物聯網與微處理機系統設計
Internet of Things and Microprocessor System Design

Lecture 09 – Autonomous Car

Lecturer: 陳彥安 Chen, Yan-Ann

YZU CSE

Outline

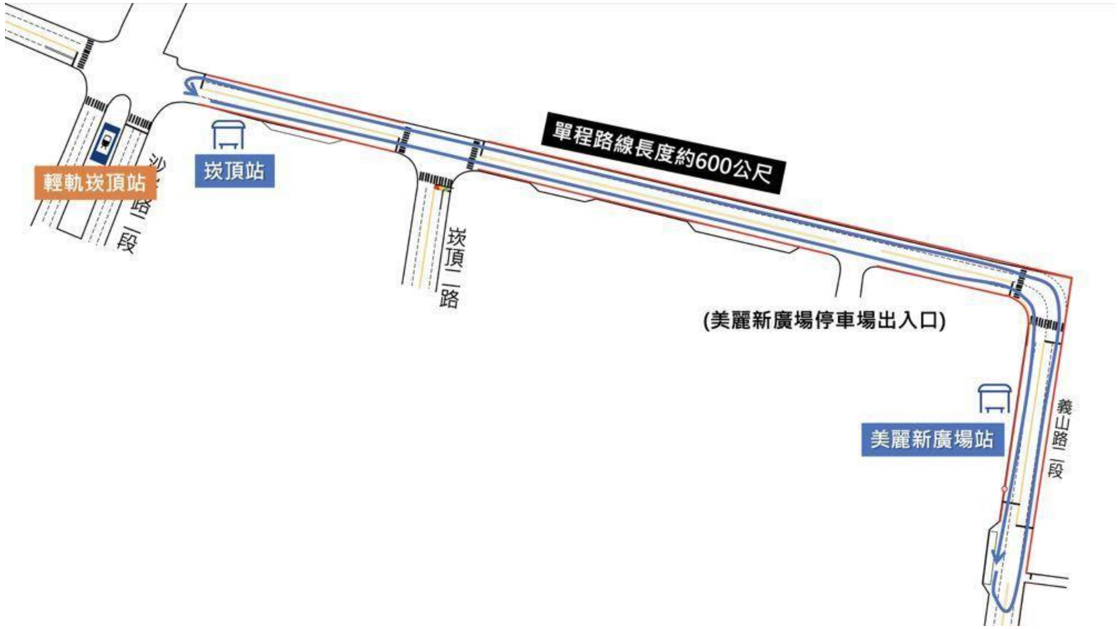
- Introduction
- Motor control
- Moving control
- Follower Car

Outline

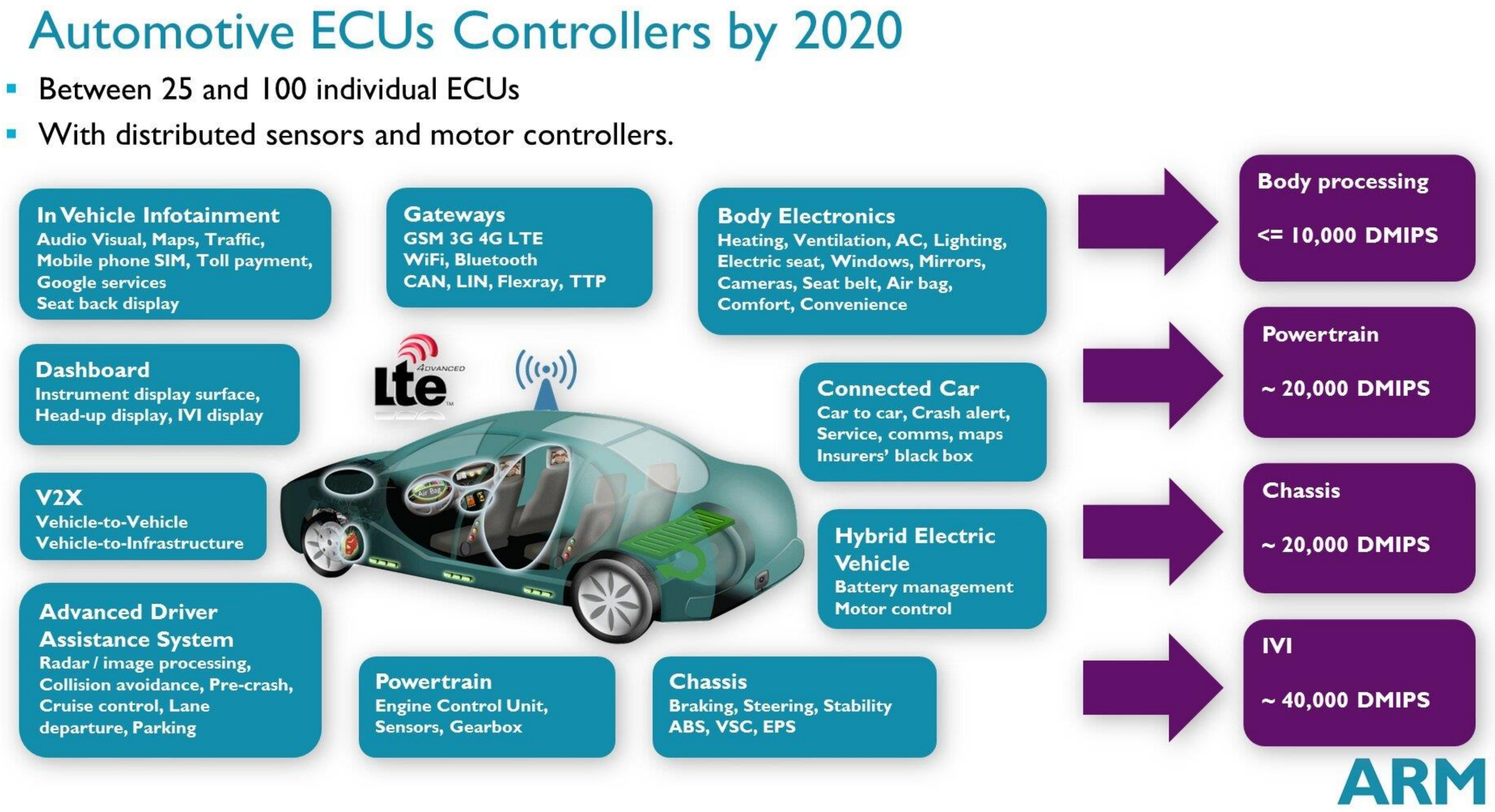
- Introduction
- Motor control
- Moving control
- Follower Car

Self-driving Car

- https://smartbus.dev.flyelephant.com.tw/



Electronic Controller Unit



Automotive ECUs Controllers by 2020

- Between 25 and 100 individual ECUs
- With distributed sensors and motor controllers.

In Vehicle Infotainment
Audio Visual, Maps, Traffic, Mobile phone SIM, Toll payment, Google services
Seat back display

Gateways
GSM 3G 4G LTE
WiFi, Bluetooth
CAN, LIN, Flexray, TTP

Body Electronics
Heating, Ventilation, AC, Lighting, Electric seat, Windows, Mirrors, Cameras, Seat belt, Air bag, Comfort, Convenience

Dashboard
Instrument display surface, Head-up display, IVI display

Connected Car
Car to car, Crash alert, Service, comms, maps
Insurers' black box

V2X
Vehicle-to-Vehicle
Vehicle-to-Infrastructure

Hybrid Electric Vehicle
Battery management
Motor control

Advanced Driver Assistance System
Radar / image processing, Collision avoidance, Pre-crash, Cruise control, Lane departure, Parking

Powertrain
Engine Control Unit, Sensors, Gearbox

Chassis
Braking, Steering, Stability
ABS, VSC, EPS

Body processing
<= 10,000 DMIPS

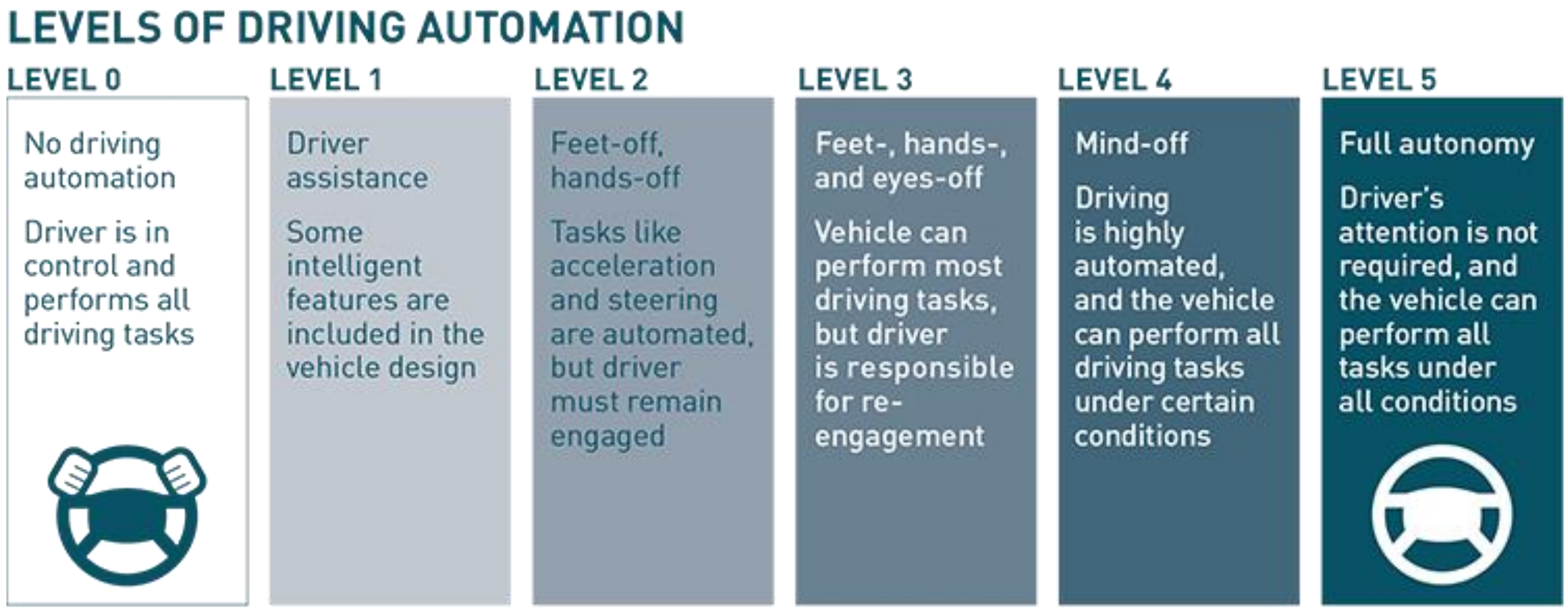
Powertrain
~ 20,000 DMIPS

Chassis
~ 20,000 DMIPS

IVI
~ 40,000 DMIPS

Self-driving Car

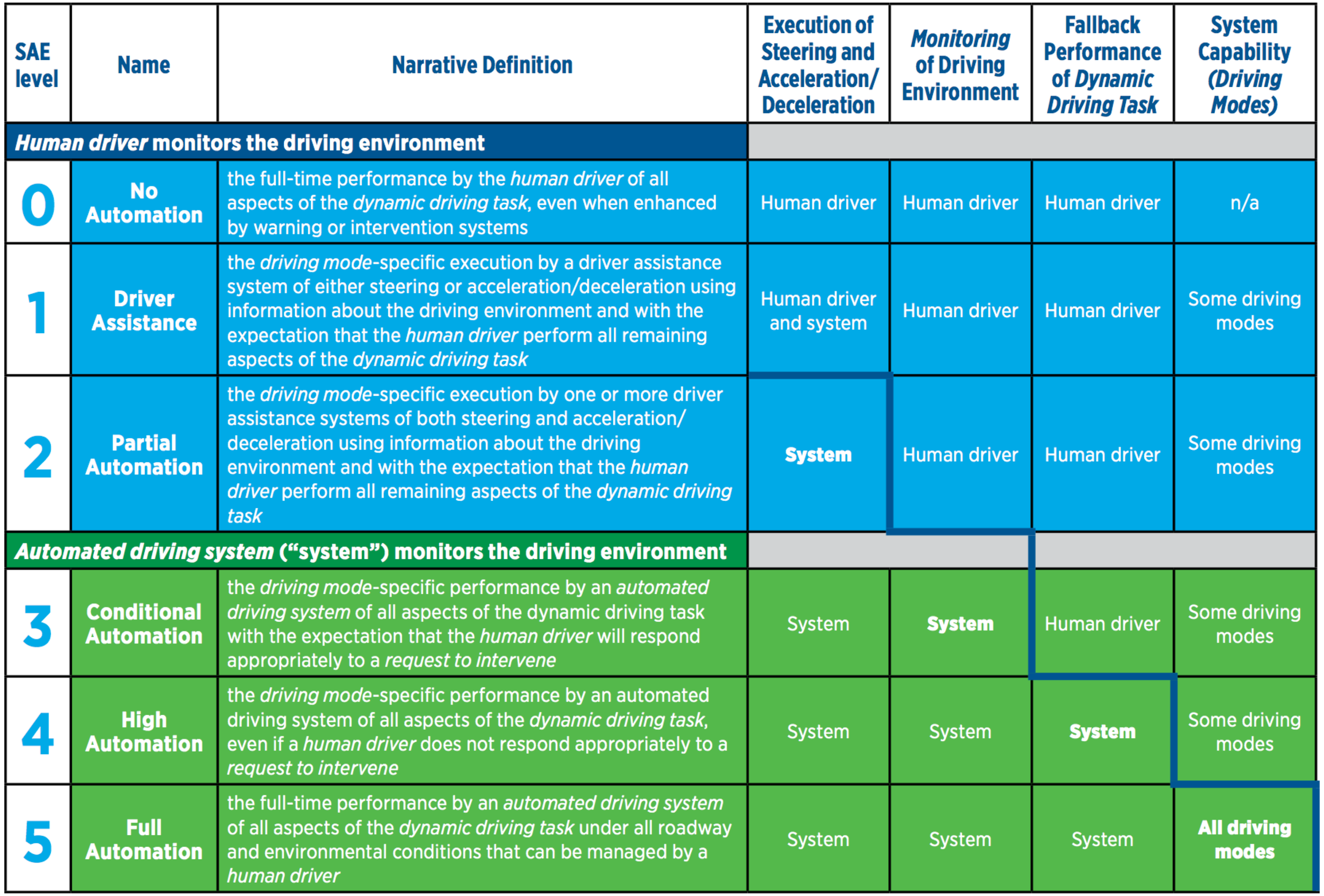
- Levels of driving automation defined by SAE (Society of Automotive Engineers)



LEVELS OF DRIVING AUTOMATION

LEVEL 0	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
No driving automation	Driver assistance	Feet-off, hands-off	Feet-, hands-, and eyes-off	Mind-off	Full autonomy
Driver is in control and performs all driving tasks	Some intelligent features are included in the vehicle design	Tasks like acceleration and steering are automated, but driver must remain engaged	Vehicle can perform most driving tasks, but driver is responsible for re-engagement	Driving is highly automated, and the vehicle can perform all driving tasks under certain conditions	Driver's attention is not required, and the vehicle can perform all tasks under all conditions

Src: https://www.osa-opn.org/home/articles/volume_30/september_2019/features/integrated_lidar_transforming_transportation/

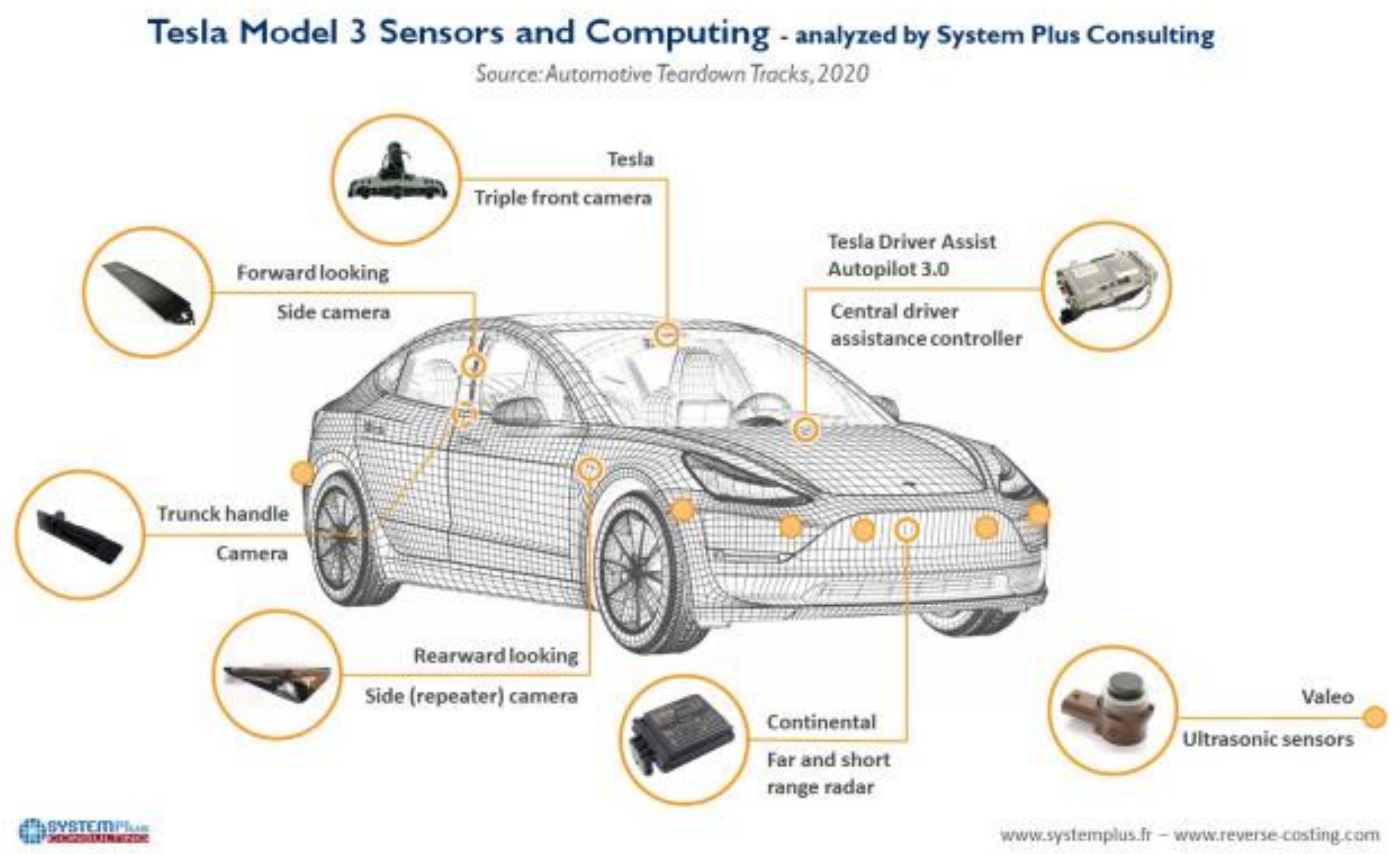


SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
***Human driver* monitors the driving environment**						
0	**No Automation**	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	**Driver Assistance**	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	**Partial Automation**	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	**Conditional Automation**	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	**High Automation**	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	**Full Automation**	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Tesla Autopilot

- https://youtu.be/tlThdr3O5Qo

Tesla Sensors



Tesla Model 3 Sensors and Computing - analyzed by System Plus Consulting

Source: Automotive Teardown Tracks, 2020

Tesla
Triple front camera

Forward looking
Side camera

Tesla Driver Assist
Autopilot 3.0

Central driver
assistance controller

Trunck handle
Camera

Rearward looking
Side (repeater) camera

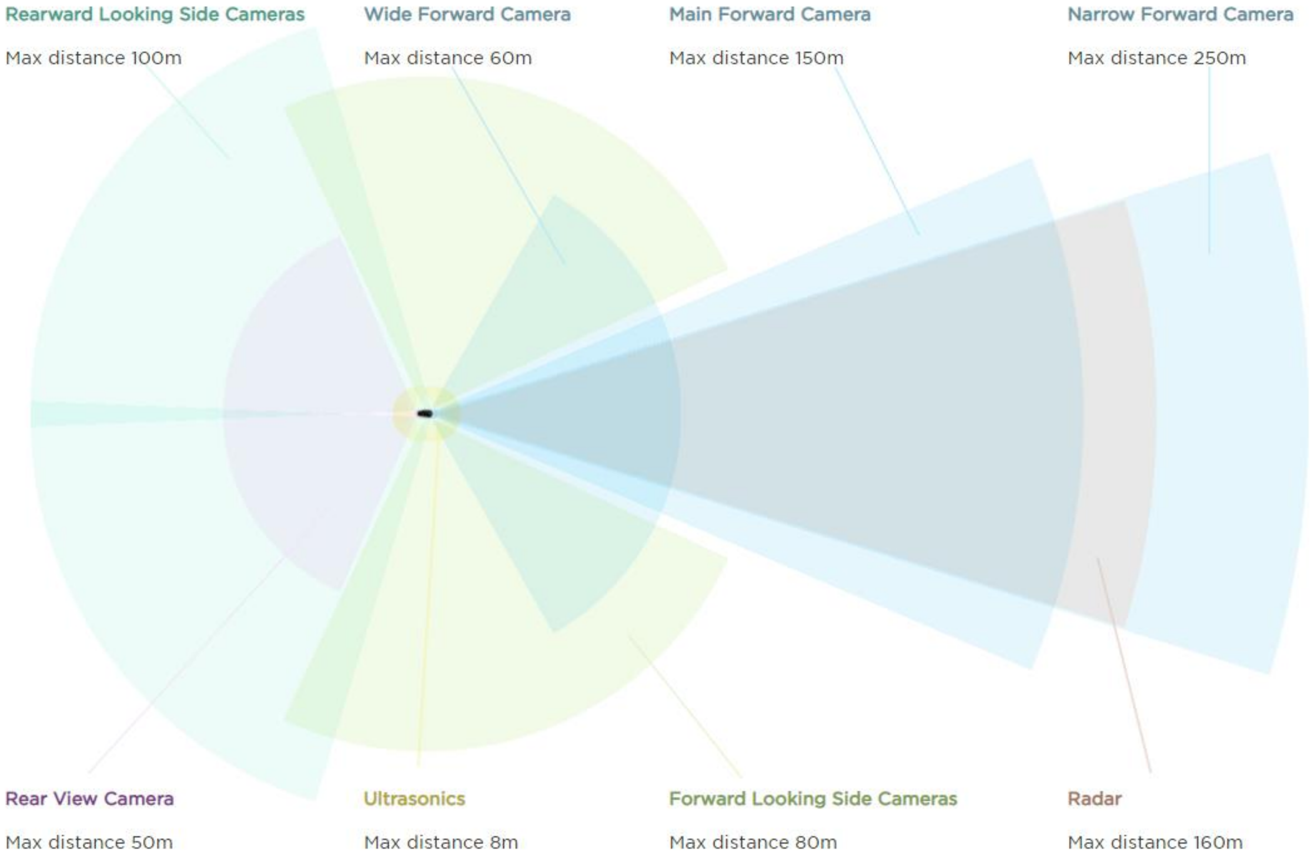
Continental
Far and short
range radar

Valeo
Ultrasonic sensors

www.systemplus.fr – www.reverse-costing.com

Tesla Sensors

- Eight surround cameras
  - Provide 360 degrees of visibility around the car at up to 250 meters of range.
- Twelve ultrasonic sensors
  - Allowing for detection of both hard and soft objects
- A forward-facing radar with enhanced processing
  - Be able to see through heavy rain, fog, dust and even the car ahead.



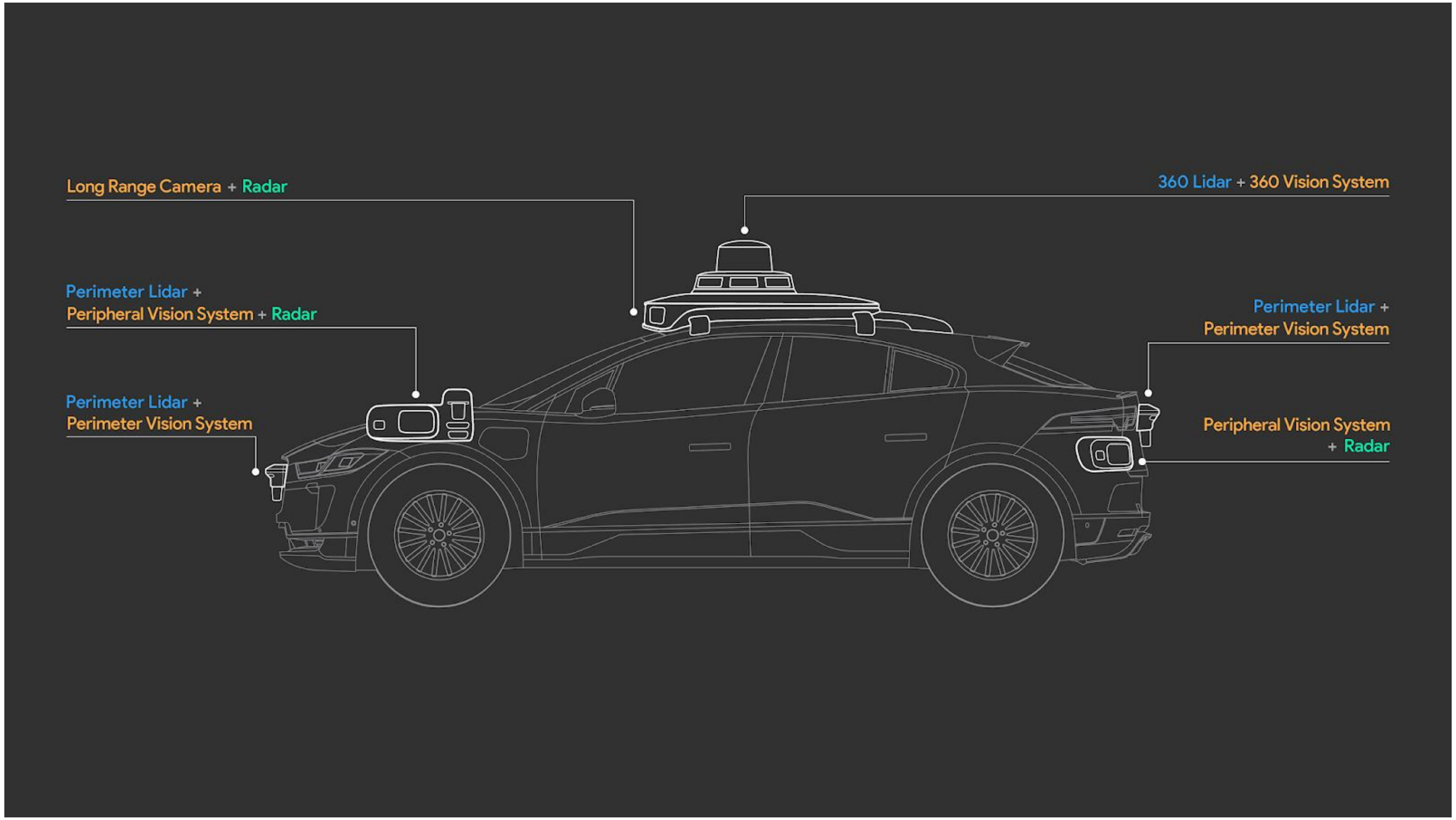
Waymo

- Waymo began as the Google self-driving car project in 2009.
- Make it safe and easy to get around - without the need for anyone in the driver’s seat.
- Waymo driver

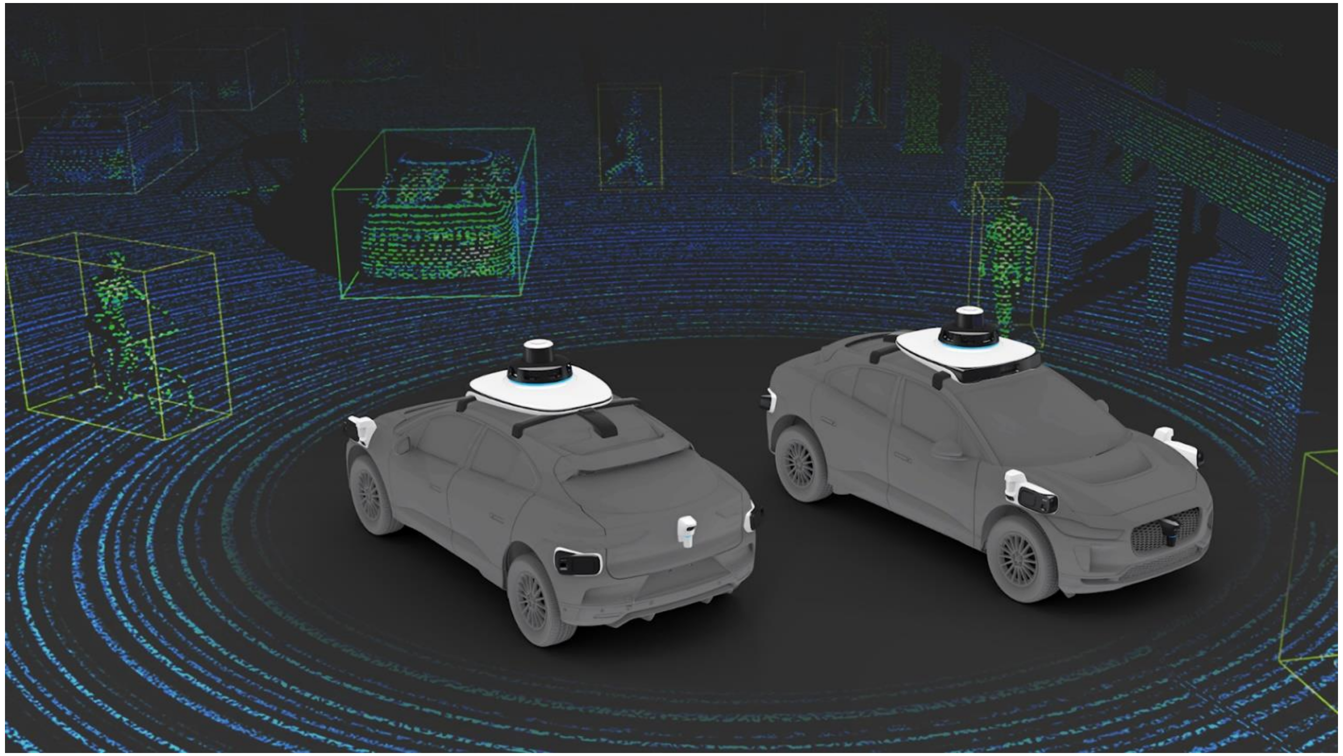
Waymo

- Waymo Celebrates 10 Million Miles of Self-Driving
  - https://youtu.be/ROAwXEqDk7k

Waymo Driver



360 Lidar

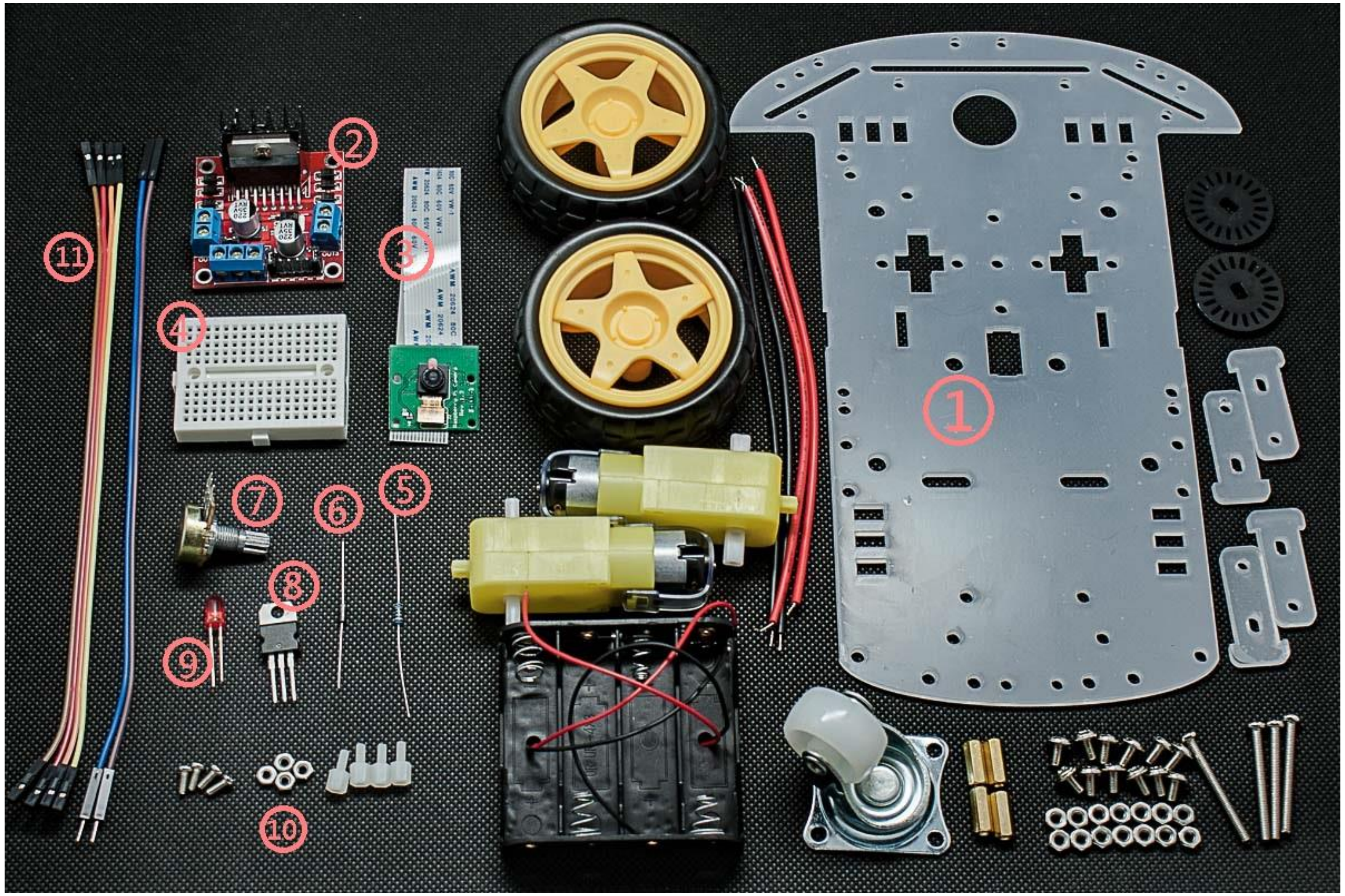


Outline

- Introduction
- Motor control
- Moving control
- Follower Car

Follower-Car

Components

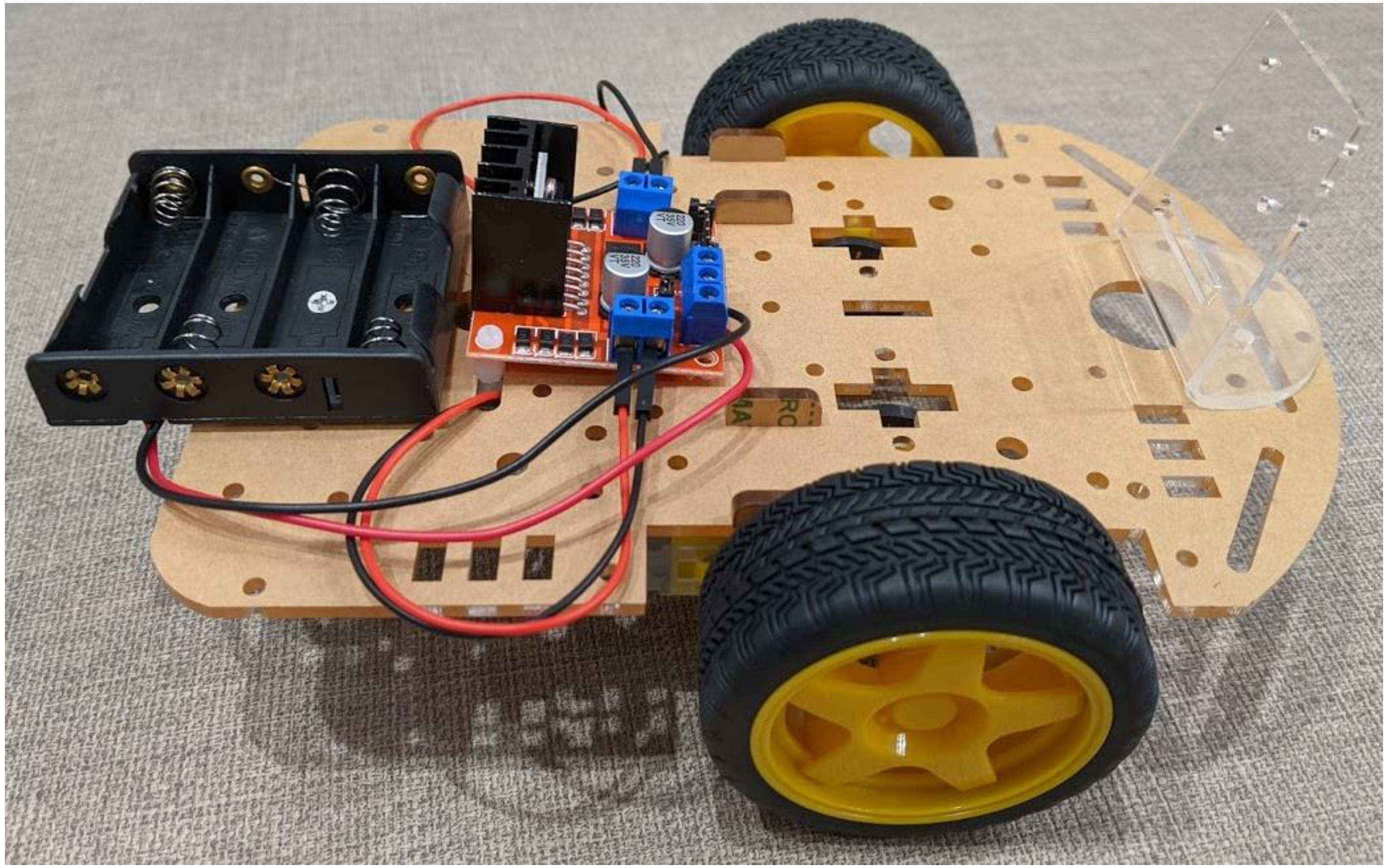


《規格》

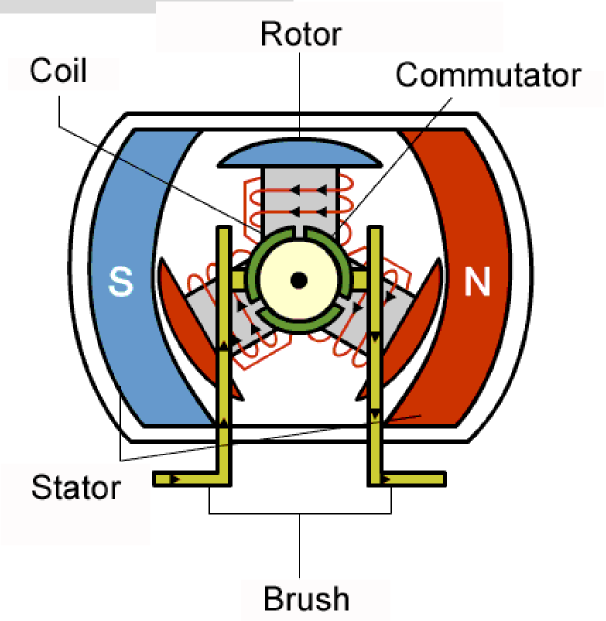
1. 單層自走車底盤(含二輪跑車胎 + 一萬向輪 + 二馬達 + 螺絲組) x1
2. L298N 馬達驅動板 x1
3. 5MP Camera for Raspberry Pi x1
4. 170 洞小型麵包板 x1
5. 1KΩ 電阻(1/4W) x1
6. 1N4004 二極體 x1
7. 16m/m 可變電阻 10Kx1
8. TIP120 電晶體 x1
9. 5mm LED x1
10. 架高螺絲組(螺絲母x4 + 塑膠架高螺絲 x4 + 圓頭螺絲 x4) x1
11. 公對母排線(20cm)x2, 母對母排線(20cm)x4

Car Frame

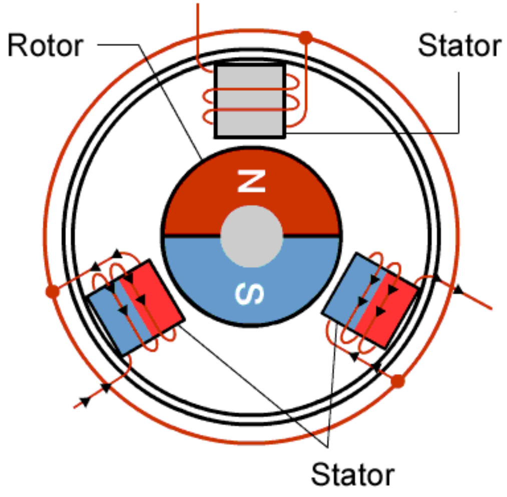
- The car frame that you will use.



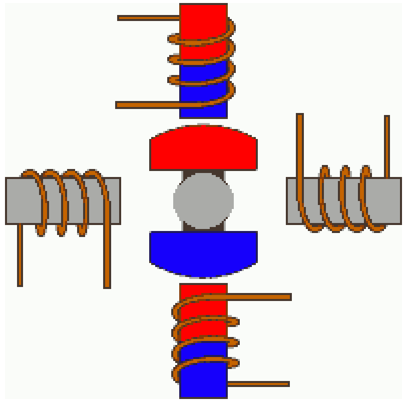
Motors



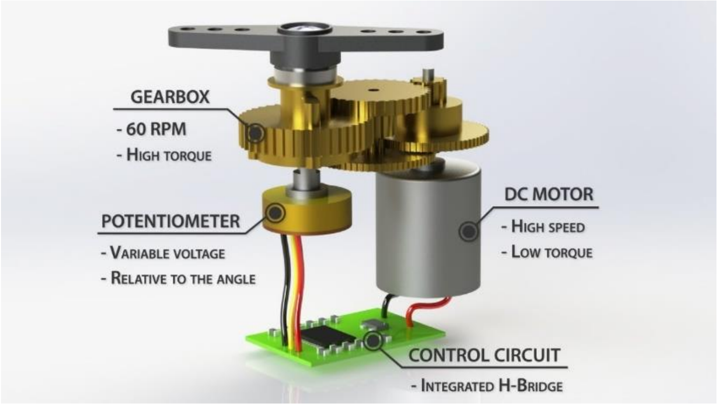
brush DC motor



brushless DC motor



stepper motor

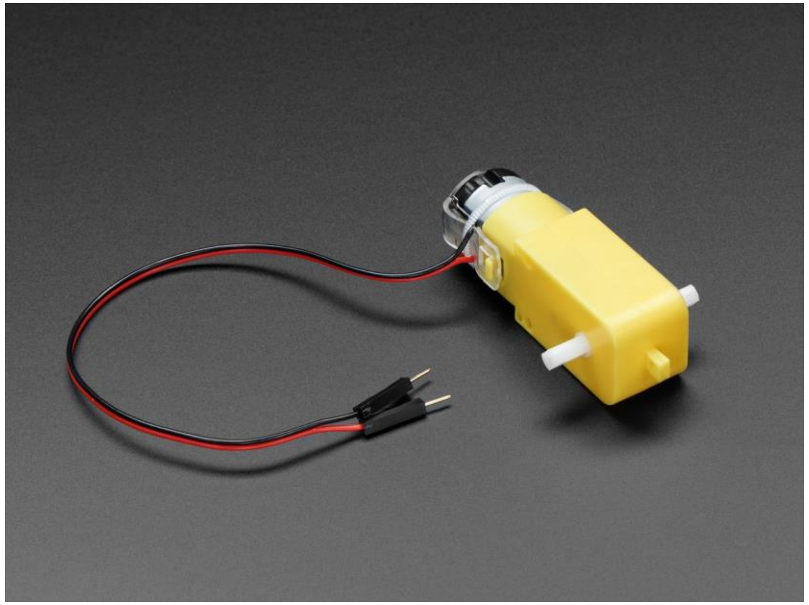


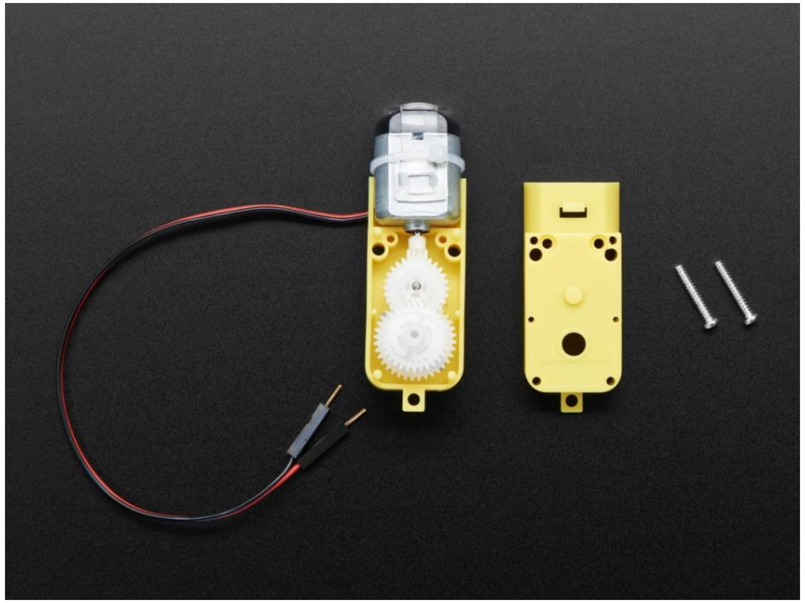
servo motor

ref: https://www.seeedstudio.com/blog/2019/04/01/choosing-the-right-motor-for-your-project-dc-vs-stepper-vs-servo-motors/

DC Gearbox Motor

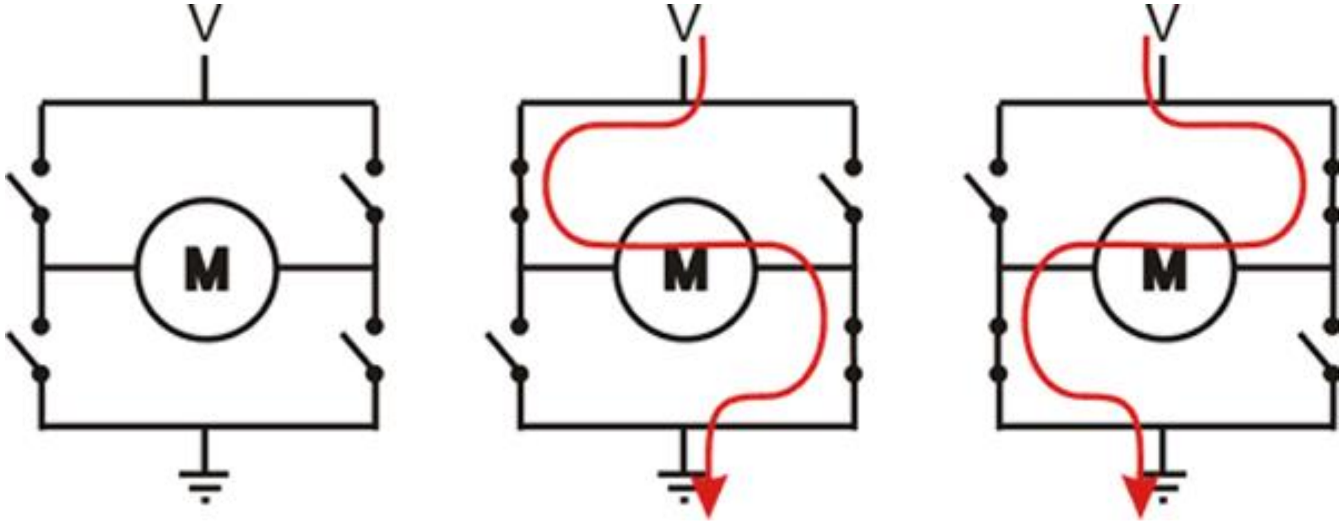
- TT DC Gearbox Motor with a gear ratio of 1:48, and it comes with 2 x 200mm wires with breadboard-friendly 0.1" male connectors.
  - https://www.adafruit.com/product/3777
- At 3VDC we measured 150mA @ 120 RPM no-load, and 1.1 Amps when stalled.
- At 4.5VDC we measured 155mA @ 185 RPM no-load, and 1.2 Amps when stalled.
- At 6VDC we measured 160mA @ 250 RPM no-load, and 1.5 Amps when stalled.

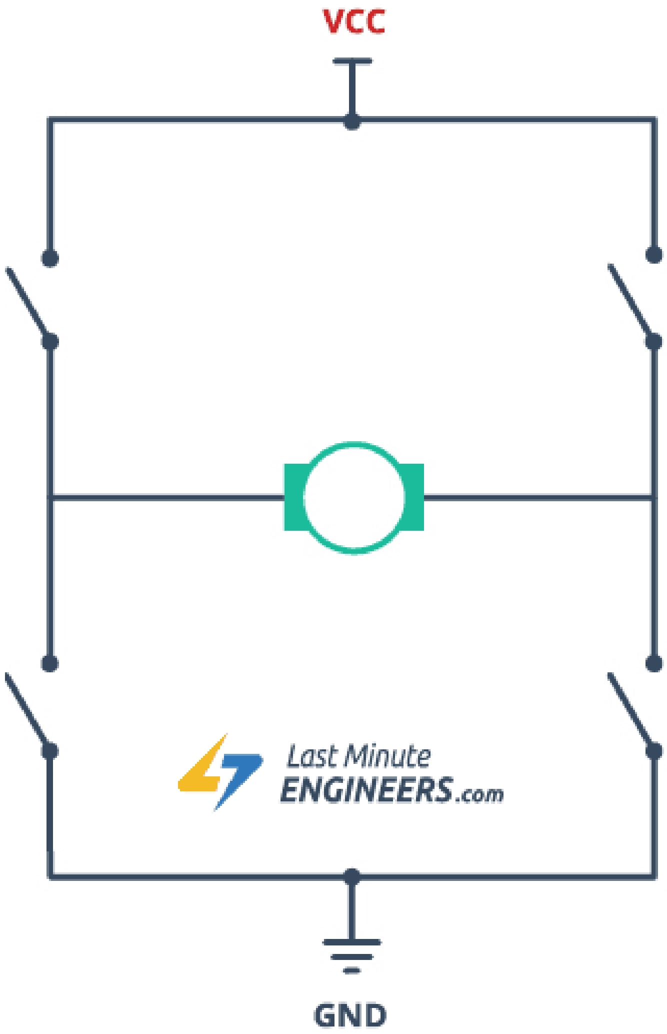




H-bridge

- Switch the polarity of a voltage applied to a load

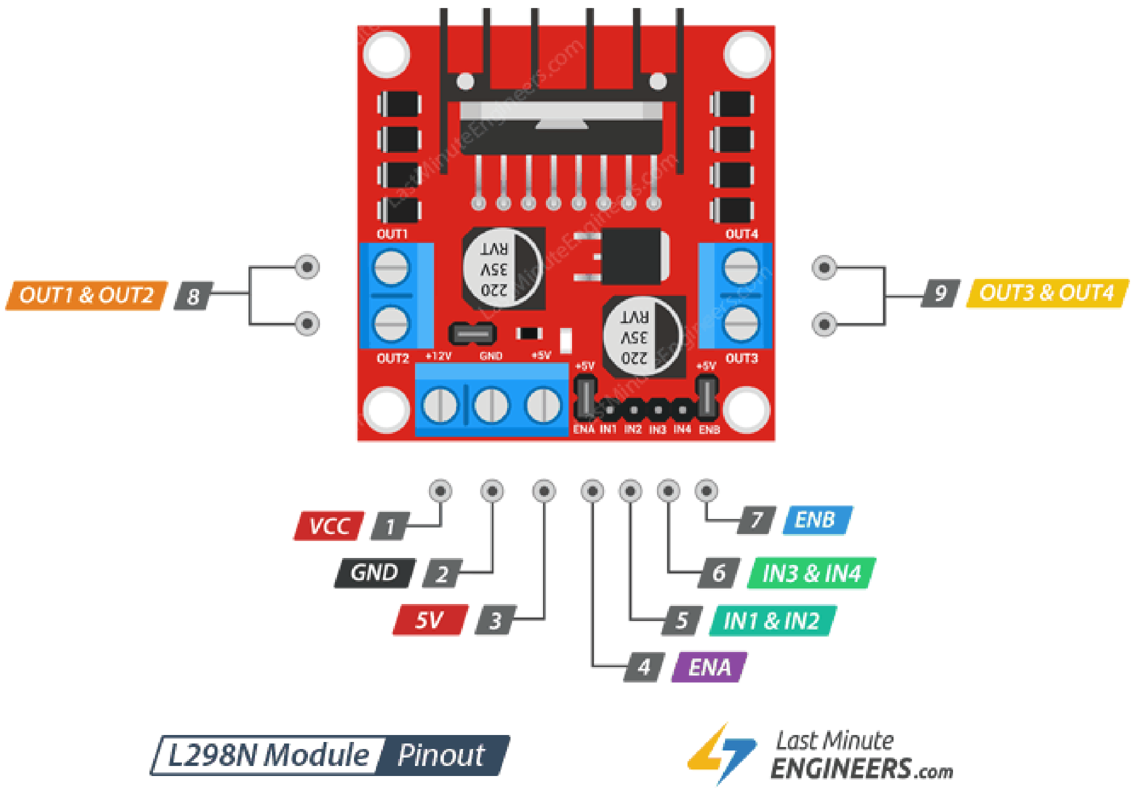




ref: https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

Dual H-Bridge Motor Driver

- L298N



VCC pin supplies power for the motor. It can be anywhere between 5 to 35V. Remember, if the 5V-EN jumper is in place, you need to supply 2 extra volts than motor's actual voltage requirement, in order to get maximum speed out of your motor.

GND is a common ground pin.

5V pin supplies power for the switching logic circuitry inside L298N IC. If the 5V-EN jumper is in place, this pin acts as an output and can be used to power up your Arduino. If the 5V-EN jumper is removed, you need to connect it to the 5V pin on Arduino.

ENA pins are used to control speed of Motor A. Pulling this pin HIGH(Keeping the jumper in place) will make the Motor A spin, pulling it LOW will make the motor stop. Removing the jumper and connecting this pin to PWM input will let us control the speed of Motor A.

IN1 & IN2 pins are used to control spinning direction of Motor A. When one of them is HIGH and other is LOW, the Motor A will spin. If both the inputs are either HIGH or LOW the Motor A will stop.

IN3 & IN4 pins are used to control spinning direction of Motor B. When one of them is HIGH and other is LOW, the Motor B will spin. If both the inputs are either HIGH or LOW the Motor B will stop.

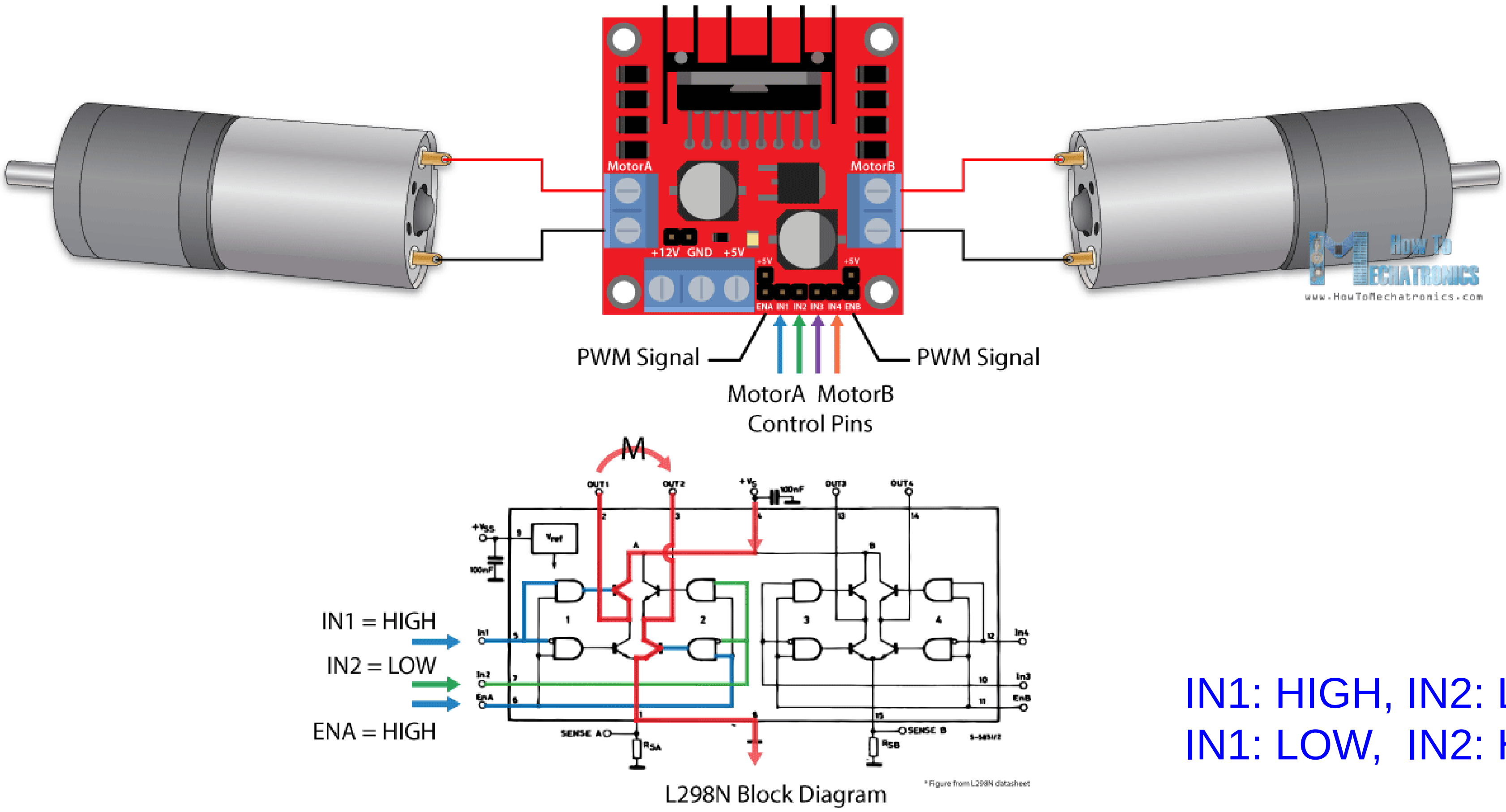
ENB pins are used to control speed of Motor B. Pulling this pin HIGH(Keeping the jumper in place) will make the Motor B spin, pulling it LOW will make the motor stop. Removing the jumper and connecting this pin to PWM input will let us control the speed of Motor B.

OUT1 & OUT2 pins are connected to Motor A.

OUT3 & OUT4 pins are connected to Motor B.

ref: https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

Dual H-Bridge Motor Driver

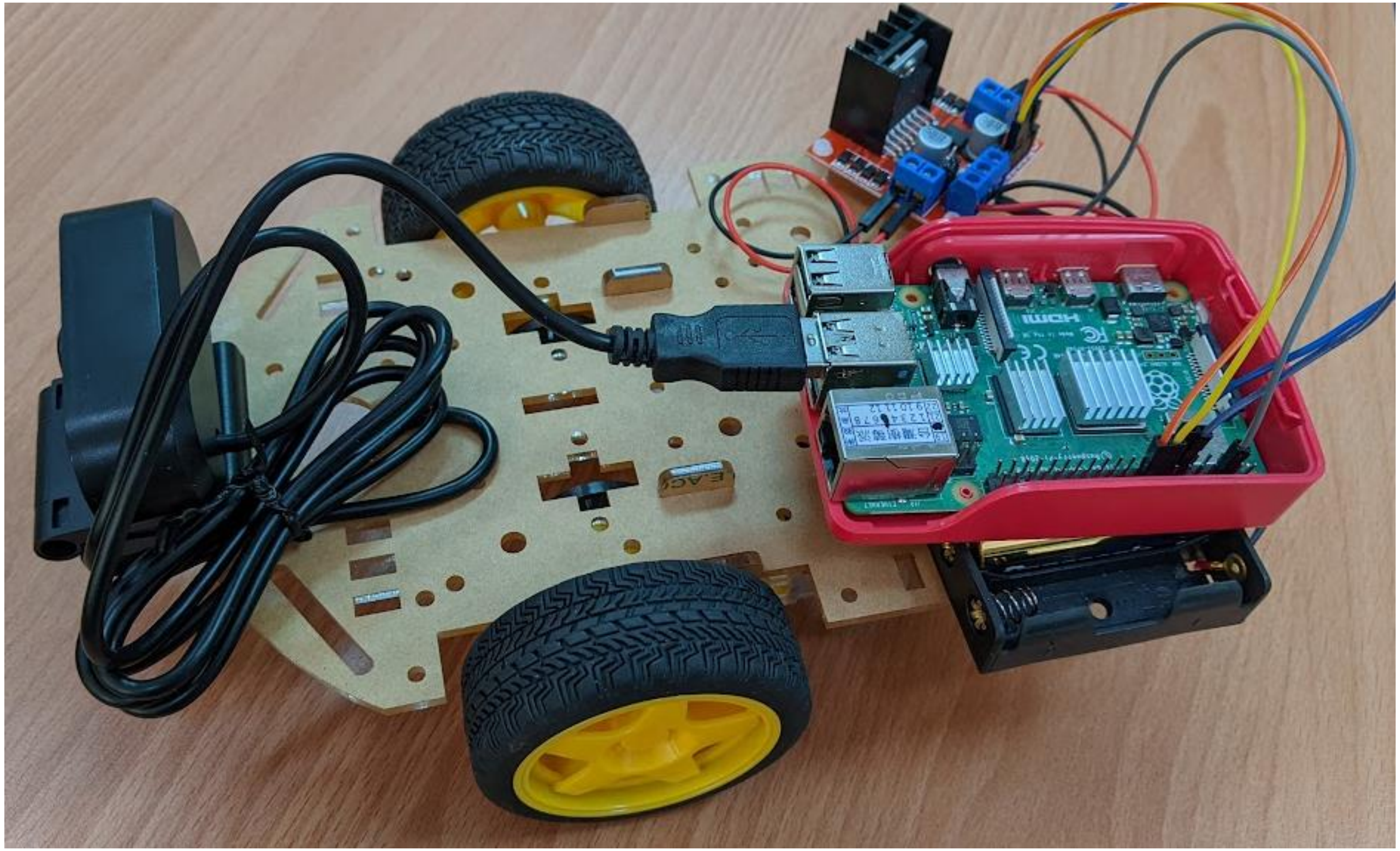


L298N Block Diagram

IN1: HIGH, IN2: LOW => Clockwise
IN1: LOW, IN2: HIGH => Counter clockwise

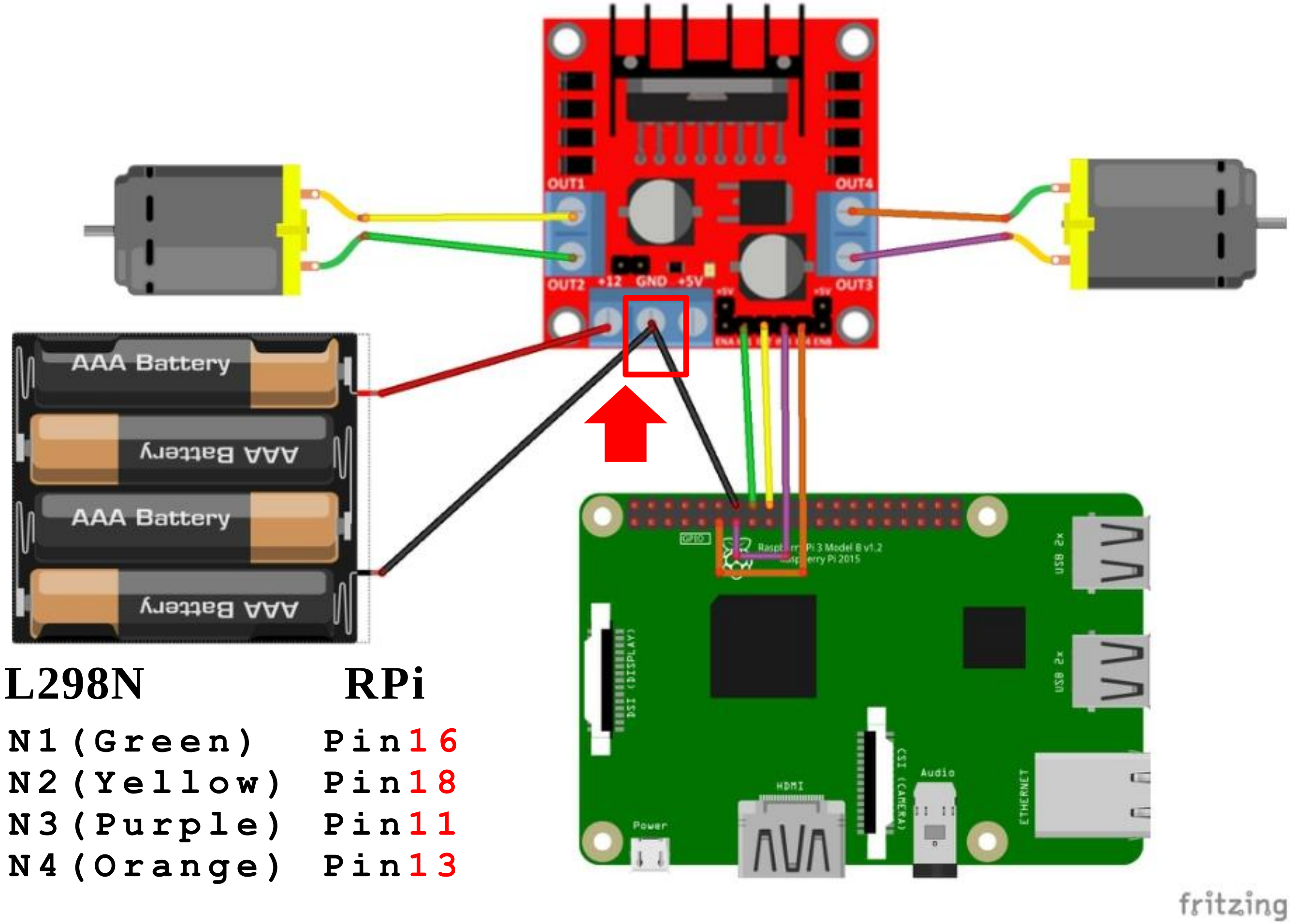
ref: https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/

Assembly

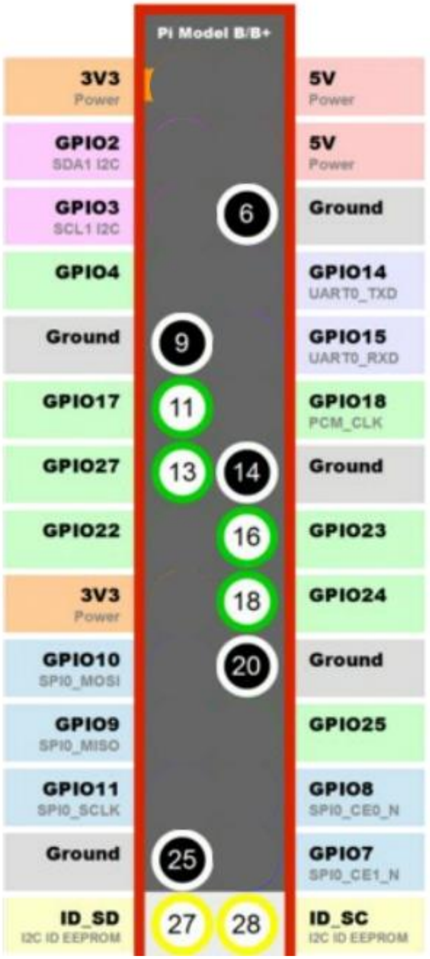


Wiring

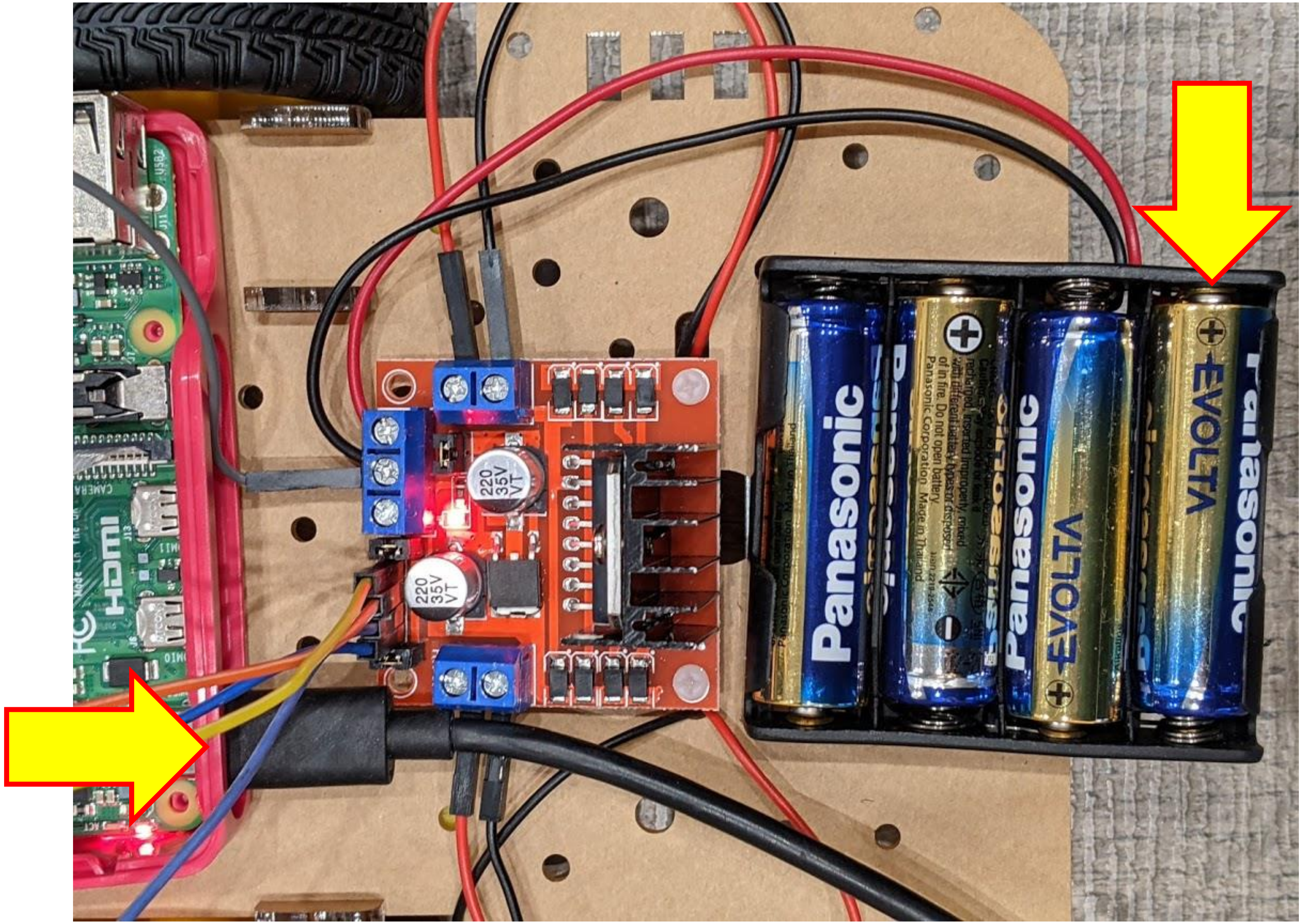
- Wire IN1~4 of L298N to RPi
- Wire GNDs



L298N	RPi
IN1(Green)	Pin16
IN2(Yellow)	Pin18
IN3(Purple)	Pin11
IN4(Orange)	Pin13



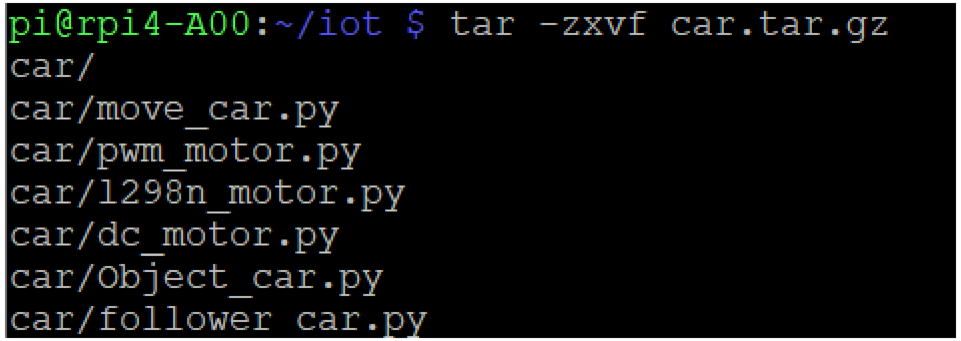
Powering On



Sample Codes

```
$ wget https://github.com/yachentw/yzucseiot/raw/main/lec09/car.tar.gz
```

```
$ tar -zxvf car.tar.gz
```



```
pi@rpi4-A00:~/iot $ tar -zxvf car.tar.gz
car/
car/move_car.py
car/pwm_motor.py
car/l298n_motor.py
car/dc_motor.py
car/Object_car.py
car/follower_car.py
```

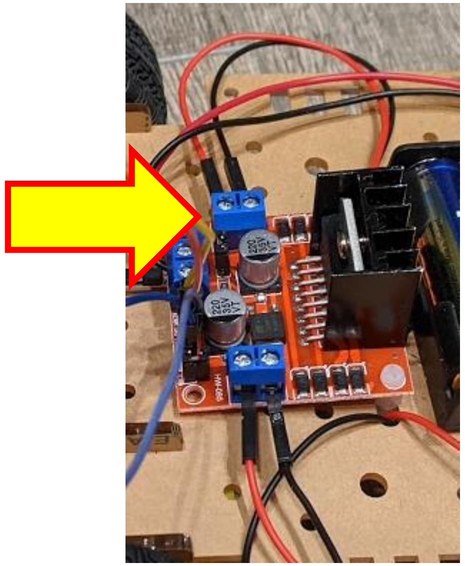
```
$ cd car
```

Motor Test (Right)

- l298n_motor.py

$ python3 l298n_motor.py

- Check if your right wheel rotates forward and the backward?
- If not, swap the wires of the motor.



l298n_motor.py

```
import RPi.GPIO as GPIO
import time

Motor_Pin1 = 16
Motor_Pin2 = 18

GPIO.setmode(GPIO.BOARD)
GPIO.setup(Motor_Pin1, GPIO.OUT)
GPIO.setup(Motor_Pin2, GPIO.OUT)

try:
    GPIO.output(Motor_Pin1, True)     # clockwise
    time.sleep(3)
    GPIO.output(Motor_Pin1, False)

    time.sleep(1)                     # protect motor

    GPIO.output(Motor_Pin2, True)     # counterclockwise
    time.sleep(3)
    GPIO.output(Motor_Pin2, False)

finally:
    GPIO.cleanup()
```

Motor Test (Left)

- Set Pin1 and Pin2 to 11 and 13.

```
Motor_Pin1 = 11
Motor_Pin2 = 13
```

```
$ python3 l298n_motor.py
```

- Check if your left wheel rotates forward and the backward?
- If not, swap the wires of the motor.

Outline

- Introduction
- Motor control
- Moving control
- Follower Car

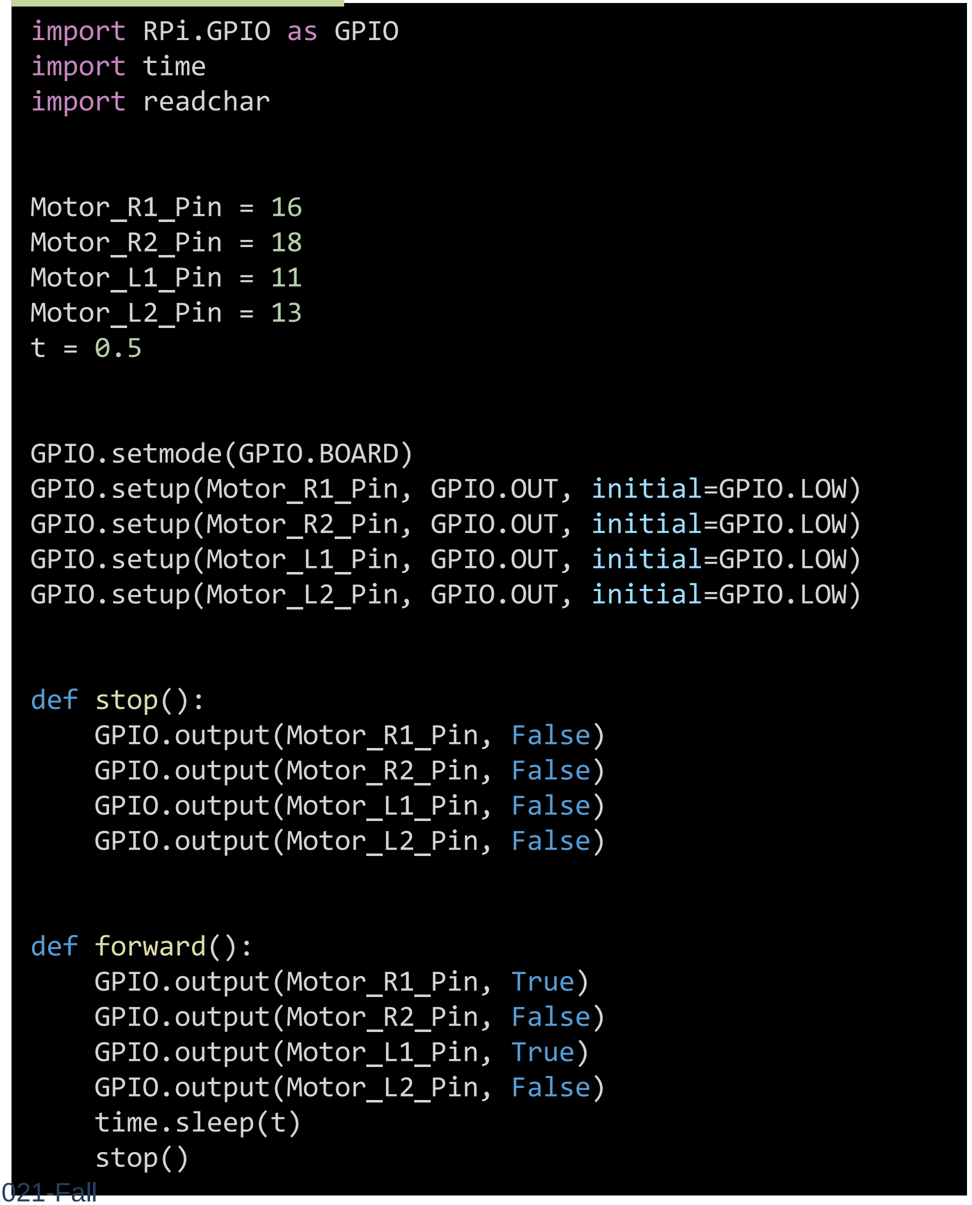
Moving Control

- Use keyboard input to control the movement of your car
  - w: move forward
  - s: move backward
  - d: turn right
  - a: turn left

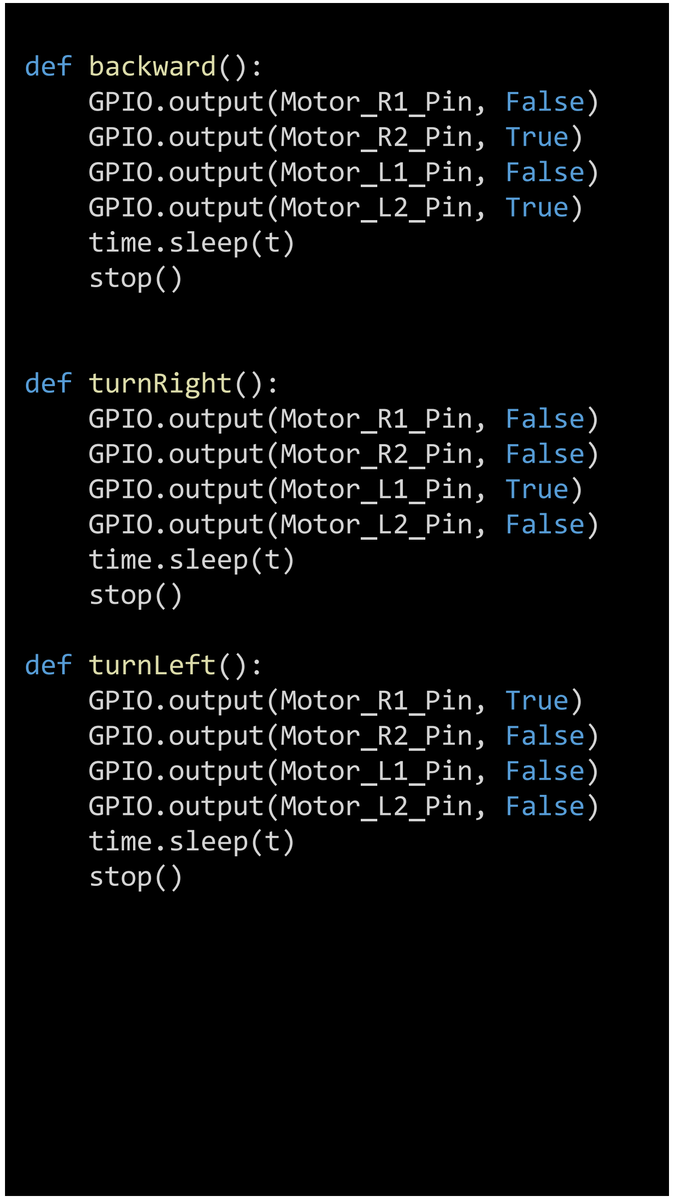
```
$ pip3 install readchar
$ python3 move_car.py
```

- Test if the car is moving toward the right direction.

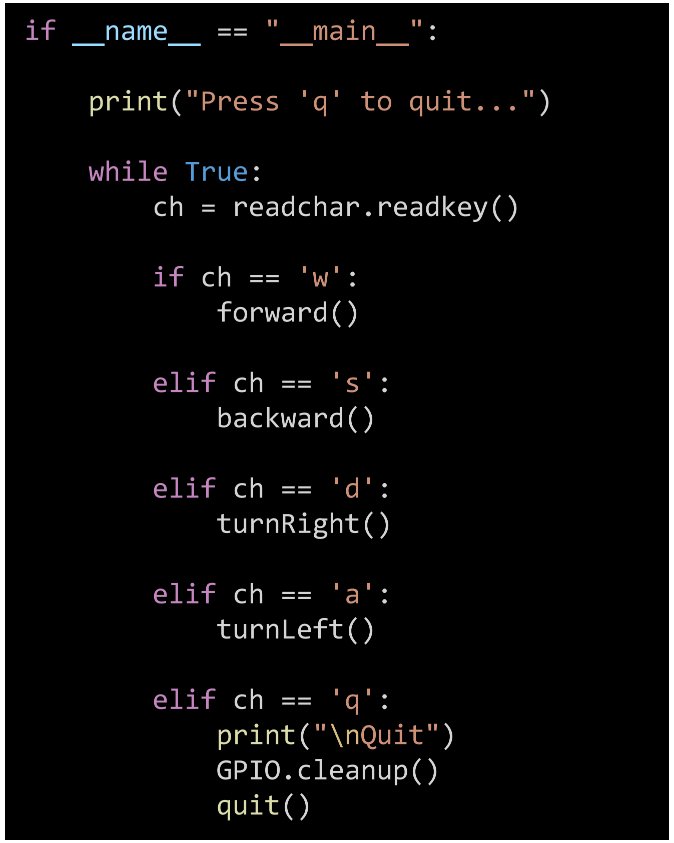
move_car.py



```python
import RPi.GPIO as GPIO
import time
import readchar


Motor_R1_Pin = 16
Motor_R2_Pin = 18
Motor_L1_Pin = 11
Motor_L2_Pin = 13
t = 0.5


GPIO.setmode(GPIO.BOARD)
GPIO.setup(Motor_R1_Pin, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(Motor_R2_Pin, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(Motor_L1_Pin, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(Motor_L2_Pin, GPIO.OUT, initial=GPIO.LOW)


def stop():
    GPIO.output(Motor_R1_Pin, False)
    GPIO.output(Motor_R2_Pin, False)
    GPIO.output(Motor_L1_Pin, False)
    GPIO.output(Motor_L2_Pin, False)


def forward():
    GPIO.output(Motor_R1_Pin, True)
    GPIO.output(Motor_R2_Pin, False)
    GPIO.output(Motor_L1_Pin, True)
    GPIO.output(Motor_L2_Pin, False)
    time.sleep(t)
    stop()
```



```python
def backward():
    GPIO.output(Motor_R1_Pin, False)
    GPIO.output(Motor_R2_Pin, True)
    GPIO.output(Motor_L1_Pin, False)
    GPIO.output(Motor_L2_Pin, True)
    time.sleep(t)
    stop()


def turnRight():
    GPIO.output(Motor_R1_Pin, False)
    GPIO.output(Motor_R2_Pin, False)
    GPIO.output(Motor_L1_Pin, True)
    GPIO.output(Motor_L2_Pin, False)
    time.sleep(t)
    stop()

def turnLeft():
    GPIO.output(Motor_R1_Pin, True)
    GPIO.output(Motor_R2_Pin, False)
    GPIO.output(Motor_L1_Pin, False)
    GPIO.output(Motor_L2_Pin, False)
    time.sleep(t)
    stop()
```



```python
if __name__ == "__main__":

    print("Press 'q' to quit...")

    while True:
        ch = readchar.readkey()

        if ch == 'w':
            forward()

        elif ch == 's':
            backward()

        elif ch == 'd':
            turnRight()

        elif ch == 'a':
            turnLeft()

        elif ch == 'q':
            print("\nQuit")
            GPIO.cleanup()
            quit()
```

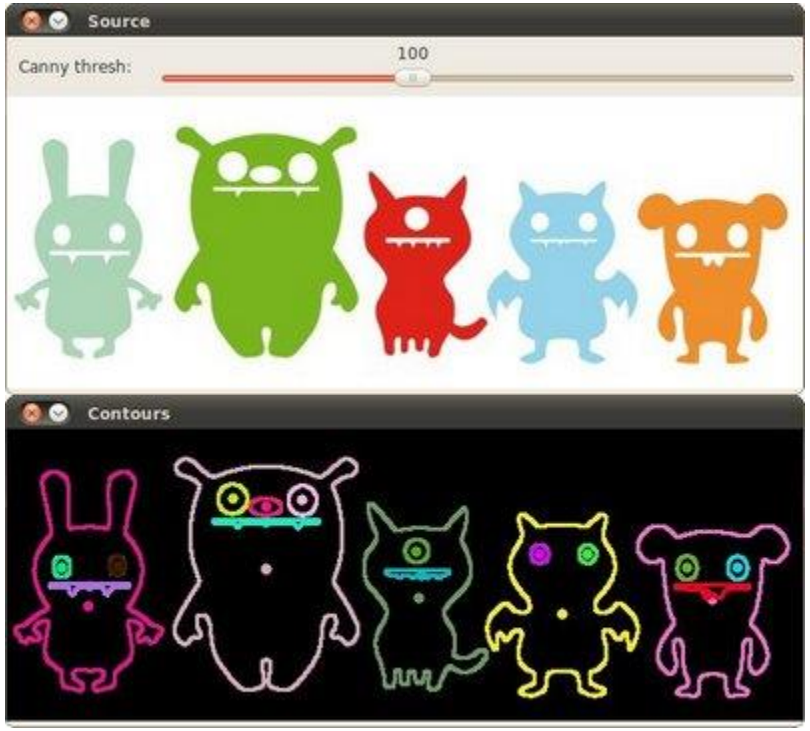
Outline

- Introduction
- Motor control
- Moving control
- Follower Car

Follower Car

- Run in VNC to see the images.
- Find the contours of objects and move toward the center of the object.

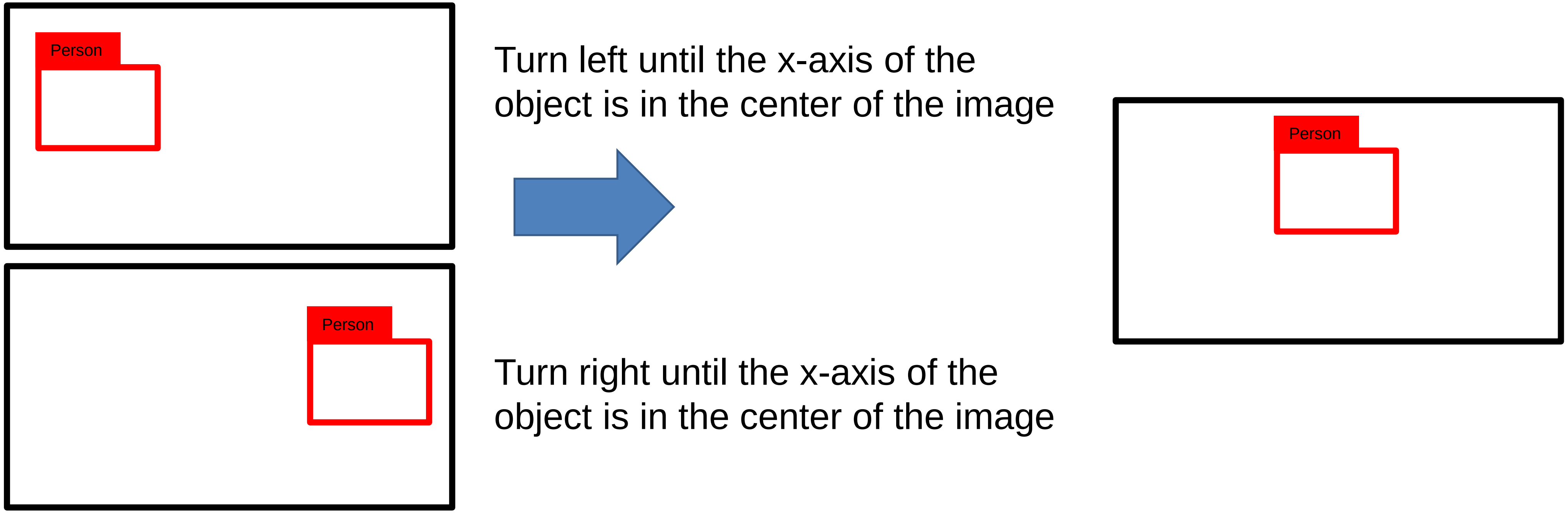
```
$ python3 follower_car.py
```



Advanced Follower Car

- Combine the function of object detection in the last lecture.

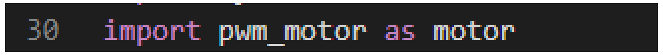
```
$ mv Object_car.py ~/tensorflow1/models/research/object_detection/
$ mv pwm_motor.py ~/tensorflow1/models/research/object_detection/
$ cd ~/tensorflow1/models/research/object_detection
$ python3 Object_car.py
```



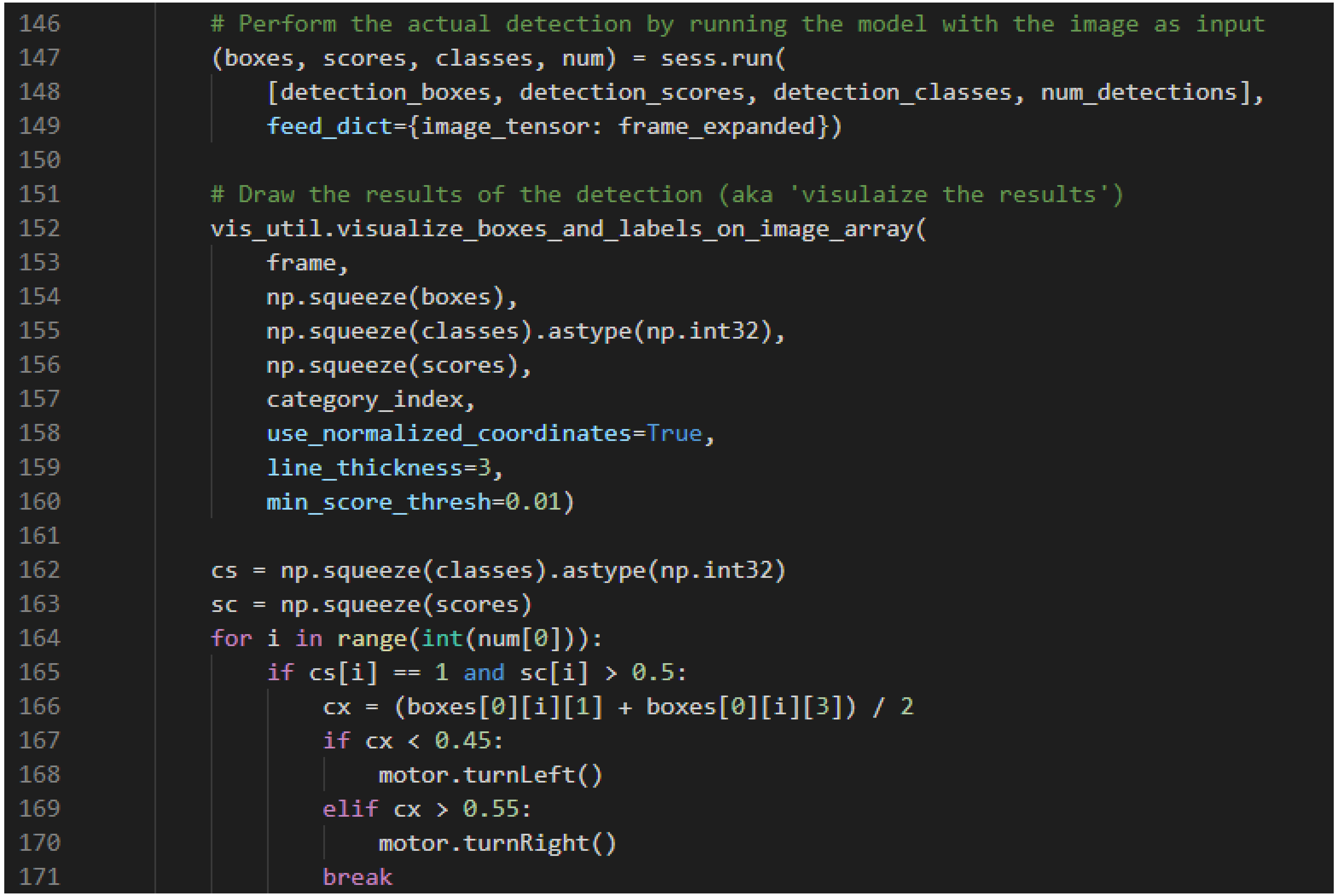
Turn left until the x-axis of the object is in the center of the image

Turn right until the x-axis of the object is in the center of the image

Object_car.py

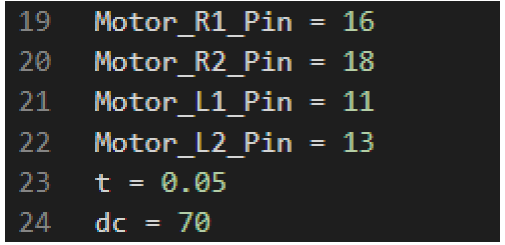


```python
import pwm_motor as motor
```

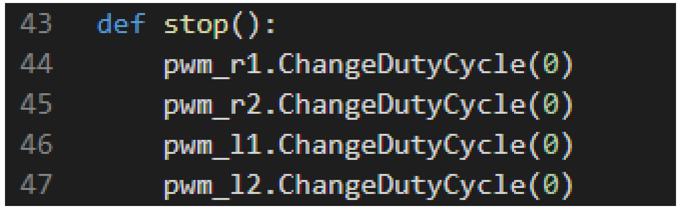


```python
        # Perform the actual detection by running the model with the image as input
        (boxes, scores, classes, num) = sess.run(
            [detection_boxes, detection_scores, detection_classes, num_detections],
            feed_dict={image_tensor: frame_expanded})

        # Draw the results of the detection (aka 'visulaize the results')
        vis_util.visualize_boxes_and_labels_on_image_array(
            frame,
            np.squeeze(boxes),
            np.squeeze(classes).astype(np.int32),
            np.squeeze(scores),
            category_index,
            use_normalized_coordinates=True,
            line_thickness=3,
            min_score_thresh=0.01)

        cs = np.squeeze(classes).astype(np.int32)
        sc = np.squeeze(scores)
        for i in range(int(num[0])):
            if cs[i] == 1 and sc[i] > 0.5:
                cx = (boxes[0][i][1] + boxes[0][i][3]) / 2
                if cx < 0.45:
                    motor.turnLeft()
                elif cx > 0.55:
                    motor.turnRight()
                break
```

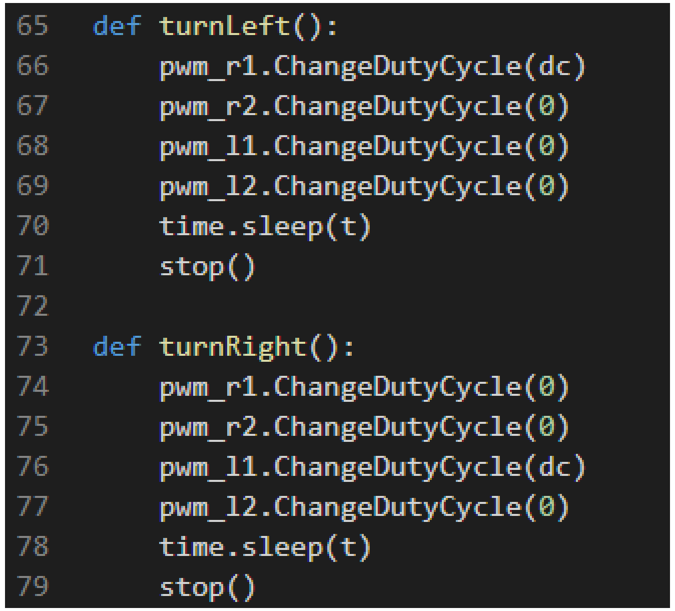
pwm_motor.py



```python
Motor_R1_Pin = 16
Motor_R2_Pin = 18
Motor_L1_Pin = 11
Motor_L2_Pin = 13
t = 0.05
dc = 70
```



```python
def stop():
    pwm_r1.ChangeDutyCycle(0)
    pwm_r2.ChangeDutyCycle(0)
    pwm_l1.ChangeDutyCycle(0)
    pwm_l2.ChangeDutyCycle(0)
```



```python
def turnLeft():
    pwm_r1.ChangeDutyCycle(dc)
    pwm_r2.ChangeDutyCycle(0)
    pwm_l1.ChangeDutyCycle(0)
    pwm_l2.ChangeDutyCycle(0)
    time.sleep(t)
    stop()

def turnRight():
    pwm_r1.ChangeDutyCycle(0)
    pwm_r2.ChangeDutyCycle(0)
    pwm_l1.ChangeDutyCycle(dc)
    pwm_l2.ChangeDutyCycle(0)
    time.sleep(t)
    stop()
```

Obstacle Avoiding Robot

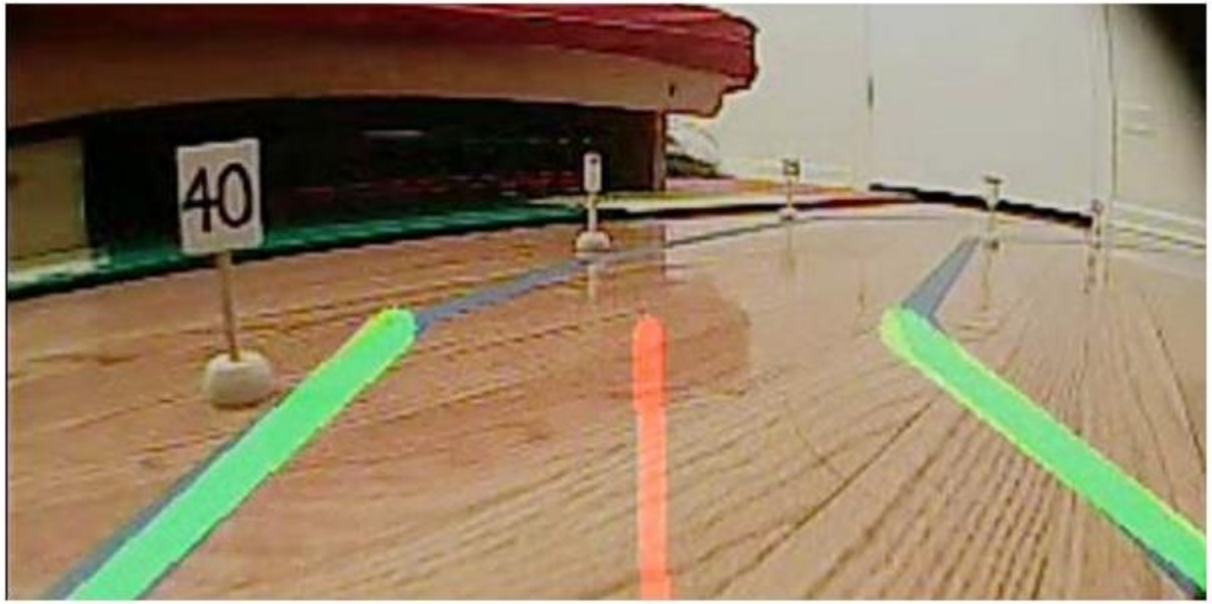
- https://youtu.be/XQuEf6nEoEo

OpenCV Self Driving Car using NN

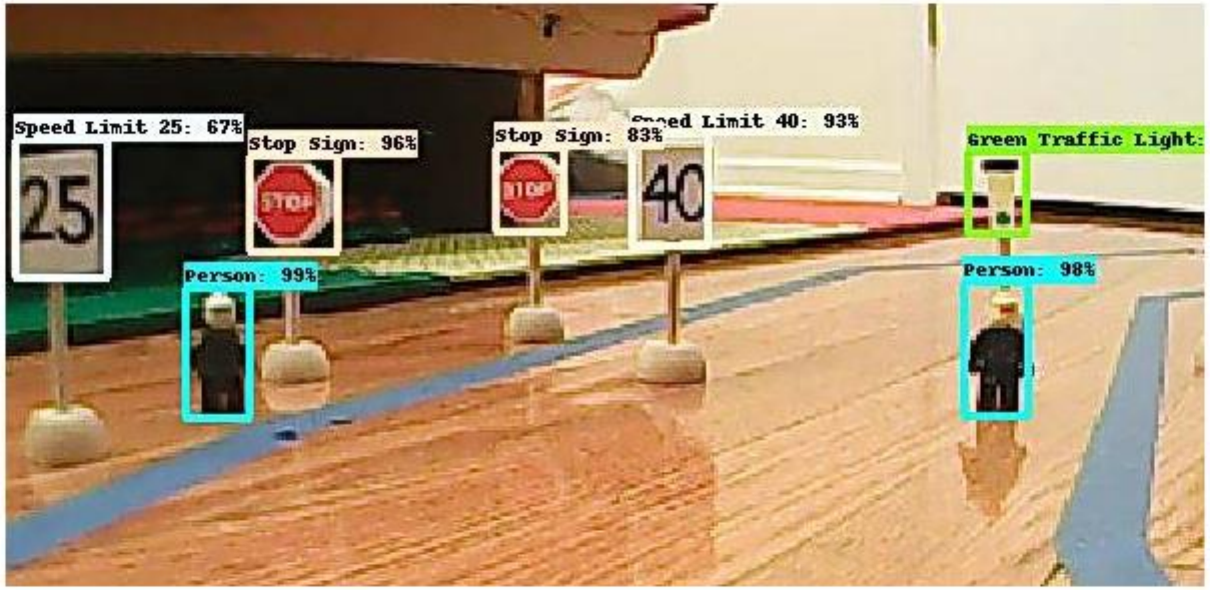
- https://www.youtube.com/watch?v=VoBsLc8V0Q0

DeepPiCar

- https://github.com/dctian/DeepPiCar



Lane Following



Traffic Sign and People Detection (right) from DeepPiCar's DashCam

Lab

- Extend the advanced follower car by moving the car toward the detected object.
  - You may use smartphone to show the photo of the object.
  - You may use change the object from person to other class.
- Stop the car when the object is close enough (decide by yourself).