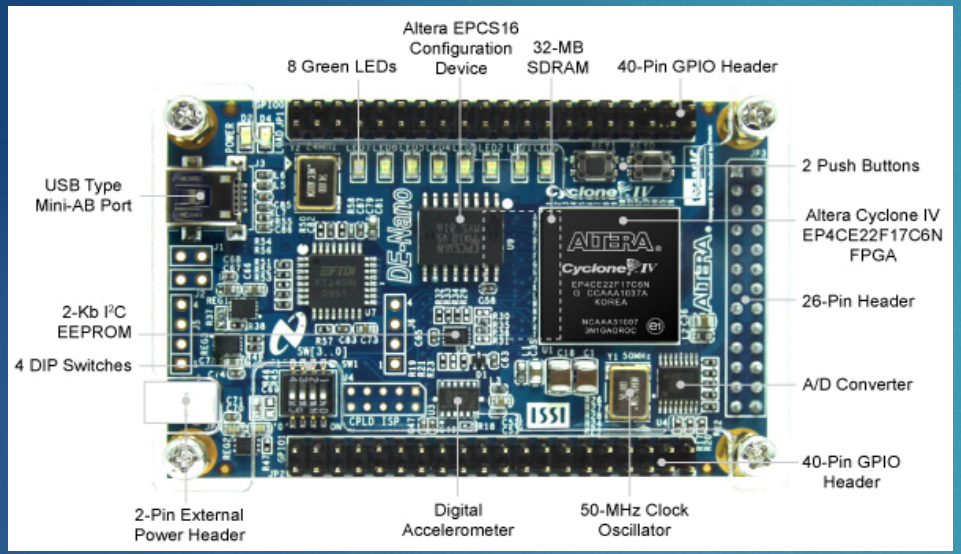


Why do we need the Nano Board?

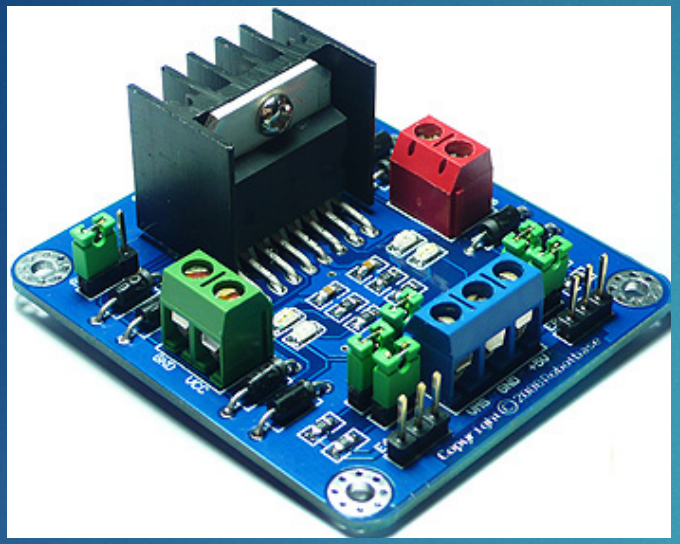


We use the DE0-Nano Board to provide an interface for the Raspberry Pi 3 to control the car

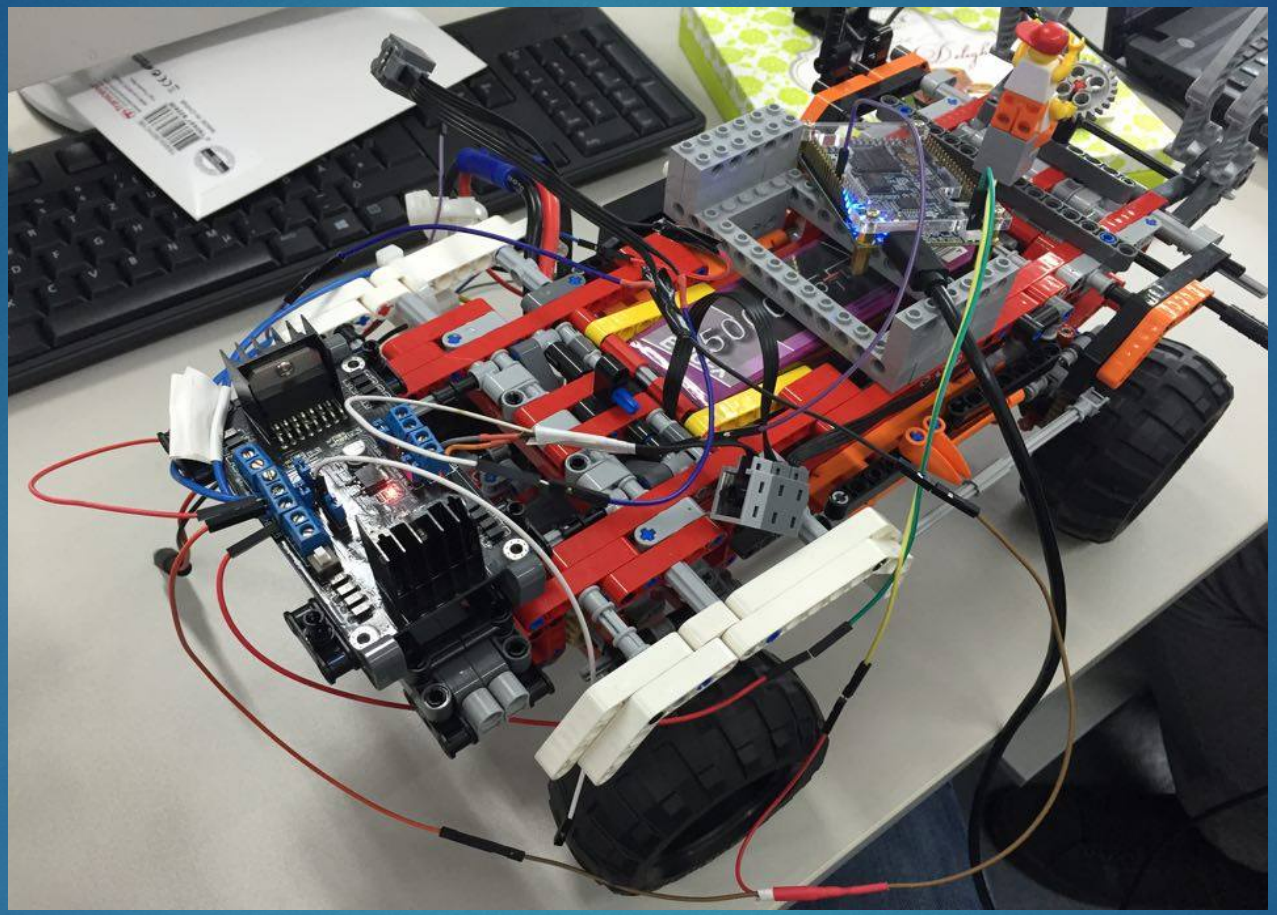
Features of the Board



- **Memory Devices**
  32-MB SDRAM
  2Kb I2C EEPROM
- **General user input/output**
  8 green LEDs
  2 debounced pushbuttons
  4-position DIP switches
- **Expansion header**
  Two 40-pin Headers (GPIOs) provide 72 I/O pins, 5V power pins, two 3.3V power pins and four ground pins
- **Clock system**
  On-board 50MHz clock oscillator
- **Power Supply**
  USB Type mini-AB port (5V)
  DC 5V pin for each GPIO header (2 DC 5V pins)
  2-pin external power header (3.6-5.7V)



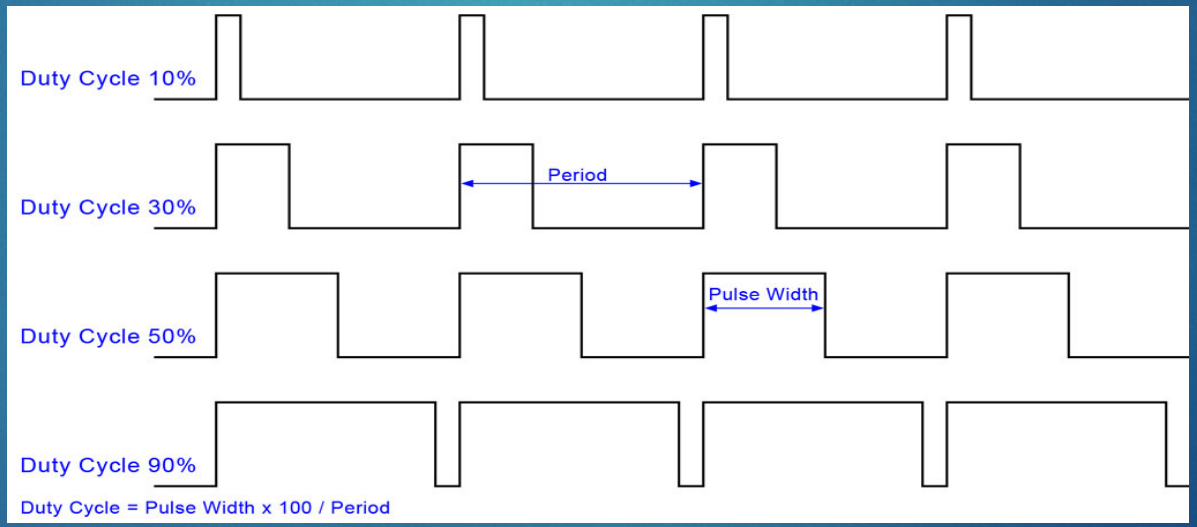
Our Nano board works at 5V, so we use an H-bridge to convert from the battery coming current to 5V.



Connection between the H-Bridge and Nano Board

PWM(Pulse-width Modulation)

- PWM is a modulation technique used to encode a message into a pulsing signal
- The width of the pulse is modulated depending on the requirement



Why do we use the PWM?

The voltage can be changed to control the speed of our motor

Why do we use the PWM?

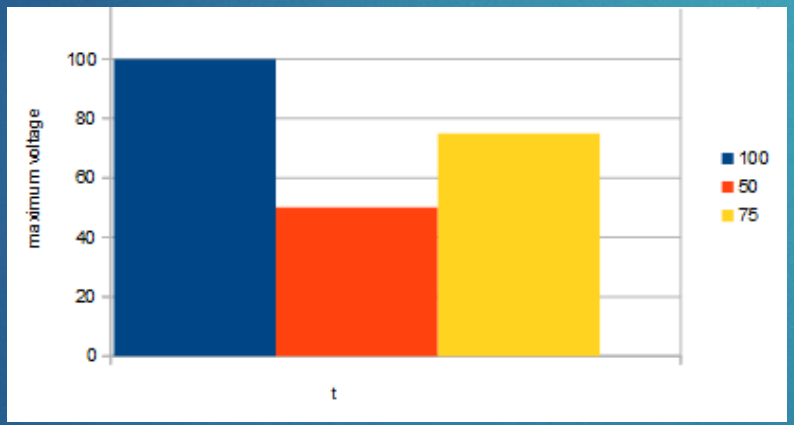
The voltage can be changed to control the speed of our motor

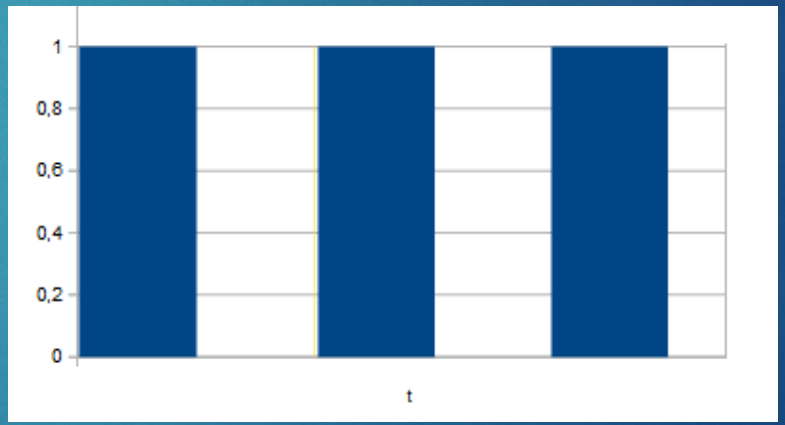
Analog vs Digital Voltage Control

Why do we use the PWM?

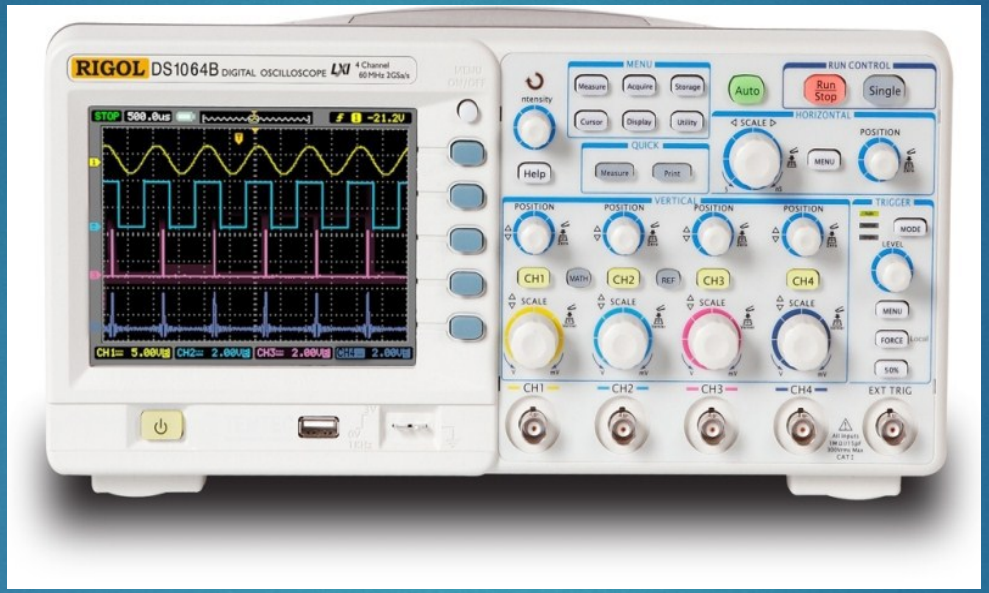
The voltage can be changed to control the speed of our motor

Analog vs Digital Voltage Control



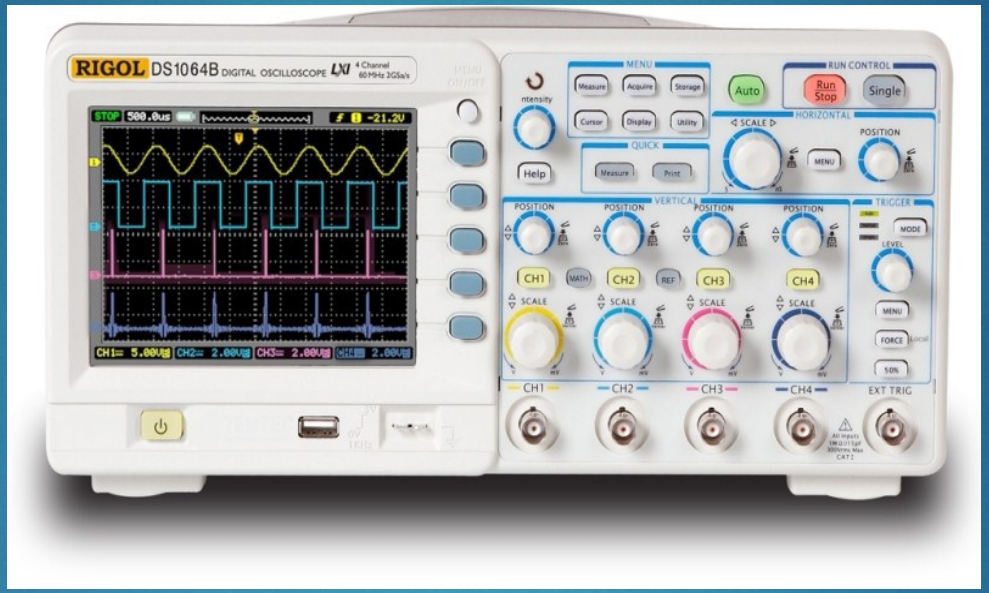


Oscilloscope



We use the oscilloscope to verify the duty cycle

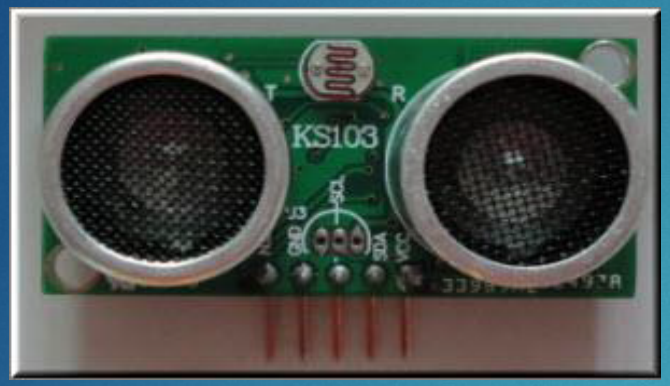
Oscilloscope



We use the oscilloscope to verify the duty cycle

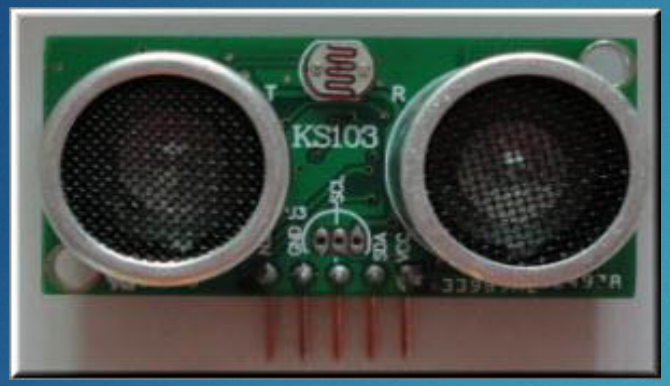
Link to Lego Car

KS103 Ultrasound Module



We use this device to measure the distance between our car and the obstacle

KS103 Ultrasound Module

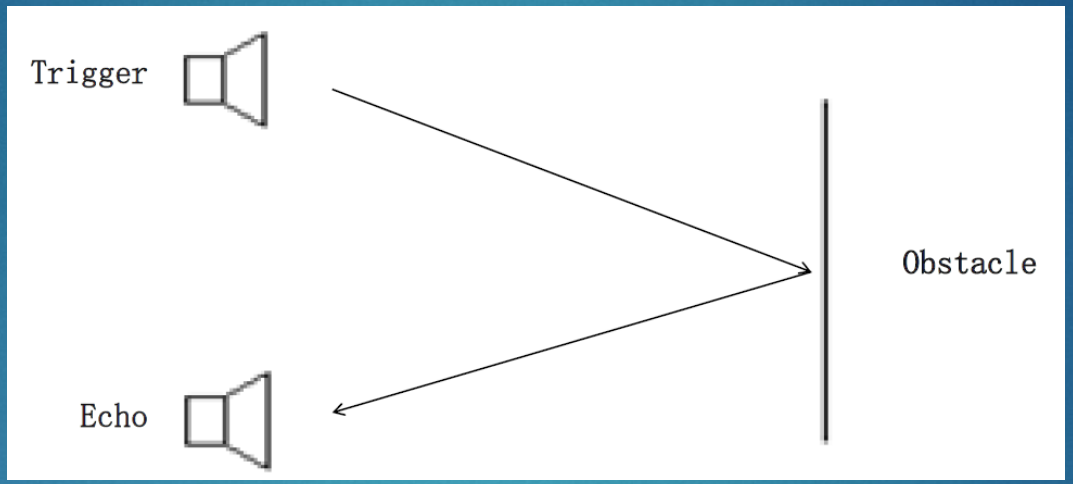


We use this device to measure the distance between our car and the obstacle

Pins

- VCC: power pin
- SDA/TX: data pin
- SCL/RX: clock pin
- GND: power ground pin
- Mode: selects the communication mode

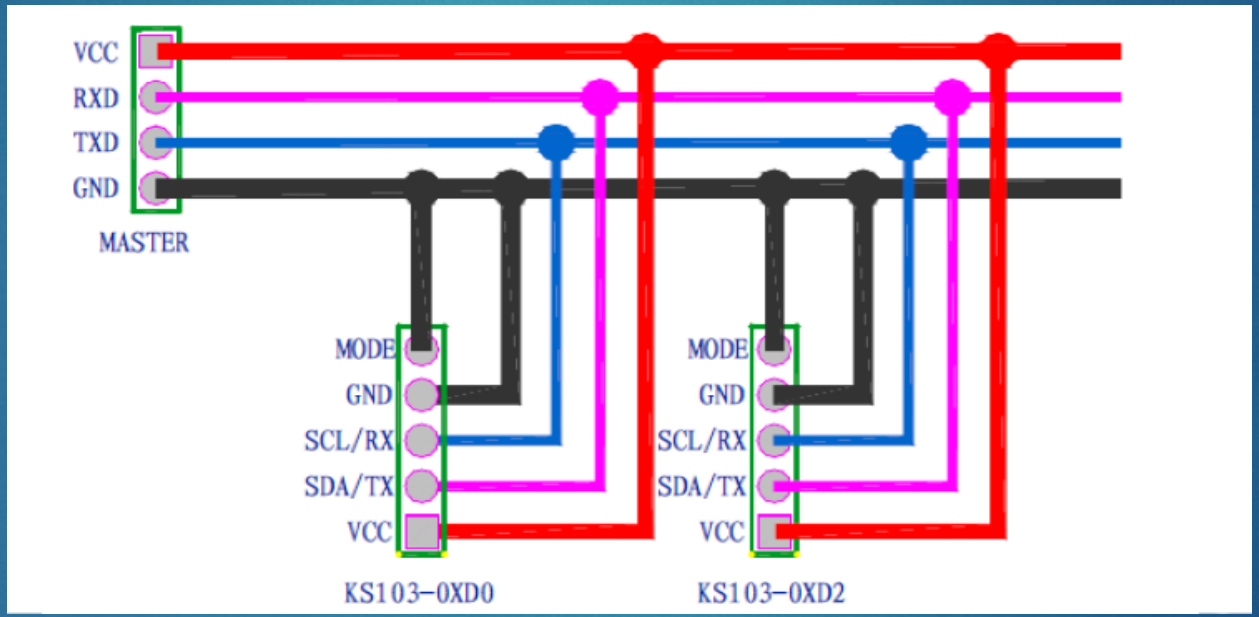
How It Works?



The trigger sends chirps and then the device calculates the distance by using this formula: (340(m/s) x Δt(s)) / 2)

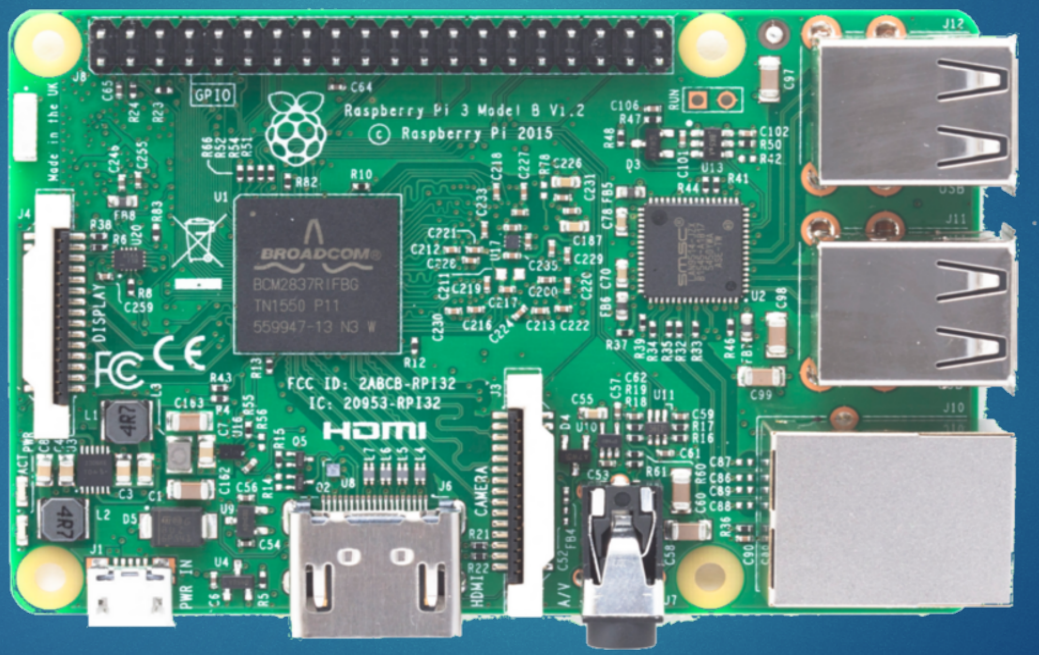
Connection of Three Modules

Our task was to connect three modules with each other



Raspberry Pi

- Raspberry Pi Version: Raspberry Pi 3
- Capable little computer that can be used for electronics projects
- Used to connect the camera with the nano-board for the line detection



Raspberry Pi

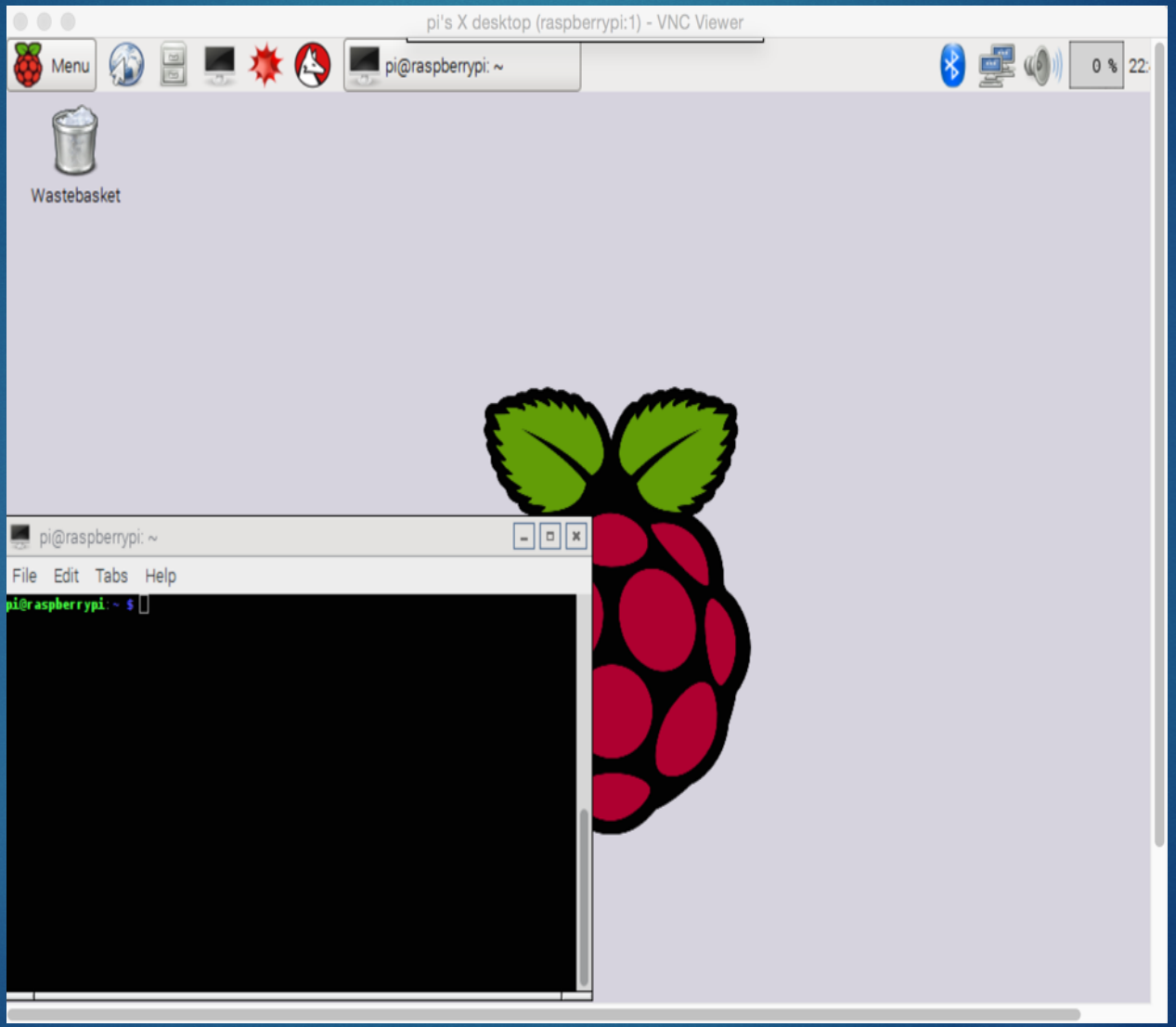
Carried out steps:

1. Raspbian installed (operating system for the Raspberry Pi)
2. Connected to the laptop using Ethernet
3. Used SSH (Secure Shell) to gain access to the command line of the Raspberry Pi
4. Controlled the Raspberry Pi using VNC (a graphical desktop sharing system)
5. Downloaded OpenCV and connected the camera
6. Tested the Code for the line detection

To-Do:

1. Connect the Raspberry Pi to the nano-board
2. Align the line detection with the motor control

Raspberry Pi



Line Detection

Approach

- Approximate line
- Calculate direction

Line Detection

Approach

- Approximate line
- Calculate direction

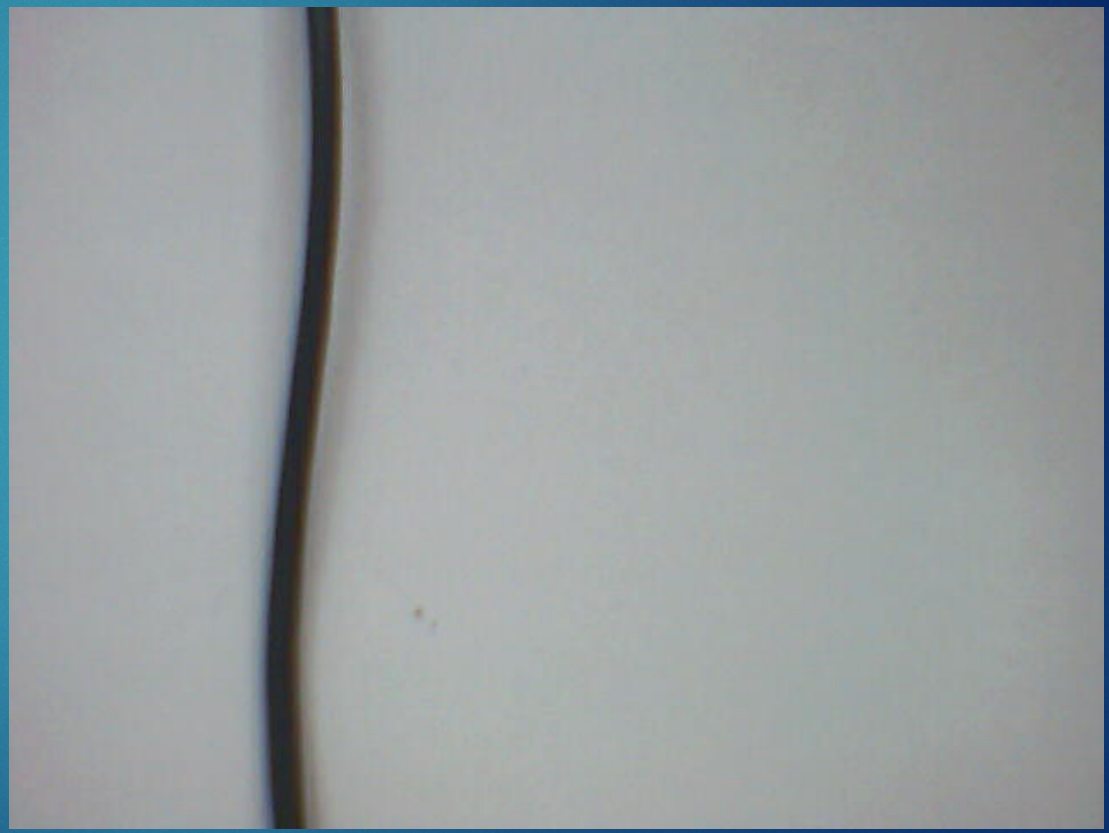
Assumptions

- Vertical line
- Car position
- Line is highest contrast on image
- Width of line (optional)

Line Detection

Steps

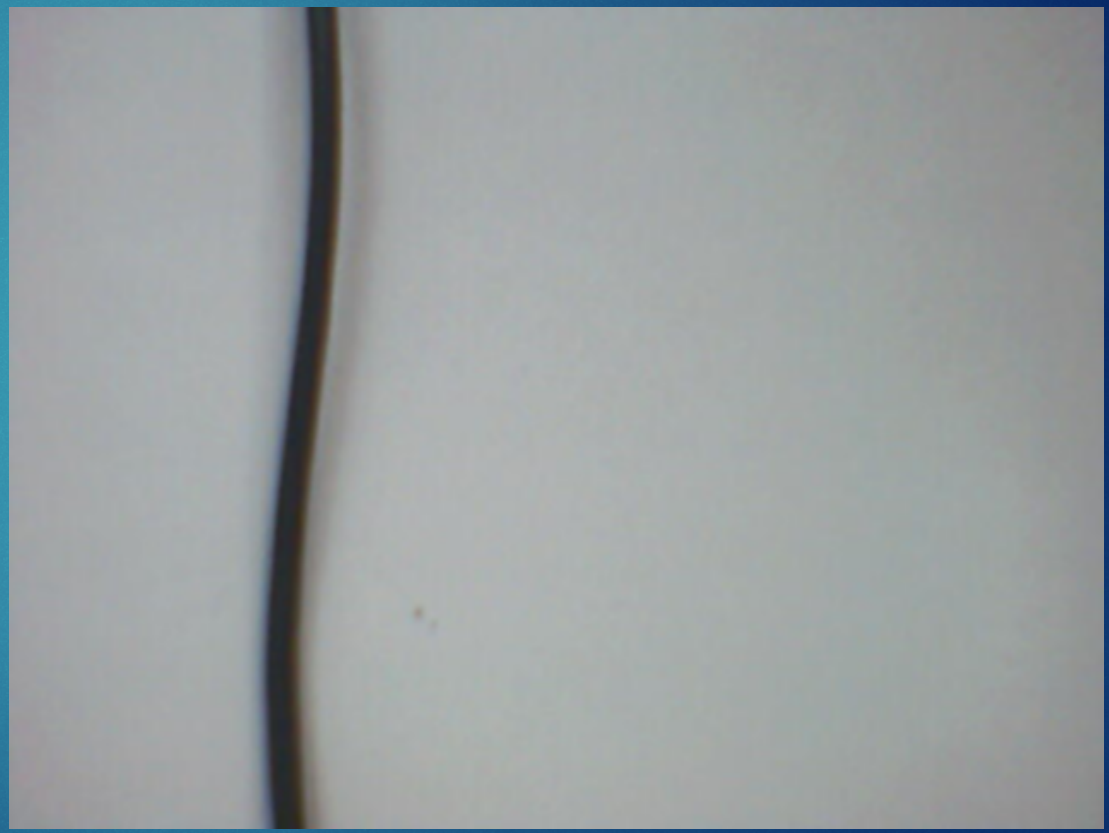
- Get Frame from camera



Line Detection

Steps

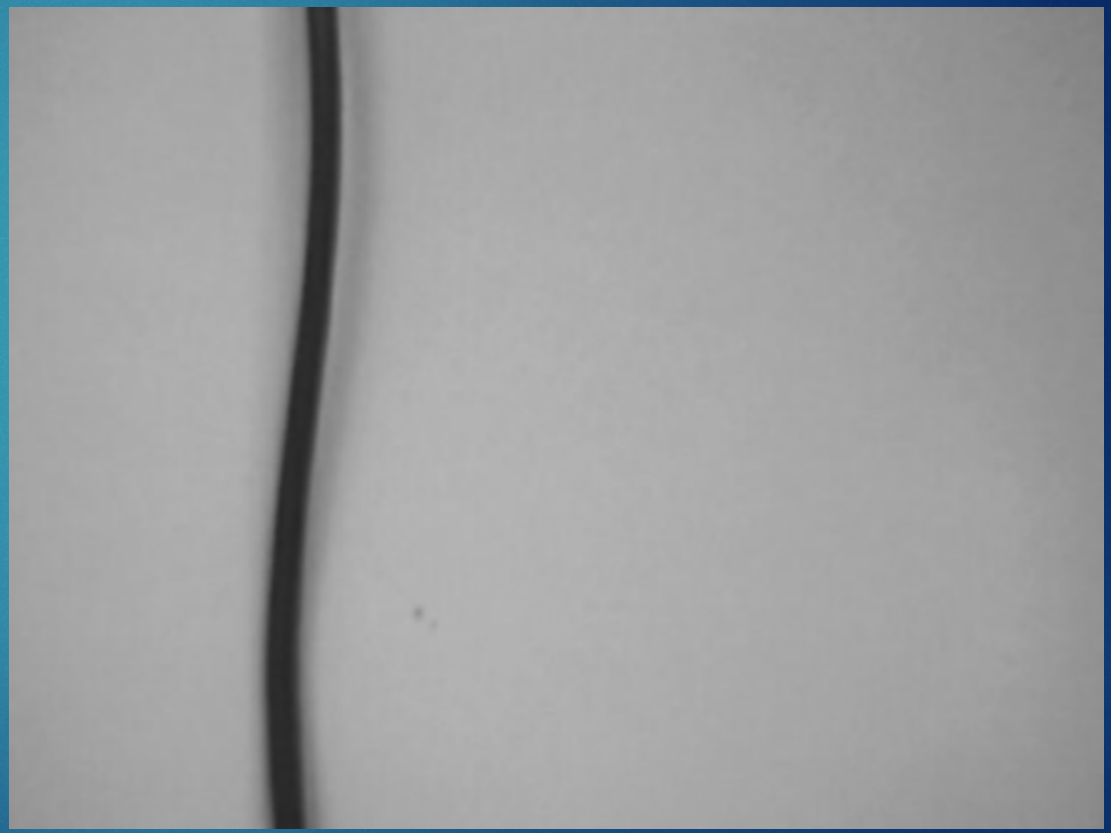
- Get Frame from camera
- Apply Gaussian Filter



Line Detection

Steps

- Get Frame from camera
- Apply Gaussian Filter
- Transform image from RGB to grey

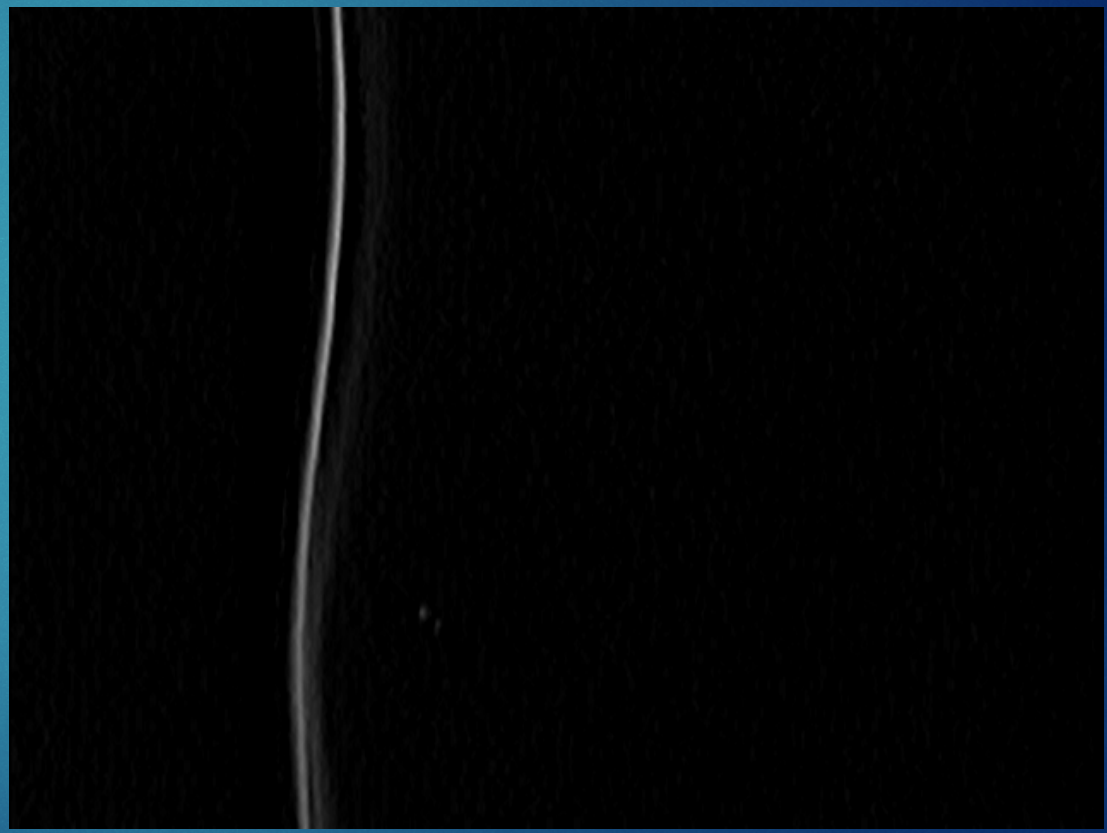


Line Detection

Steps

- Get Frame from camera
- Apply Gaussian Filter
- Transform image from RGB to grey
- Approximate Gradient

$$M = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



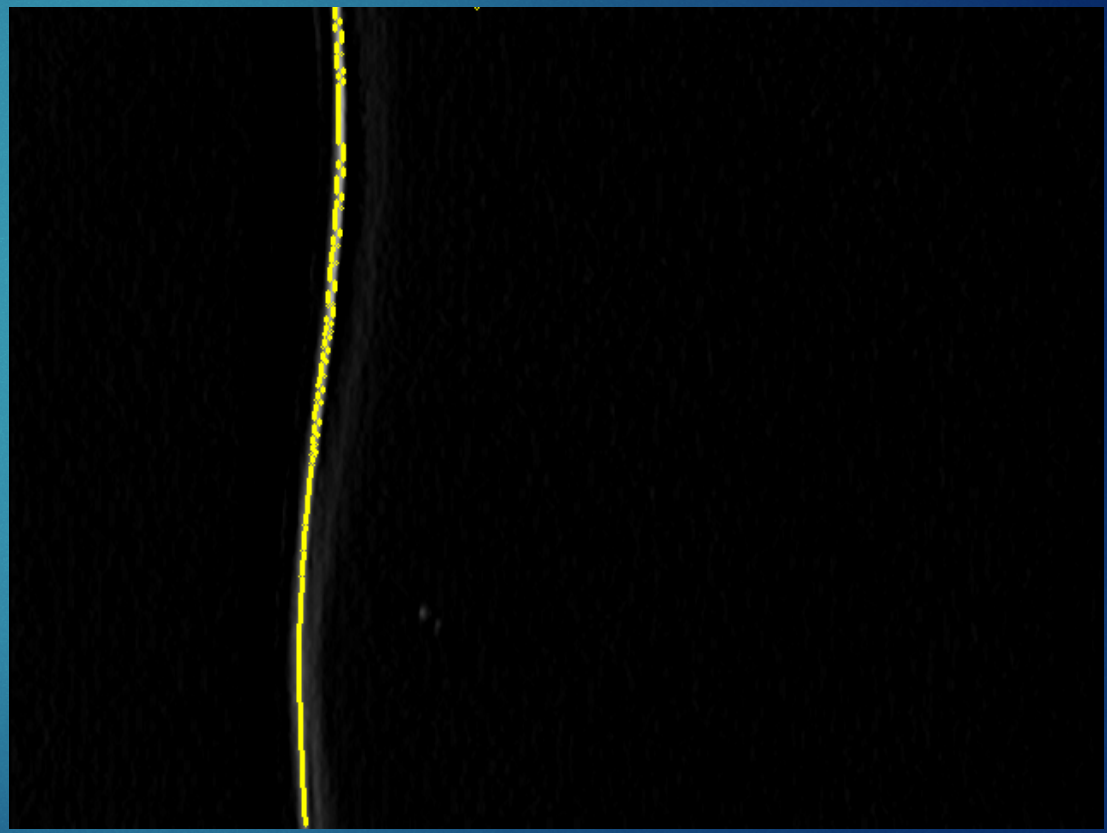
Line Detection

Steps

- Get Frame from camera
- Apply Gaussian Filter
- Transform image from RGB to grey
- Approximate Gradient

$$M = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- Calculate maximal points for each row



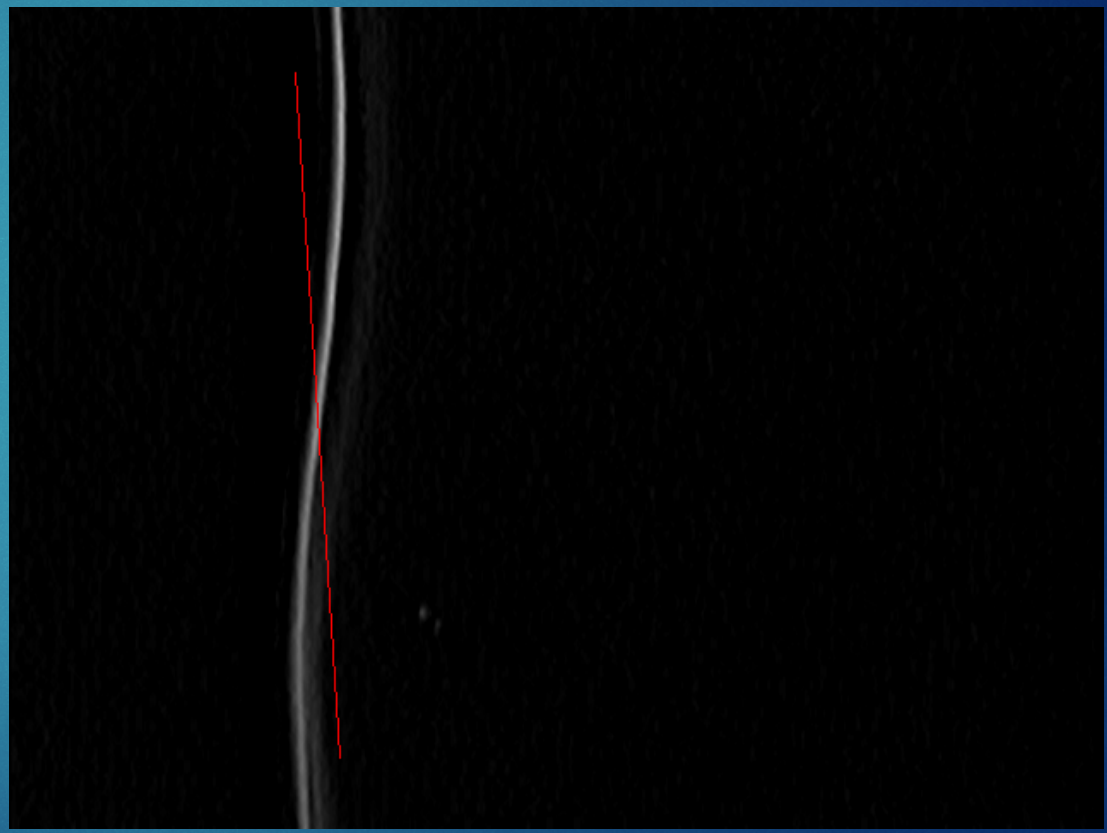
Line Detection

Steps

- Get Frame from camera
- Apply Gaussian Filter
- Transform image from RGB to grey
- Approximate Gradient

$$M = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- Calculate maximal points for each row
- Fit line on points



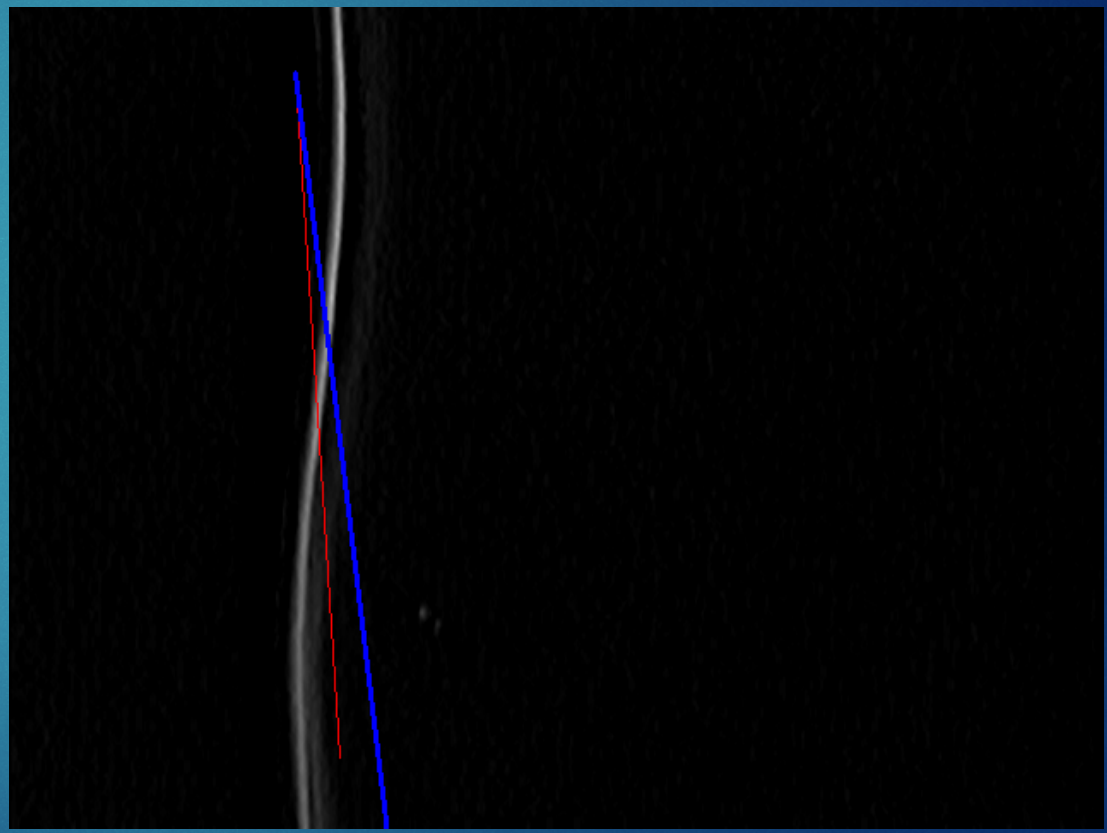
Line Detection

Steps

- Get Frame from camera
- Apply Gaussian Filter
- Transform image from RGB to grey
- Approximate Gradient

$$M = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- Calculate maximal points for each row
- Fit line on points
- Calculate direction of car



Line Detection

Pros/Cons

- Fast calculation time
- Vulnerable to noise

TODO

- Calculate discrete angle
- Attaching camera
- Testing/Optimization