

Lab6 - Line Following Car

NTHU Hardware Design and Laboratory (Fall 2023)

11/28/2023

By Prof. Chun-Yi Lee

Agenda

Introduction

Materials overview

FPGA configuration

Grading

Agenda

Introduction

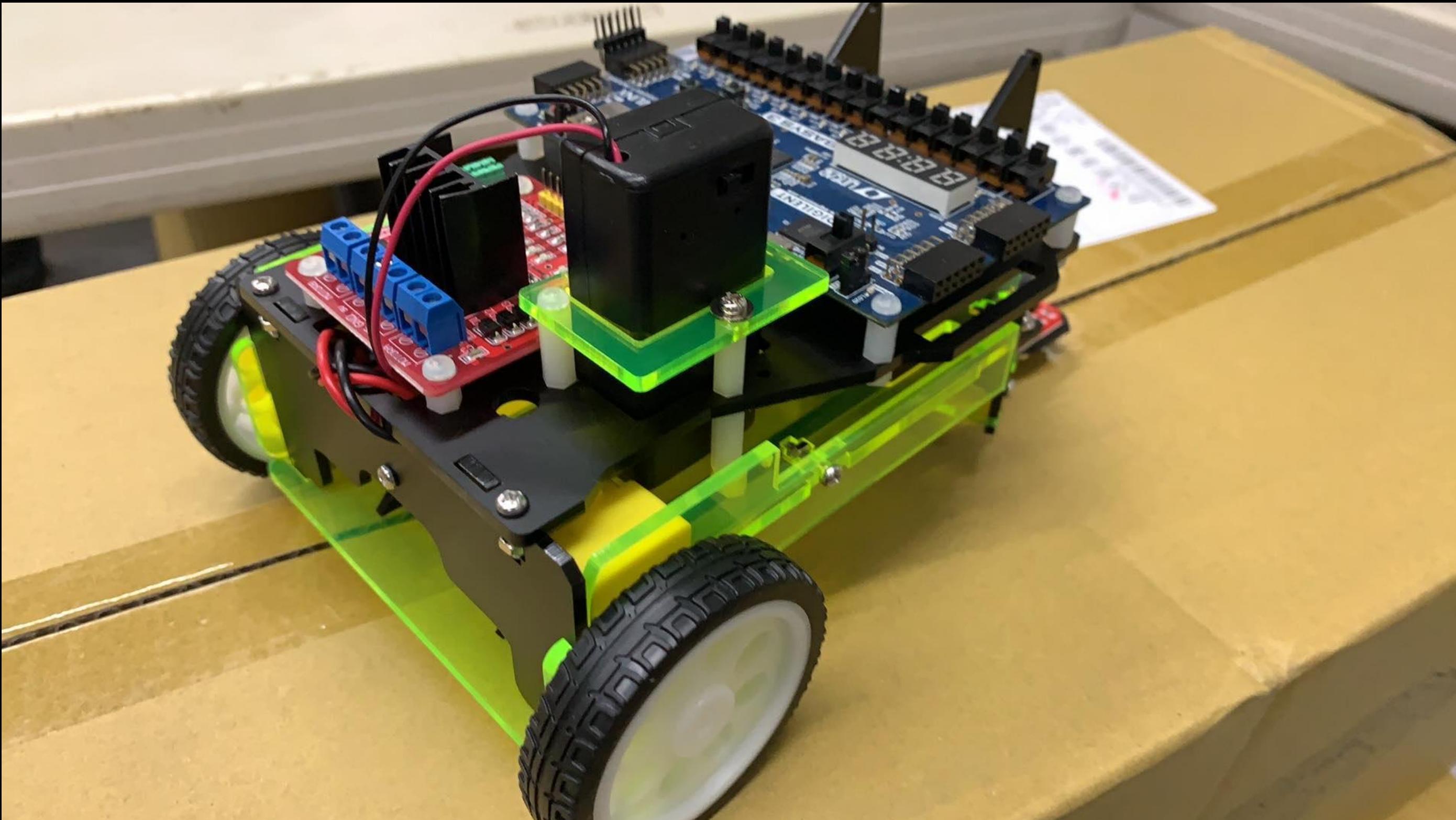
Materials overview

FPGA configuration

Grading

Introduction (1/1)

- . One of the advanced question in Lab6 is to implement a line following car.



Agenda

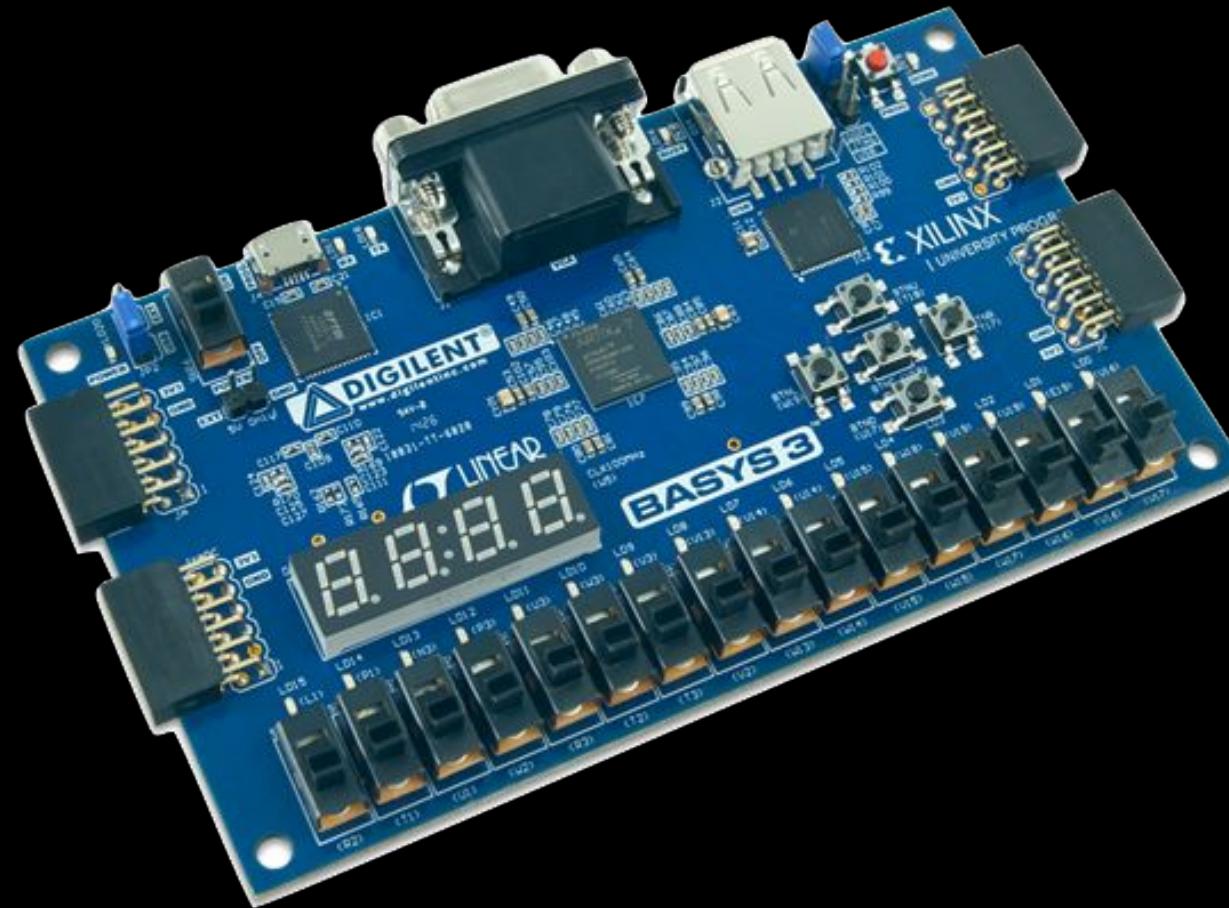
Introduction

Materials overview

FPGA configuration

Grading

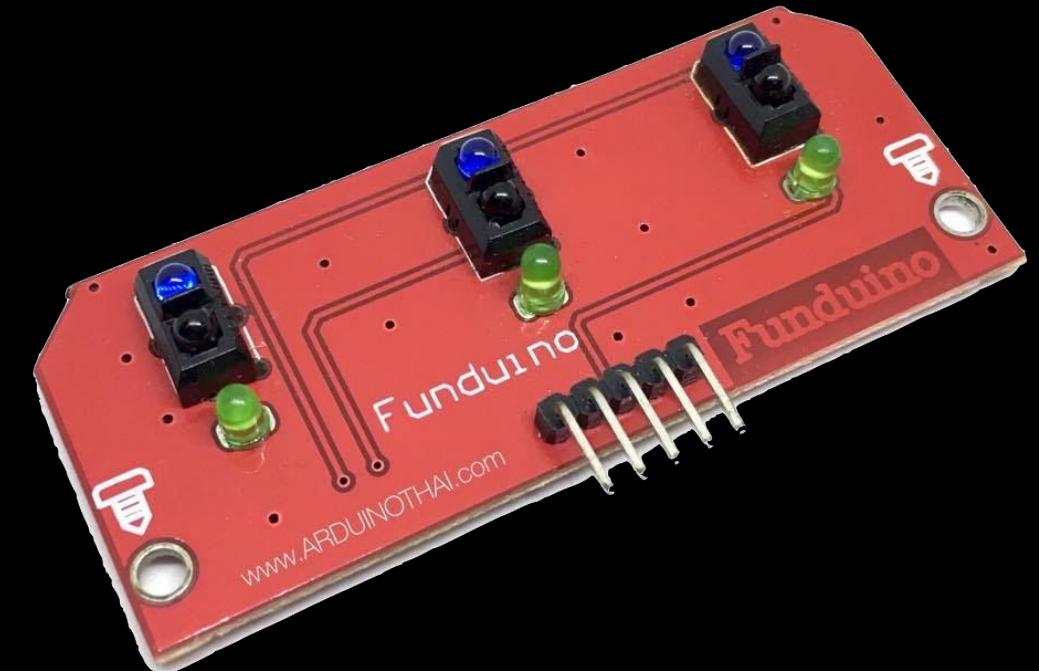
Materials overview (1/17)



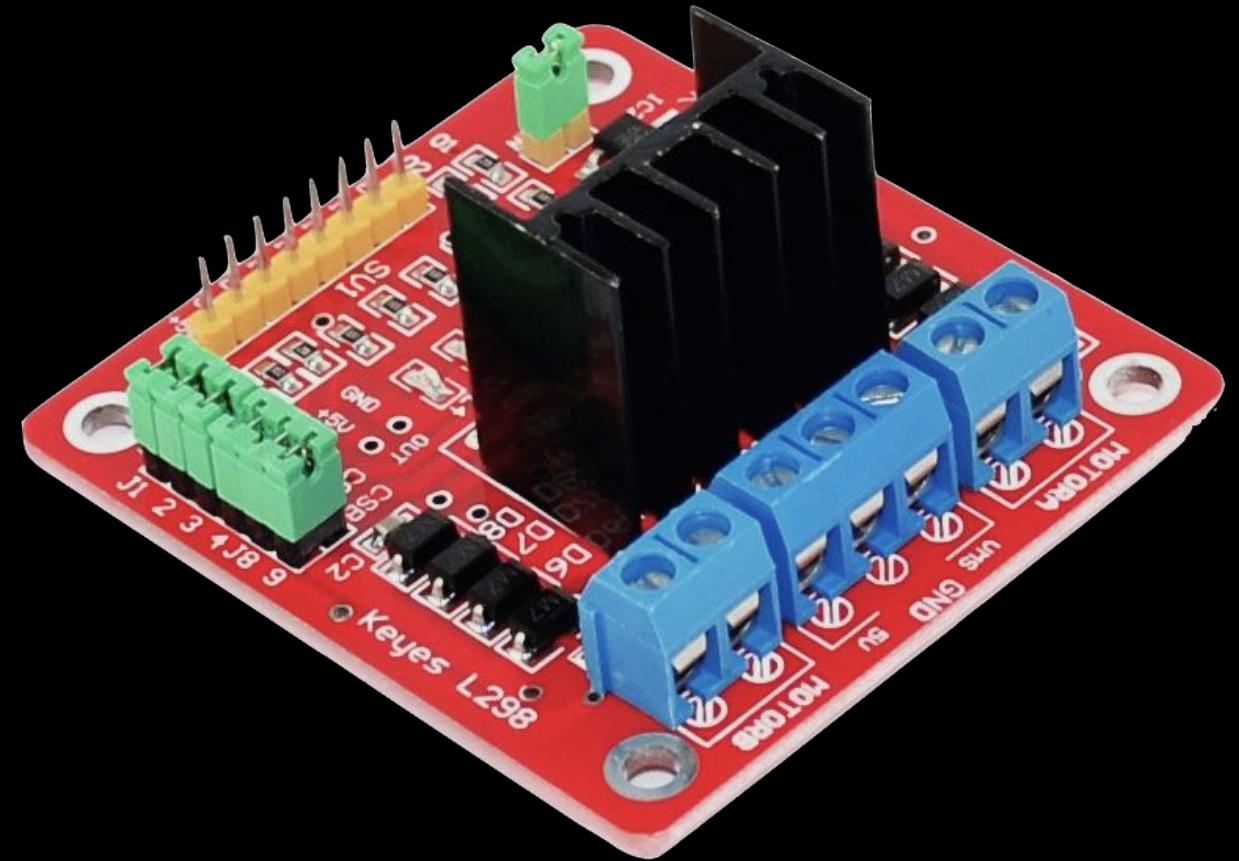
FPGA



HC-SR04



TCRT5000
3-way line tracking IR



L298N motor driver module

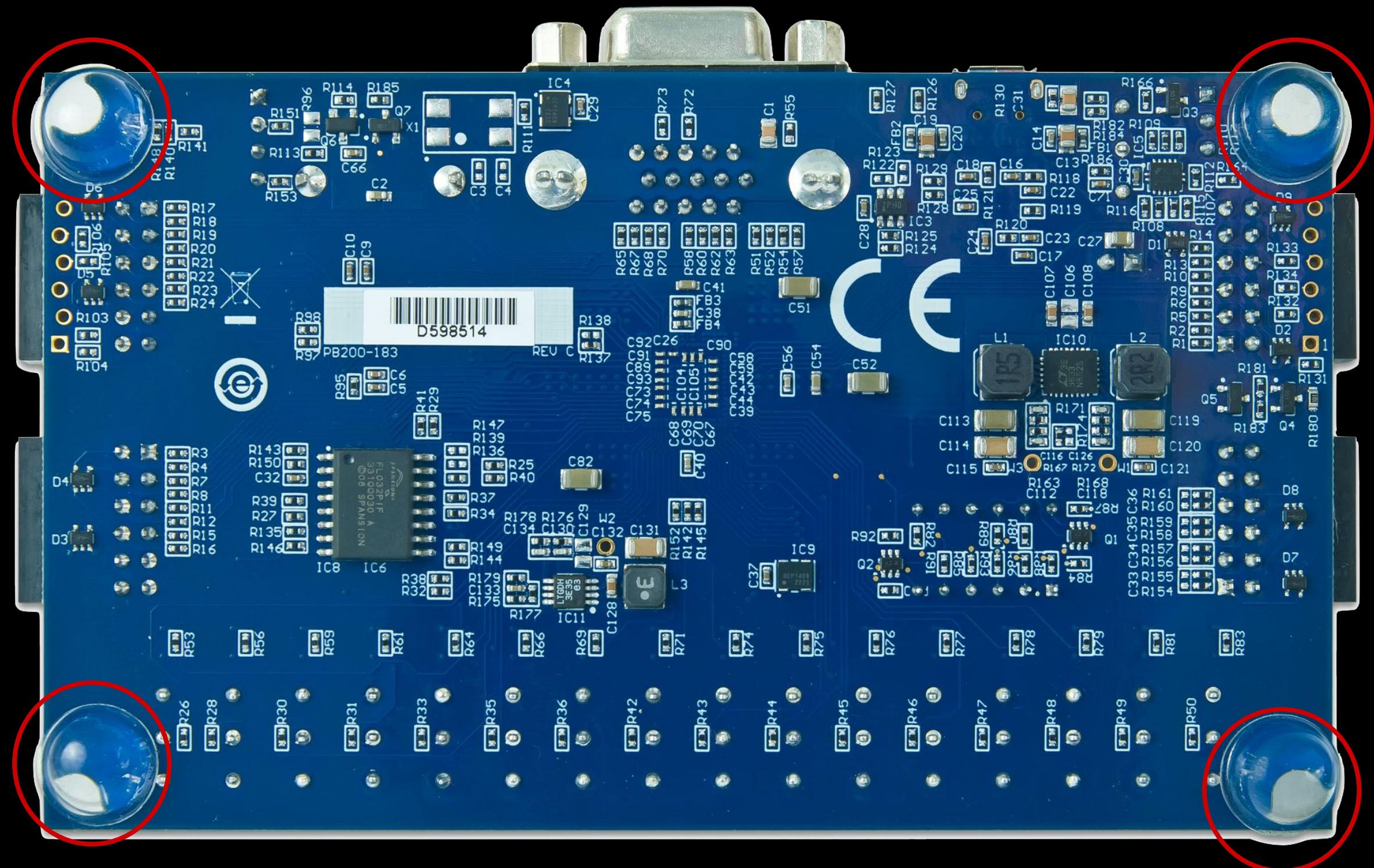


Gear motor

Materials overview (2/17)

FPGA

- Remove the rubber mats at four corners.
- BE CAREFUL !!
- Don't lose the mats.
- Screw fpga to the car.



Materials overview (3/17)

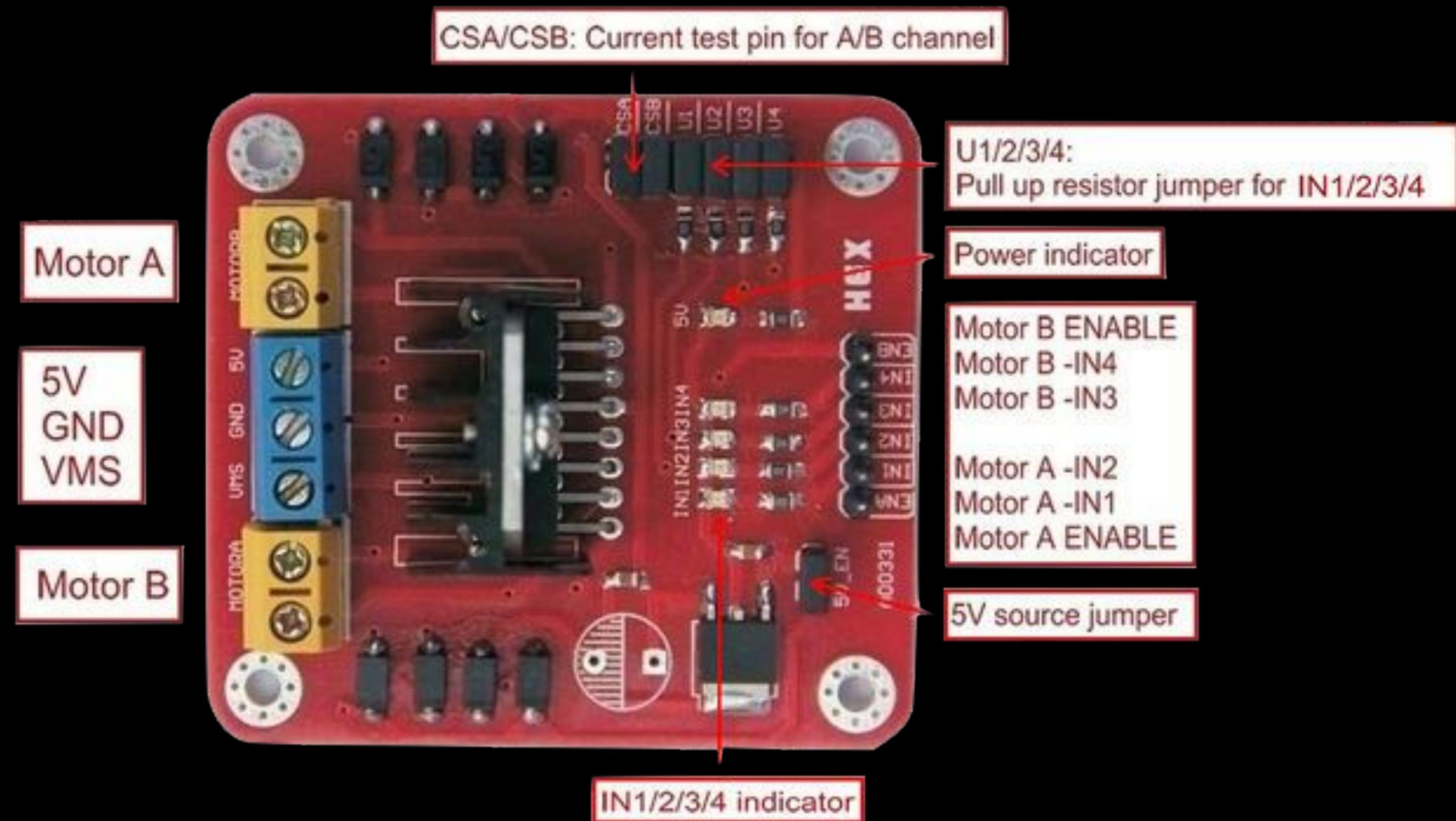
Gear motor + L298N motor driver module

- Why do we need a motor driver module?
 - Motor require high amount of current whereas the controller circuit works on low current signals.
 - We want to control these motors using other controller devices, such as FPGA.
 - Motor drivers acts as an interface between the motors and the control circuits.

Materials overview (4/17)

L298N motor driver module

- Able to control 2 motors.
- VMS for motor.
- 5V pin which can either be an input or output.



Materials overview (5/17)

L298N motor driver module

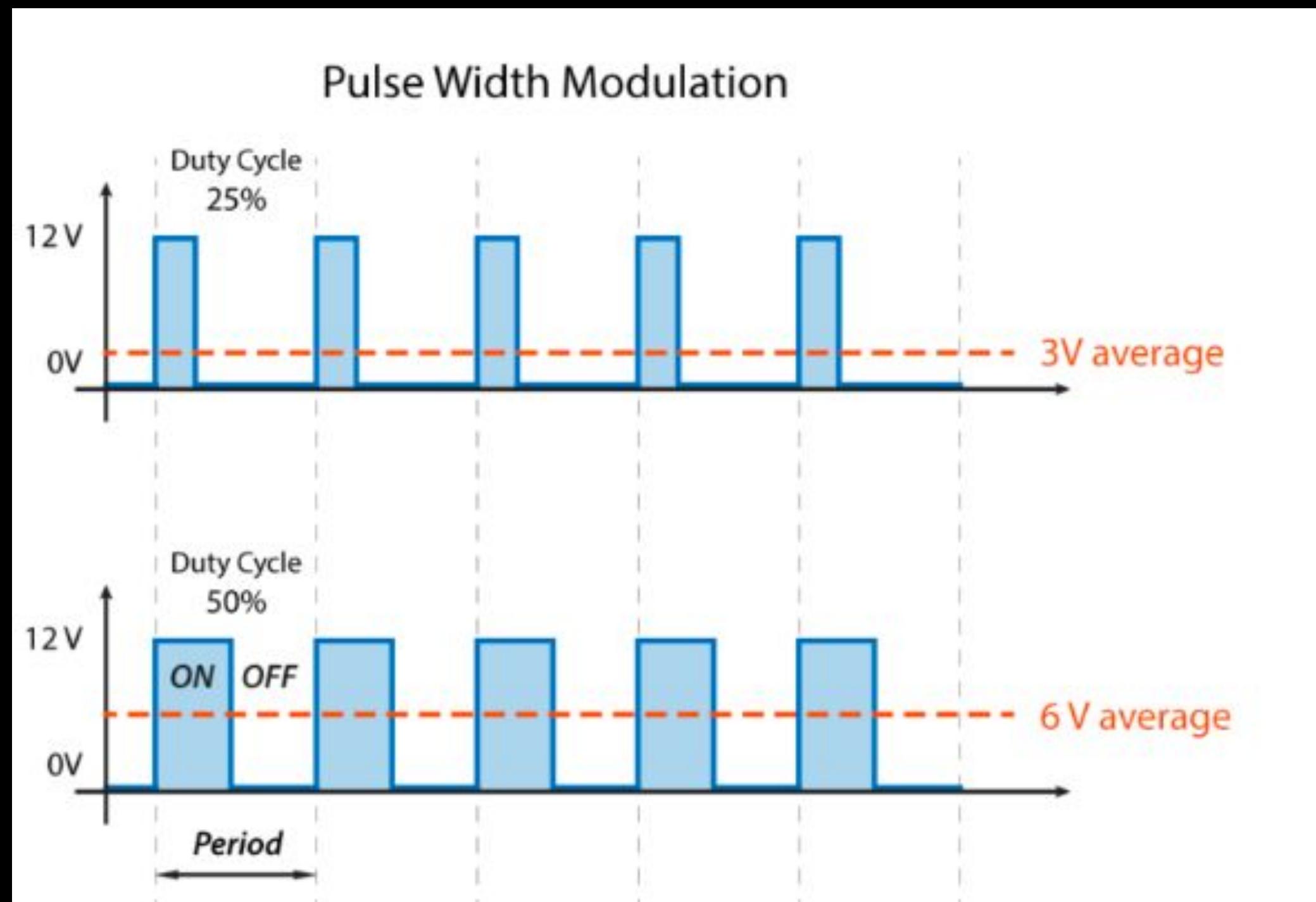
- IN1 and IN2 pins control the spinning direction of the motor A, while IN3 and IN4 control motor B.

| Input1 | Input2 | Spinning Direction |
|---------|---------|--------------------|
| Low(0) | Low(0) | Motor OFF |
| High(1) | Low(0) | Forward |
| Low(0) | High(1) | Backward |
| High(1) | High(1) | Motor OFF |

Materials overview (6/17)

L298N motor driver module

- ENA and ENB are used to control speed by PWM.
- Run faster with higher duty.



Materials overview (7/17)

Code for motor (motor.v) :

```
module motor(
    input clk,
    input rst,
    output [1:0] pwm
);

reg [9:0]next_left_motor, next_right_motor;
reg [9:0]left_motor, right_motor;
wire left_pwm, right_pwm;

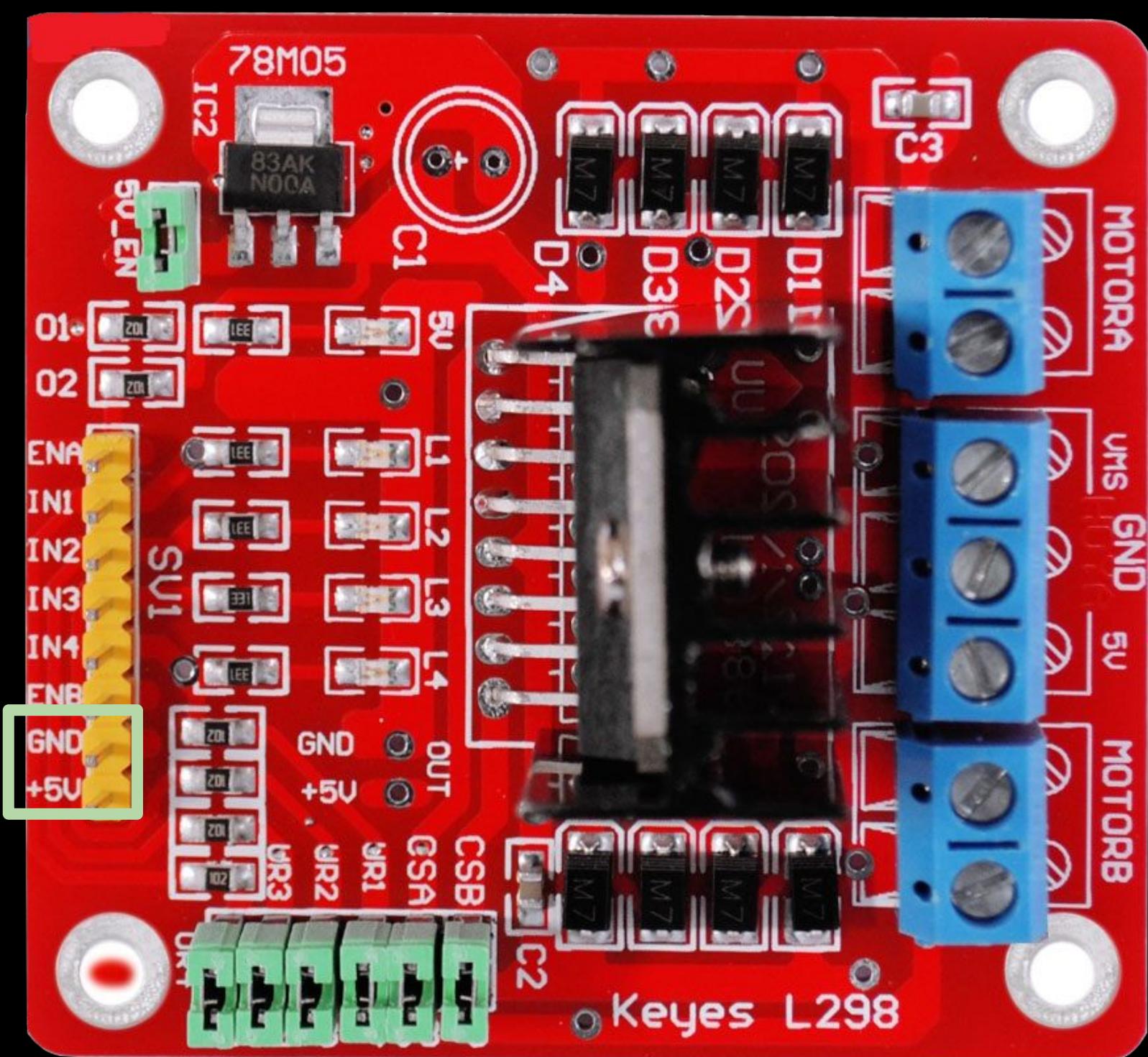
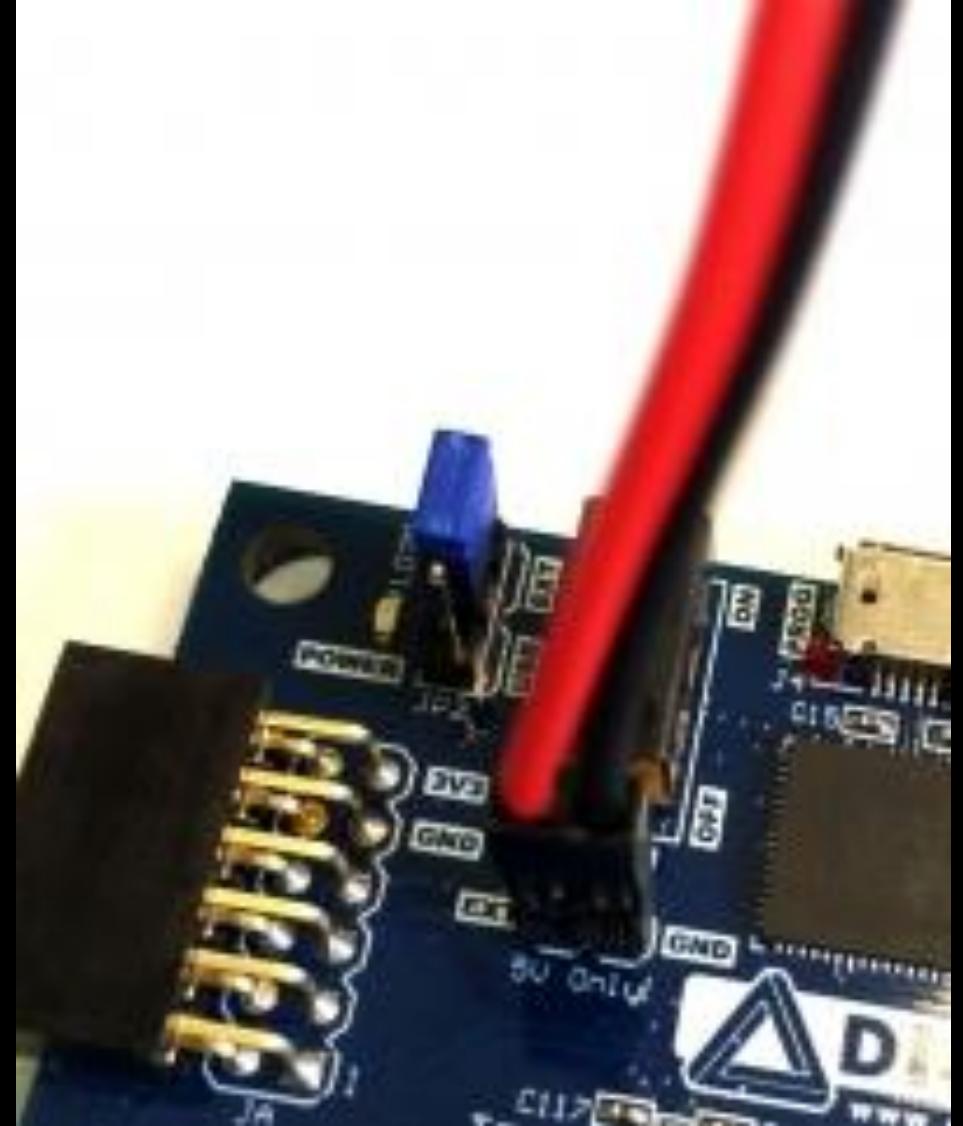
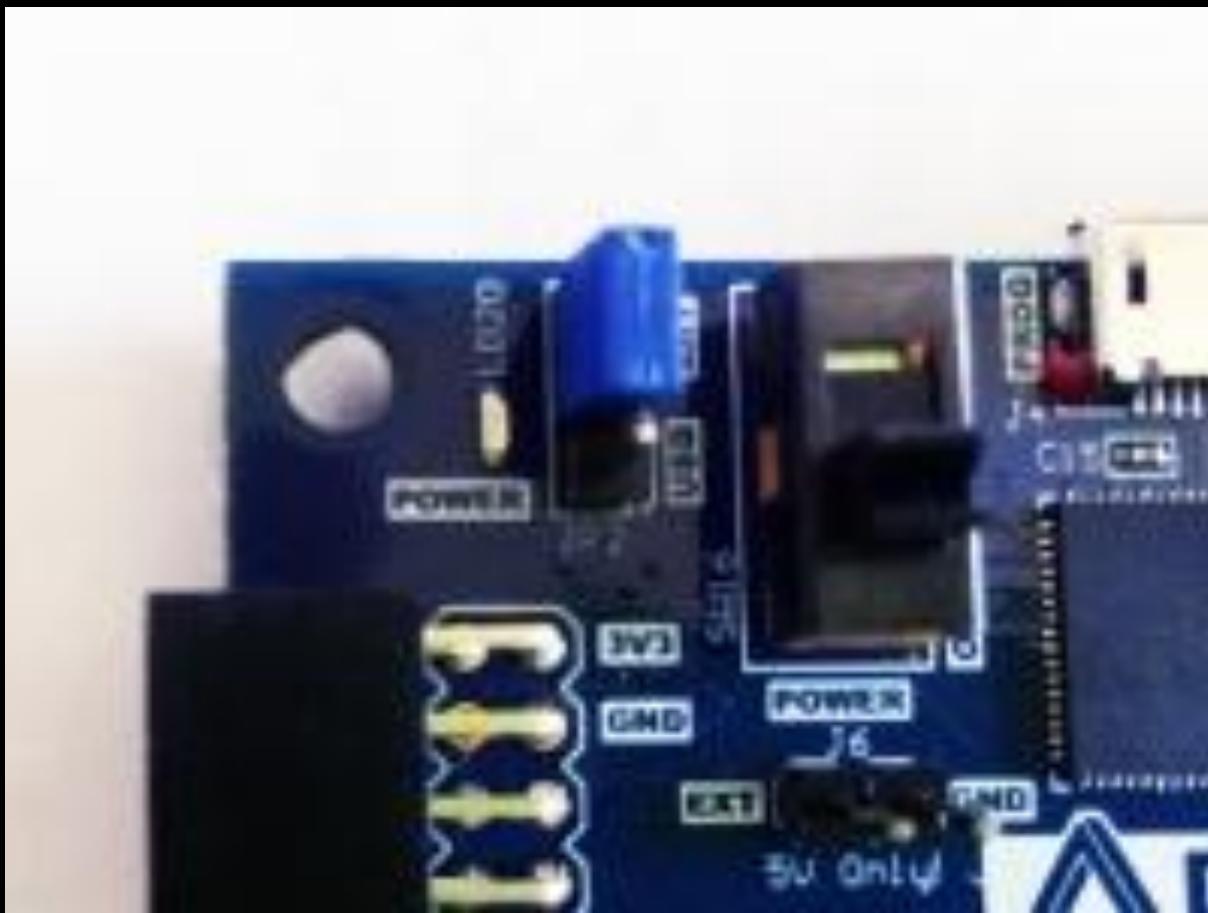
motor_pwm m0(clk, rst, left_motor, left_pwm);
motor_pwm m1(clk, rst, right_motor, right_pwm);

always@(posedge clk)begin
    if(rst)begin
        left_motor <= 10'd0;
        right_motor <= 10'd0;
    end else begin
        left_motor <= next_left_motor;
        right_motor <= next_right_motor;
    end
end
```

Materials overview (8/17)

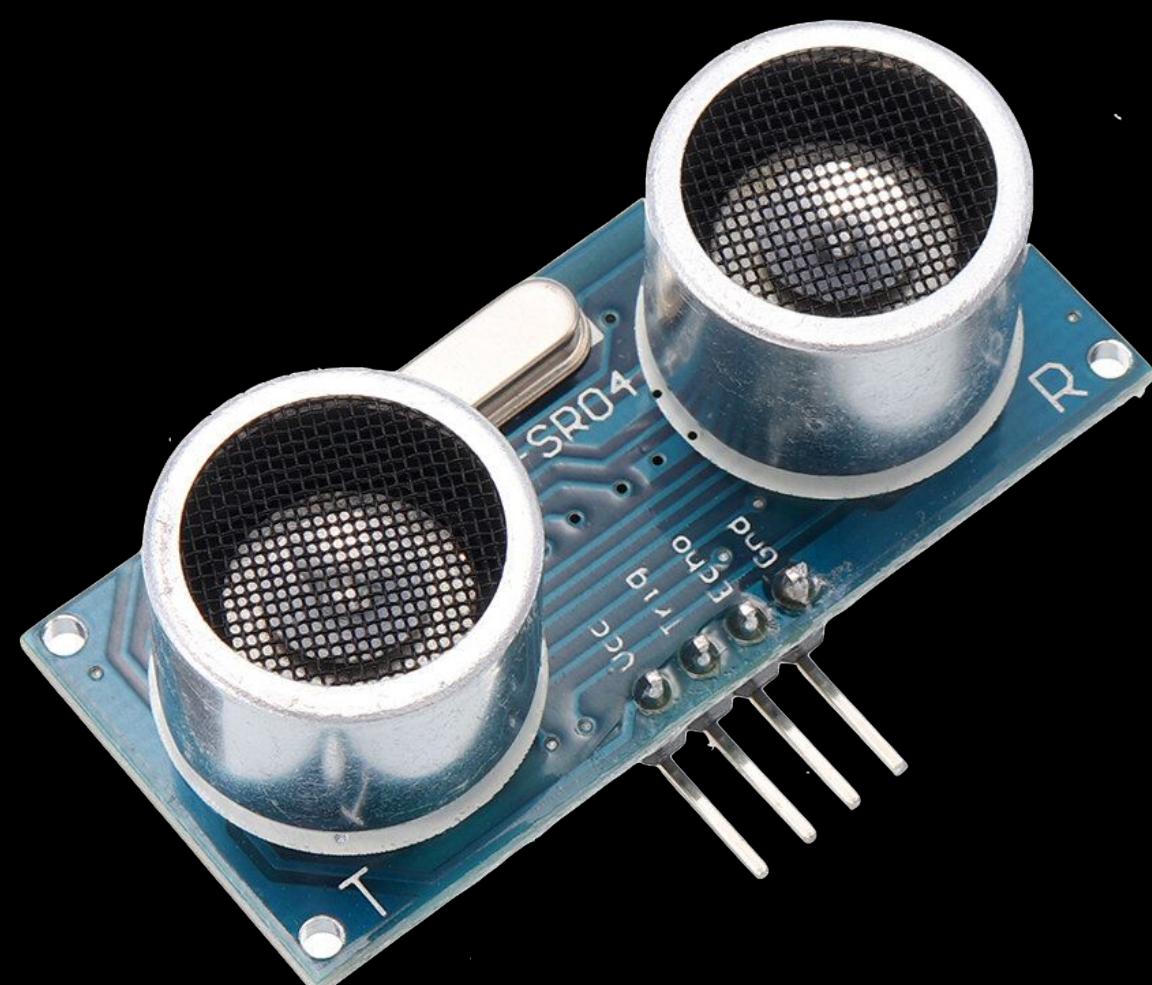
L298N motor driver module

- Supply power for FPGA.
 - Connect pin GND, +5v with external power header.
 - Set jumper JP2 to "EXT".



Materials overview (9/17)

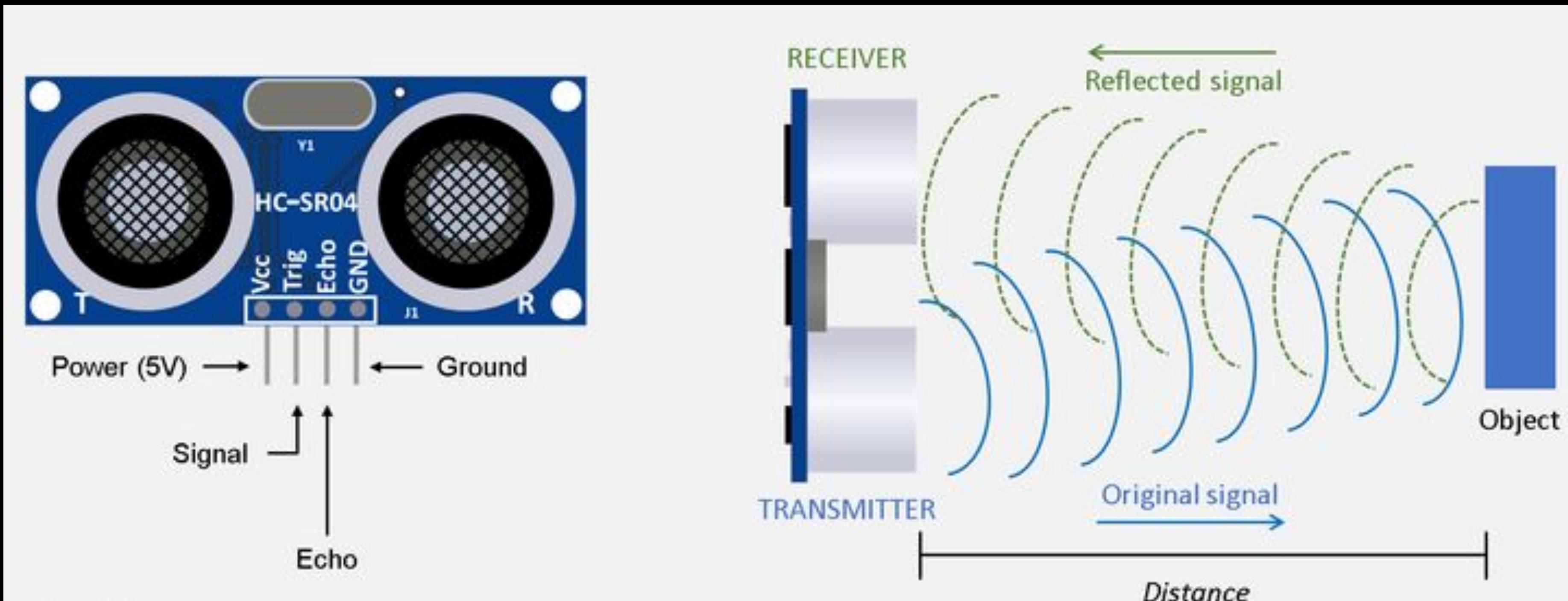
Ultrasonic sensor: HC-SR04



| Parameter | Value |
|---------------------|------------------------|
| Main Parts | Transmitter & Receiver |
| Technology Used | Non-Contact Technology |
| Operating Voltage | 5 V |
| Operating Frequency | 4 MHz |
| Detection Range | 2cm to 400cm |
| Measuring Angle | 30° |
| Resolution | 3mm |
| Operating Current | <15mA |
| Sensor Dimensions | 45mm x 20mm x 15mm |

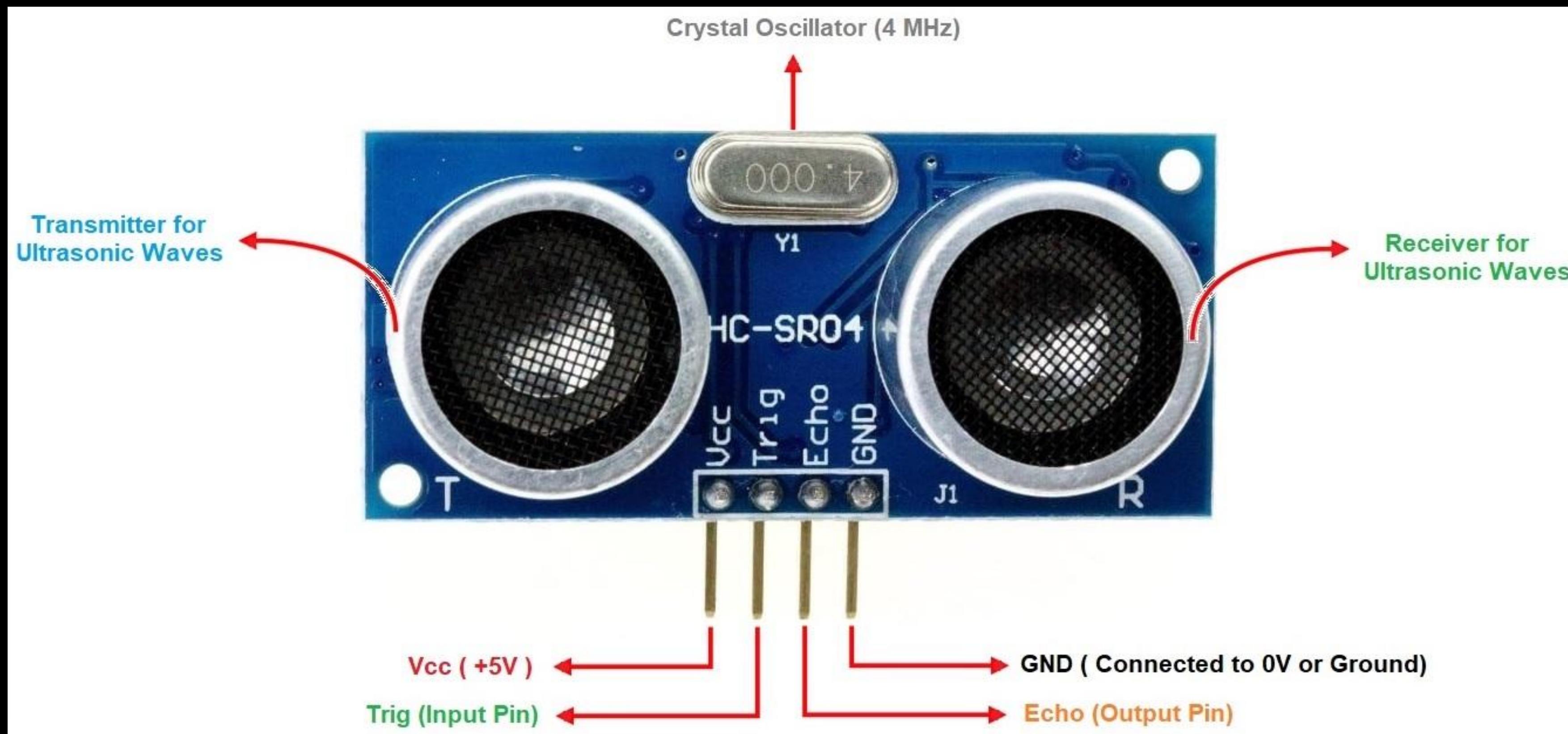
Materials overview (10/17)

HC-SR04



Materials overview (11/17)

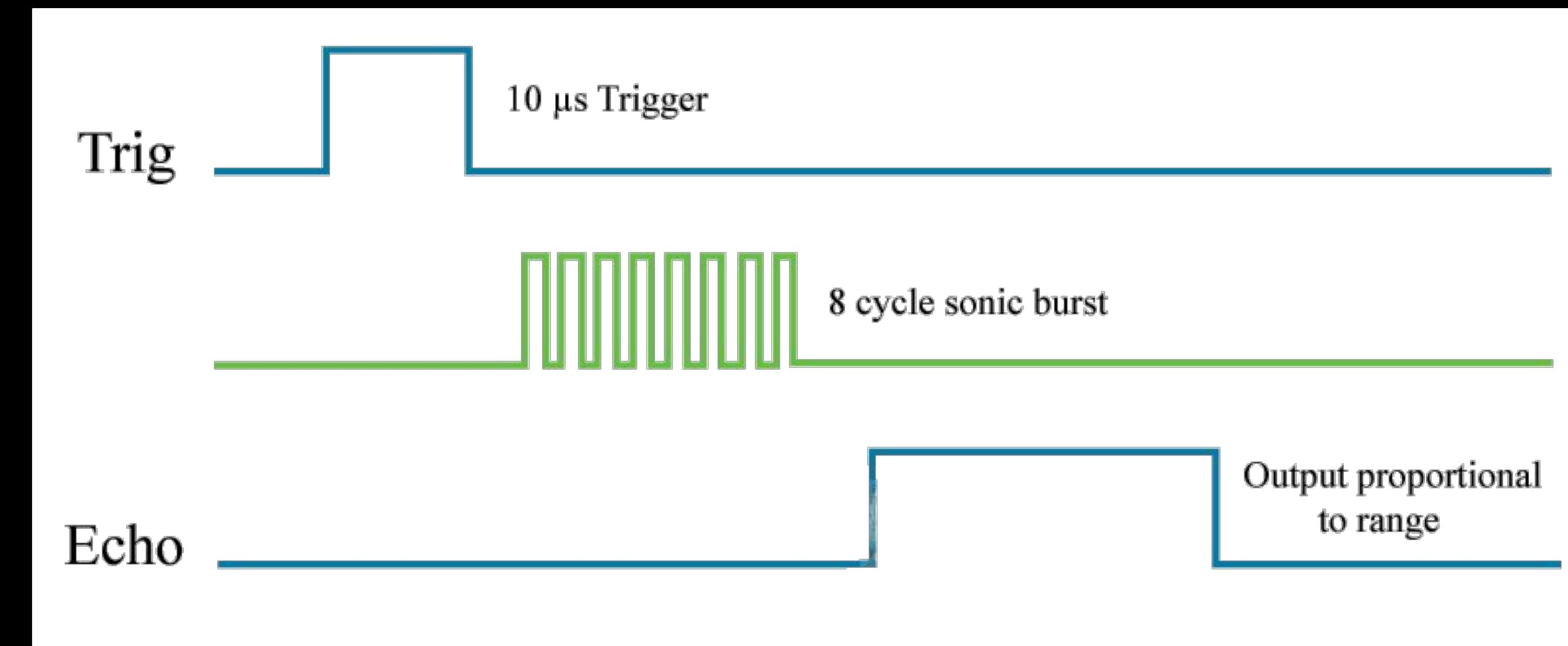
HC-SR04 Pinout



Materials overview (12/17)

HC-SR04

- To generate the ultrasound, set the Trig on a High State for 10 μs .
- Transmitter will send out 8 cycle sonic burst, and be received by Receiver.
- The Echo pin will output the time in microseconds the sound wave traveled.
- $S = t * v / 2$



Materials overview (13/17)

Code for HC-SR04 (sonic.v) :

```
module sonic_top(clk, rst, Echo, Trig, stop);
    input clk, rst, Echo;
    output Trig, stop;

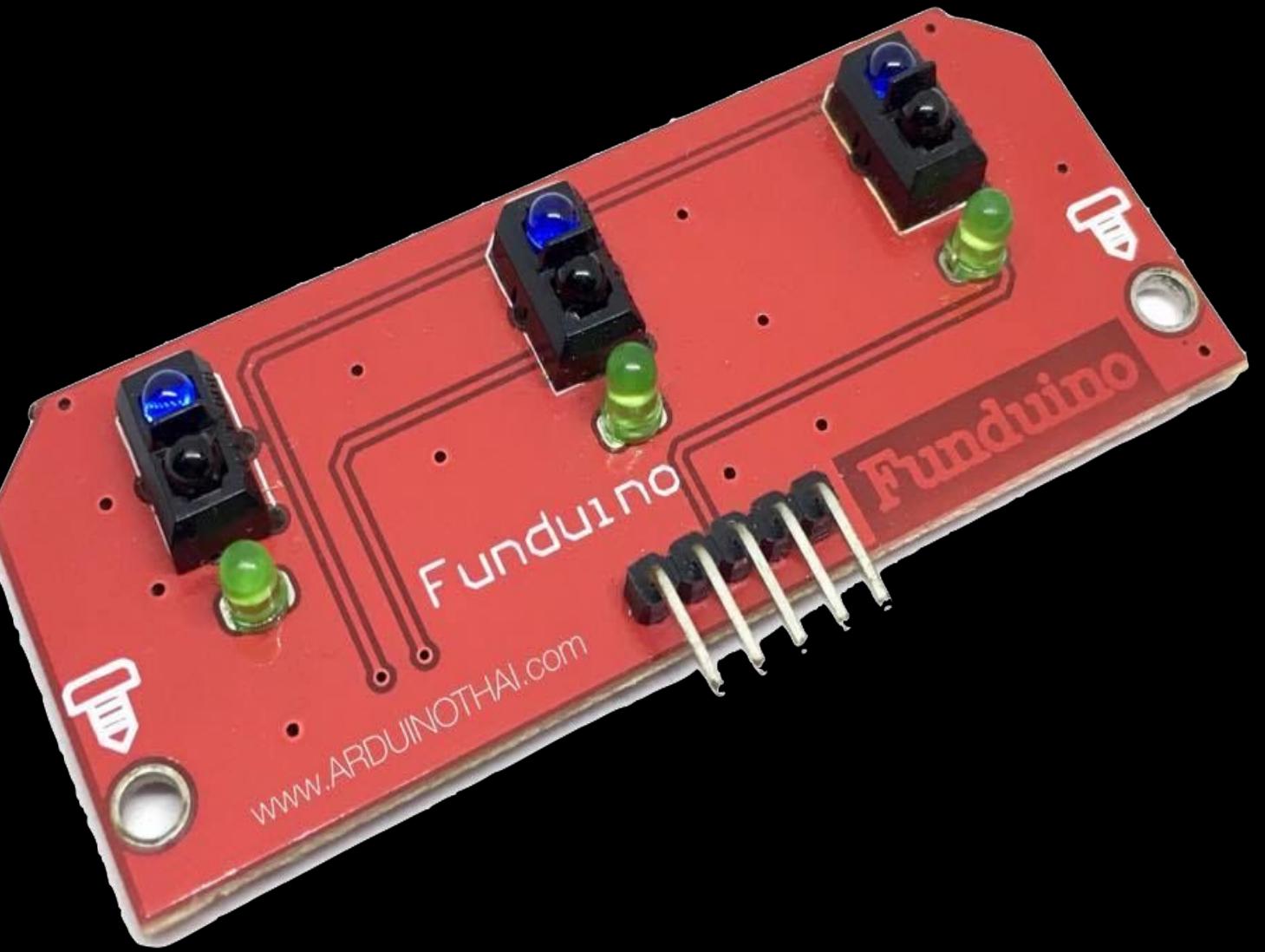
    wire[19:0] dis;
    wire[19:0] d;
    wire clk1M;
    wire clk_2_17;

    div clk1(clk ,clk1M);
    TrigSignal u1(.clk(clk), .rst(rst), .trig(Trig));
    PosCounter u2(.clk(clk1M), .rst(rst), .echo(Echo), .distance_count(dis));
```

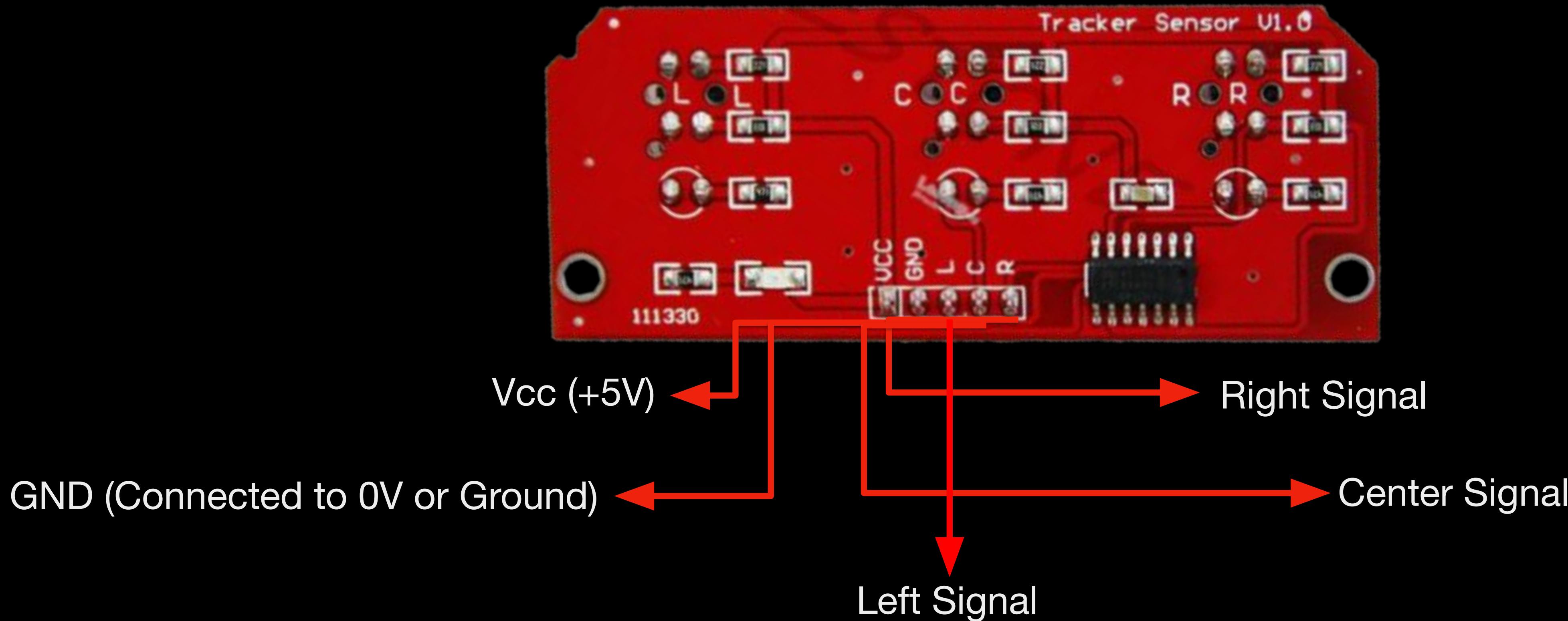
Materials overview (14/17)

3-Way Line Tracking IR : TCRT5000

- Include 3 TCRT5000 - IR Proximity Sensor
- Operating Voltage : 5V
- Output low black line, a white line output high



Materials overview (15/17)



Materials overview (16/17)

Document link

FEATURES

- Package type: leaded
- Detector type: phototransistor
- Dimensions (L x W x H in mm): 10.2 x 5.8 x 7
- Peak operating distance: 2.5 mm
- Operating range within > 20 % relative collector current: 0.2 mm to 15 mm
- Typical output current under test: $I_C = 1 \text{ mA}$
- Daylight blocking filter
- Emitter wavelength: 950 nm
- Lead (Pb)-free soldering released
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC



RoHS
COMPLIANT

Materials overview (17/17)

Code for Line Tracking IR (tracker_sensor.v) :

```
module tracker_sensor(clk, reset, left_signal, right_signal, mid_signal, state);
    input clk;
    input reset;
    input left_signal, right_signal, mid_signal;
```

Agenda

Introduction

Materials overview

FPGA configuration

Grading

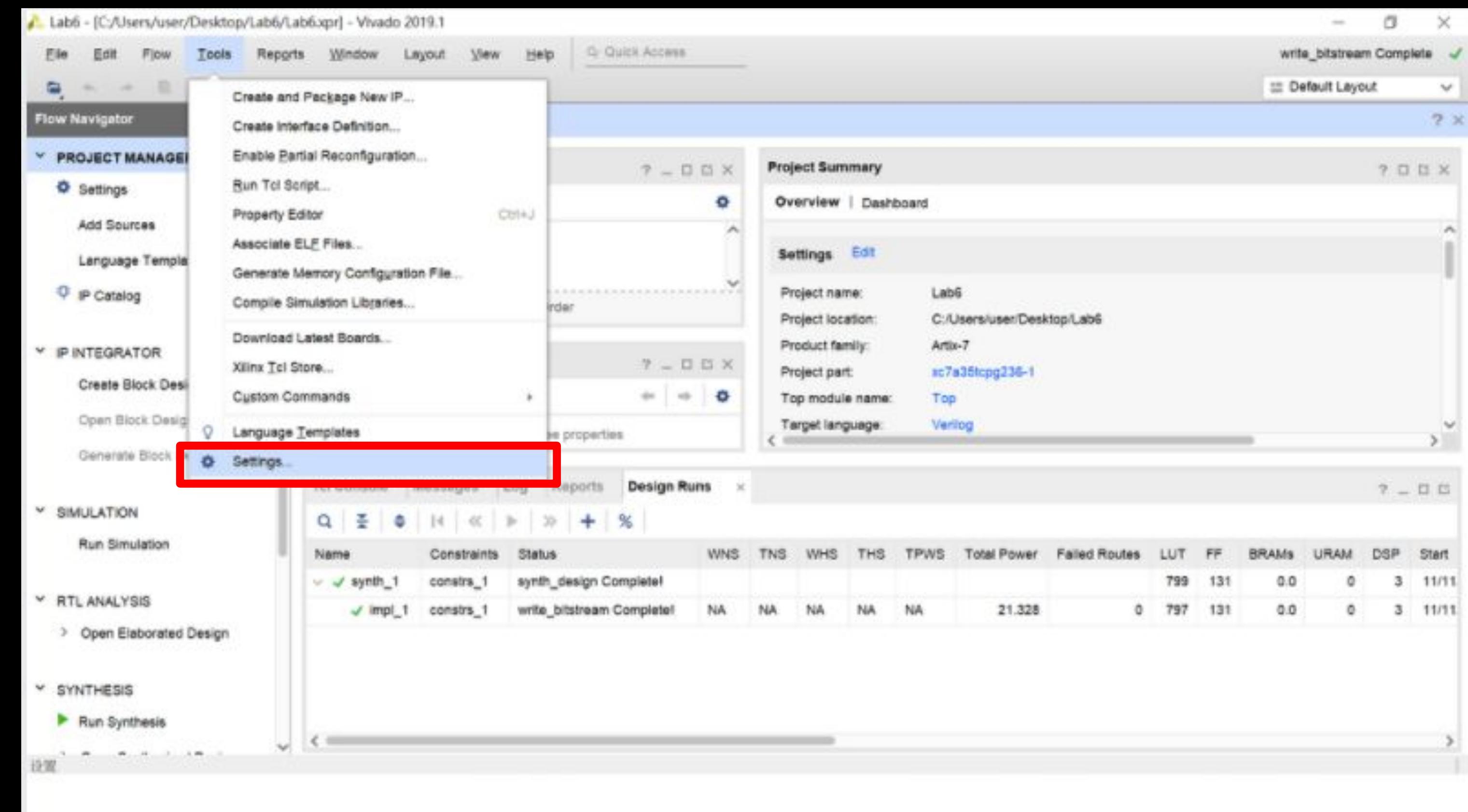
FPGA configuration (1/8)

- Bitstream Configuration
- Flash Memory Setting

FPGA configuration (2/8)

Bitstream Configuration

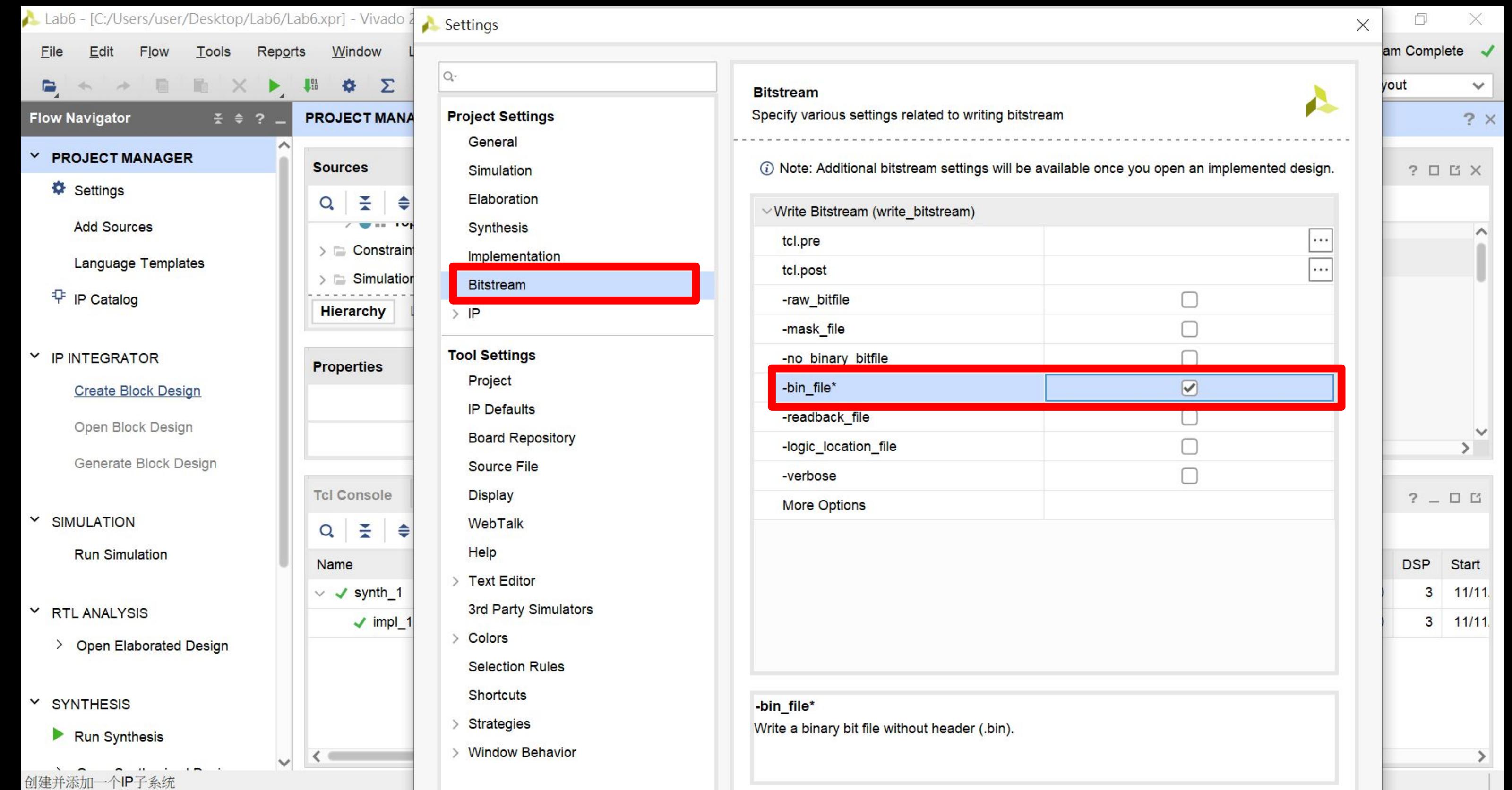
- Tools -> Settings



FPGA configuration (3/8)

Bitstream Configuration

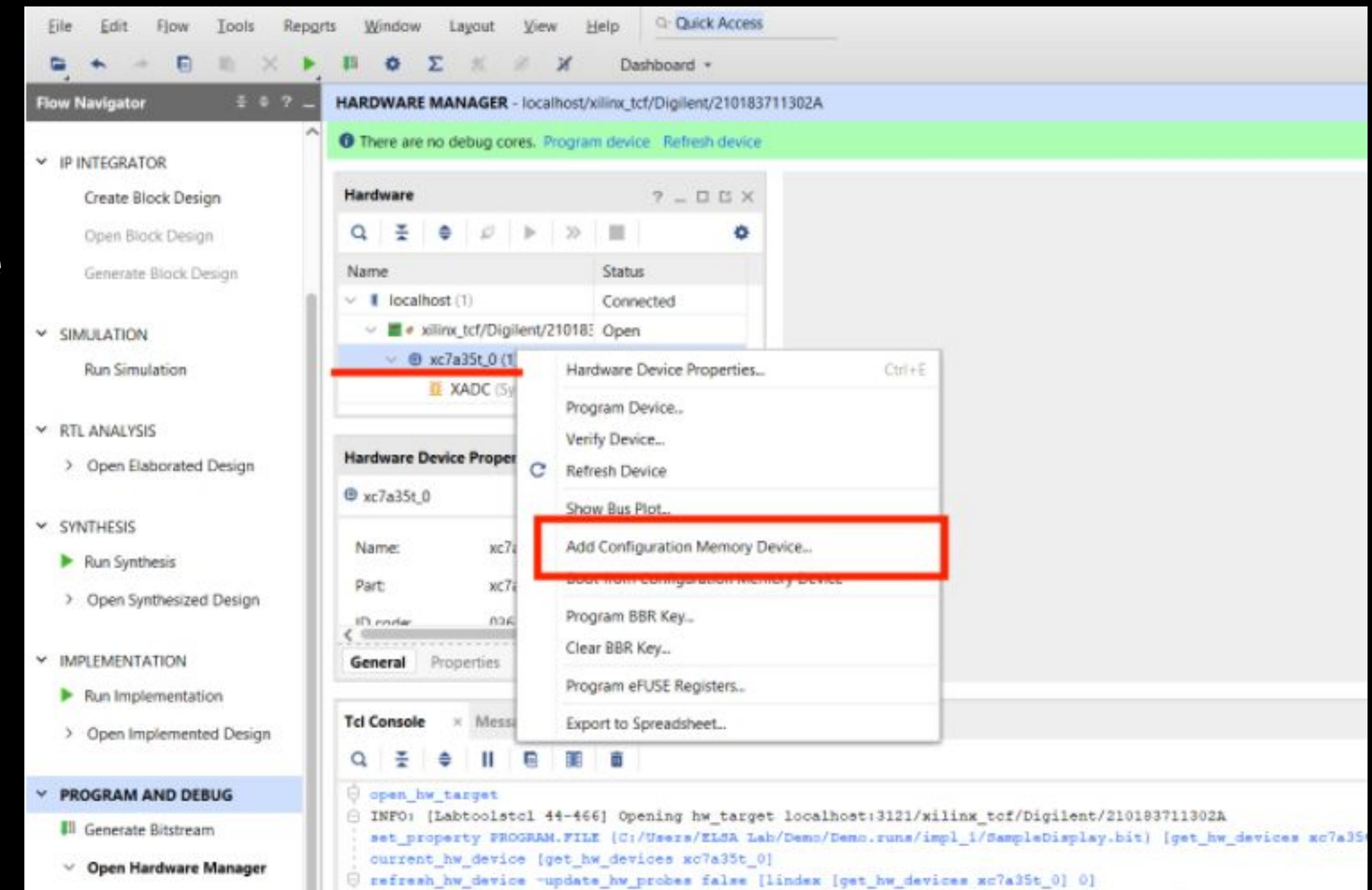
- Bitstream -> bin_file
- Apply -> OK



FPGA configuration (4/8)

Flash Memory Setting

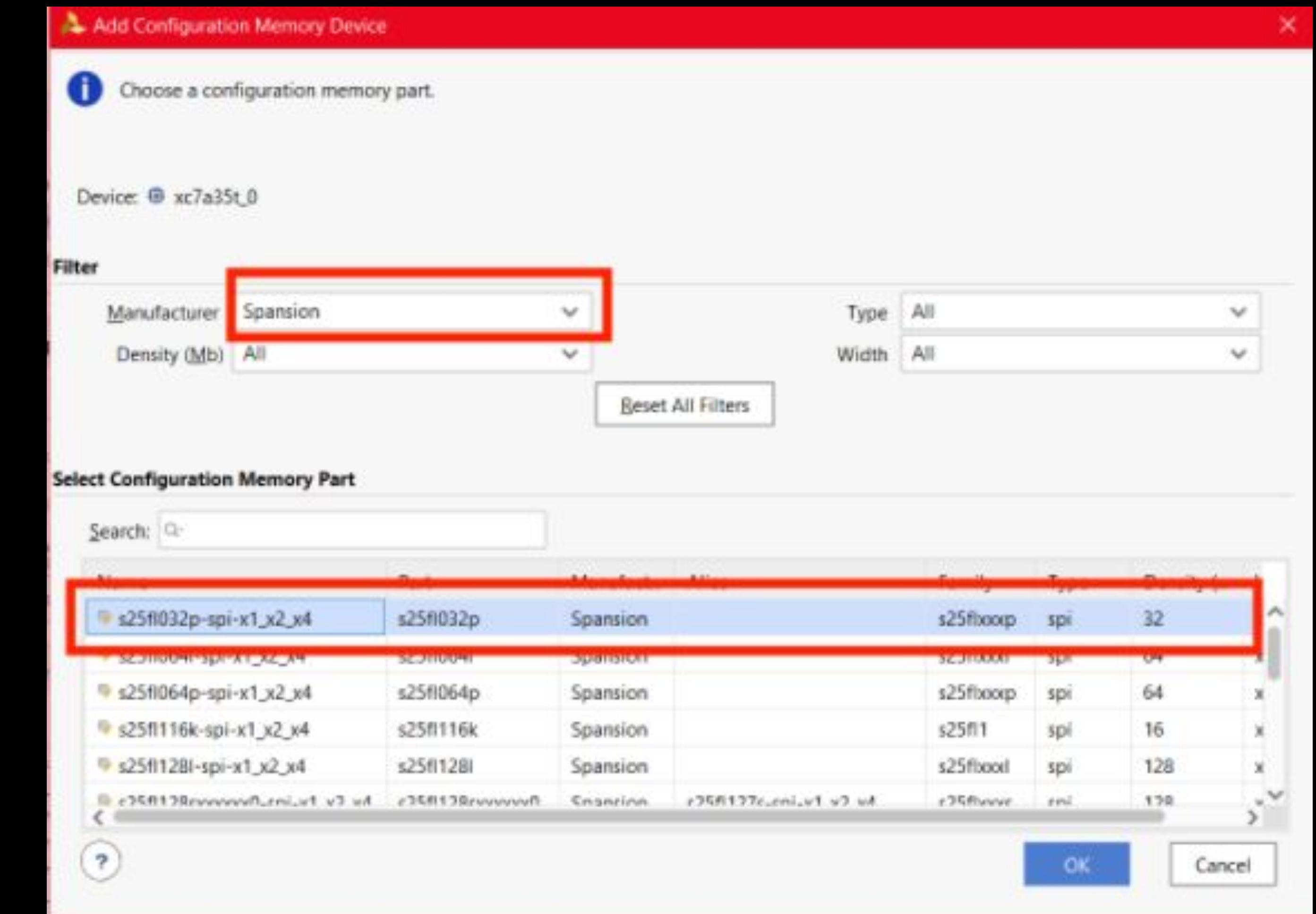
- Hardware Manager ->
Add Configuration Memory Device
- If you have previous configuration
memory, you need to delete it first



FPGA configuration (5/8)

Flash Memory Setting

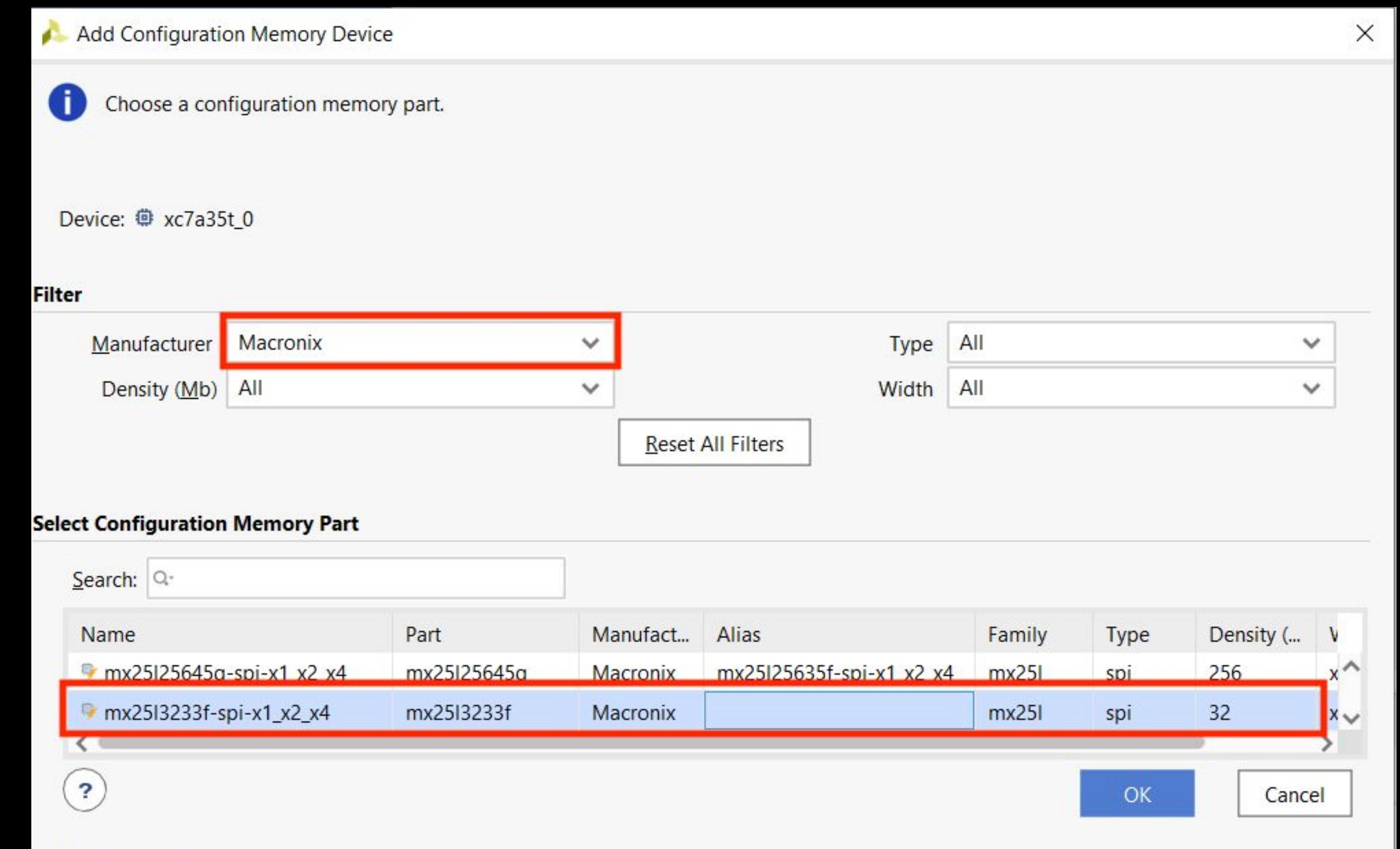
- [Filter] -> [manufactureer]
-> [Spansion]
- [Configuration Memory Part]
-> [s25fl032p]



FPGA configuration (5/8) - new board

Flash Memory Setting

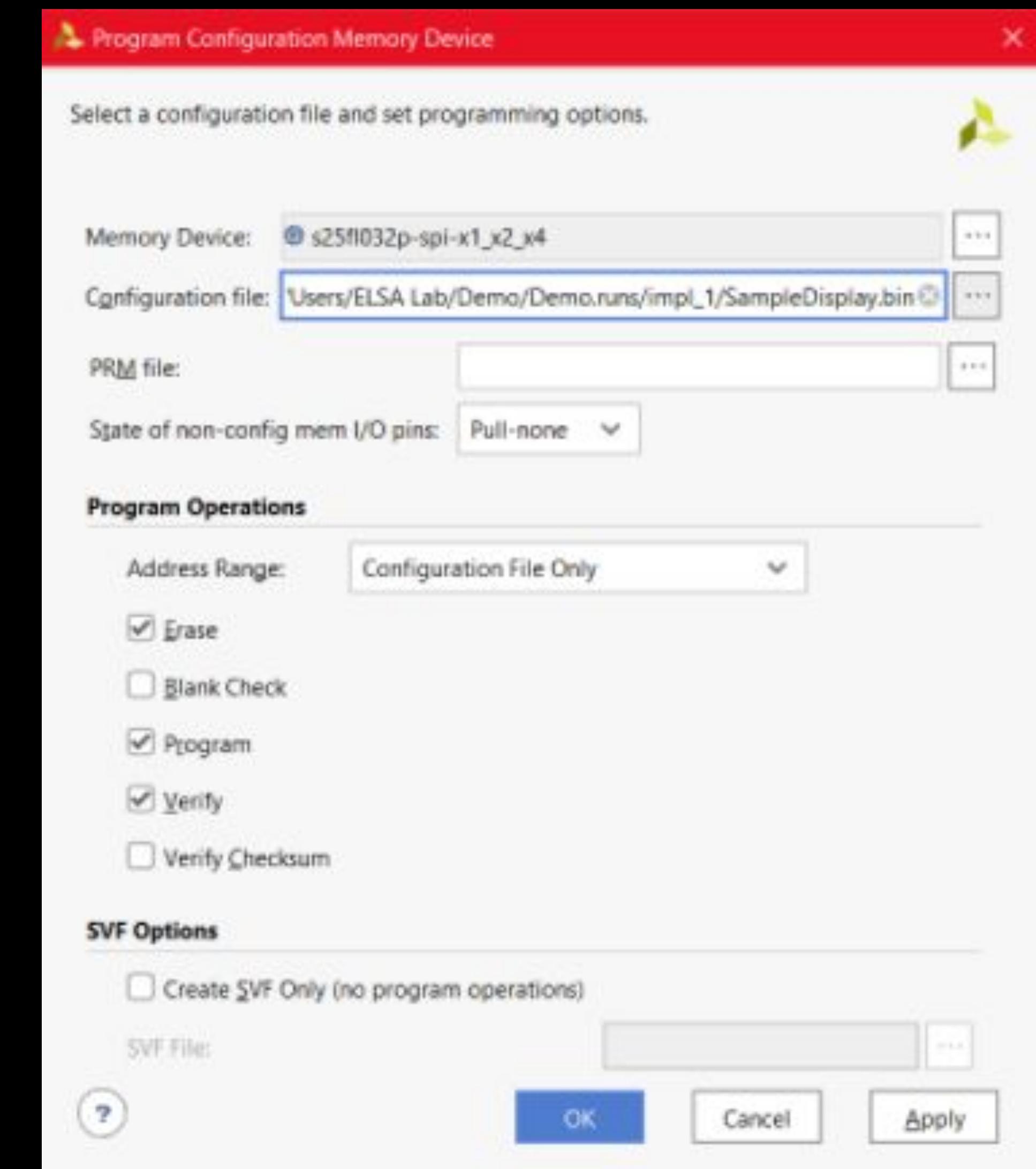
- [Filter] -> [manufactureer]
-> [Macronic]
- [Configuration Memory Part]
-> [mx25l3233f]



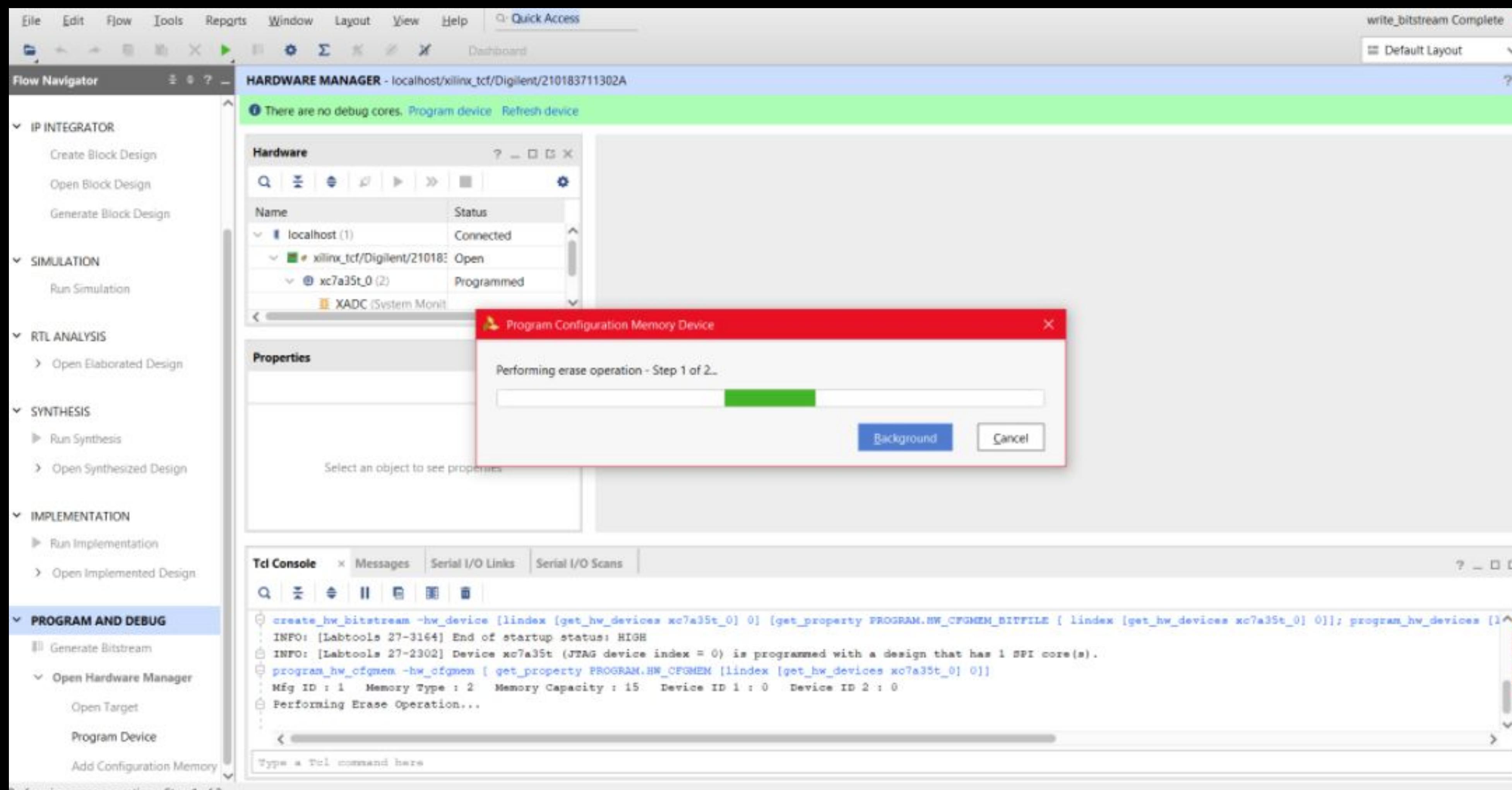
FPGA configuration (6/8)

Flash Memory Setting

- Select configuration file :
Project_Name.runs/impl_1/**.bin
- Not .bit nor .mcs
- Apply -> OK



FPGA configuration (7/8)



FPGA configuration (8/8)

Make sure your

- JP1 is on **FLASH** mode
- JP2 is on **EXTERNAL**



Agenda

Introduction

Materials overview

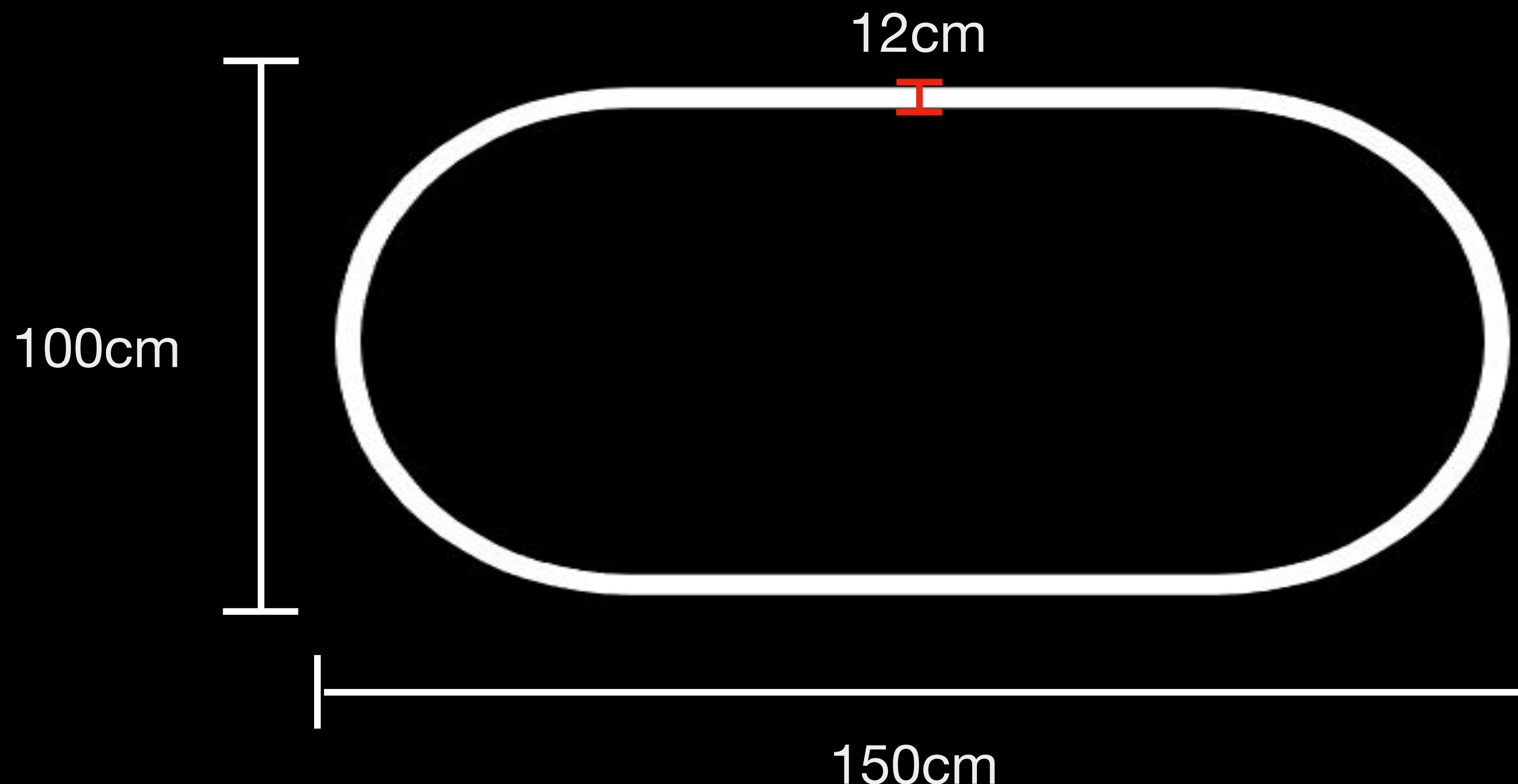
FPGA configuration

Grading

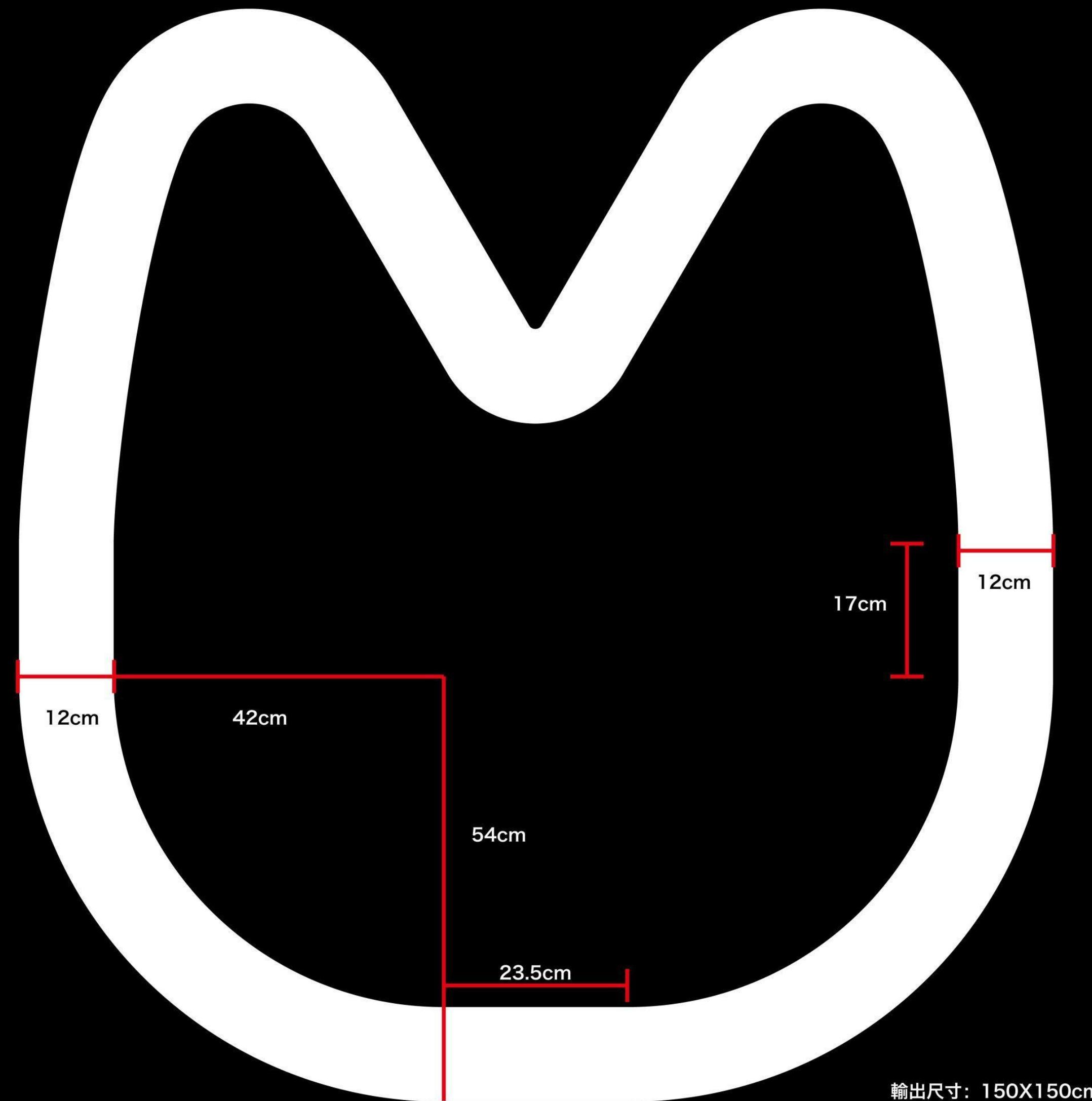
Grading (1/4)

- Use sonic sensor to detect the distance.
If distance < 40cm, you need to stop the car.
- Make sure your car can turn right and left successfully.
- We will have two basic tracks, and one bonus track.

Grading (2/4)



Grading (3/4)



Grading (4/4)

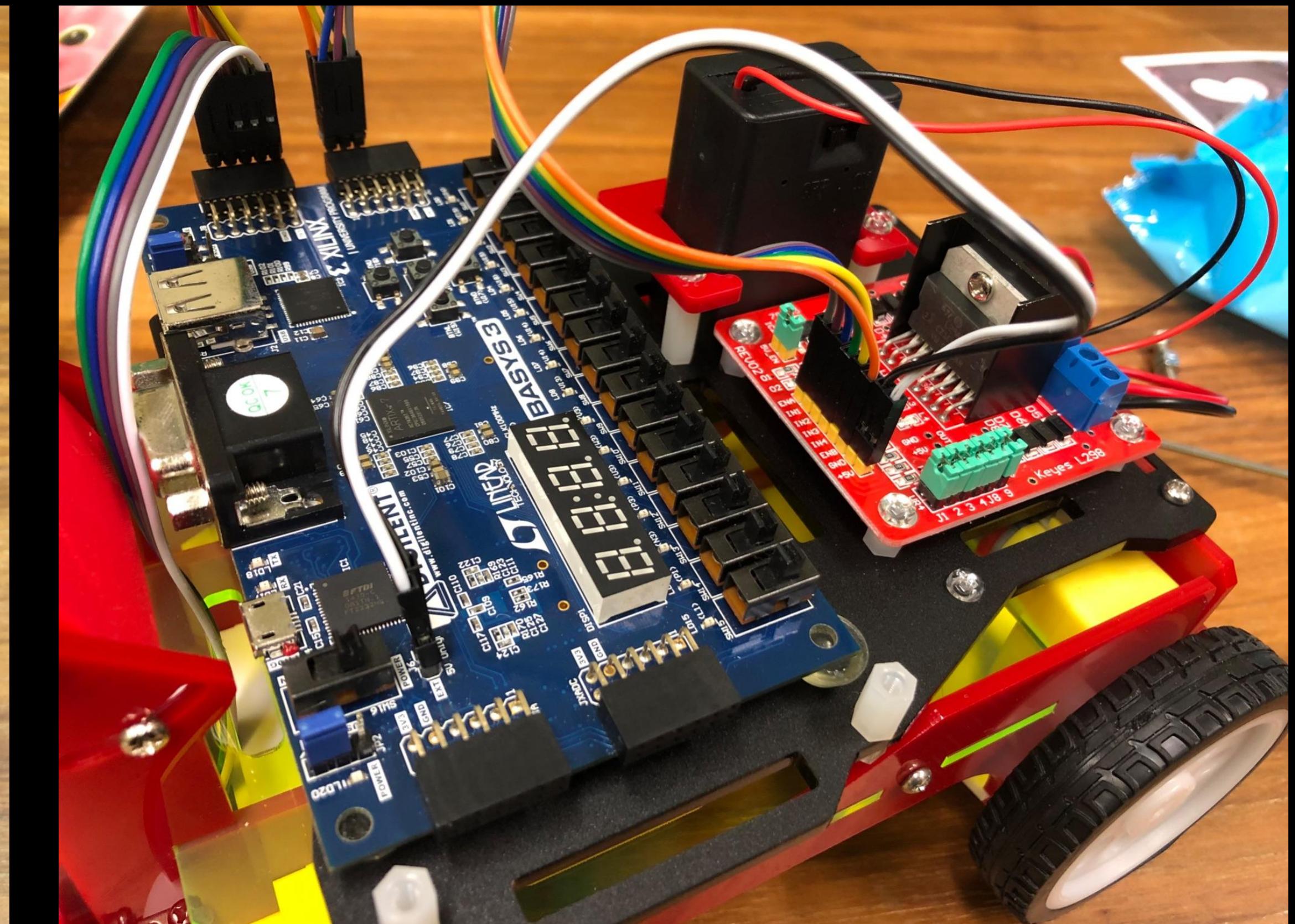
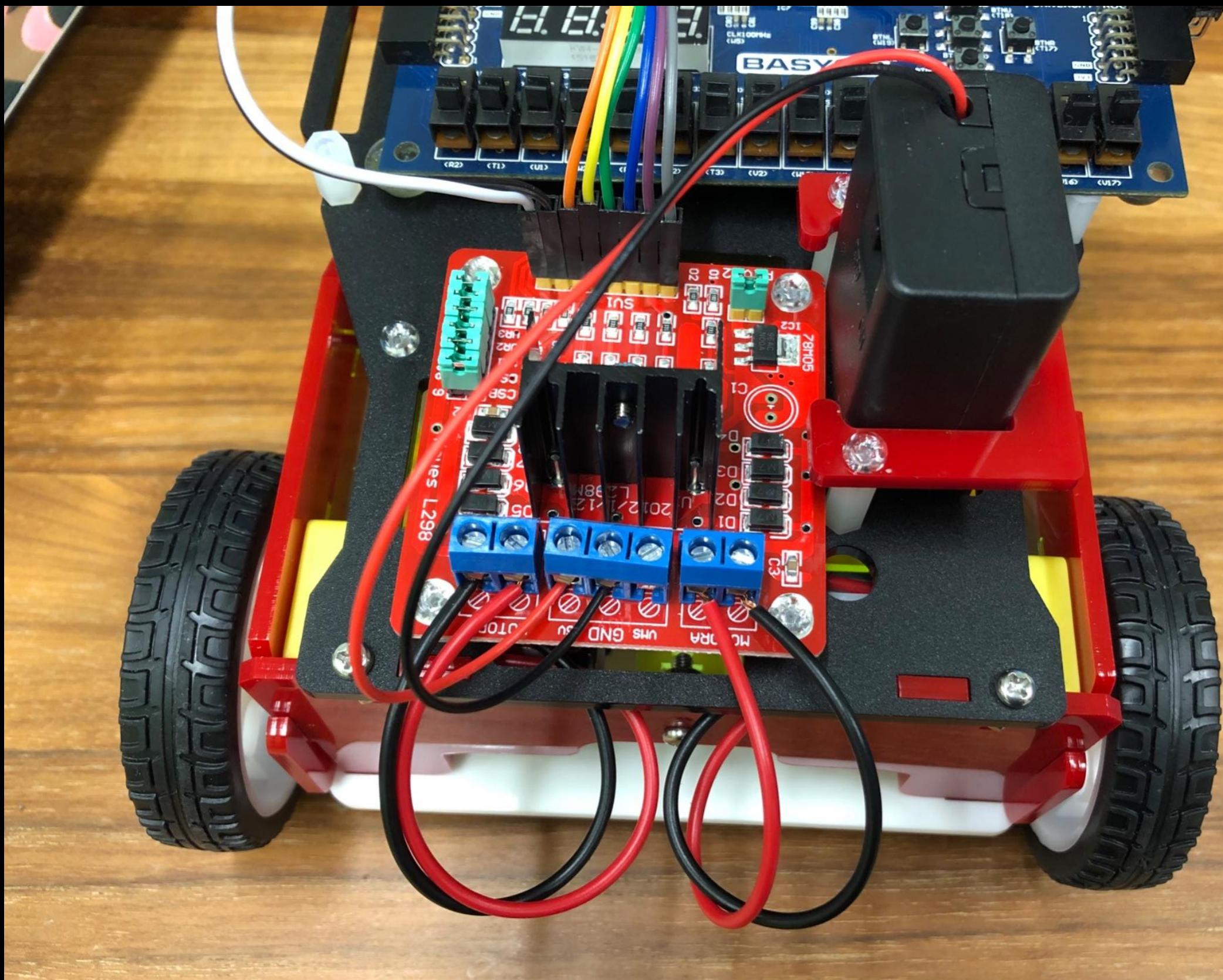
Bonus track

- Be careful on a **sharp turn**.
- You **don't** need to handle Square Corner.
- You **don't** need to reverse your car.
- Go straight if there is an intersection.
- Bonus track will first test its **correctness**,
and then test its **speed**.

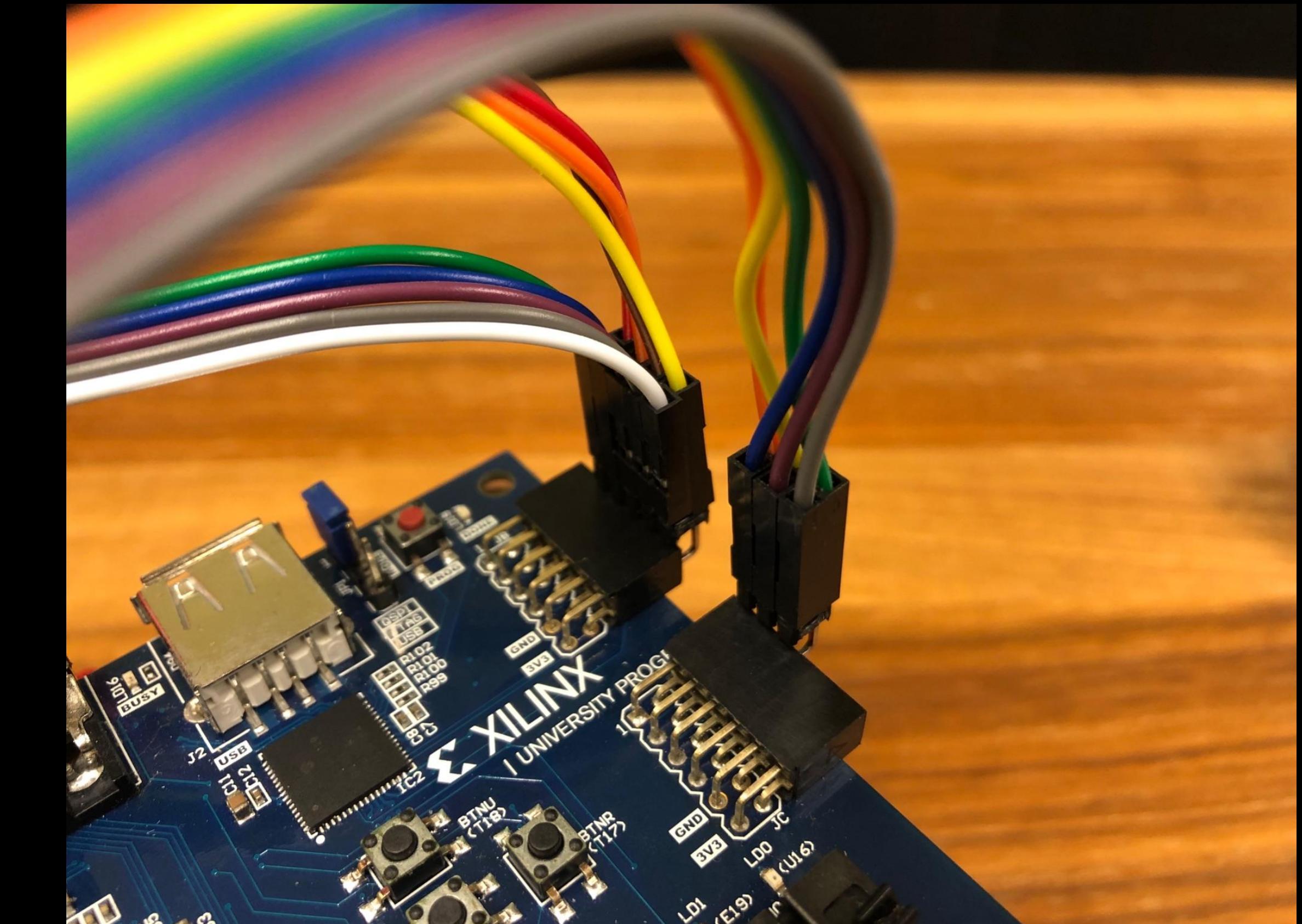
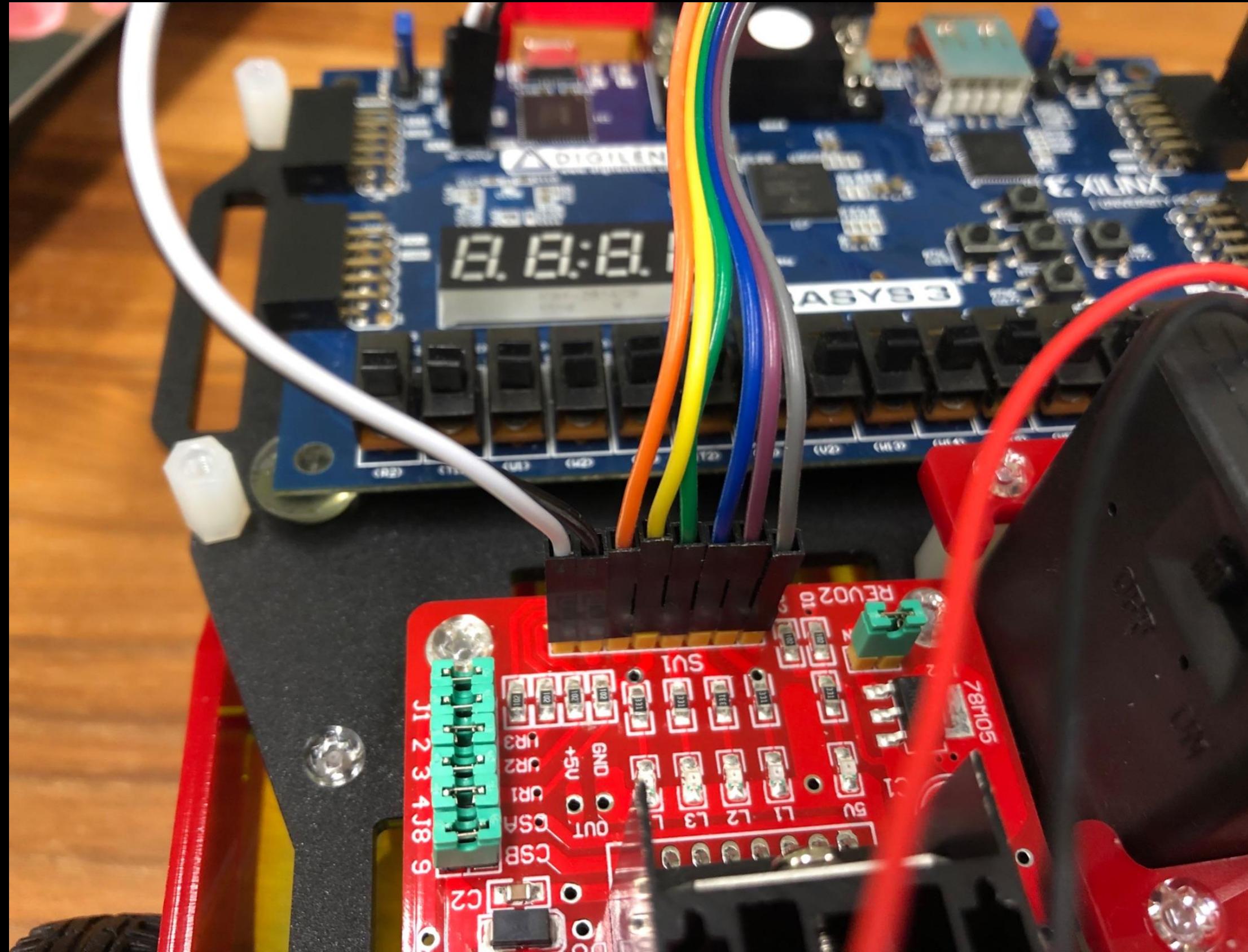


Q&A

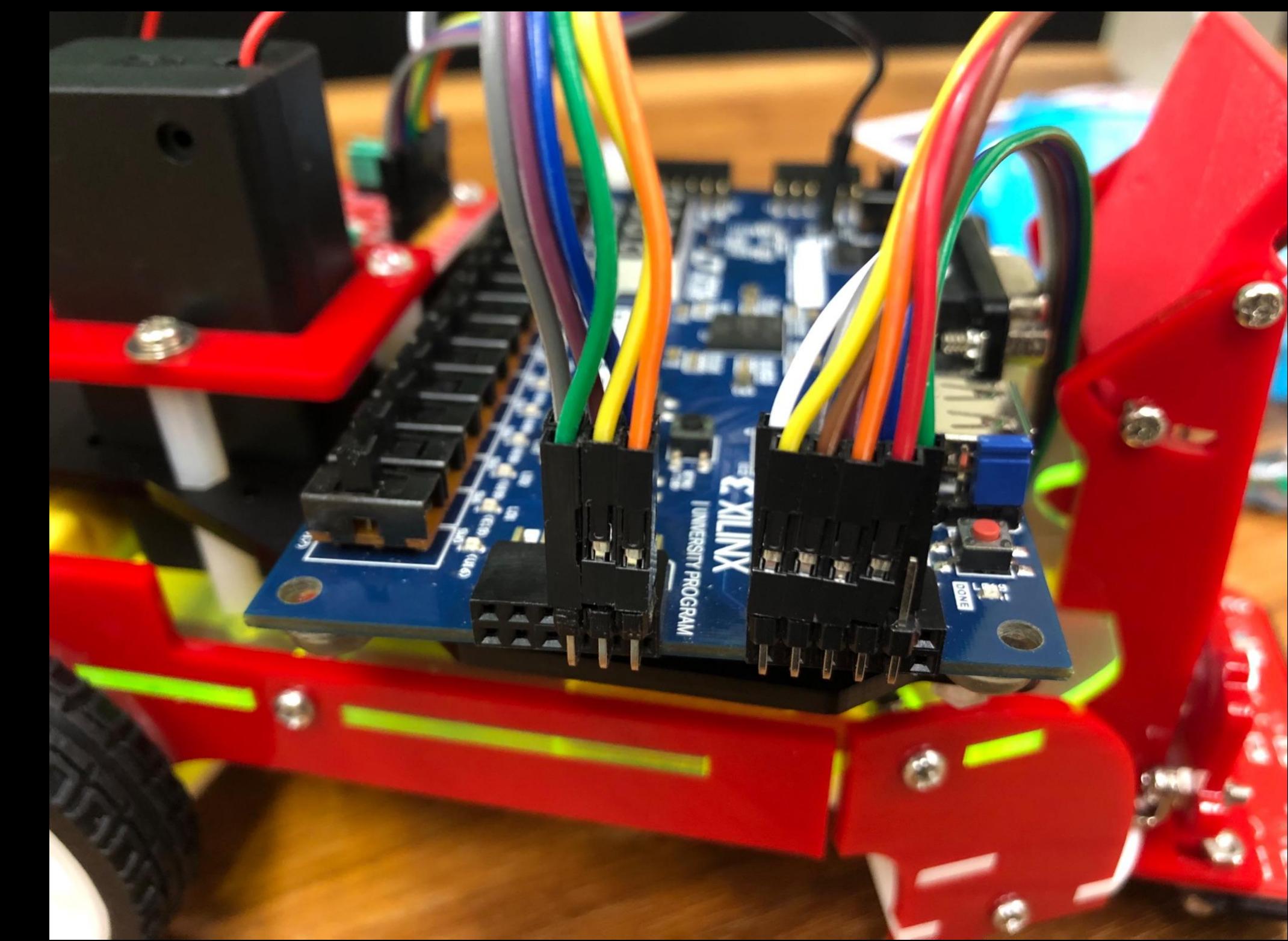
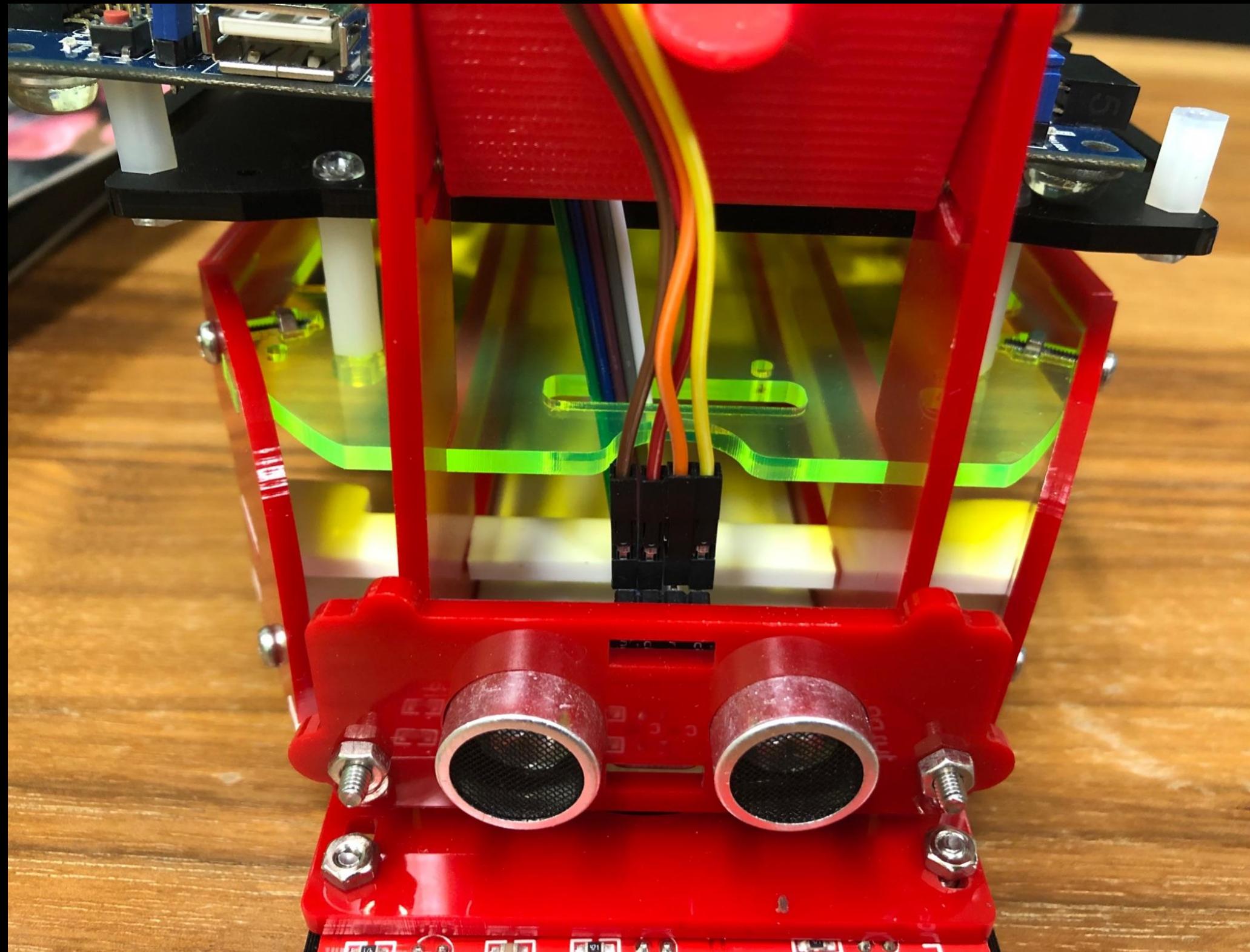
- connect the battery to motor
- supply to FPGA



- connect motor driver and FPGA



- connect HC-SR04



- connect line tracker

