Lab6 - Line Following Car

NTHU Hardware Design and Laboratory (Fall 2024)

11/12/2024

By Prof. Chun-Yi Lee

Agenda

Introduction

Materials overview

FPGA configuration

Grading

Agenda

Introduction

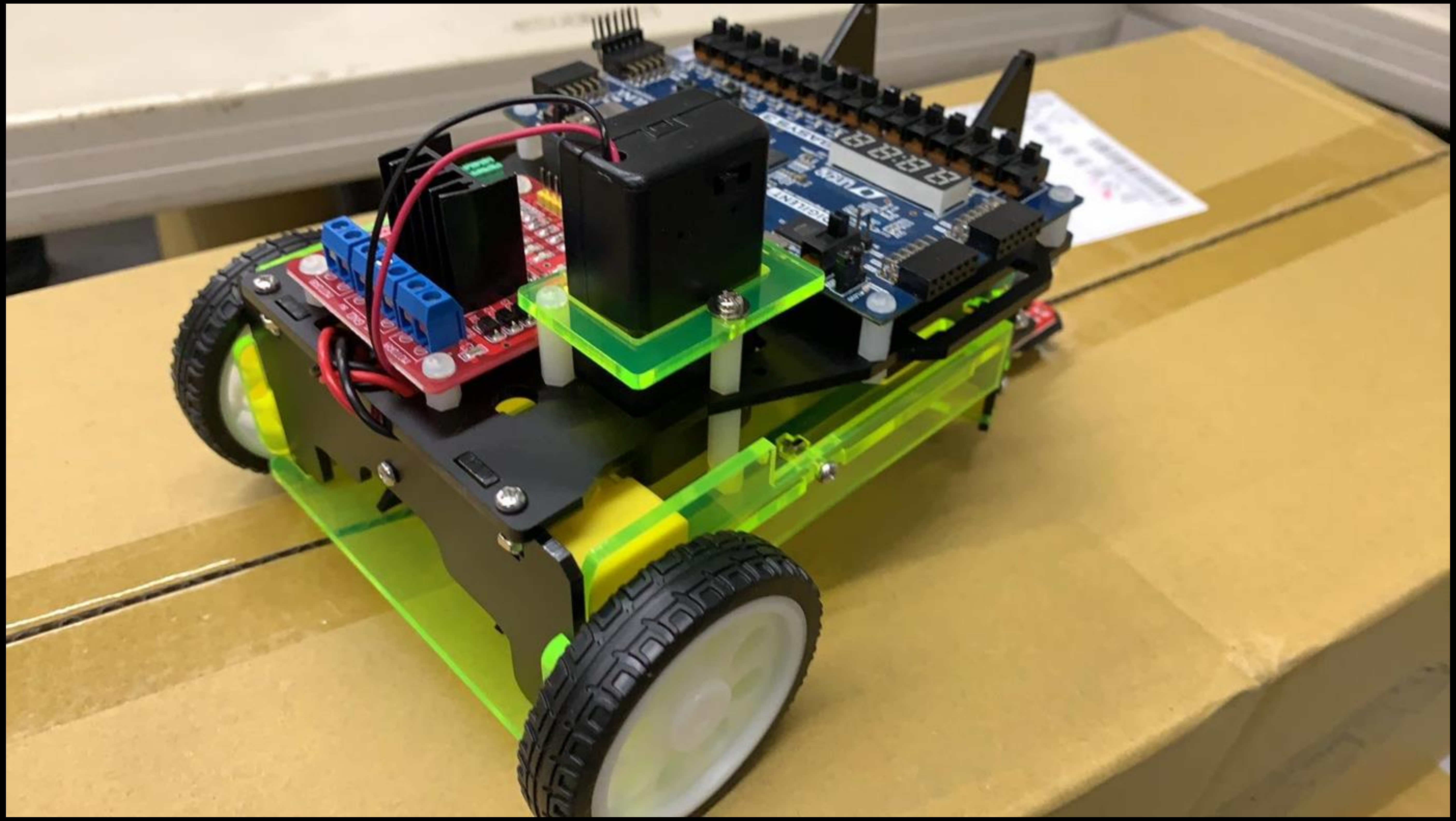
Materials overview

FPGA configuration

Grading

Introduction (1/1)

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



Agenda

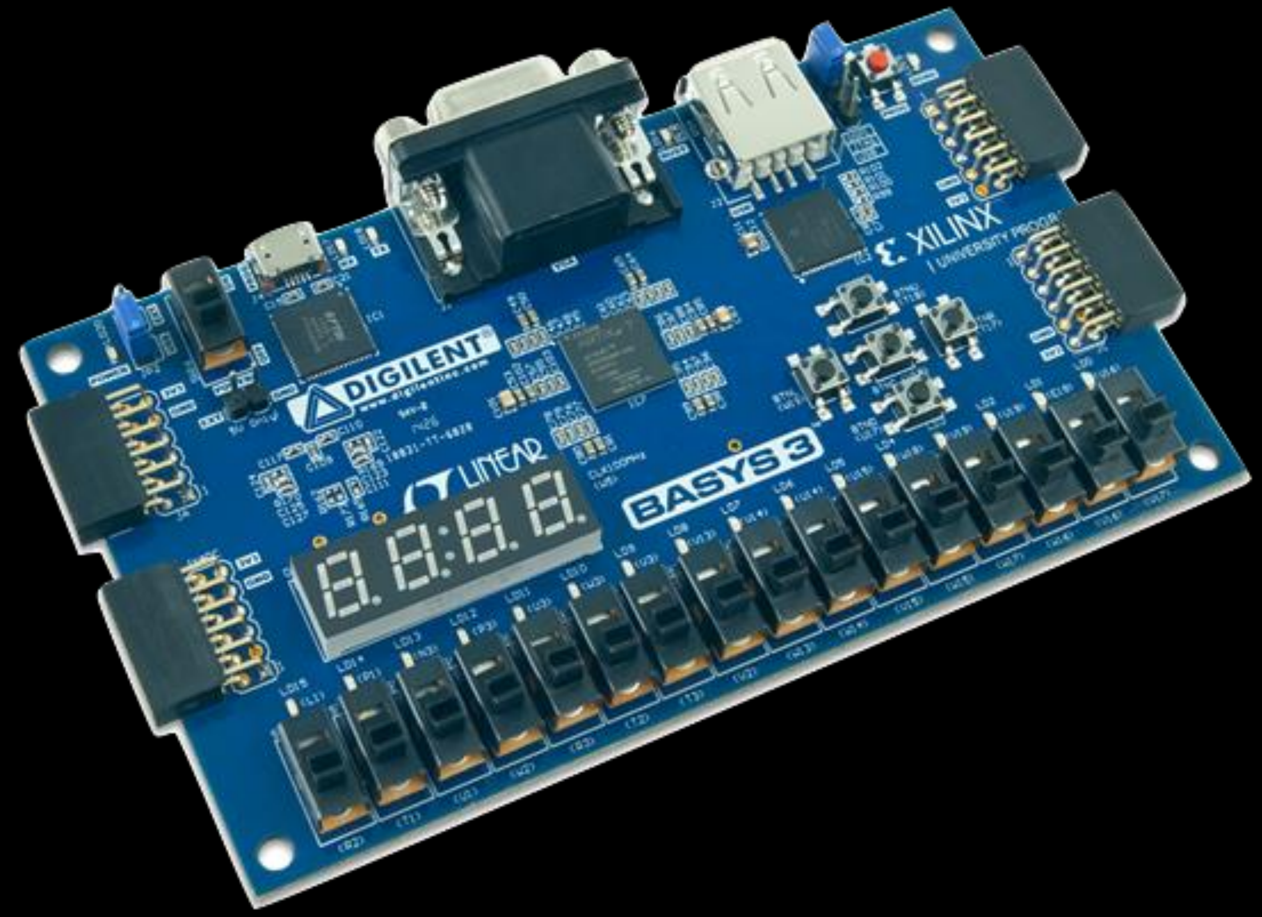
Introduction

Materials overview

FPGA configuration

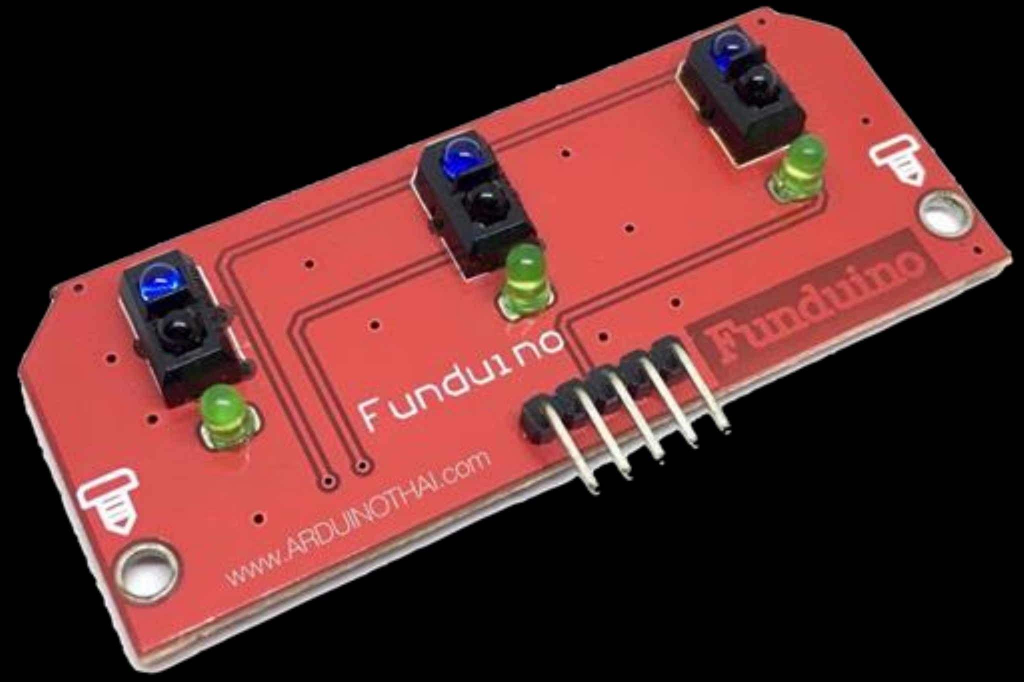
Grading

Materials overview (1/17)

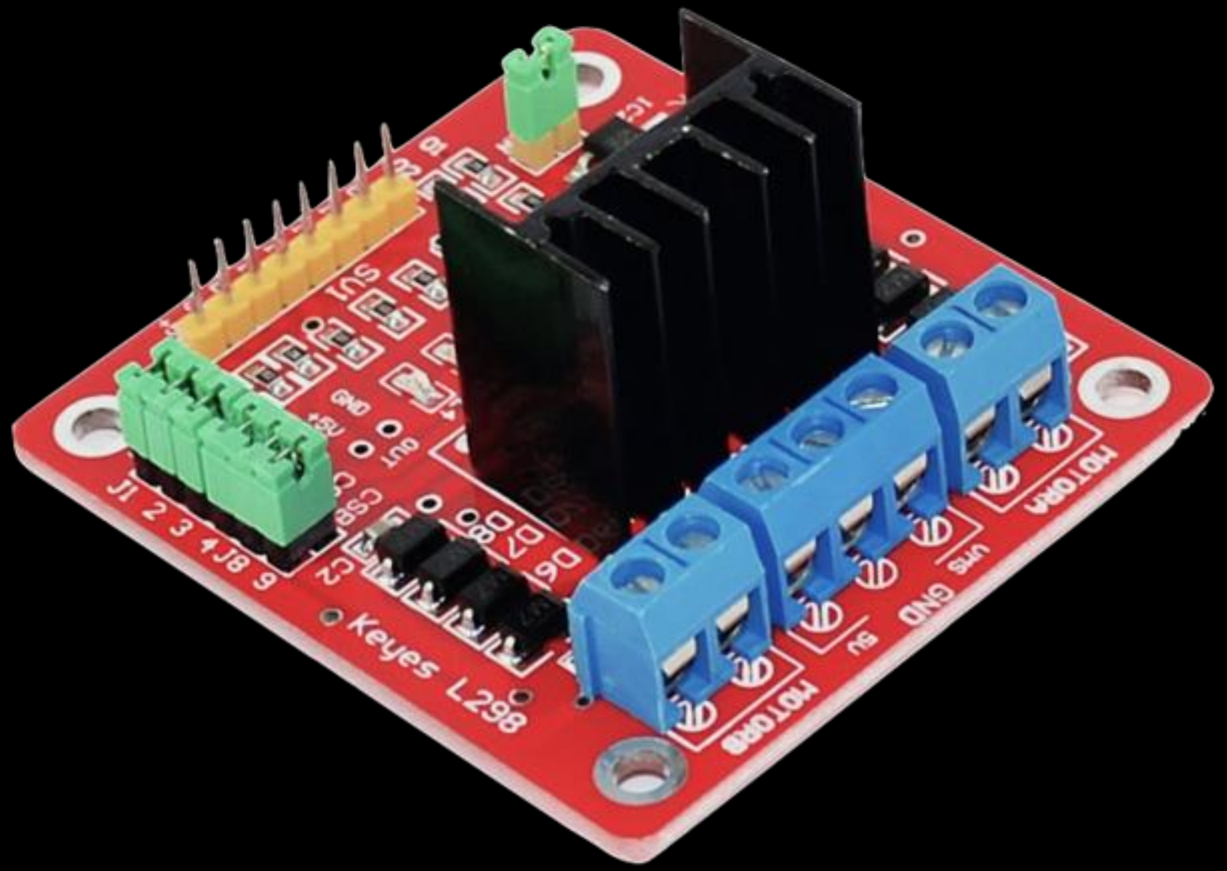


FPGA

HC-SR04



TCRT5000
3-way line tracking IR



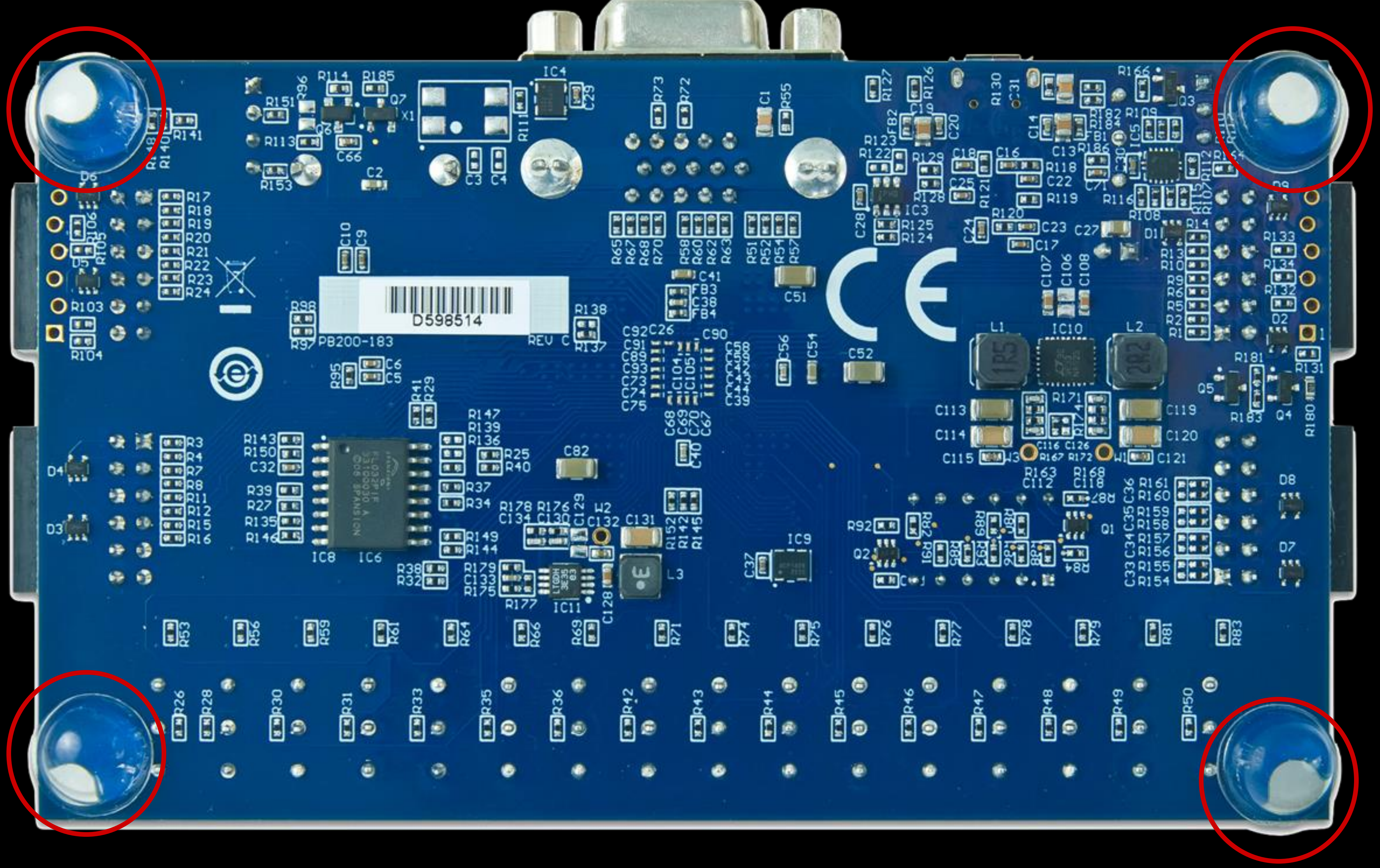
L298N motor driver module

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)

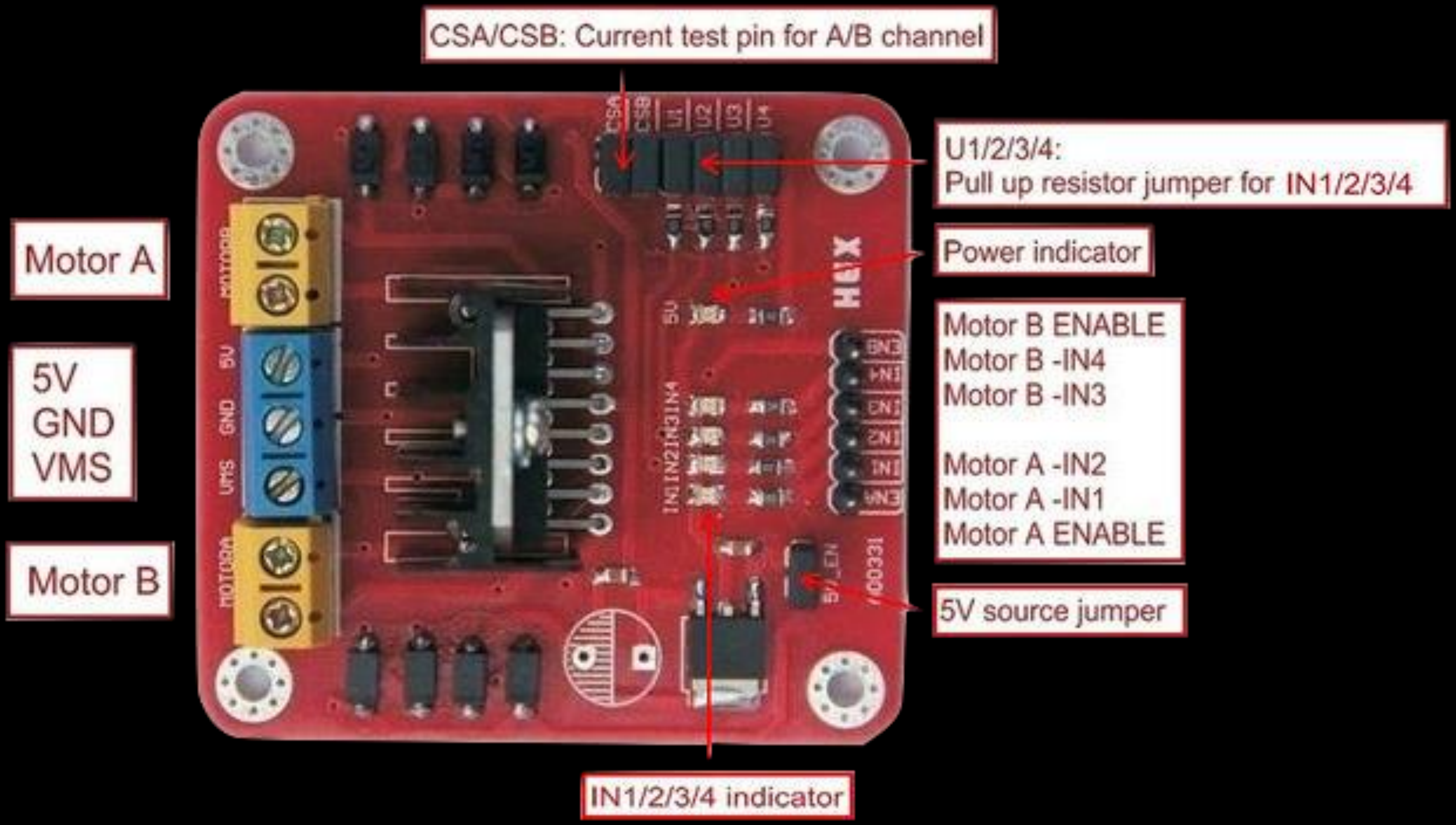
Gear motor + L298N motor driver module

- 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

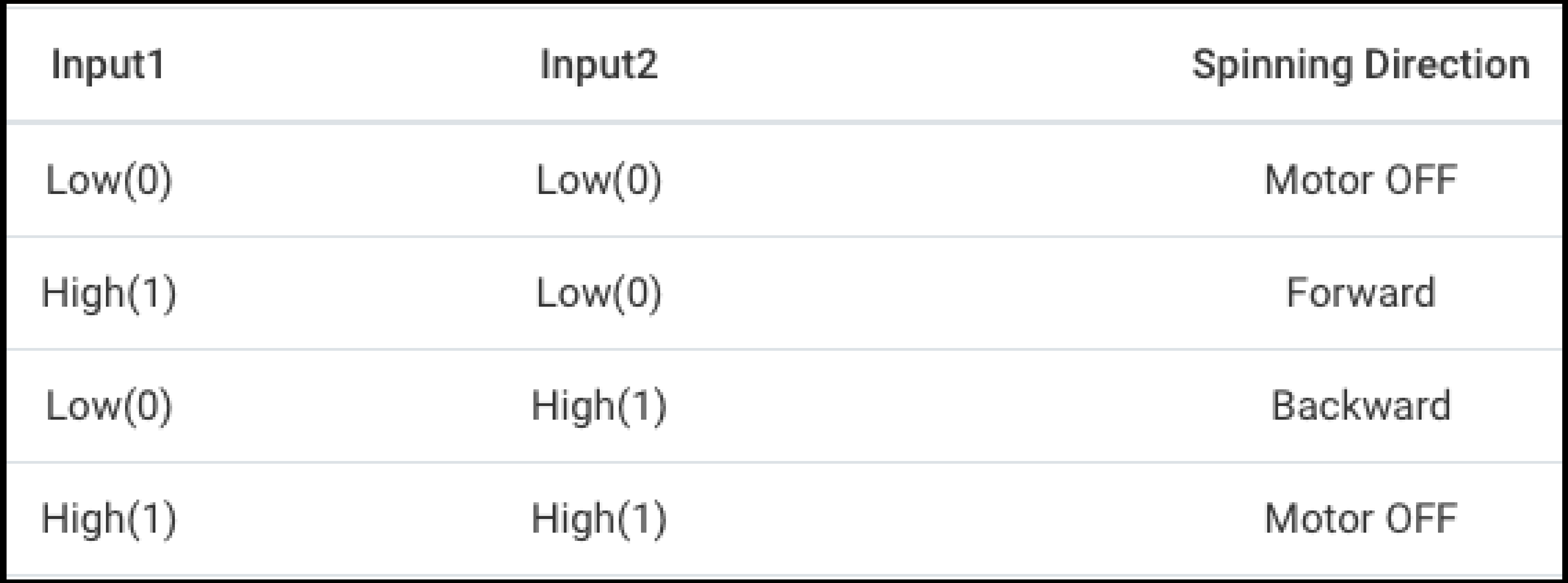
- Able to control 2 motors.
- VMS for motor.
- 5V pin which can either be an input or output.



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.

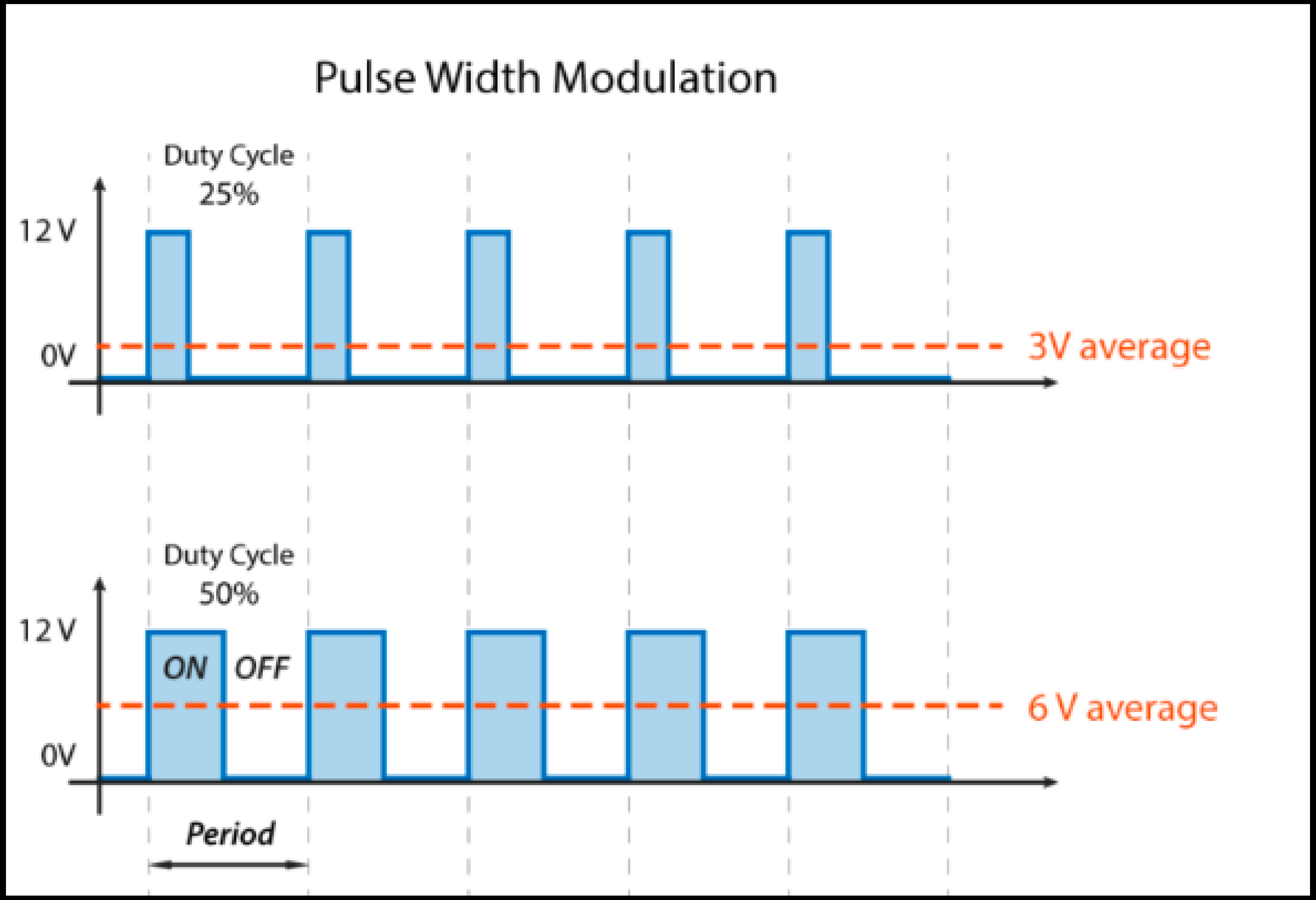


Input1	Input2	Spinning Direction
Low(0)	Low(0)	Motor OFF
High(1)	Low(0)	Forward
Low(0)	High(1)	Backward
High(1)	High(1)	Motor OFF

Materials overview (6/17)

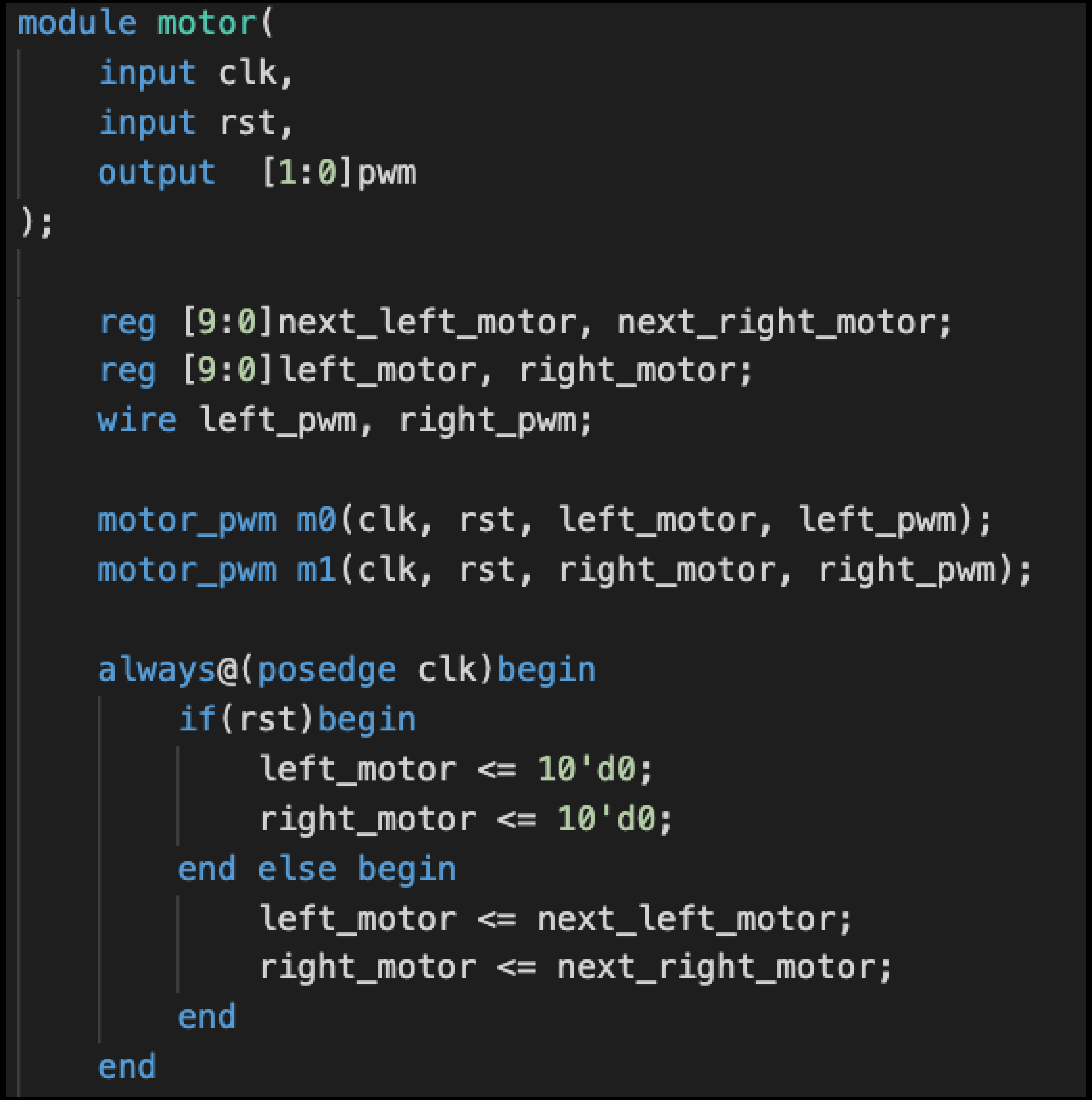
L298N motor driver module

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



Materials overview (7/17)

Code for motor (motor.v) :

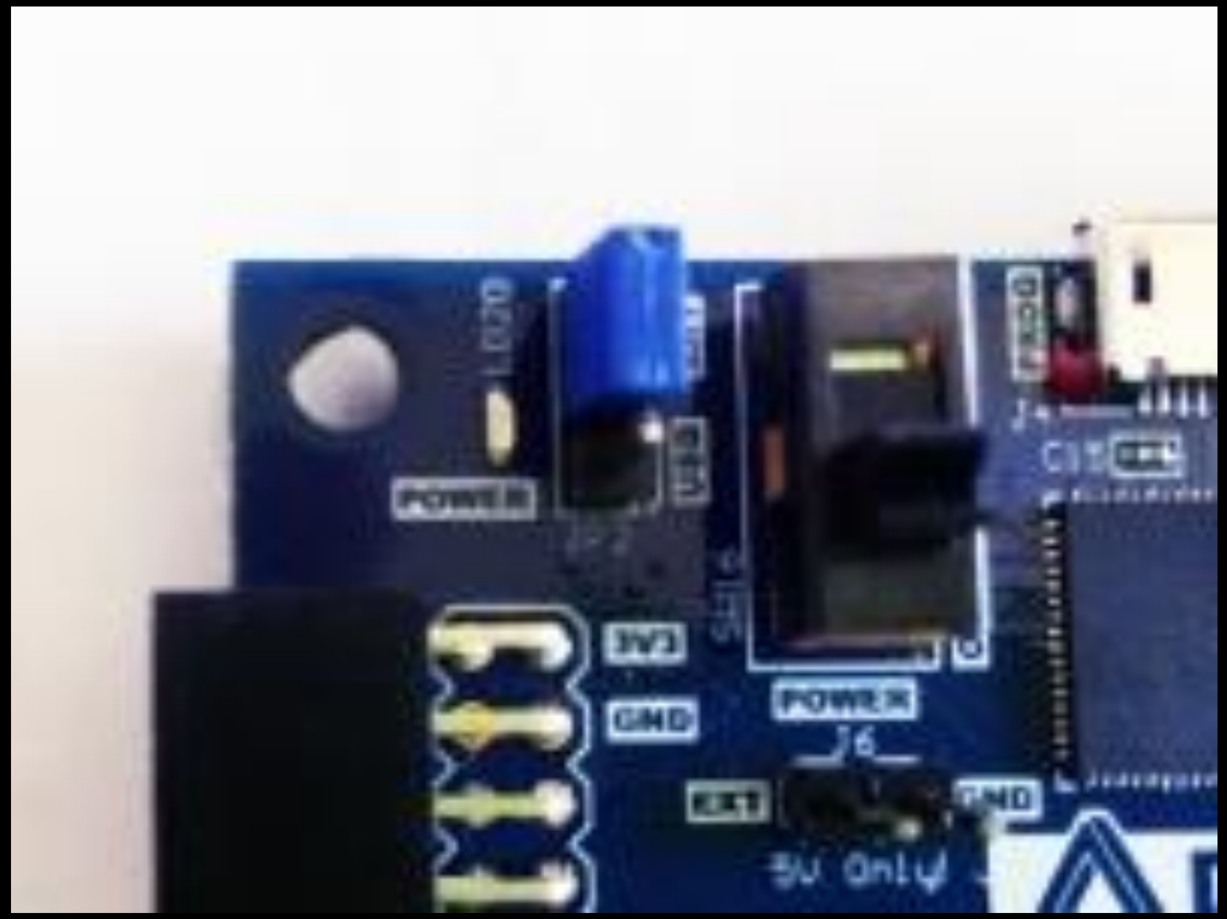


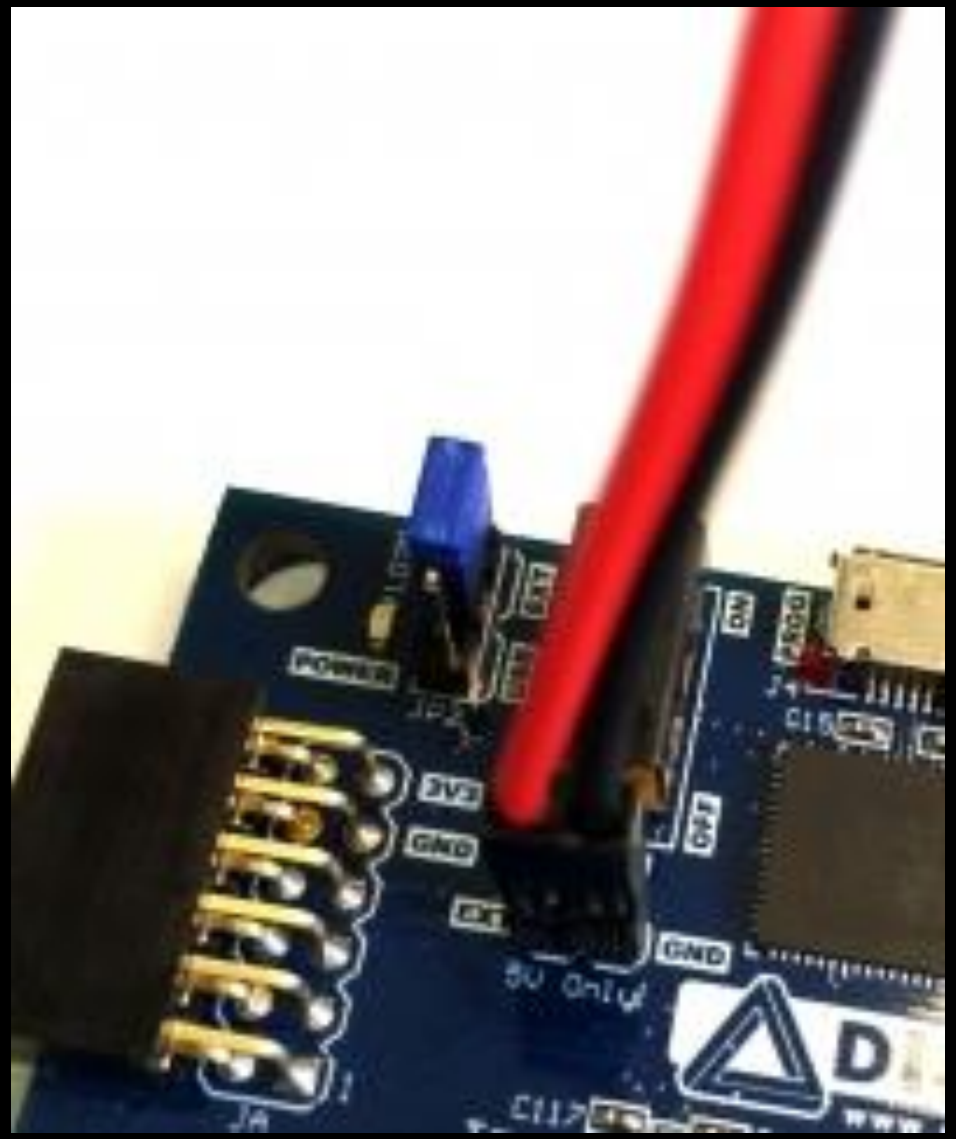
```verilog
module motor(
    input clk,
    input rst,
    output  [1:0]pwm
);

    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
```

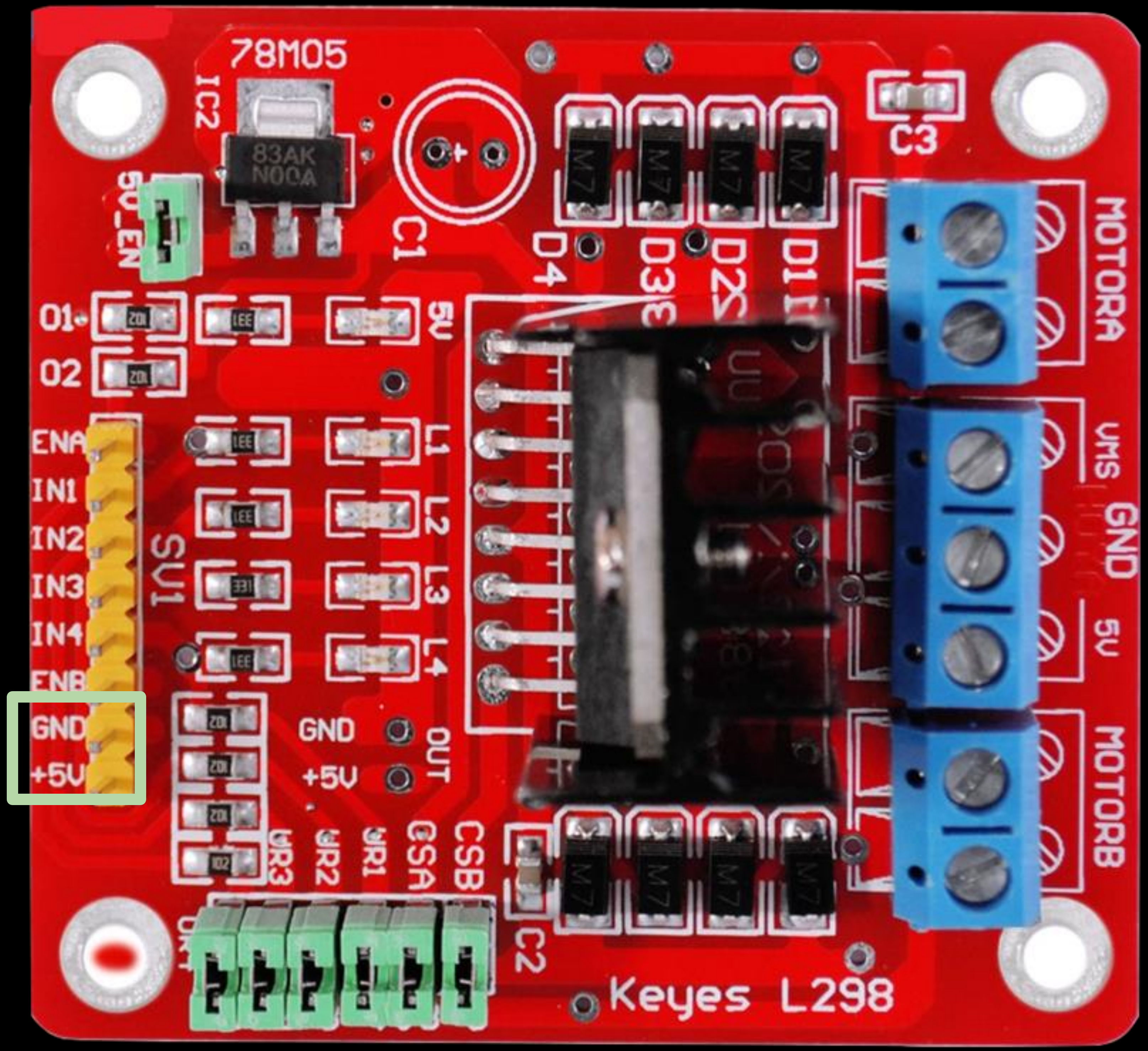
Materials overview (8/17)

L298N motor driver module

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

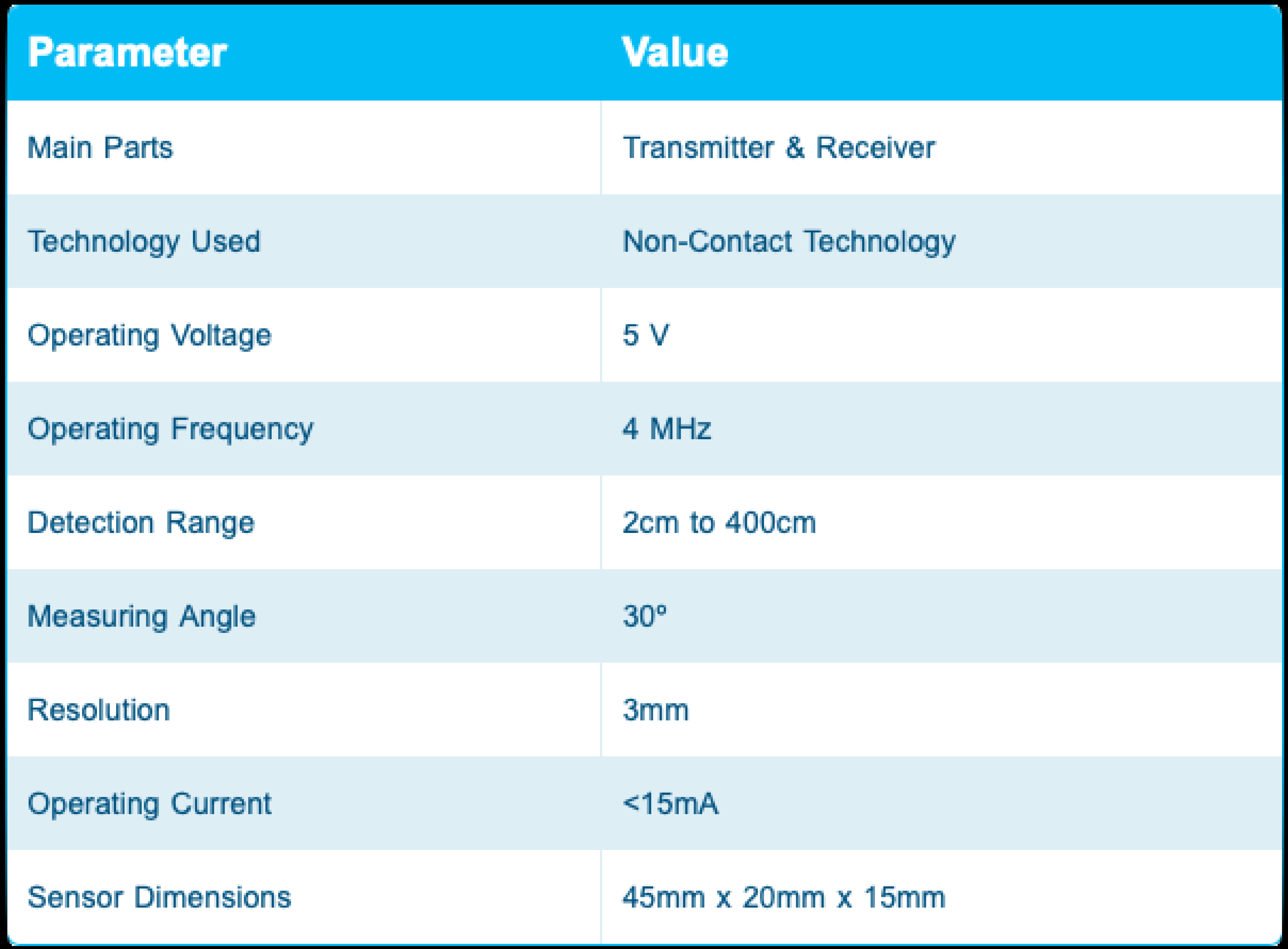






Materials overview (9/17)

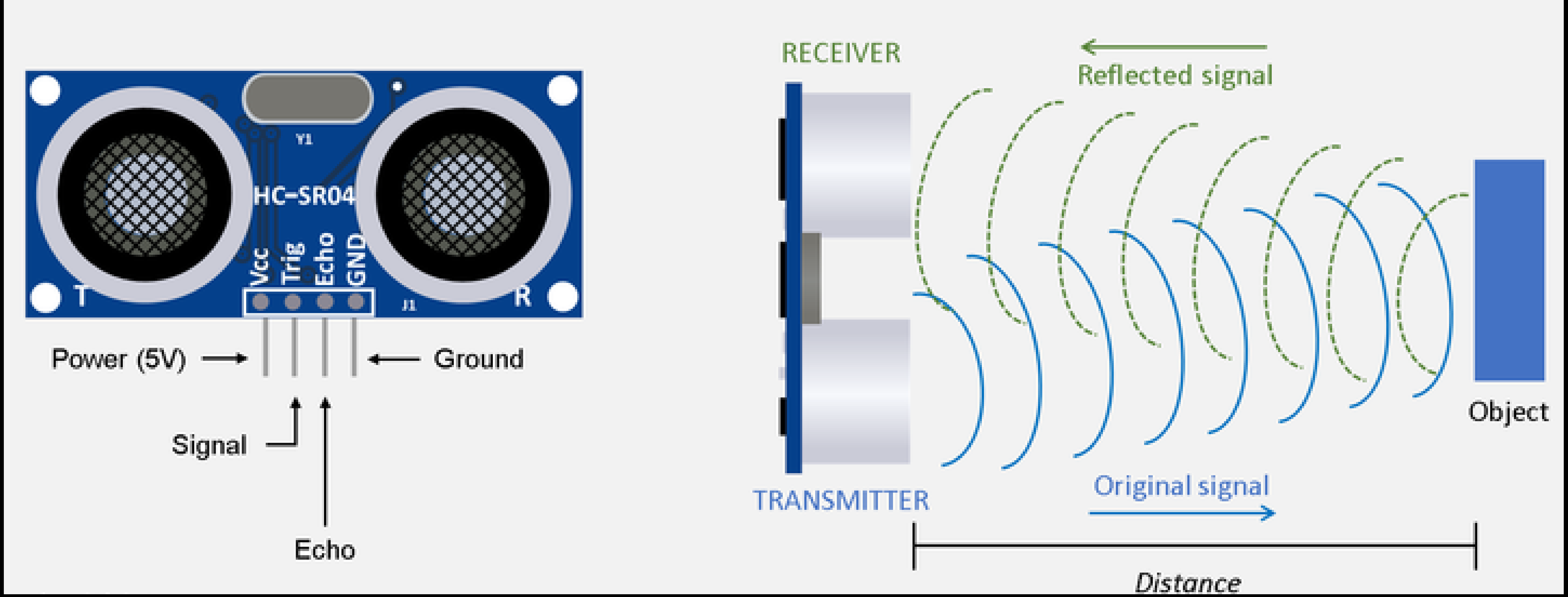
Ultrasonic sensor: HC-SR04



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

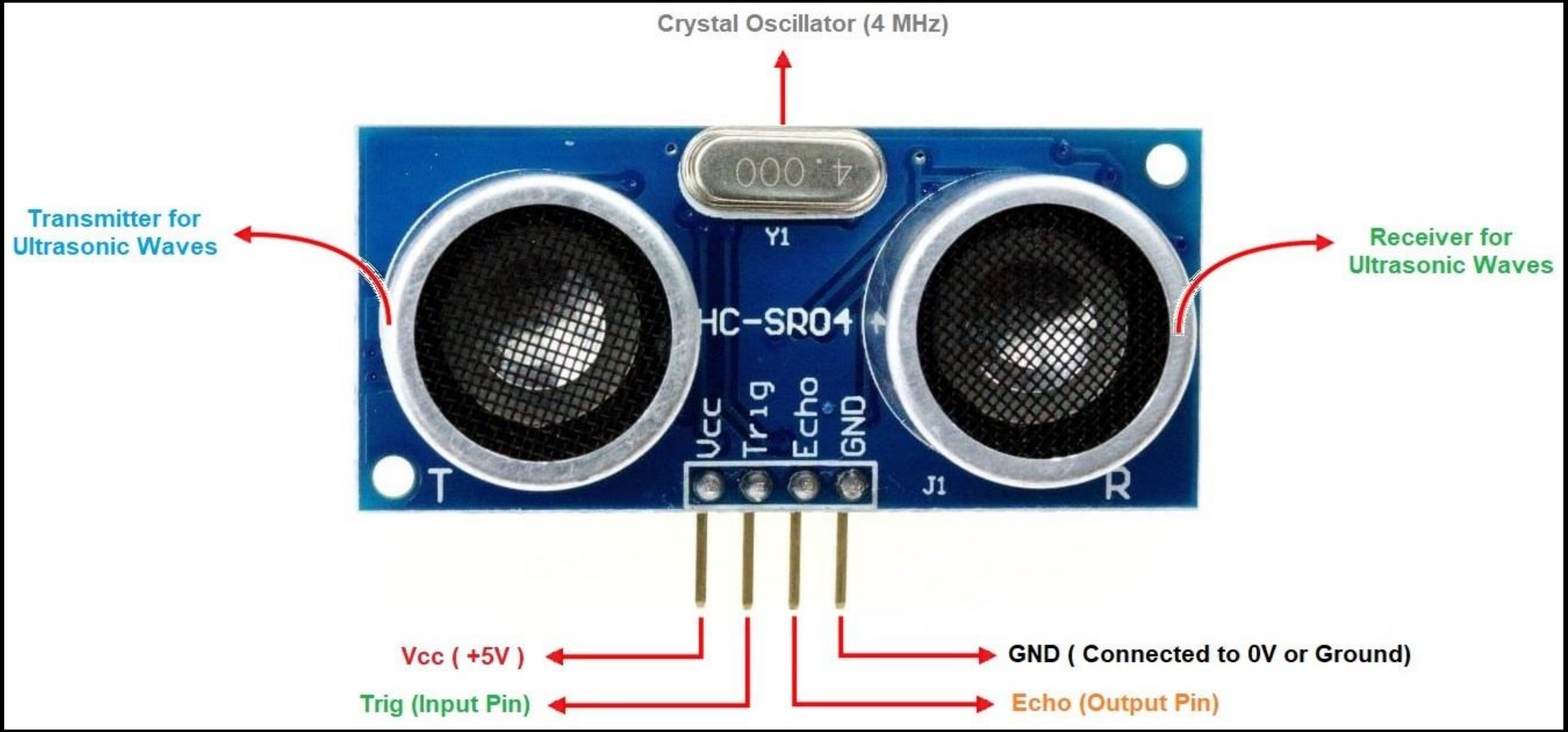
Materials overview (10/17)

HC-SR04



Materials overview (11/17)

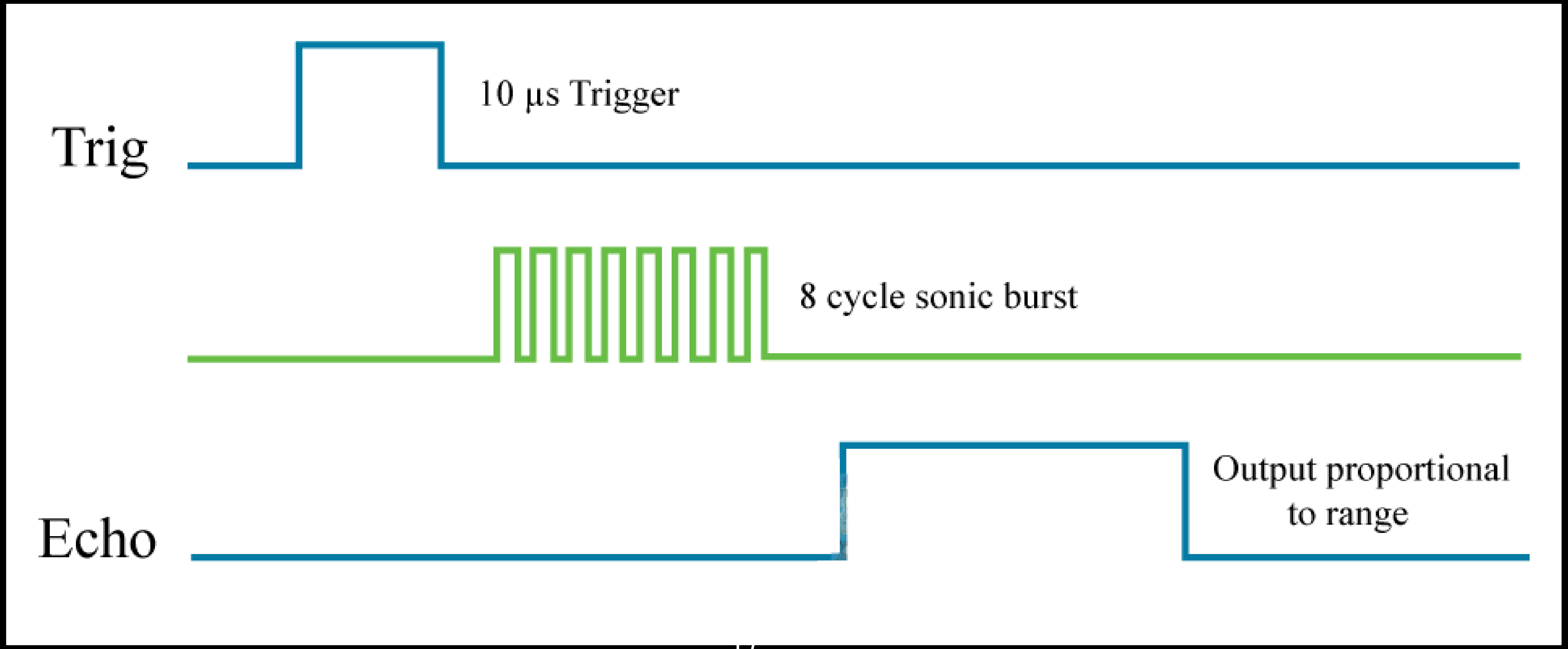
HC-SR04 Pinout


Crystal Oscillator (4 MHz)
Transmitter for Ultrasonic Waves
Receiver for Ultrasonic Waves
4.000
Y1
HC-SR04
Vcc
Trig
Echo
GND
T
J1
R
Vcc (+5V)
Trig (Input Pin)
GND (Connected to 0V or Ground)
Echo (Output Pin)

Materials overview (12/17)

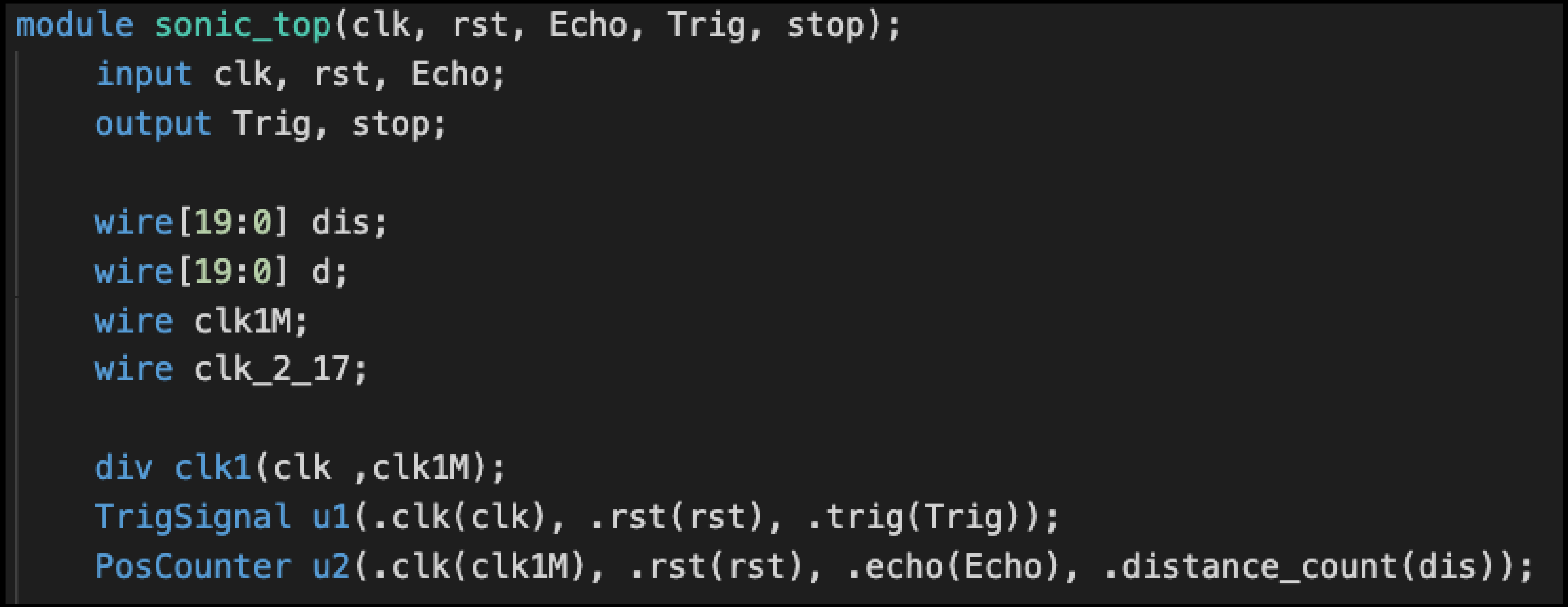
HC-SR04

- 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) :

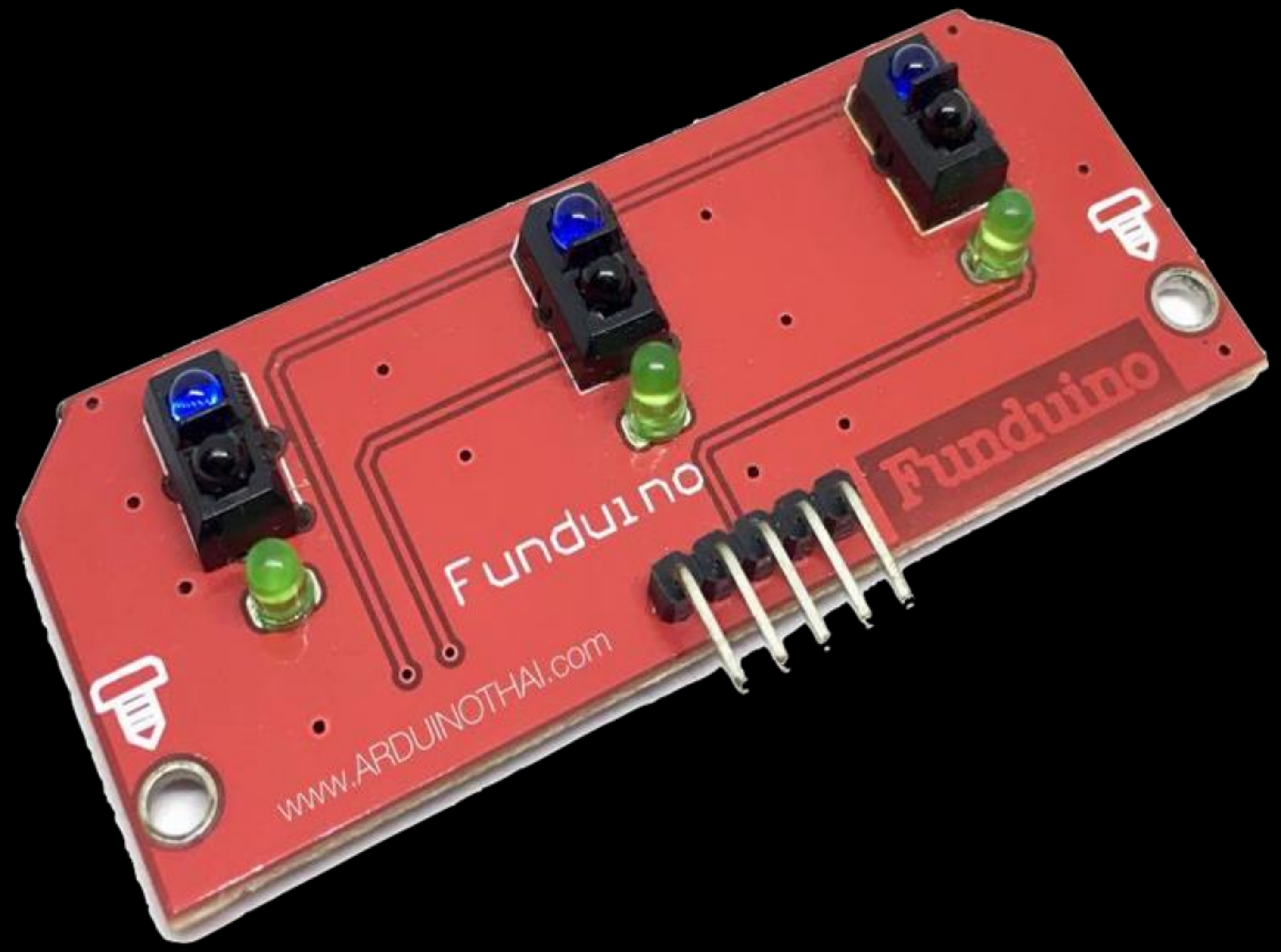


```
module sonic_top(clk, rst, Echo, Trig, stop);
    input clk, rst, Echo;
    output Trig, stop;

    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));
```

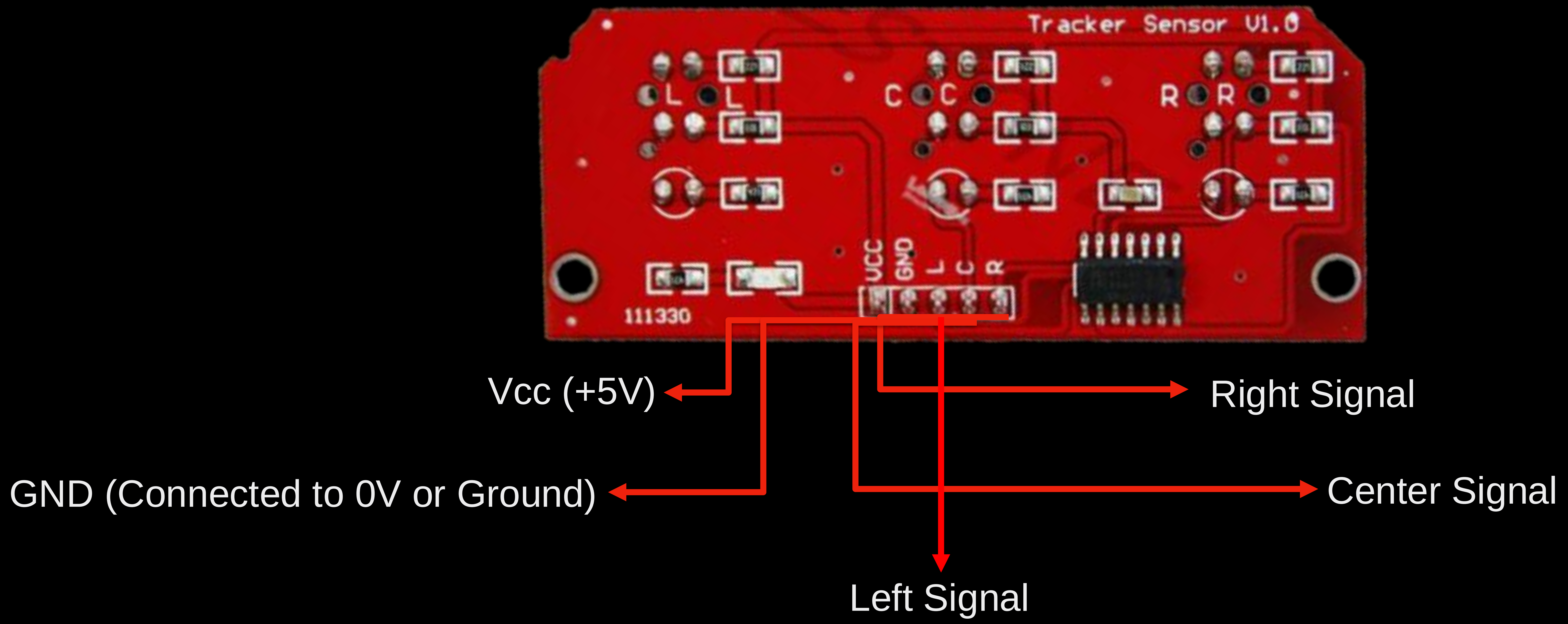
Materials overview (14/17)

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)



Vcc (+5V)

Right Signal

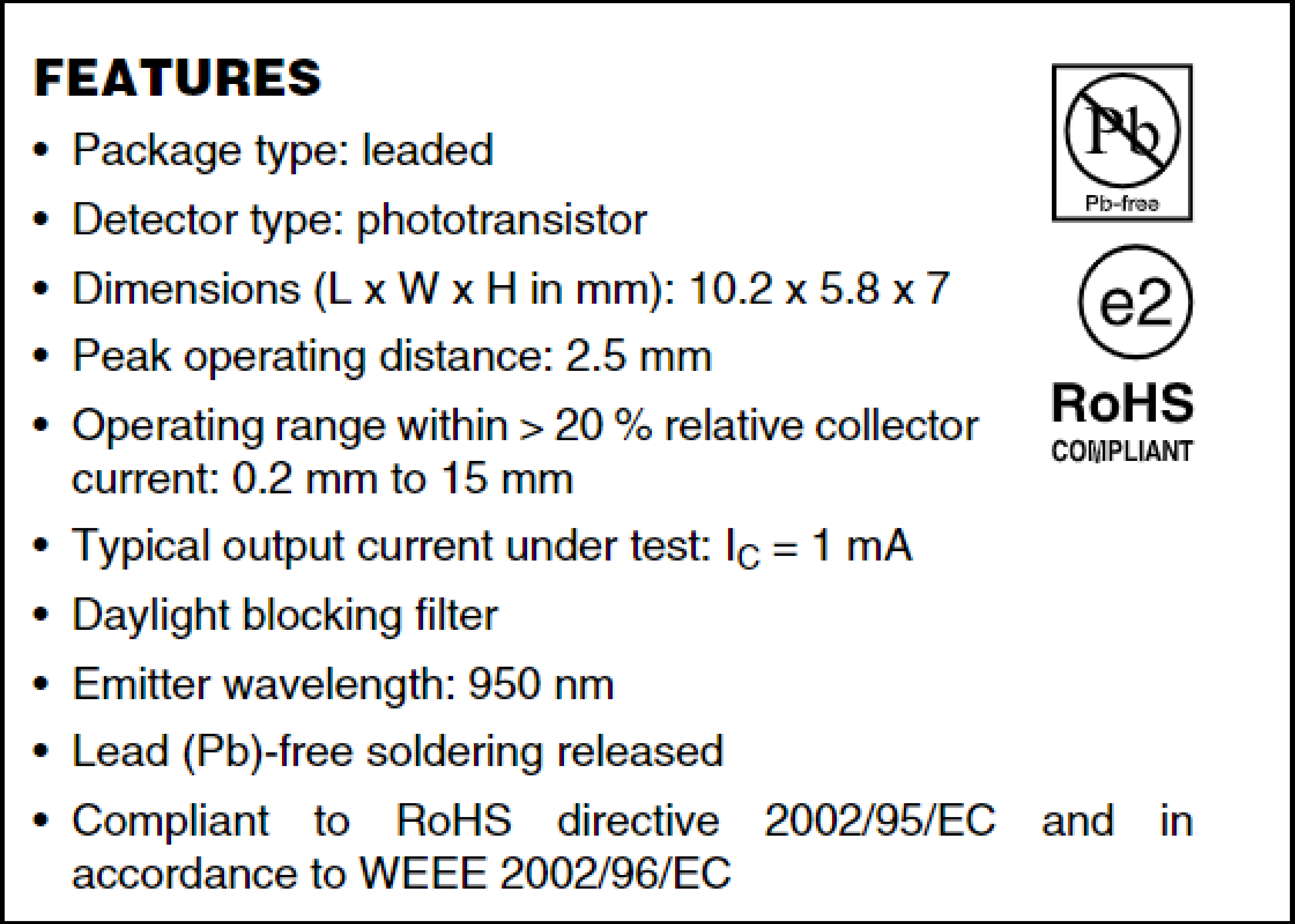
GND (Connected to 0V or Ground)

Center Signal

Left Signal

Materials overview (16/17)

Document link

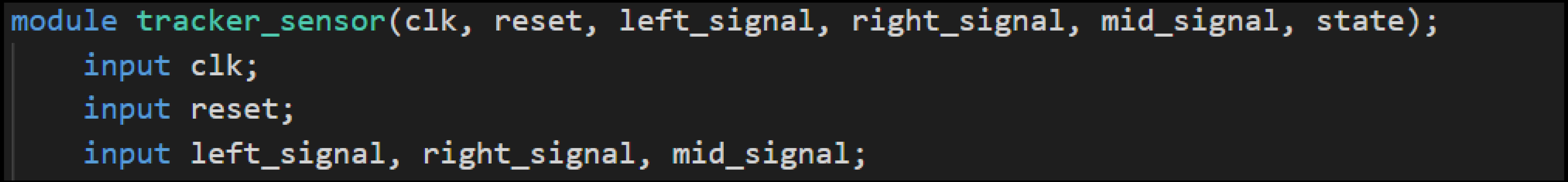


FEATURES

- 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 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

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

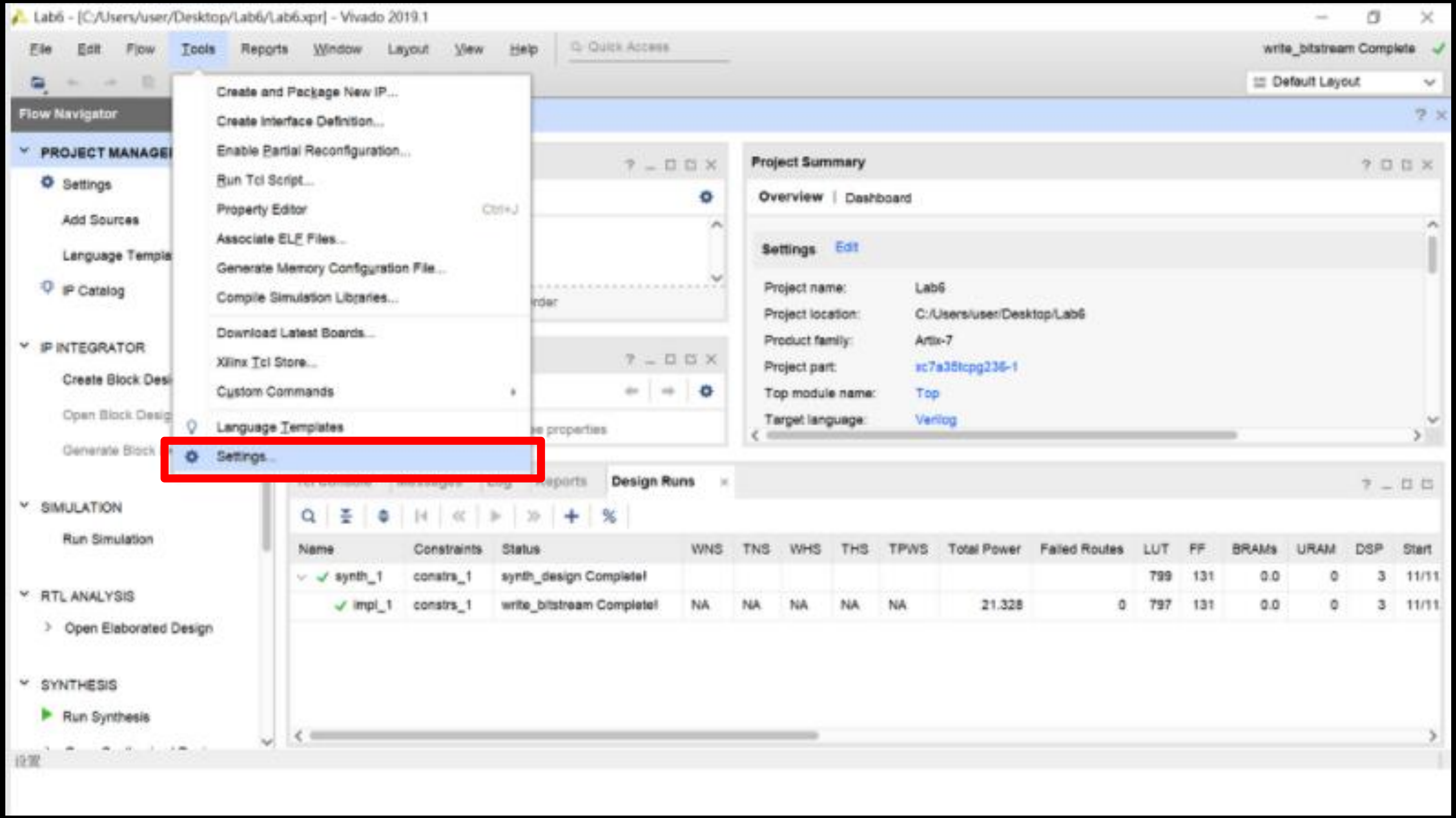
FPGA configuration (1/8)

- Bitstream Configuration
- Flash Memory Setting

FPGA configuration (2/8)

Bitstream Configuration

- Tools -> Settings



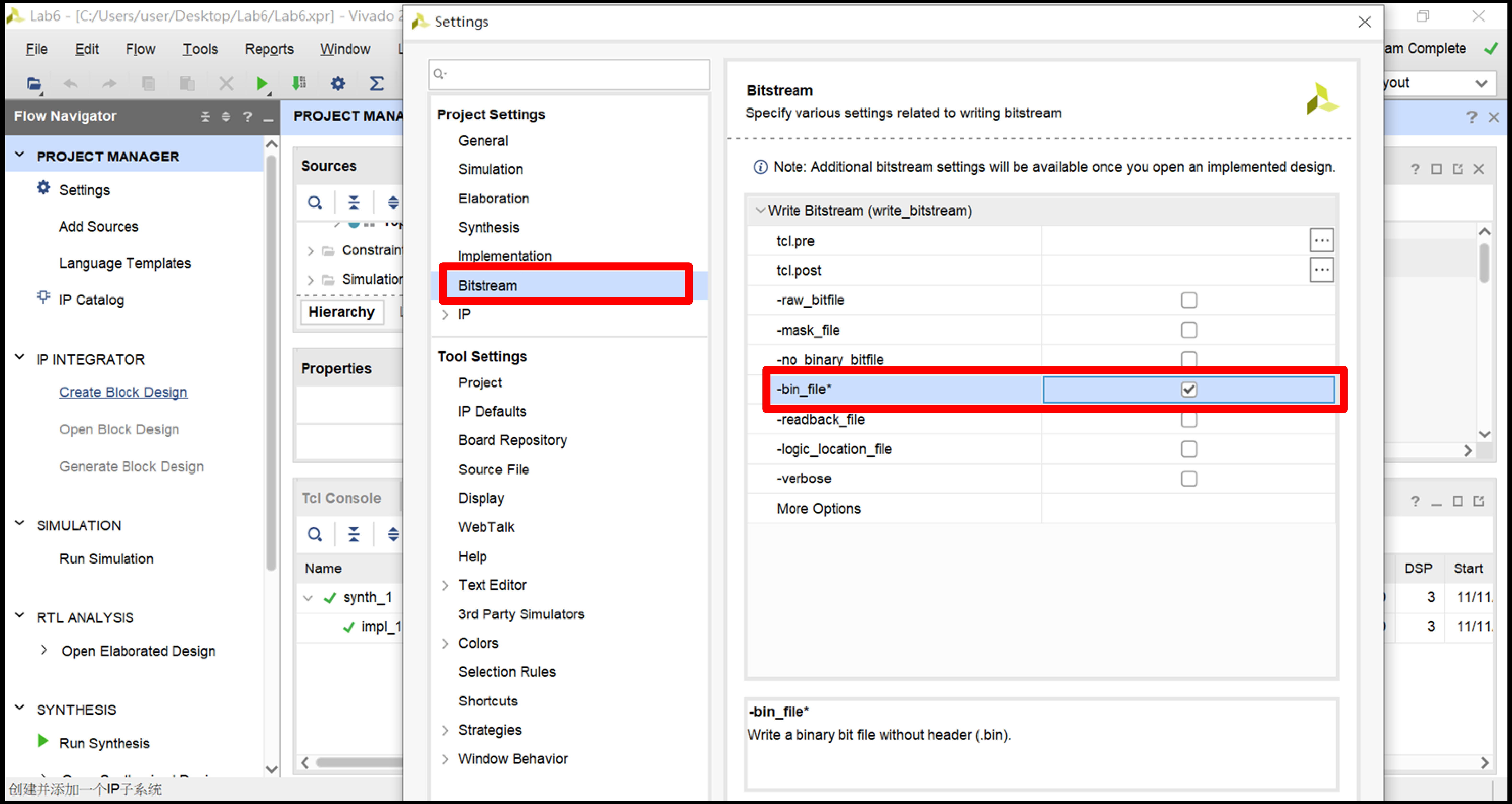
Project name:	Lab6
Project location:	C:/Users/user/Desktop/Lab6
Product family:	Artix-7
Project part:	xc7a35tcpg236-1
Top module name:	Top
Target language:	Verilog

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start
✓ synth_1	constrs_1	synth_design Complete!								799	131	0.0	0	3	11/11
✓ impl_1	constrs_1	write_bitstream Complete!	NA	NA	NA	NA	NA	21.328	0	797	131	0.0	0	3	11/11

FPGA configuration (3/8)

Bitstream Configuration

- Bitstream -> bin_file
- Apply -> OK



Bitstream

Specify various settings related to writing bitstream

ⓘ Note: Additional bitstream settings will be available once you open an implemented design.

Write Bitstream (write_bitstream)

tcl.pre	…
tcl.post	…
-raw_bitfile	☐
-mask_file	☐
-no_binary_bitfile	☐
-bin_file*	☑
-readback_file	☐
-logic_location_file	☐
-verbose	☐
More Options	

-bin_file*

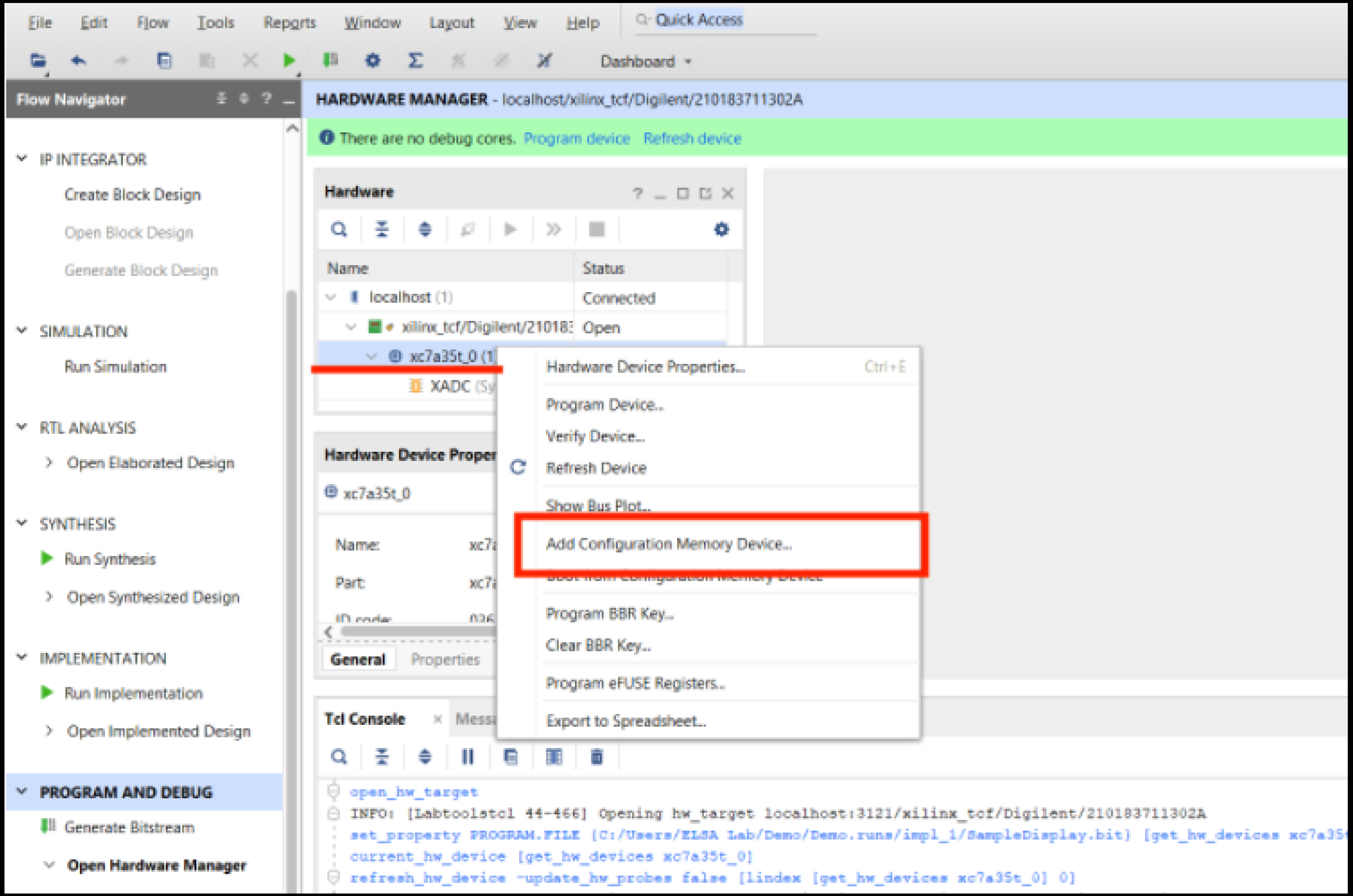
Write a binary bit file without header (.bin).

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



HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210183711302A

There are no debug cores. Program device Refresh device

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/21018	Open
xc7a35t_0 (1	
XADC (Sy	

xc7a35t_0

Name: xc7a

Part: xc7a

ID code: 036

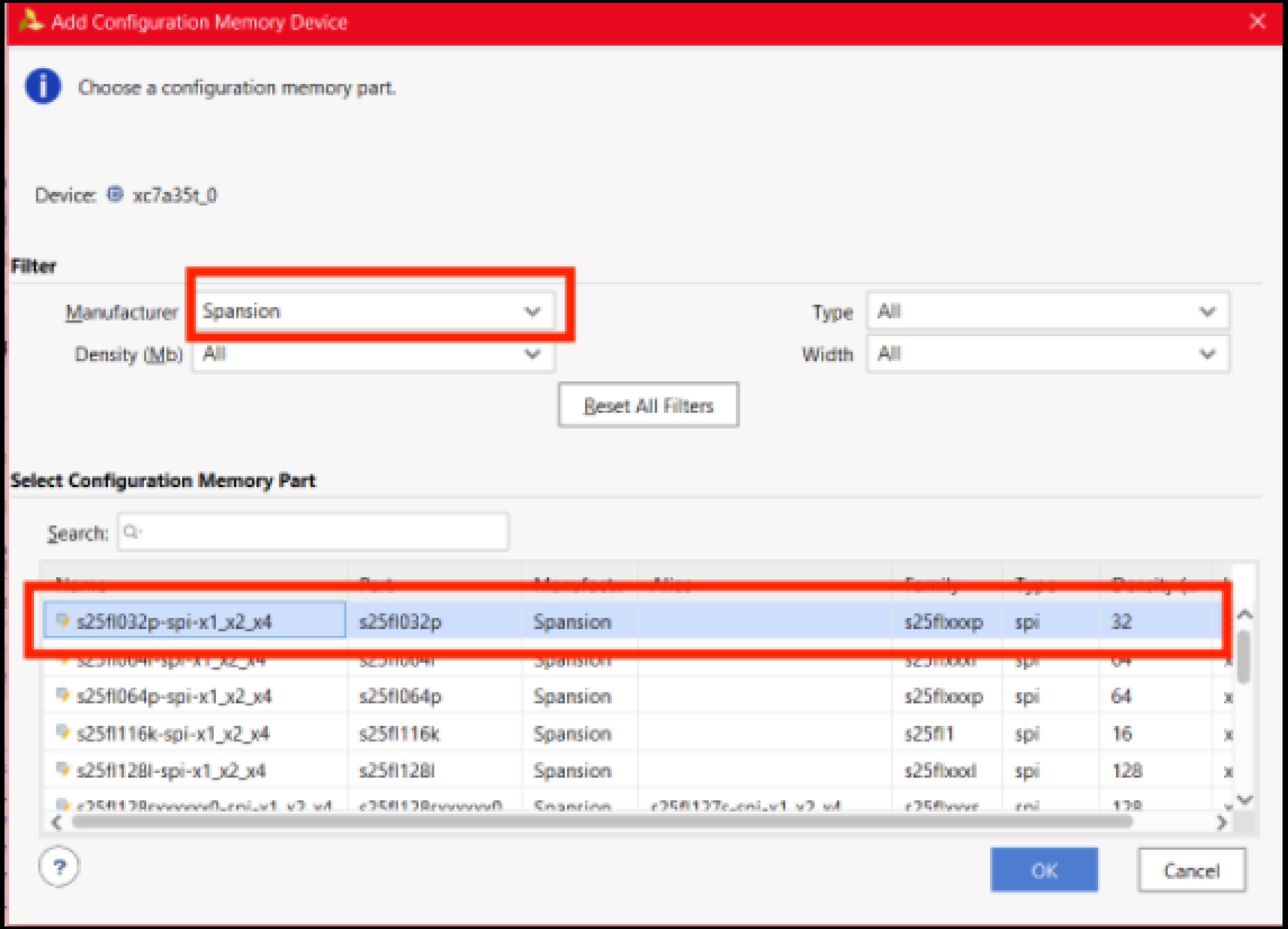
```
open_hw_target
INFO: [Labtoolstcl 44-466] Opening hw_target localhost:3121/xilinx_tcf/Digilent/210183711302A
set_property PROGRAM.FILE {C:/Users/ELSA Lab/Demo/Demo.runs/impl_1/SampleDisplay.bit} [get_hw_devices xc7a35t
current_hw_device [get_hw_devices xc7a35t_0]
refresh_hw_device -update_hw_probes false [lindex [get_hw_devices xc7a35t_0] 0]
```

FPGA configuration (5/8)

Flash Memory Setting

- [Filter] -> [manufacturer]
  -> [Spansion]

- [Configuration Memory Part]
  -> [s25fl032p]



Add Configuration Memory Device

Choose a configuration memory part.

Device: xc7a35t_0

Filter

Manufacturer: Spansion

Density (Mb): All

Type: All

Width: All

Select Configuration Memory Part

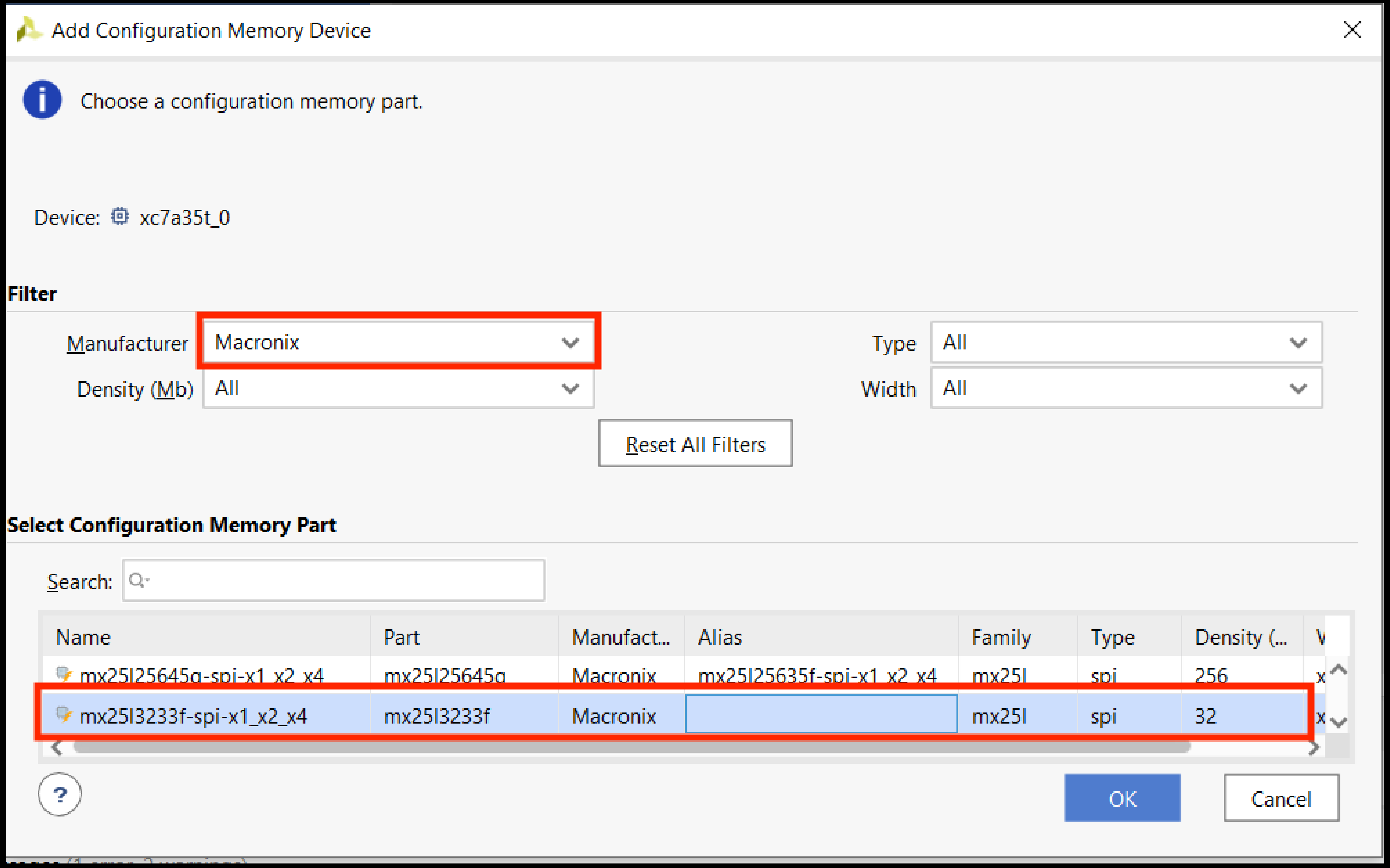
Search:

Name	Part	Manufacturer	Alias	Family	Type	Density (	
s25fl032p-spi-x1_x2_x4	s25fl032p	Spansion		s25flxxxp	spi	32	
s25fl064l-spi-x1_x2_x4	s25fl064l	Spansion		s25flxxxl	spi	64	x
s25fl064p-spi-x1_x2_x4	s25fl064p	Spansion		s25flxxxp	spi	64	x
s25fl116k-spi-x1_x2_x4	s25fl116k	Spansion		s25fl1	spi	16	x
s25fl128l-spi-x1_x2_x4	s25fl128l	Spansion		s25flxxxl	spi	128	x
s25fl128sxxxxxx0-spi-x1_x2_x4	s25fl128sxxxxxx0	Spansion	s25fl127s-spi-x1_x2_x4	s25flxxxs	spi	128	x

FPGA configuration (5/8) - new board

Flash Memory Setting

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



Add Configuration Memory Device

Choose a configuration memory part.

Device: xc7a35t_0

Filter

Manufacturer: Macronix

Density (Mb): All

Type: All

Width: All

Select Configuration Memory Part

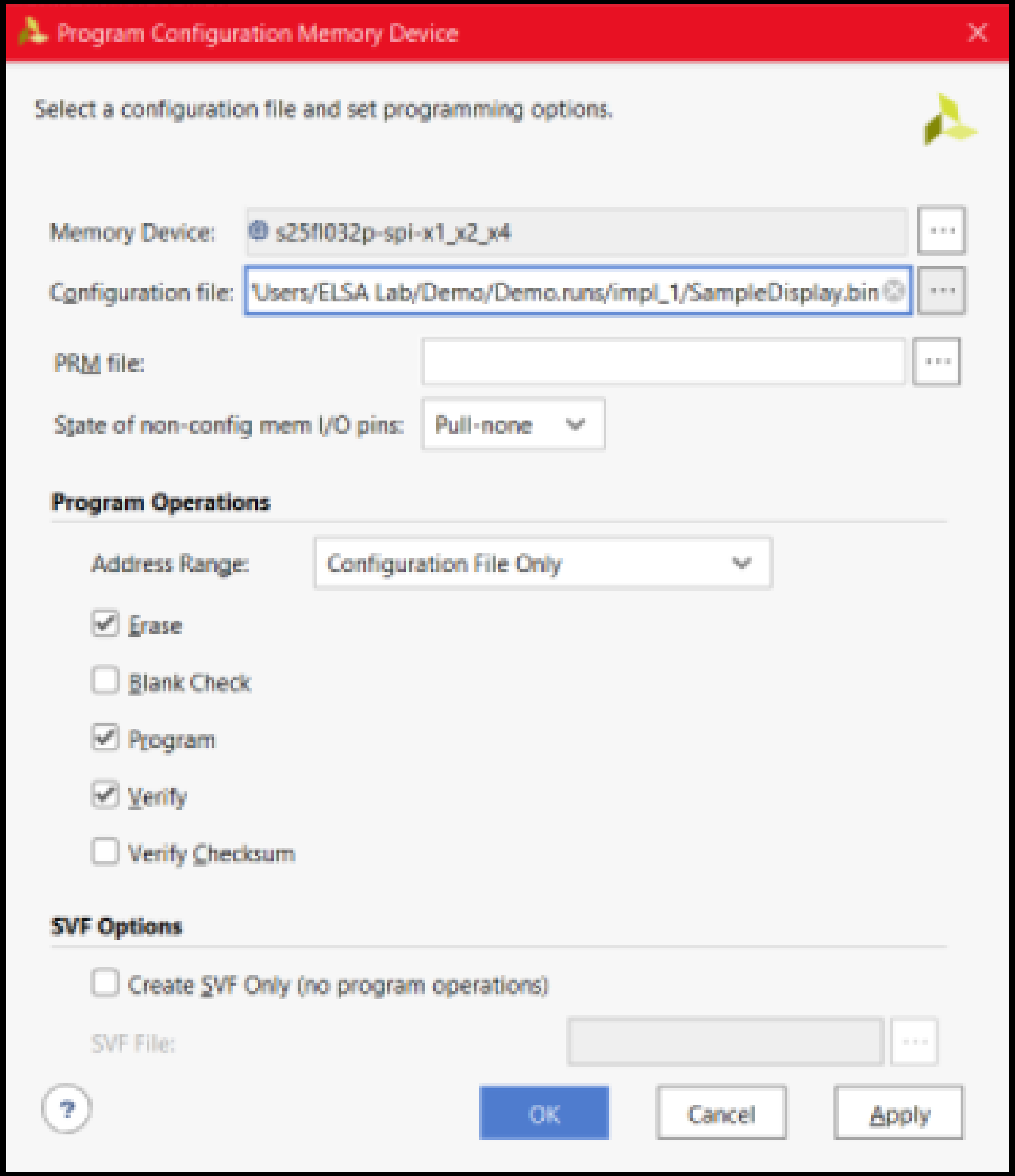
Search:

Name	Part	Manufact...	Alias	Family	Type	Density (...	V
mx25l25645g-spi-x1_x2_x4	mx25l25645g	Macronix	mx25l25635f-spi-x1_x2_x4	mx25l	spi	256	x
mx25l3233f-spi-x1_x2_x4	mx25l3233f	Macronix		mx25l	spi	32	x

FPGA configuration (6/8)

Flash Memory Setting

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



Program Configuration Memory Device

Select a configuration file and set programming options.

Memory Device: s25fl032p-spi-x1_x2_x4

Configuration file: Users/ELSA Lab/Demo/Demo.runs/impl_1/SampleDisplay.bin

PRM file:

State of non-config mem I/O pins: Pull-none

Program Operations

Address Range: Configuration File Only

☑ Erase

☐ Blank Check

☑ Program

☑ Verify

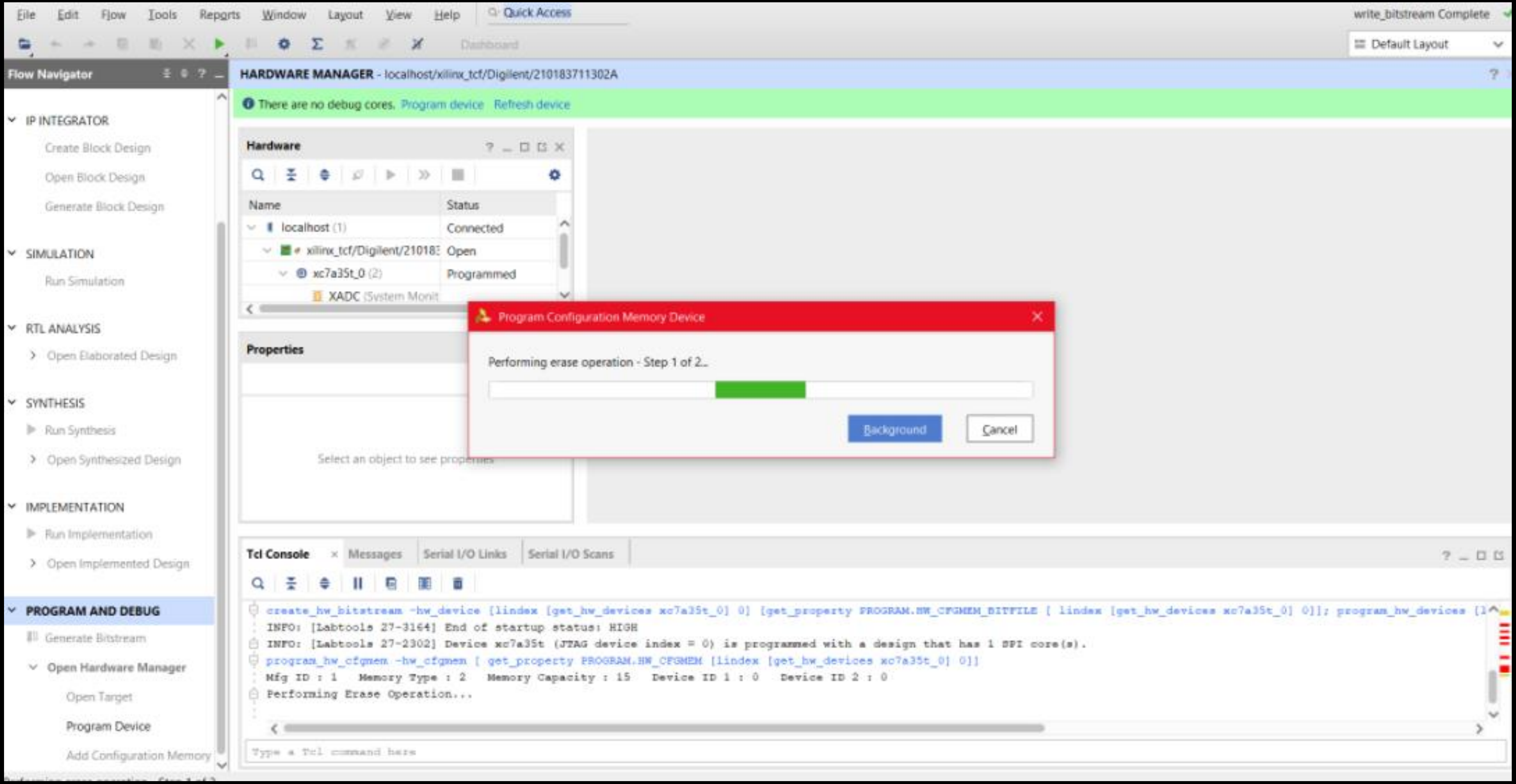
☐ Verify Checksum

SVF Options

☐ Create SVF Only (no program operations)

SVF File:

FPGA configuration (7/8)



HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210183711302A

There are no debug cores. Program device Refresh device

Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/21018:	Open
xc7a35t_0 (2)	Programmed
XADC (System Monit	

Program Configuration Memory Device

Performing erase operation - Step 1 of 2...

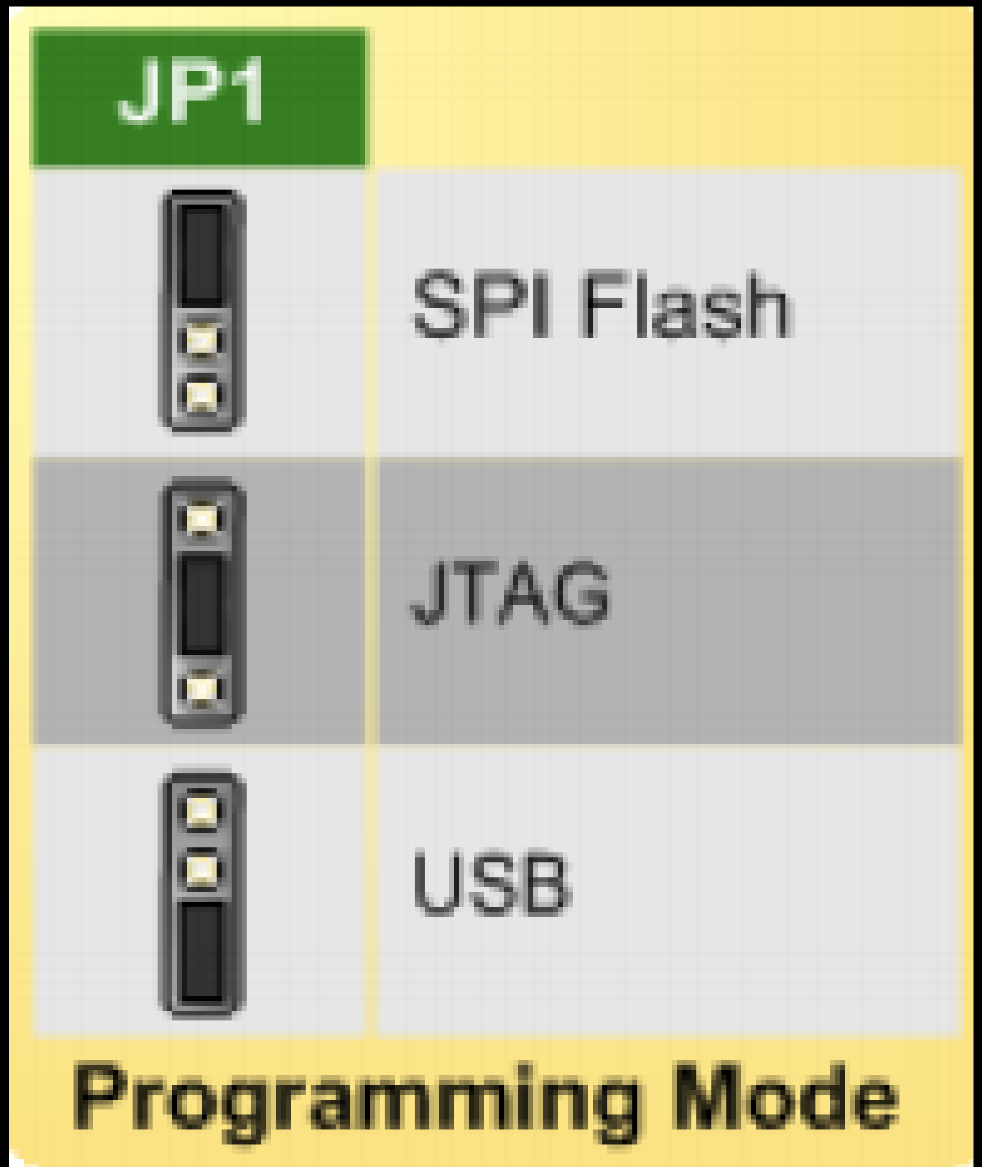
Background Cancel

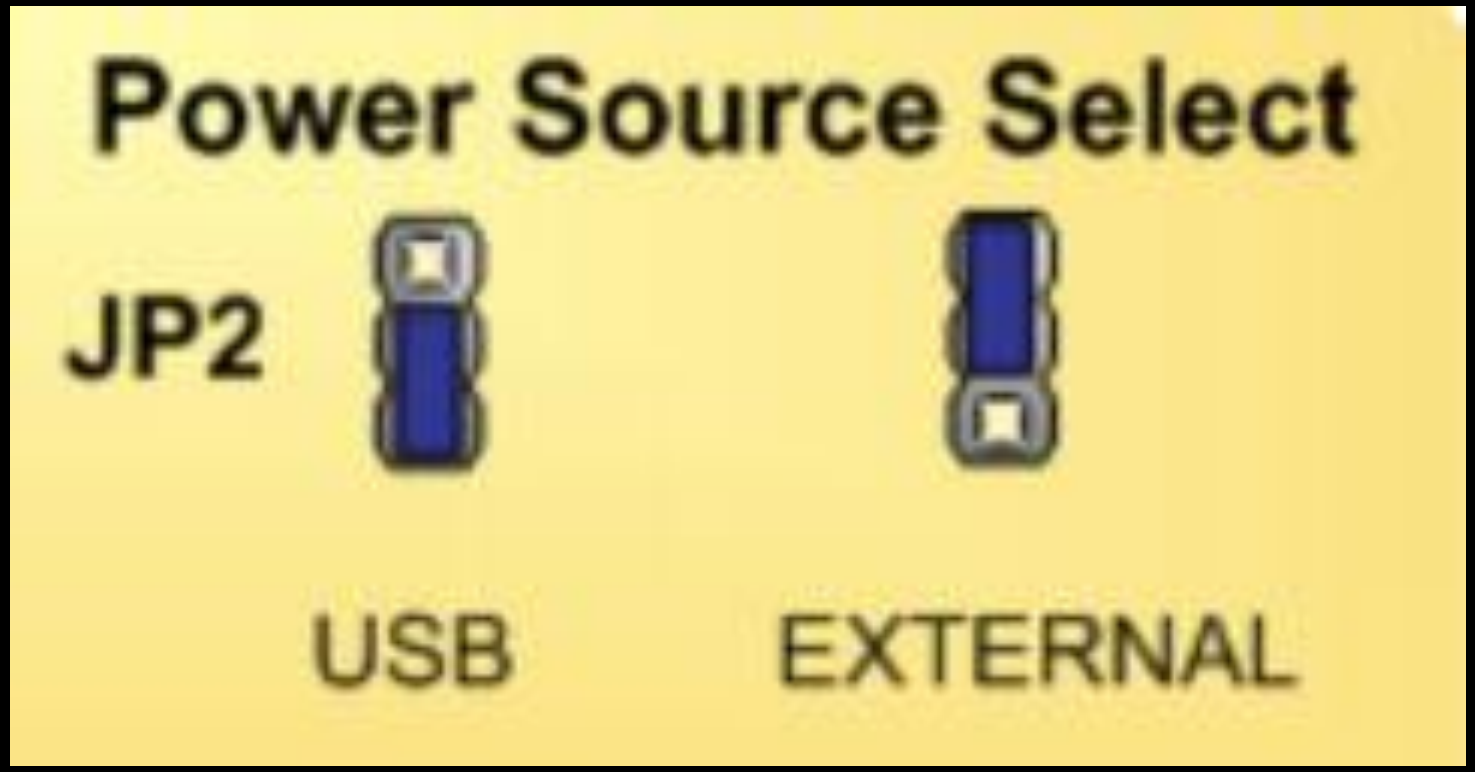
```
create_hw_bitstream -hw_device [lindex [get_hw_devices xc7a35t_0] 0] [get_property PROGRAM.HW_CFGMEM_BITFILE [ lindex [get_hw_devices xc7a35t_0] 0]]; program_hw_devices [l
INFO: [Labtools 27-3164] End of startup status: HIGH
INFO: [Labtools 27-2302] Device xc7a35t (JTAG device index = 0) is programmed with a design that has 1 SPI core(s).
program_hw_cfgmem -hw_cfgmem [ get_property PROGRAM.HW_CFGMEM [lindex [get_hw_devices xc7a35t_0] 0]]
Mfg ID : 1   Memory Type : 2   Memory Capacity : 15   Device ID 1 : 0   Device ID 2 : 0
Performing Erase Operation...
```

FPGA configuration (8/8)

Make sure your

- JP1 is on FLASH mode
- JP2 is on EXTERNAL





Agenda

Introduction

Materials overview

FPGA configuration

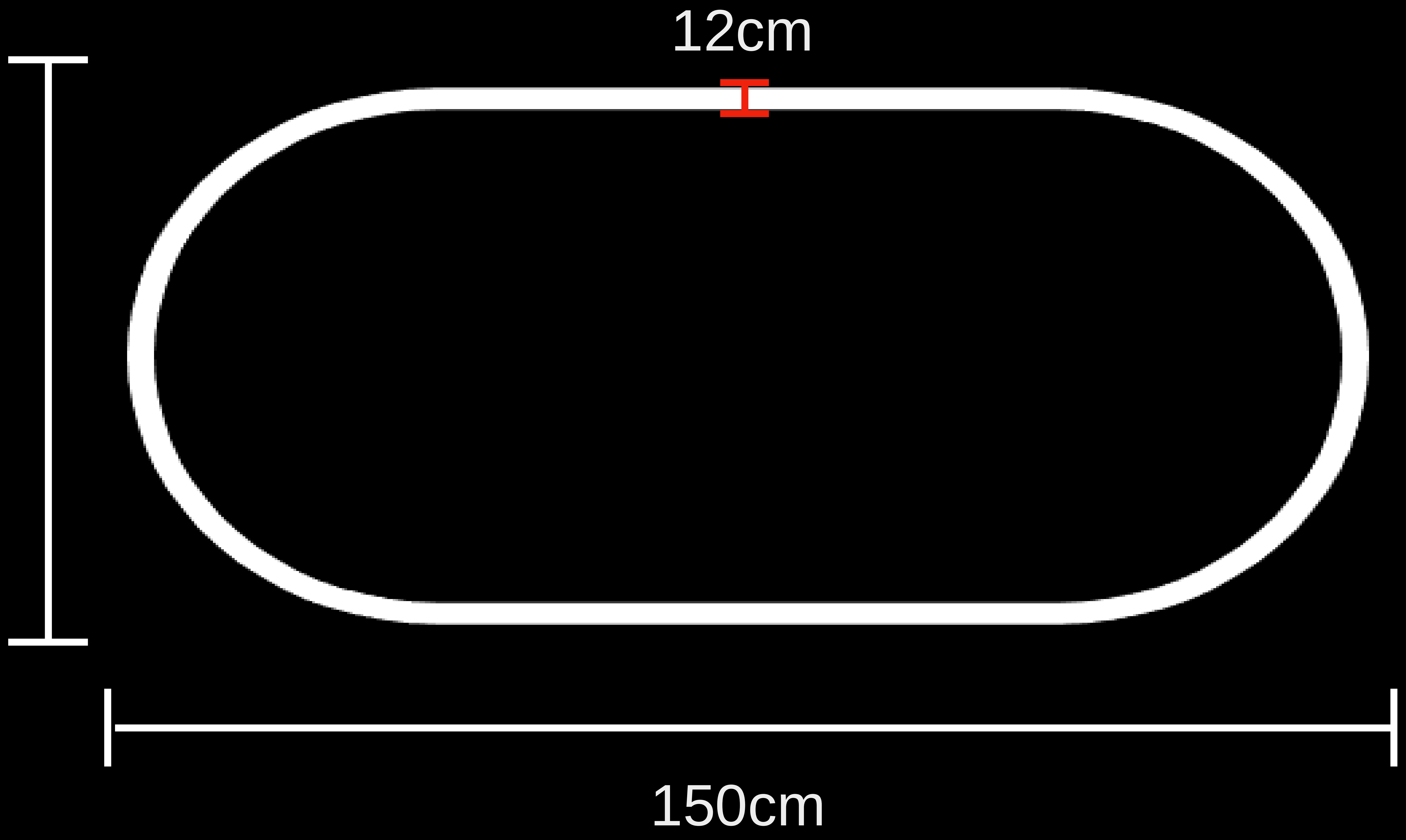
Grading

Grading (1/5)

- 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.

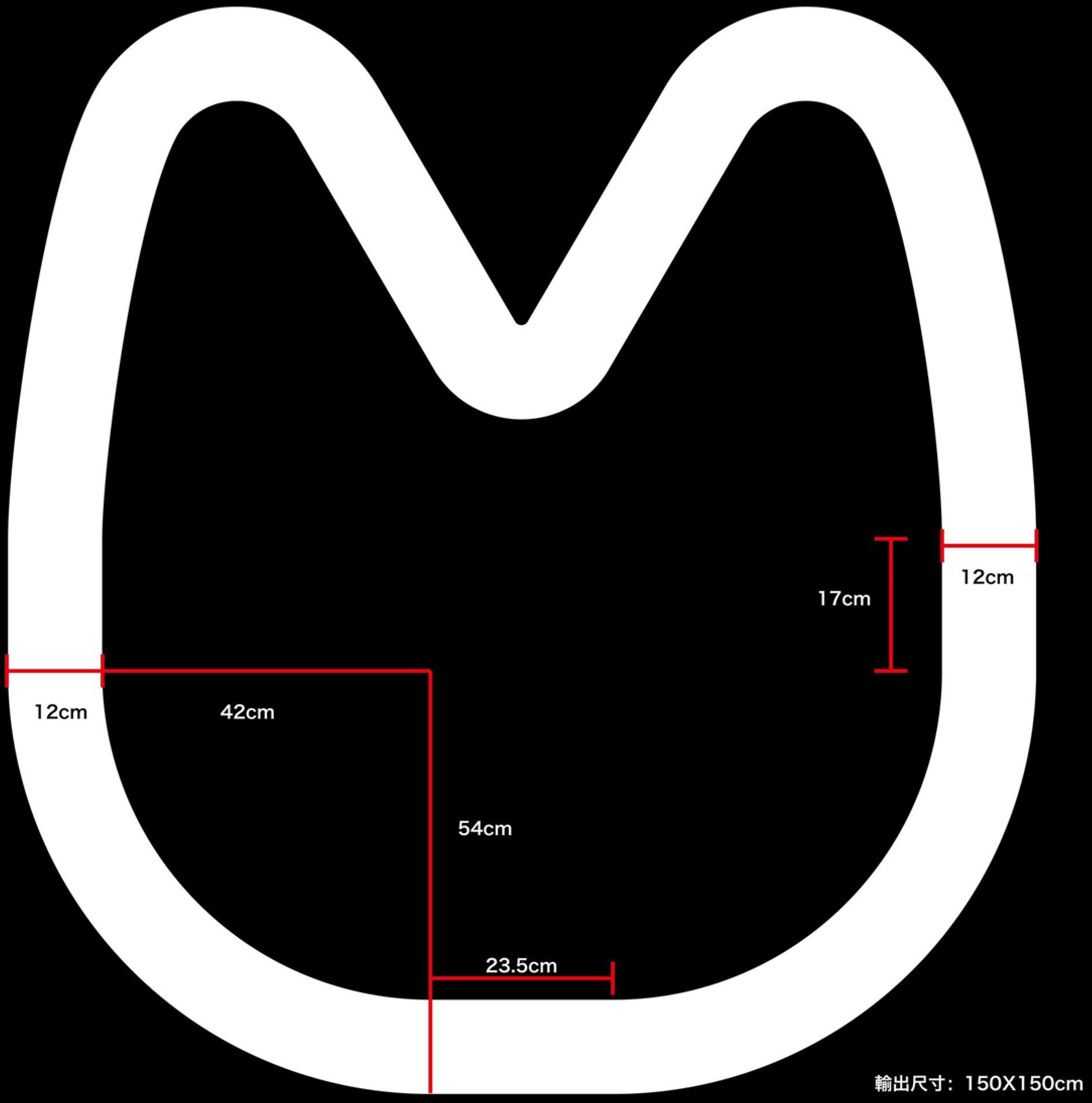
Grading (2/5)

- Track 1 (20%):



Grading (3/5)

- Track 2 (15%):



Grading (4/5)

Bonus track (10~30%):

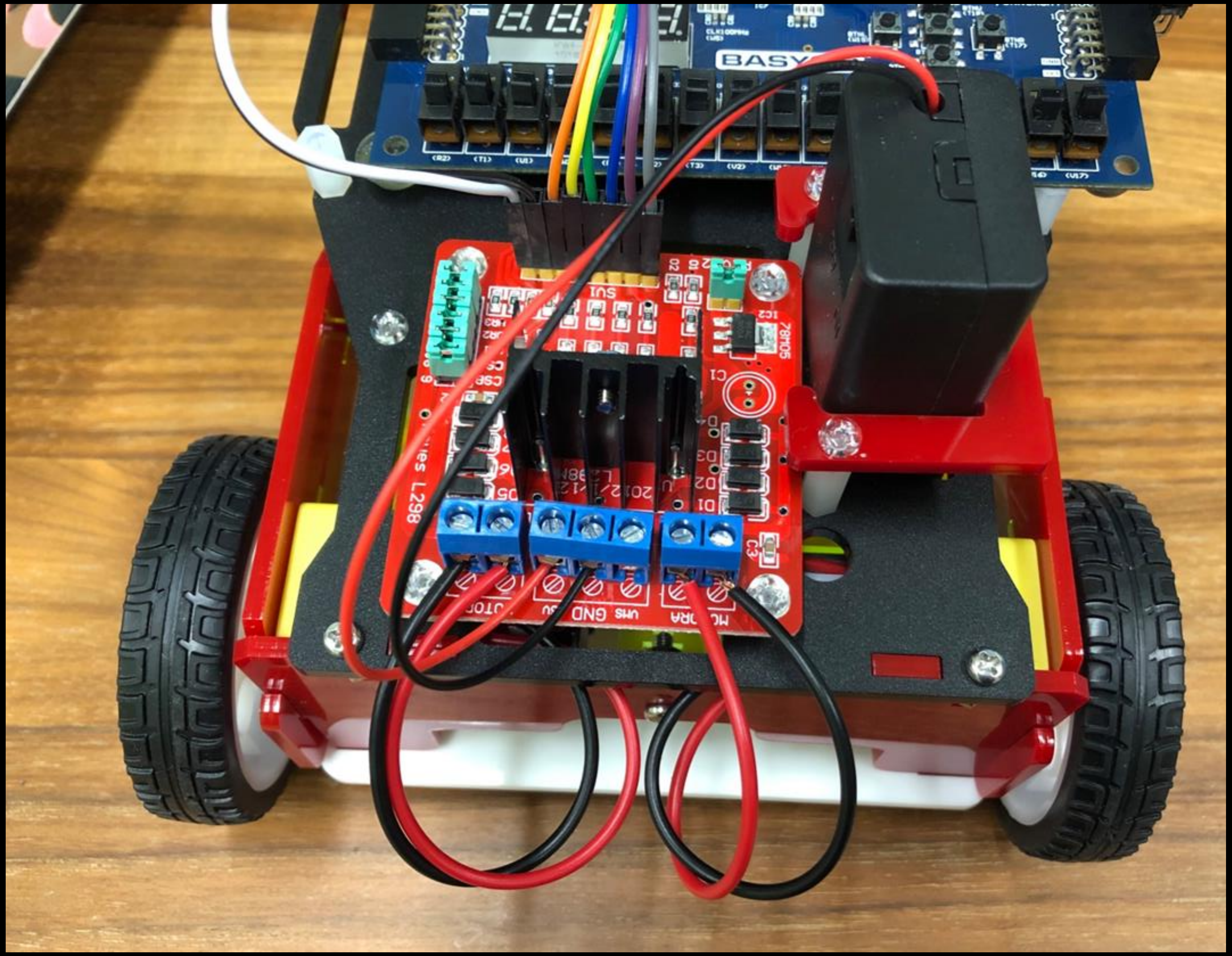
- Be careful on a sharp turn.
- You don’t need to handle Square Corner.
- You don’t need to reverse your car.
- Go straight if there is an intersection.
- Bonus track will first test its correctness, and then test its speed.

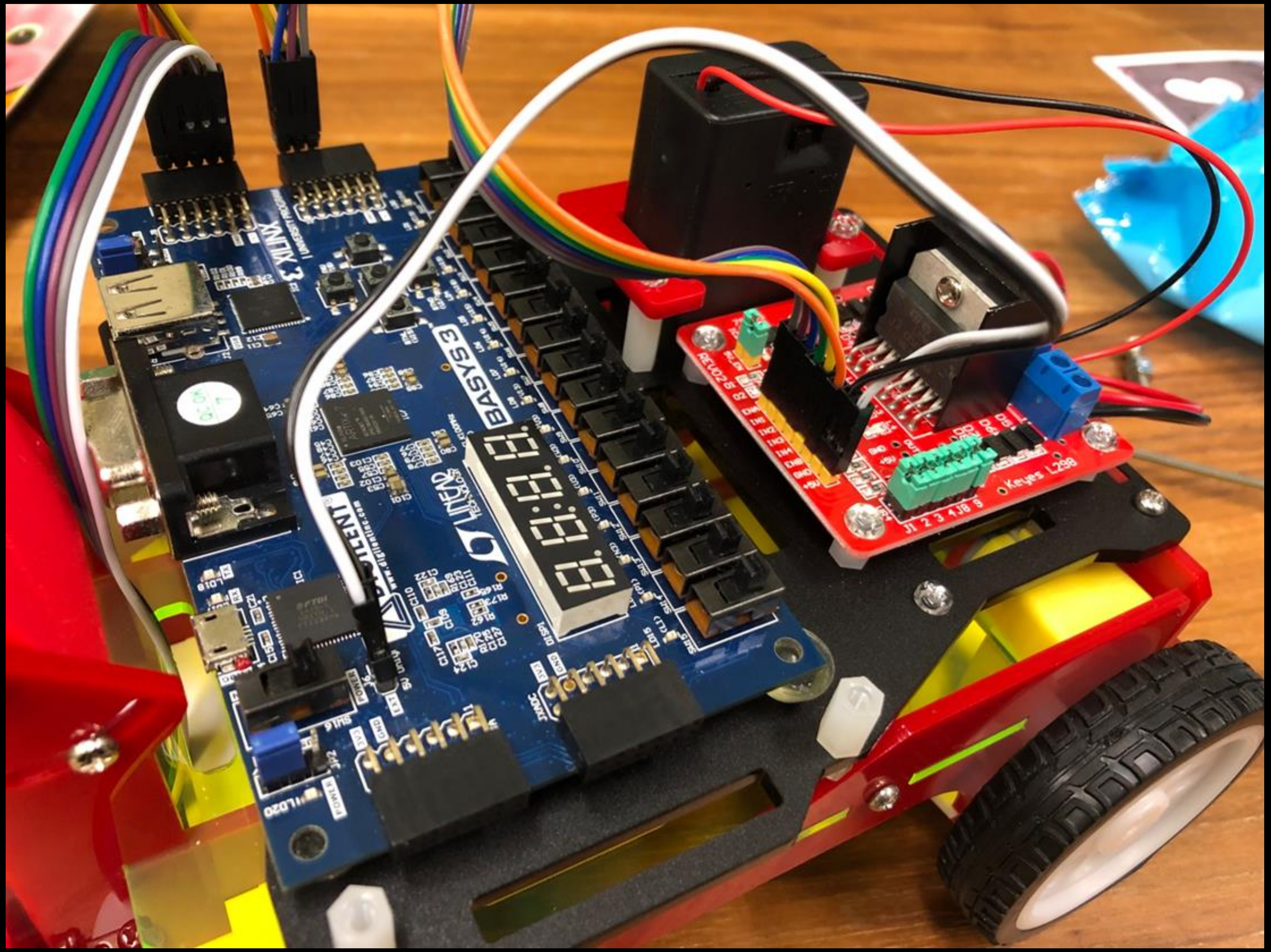
Grading (5/5)

- Report (10%):
  - Sonic:
    - Explanation (1%)
    - State transition diagram (1%)
    - Block diagram (2%)
  - Tracking Strategy:
    - Show the both motors’ PWM settings for different sensor input value (2%)
    - Block diagram for motors (2%)
  - Overall Block Diagram (2%)

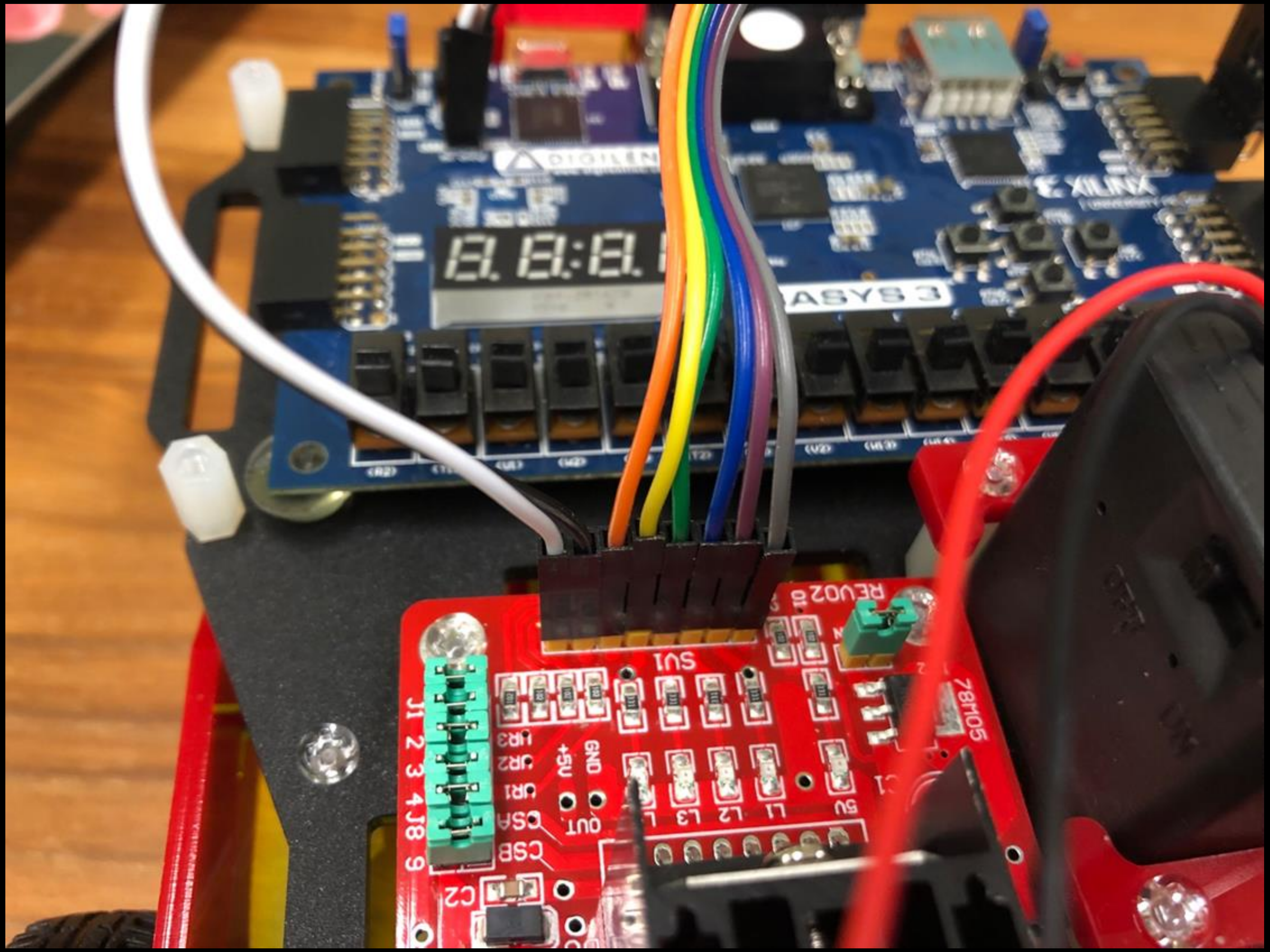
Q&A

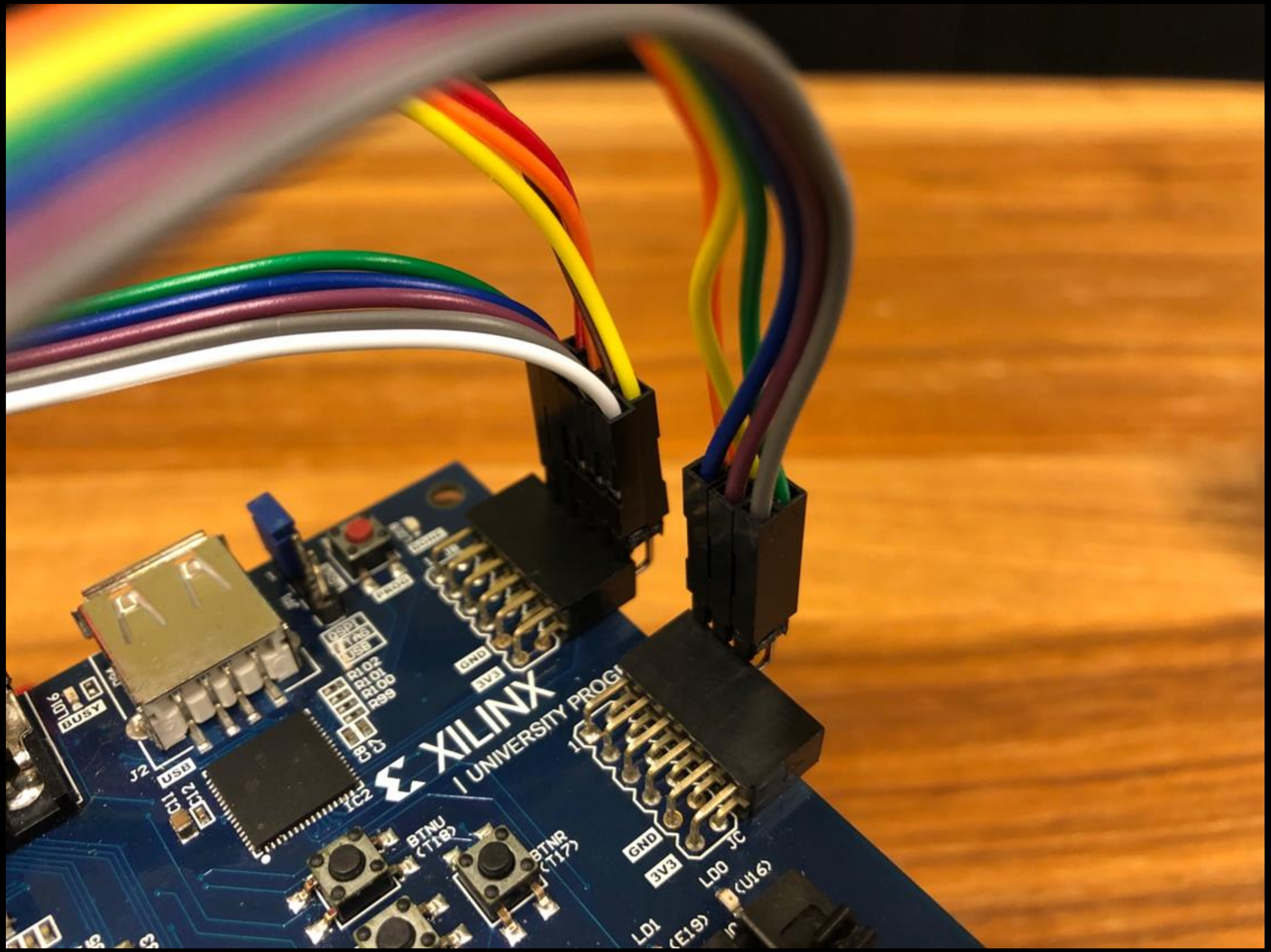
- connect the battery to motor
- supply to FPGA



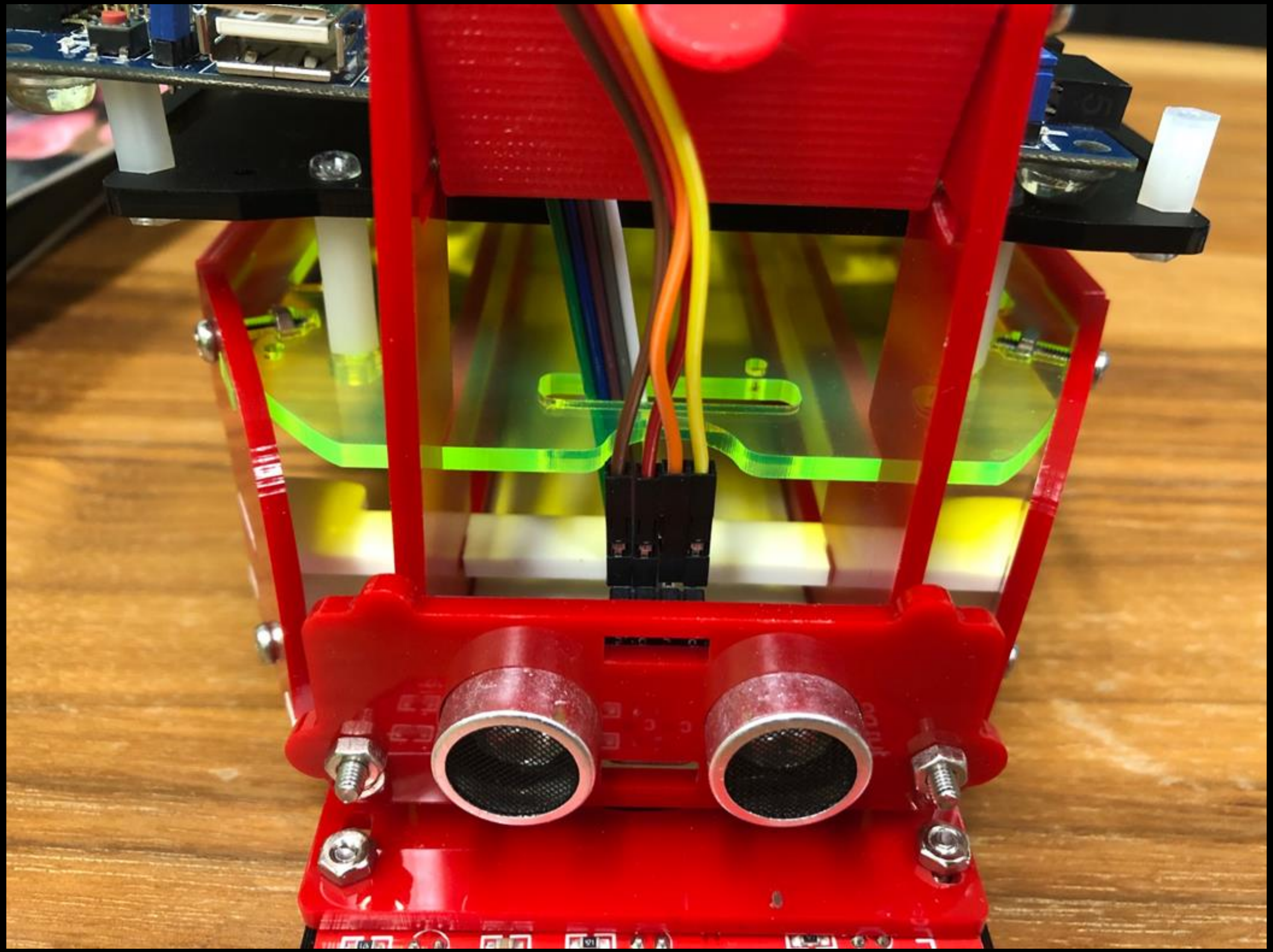


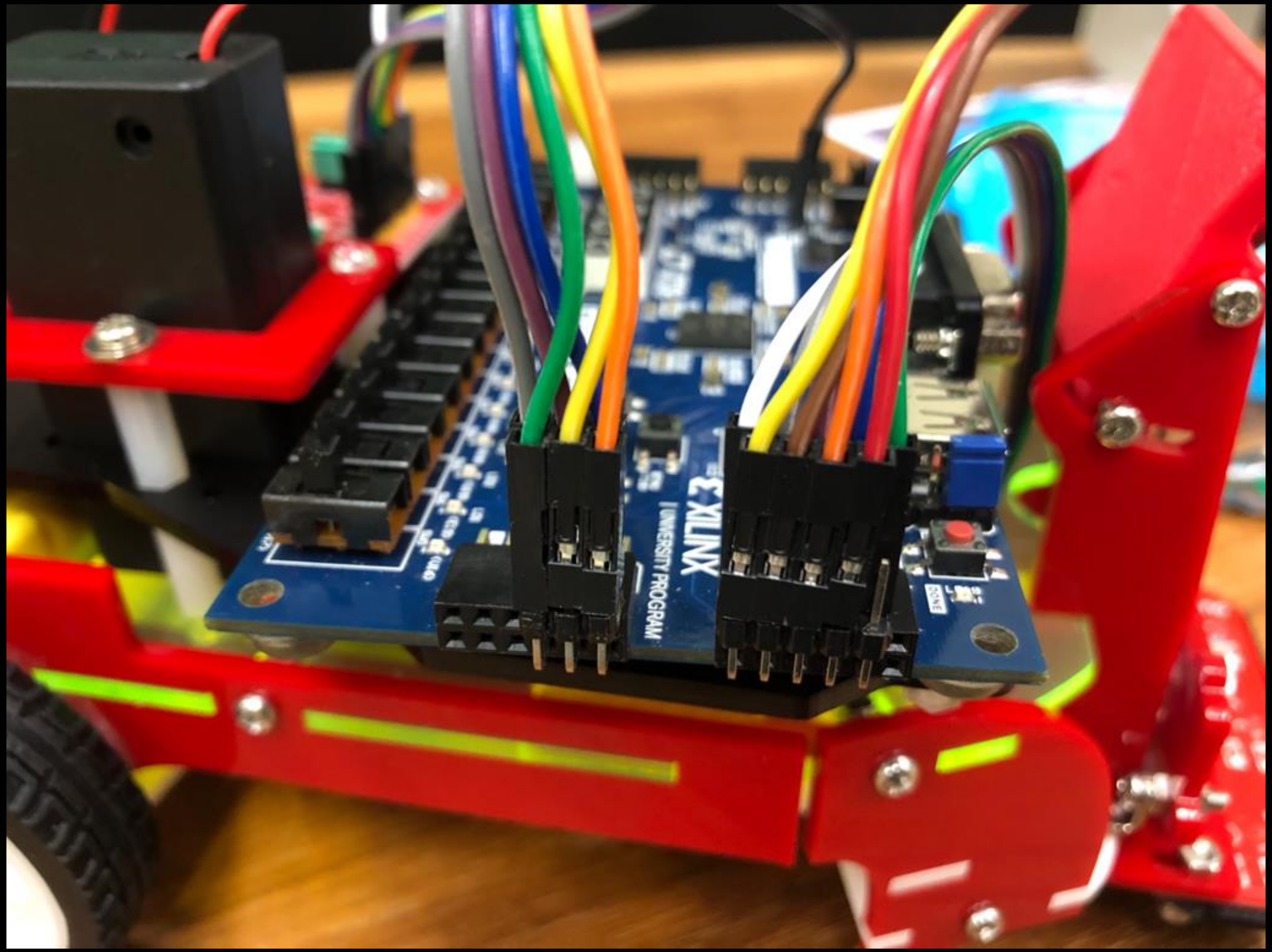
- connect motor driver and FPGA





- connect HC-SR04





- connect line tracker