[DIY] Autonomous RC Car

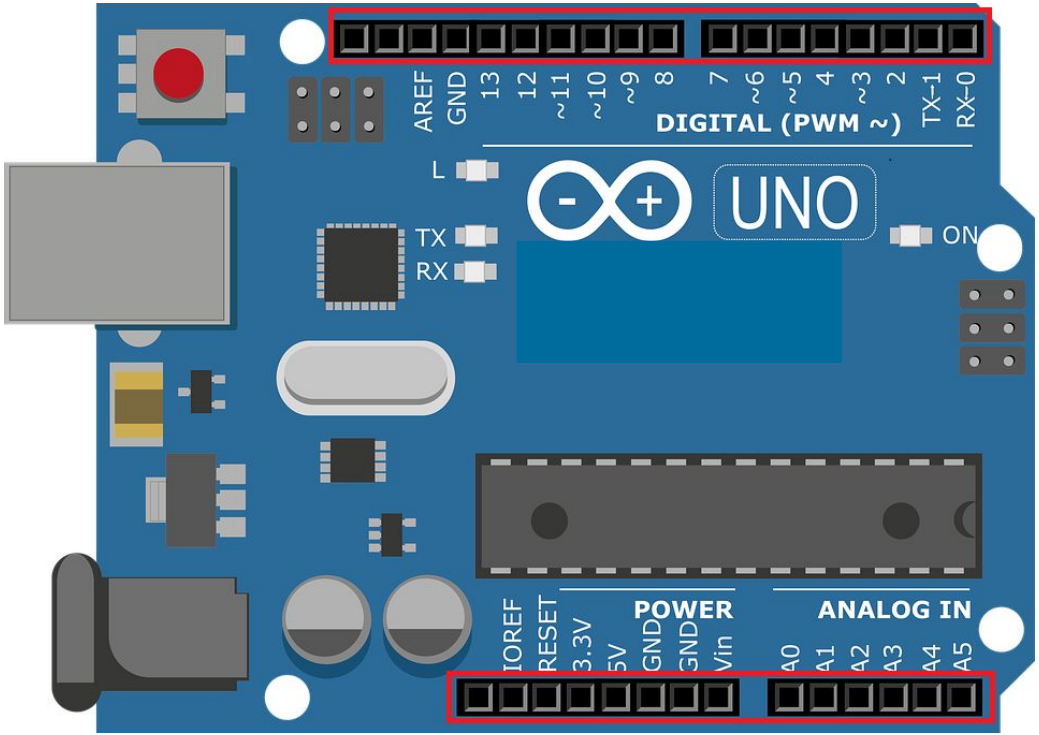
Sunggeun Han

sghan@kisti.re.kr

Principal Researcher
HPC Convergence R&D Platform
Korea Institute of Science and Technology Information

2023.08.07

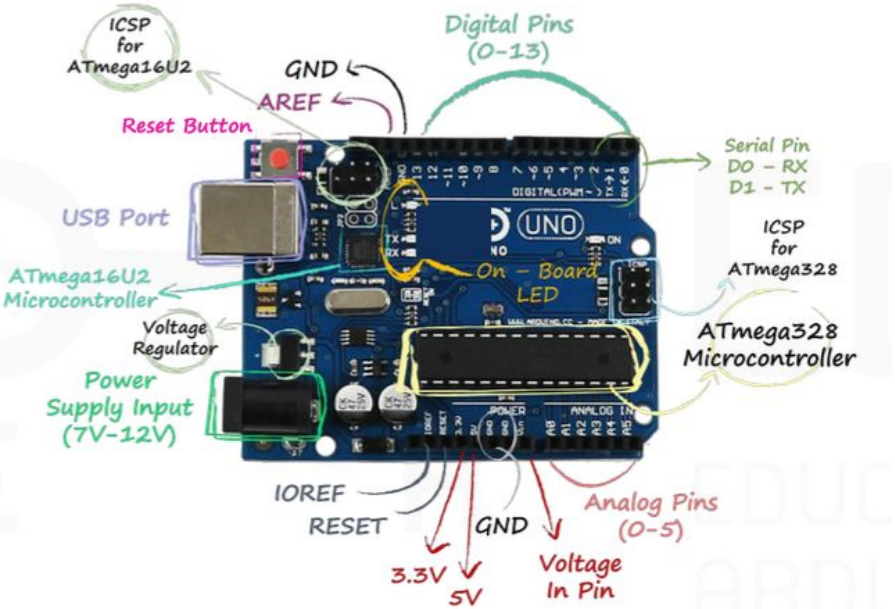
What is this?



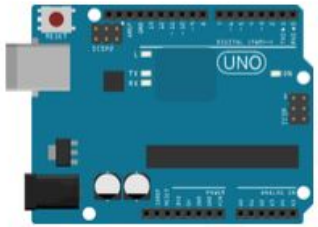
https://www.menti.com/alm8ask8grx3

ARDUINO

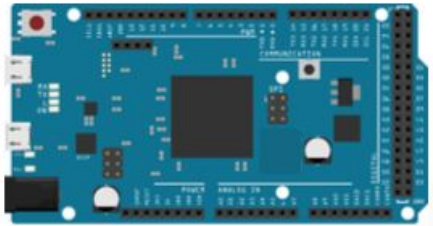
- An open-source electronics platform based on easy-to-use hardware and software
- Uses Microcontroller (MCU) for reading inputs and controlling outputs
  - MCU: a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals



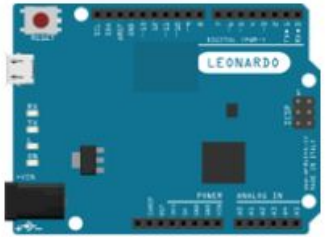
Various Types of Arduino



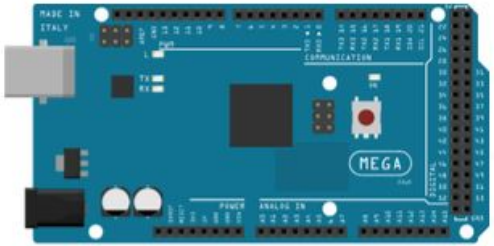
Arduino UNO R3



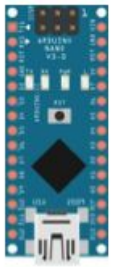
Arduino DUE



Arduino Leonardo

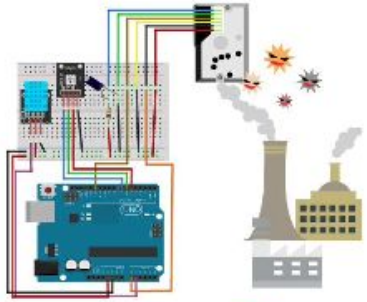


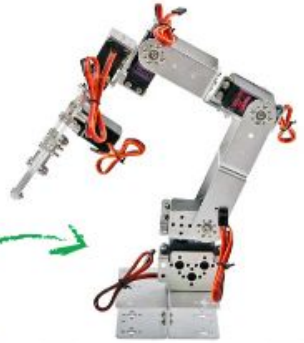
Arduino MEGA 2560



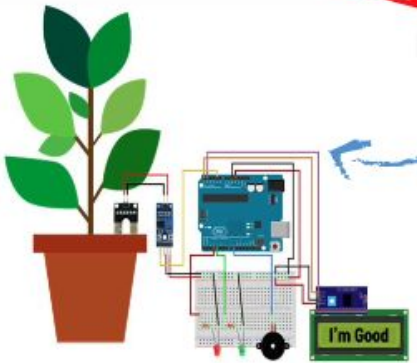
Arduino Nano

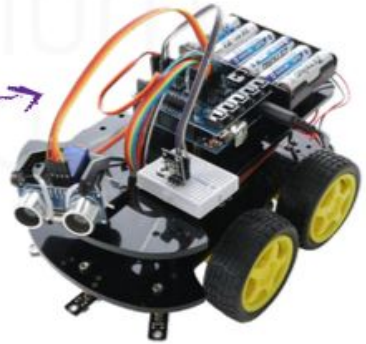
Applications



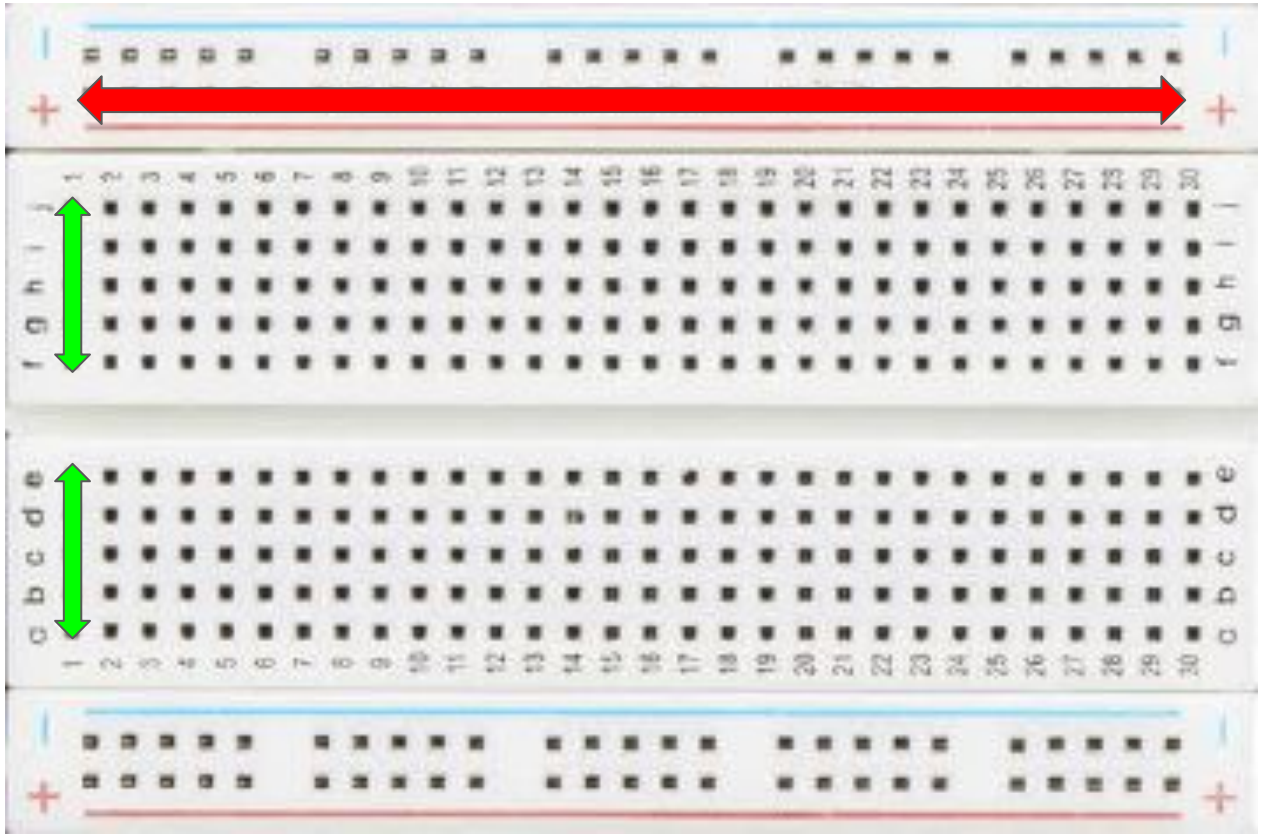


Robot, RC-car, Drone, IoT



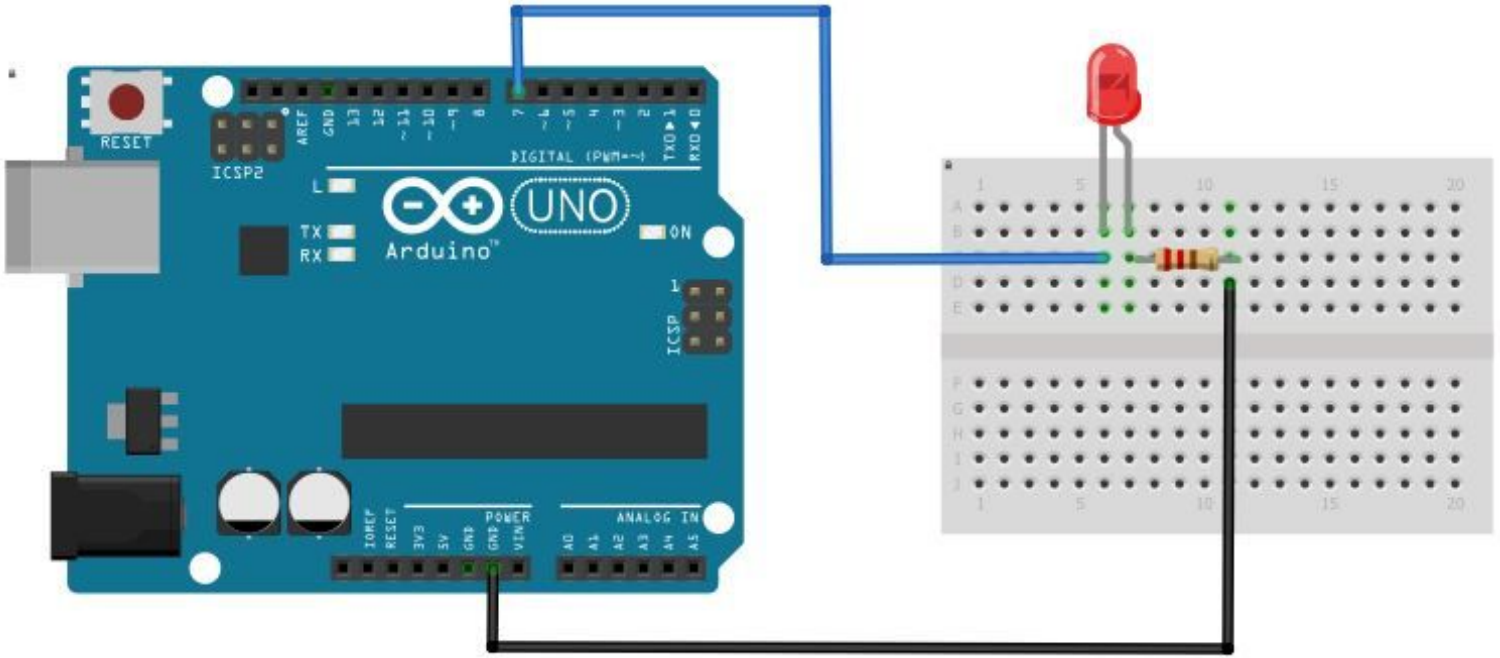


What is this?

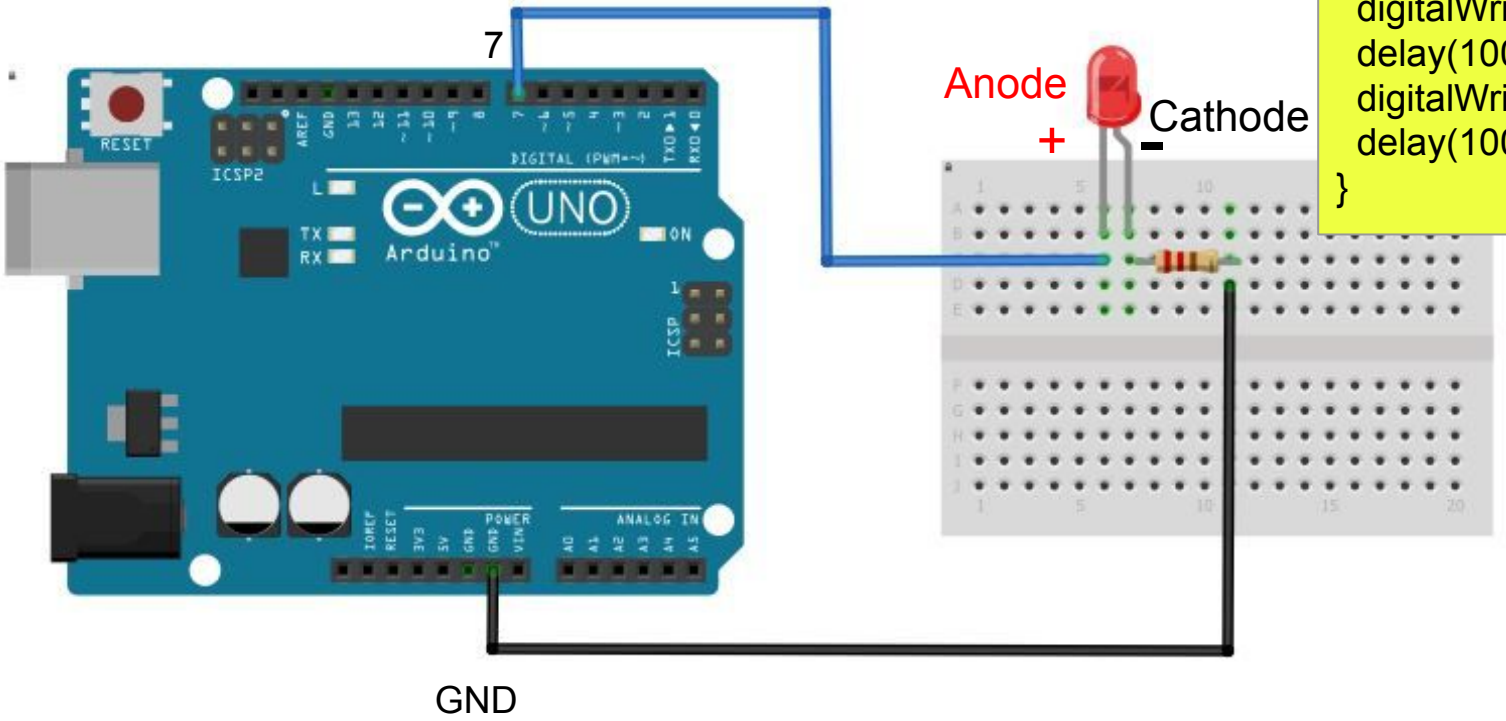


https://www.menti.com/alm8ask8grx3

Breadboards



LED On / Off




```
void setup() {
  pinMode(7, OUTPUT);
}

void loop() {
  digitalWrite(7, HIGH);
  delay(1000);
  digitalWrite(7, LOW);
  delay(1000);
}
```

Autonomous Vehicles

DEFINITION

- Driverless cars
- Equipped with **technology** that enables them to **navigate** and **drive** safely to a destination **without human intervention**.

HOW THEY WORK

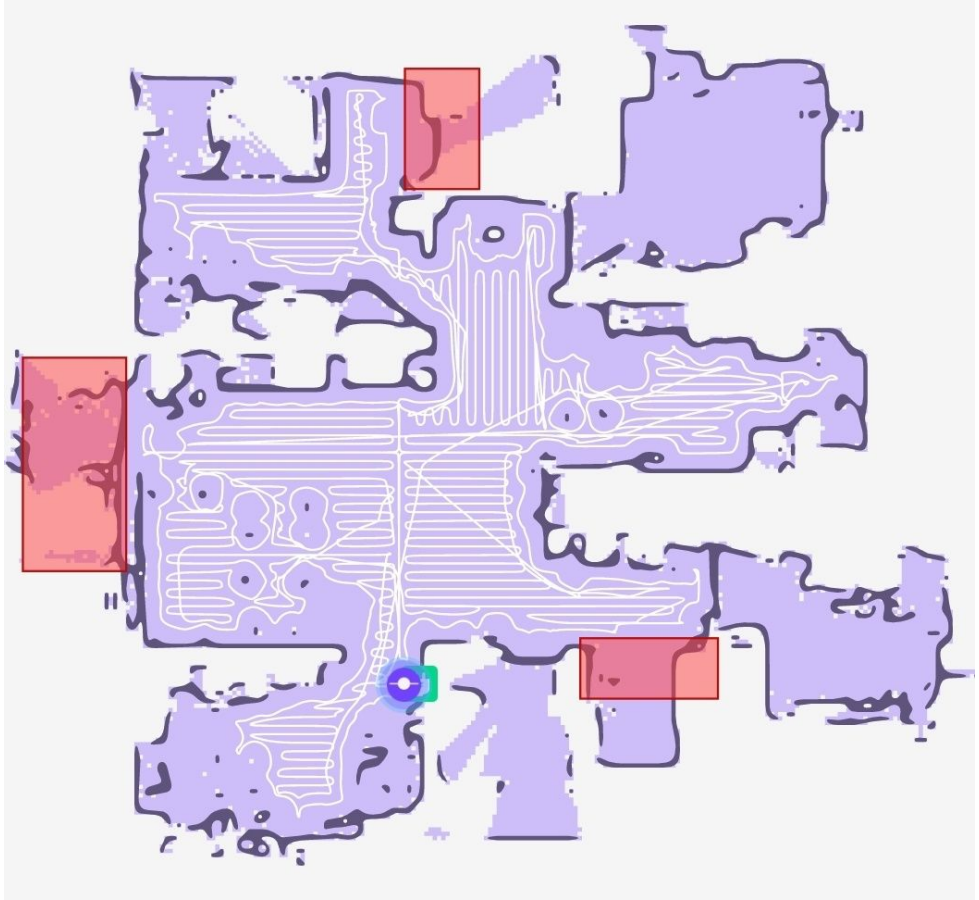
- Use a variety of sensors (perceive their surroundings & control the speed, direction, and braking of the car)
  - Radar
  - LIDAR
  - GPS
  - Infrared sensors
  - Complex algorithms

LEVELS OF AUTONOMY

- Classified into 6 levels (0 ~ 5) by the Society of Automotive Engineers(SAE)
- Level 0: No autonomy
- :
- Level 5: Complete autonomy

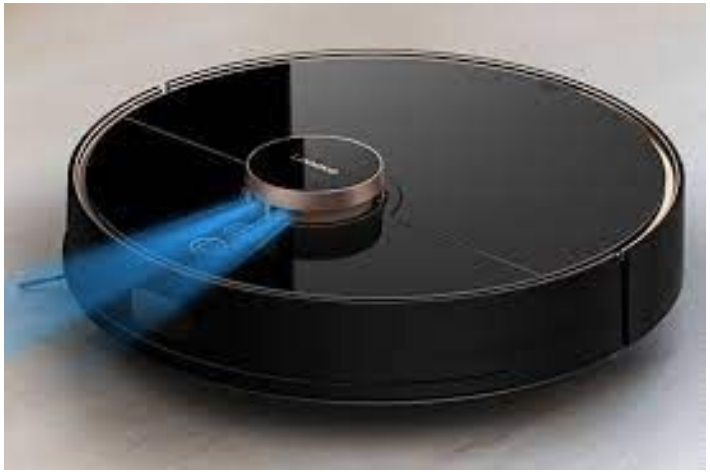
IMPORTANCE

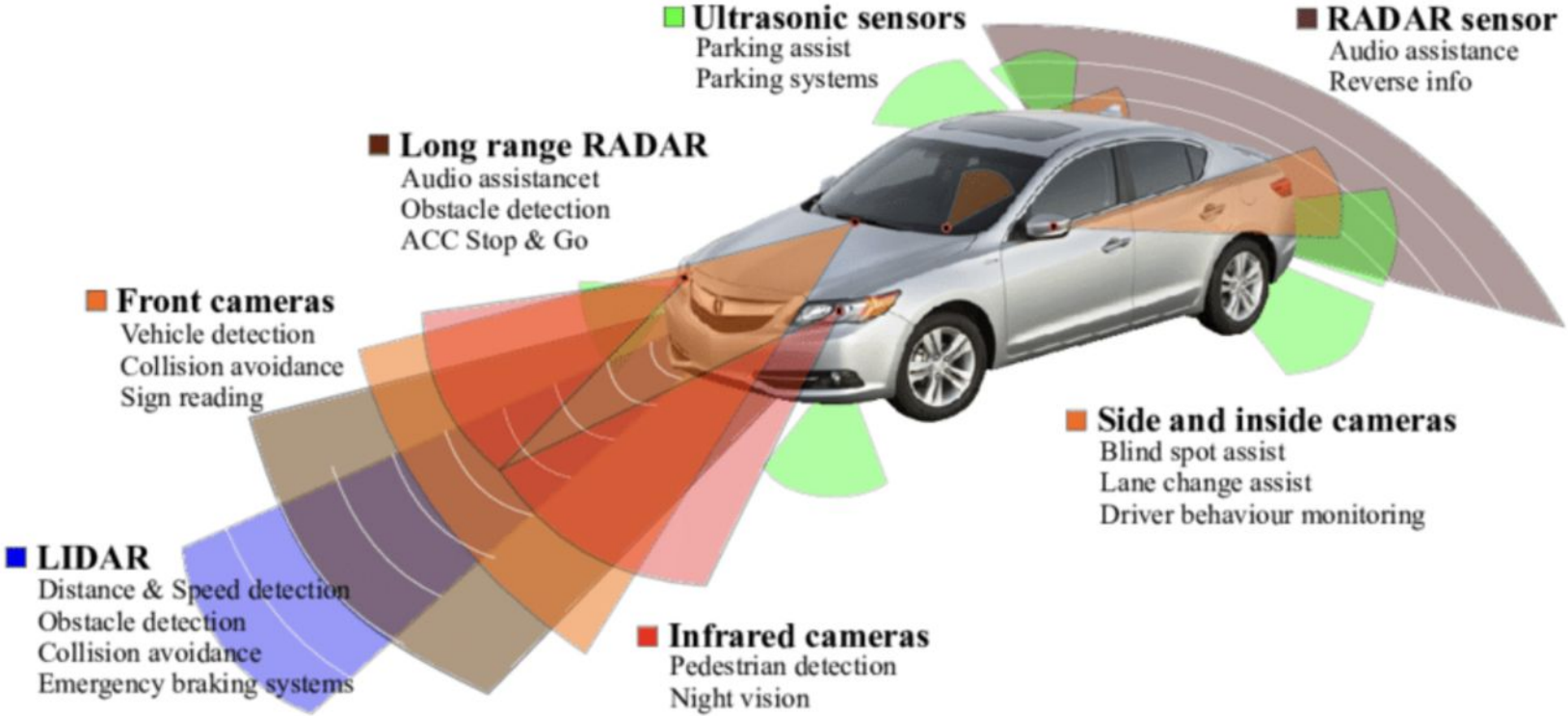
- Reducing traffic accidents
- Managing traffic efficiently
- Improving the mobility of those with travel limitations such as the disabled and elderly



Lidar

- Light Detection and Ranging
- Uses pulsed laser light to create 3D maps
- Detecting obstacles and safe navigation





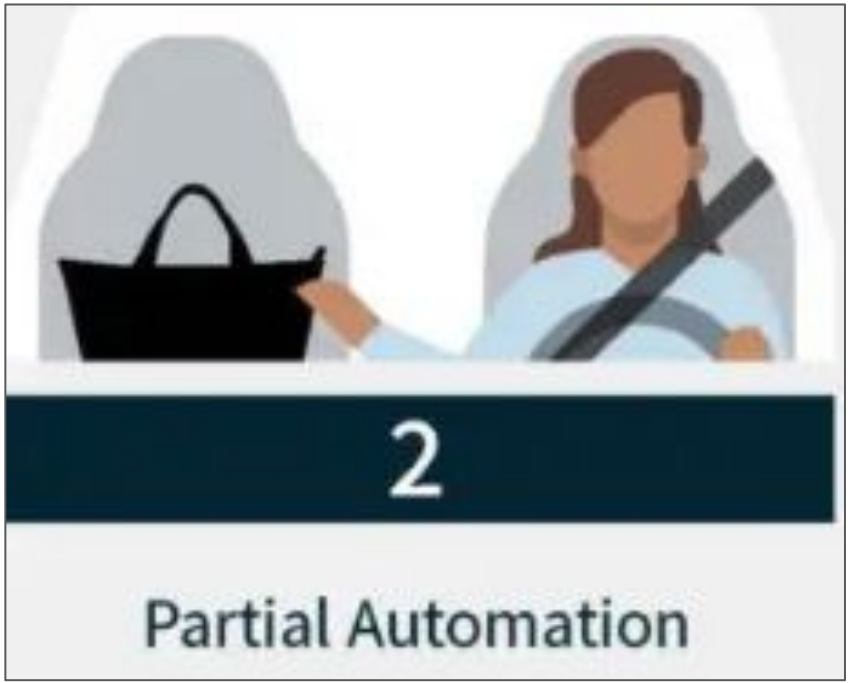
1: Typical types of sensors for ADAS.

https://www.linkedin.com/pulse/finally-tesla-reportedly-testing-lidar-sensors-which-elon-kamil-guclu

Levels of Autonomy (0 ~ 1)

https://torc.ai/understanding-levels-autonomy-3-4-5/

Levels of Autonomy (1 ~ 2)



https://torc.ai/understanding-levels-autonomy-3-4-5/

Levels of Autonomy (2 ~ 3)

https://torc.ai/understanding-levels-autonomy-3-4-5/

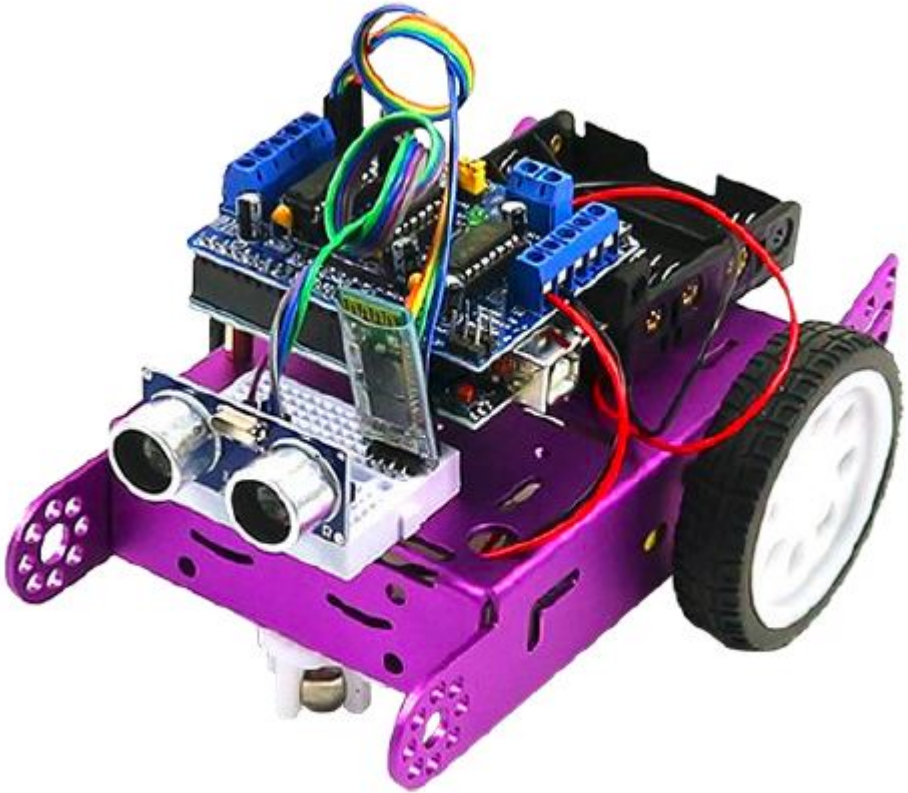
Levels of Autonomy (3 ~ 4)

https://torc.ai/understanding-levels-autonomy-3-4-5/

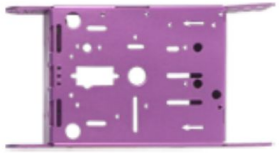
Levels of Autonomy (4 ~ 5)

https://torc.ai/understanding-levels-autonomy-3-4-5/

CAR ASSEMBLY TIME



Parts



Base

Auxiliary wheel

Bolts
Nuts
Supports

Arduino
UNO

Ultrasonic
Sensor

Breadboards

Battery Holder
(1.5V * 4)

Velcro Strips

USB Cable

Organizer box

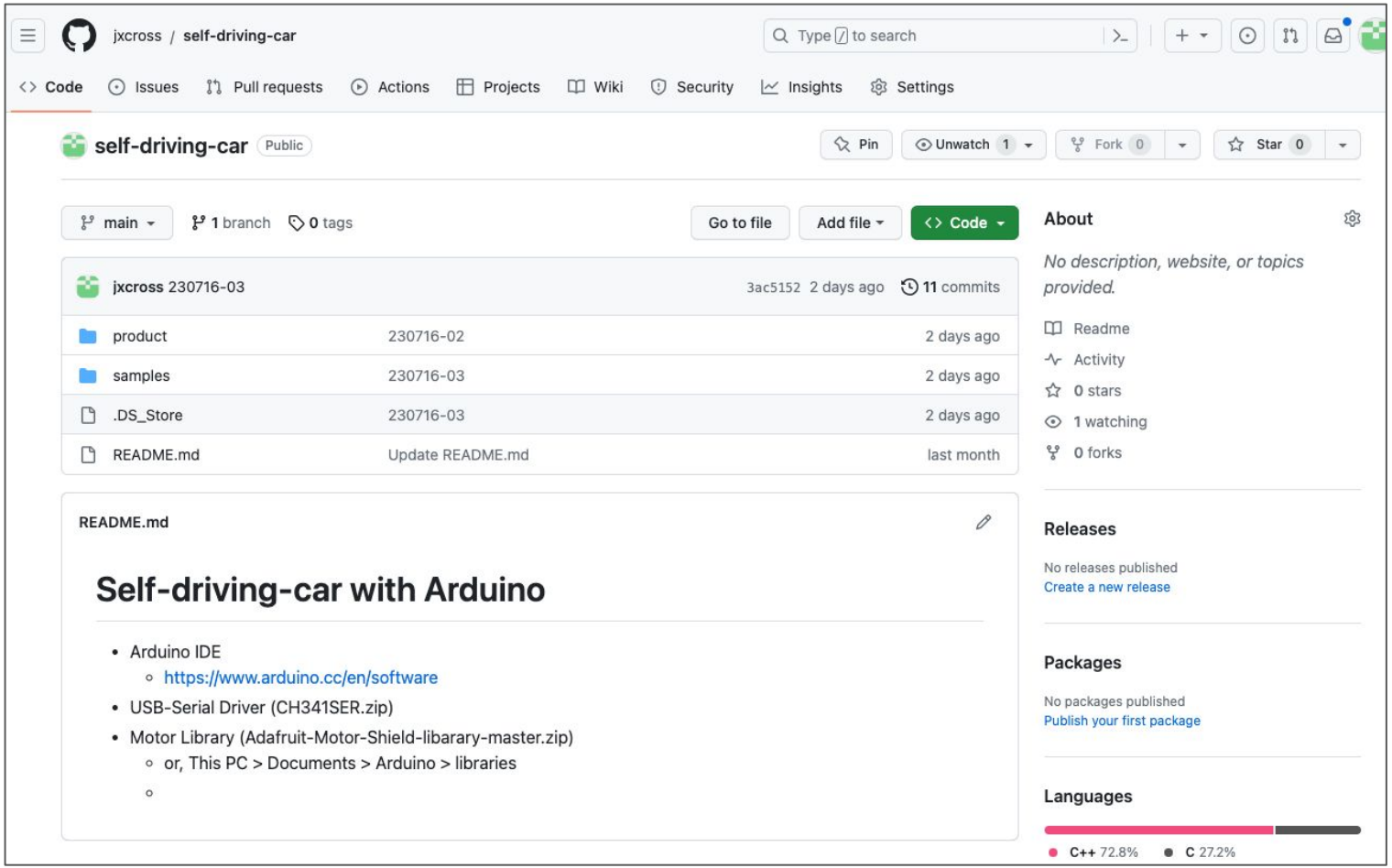
Motor Driver Shield
L293D

DC Motors

Cross Driver

Wheels

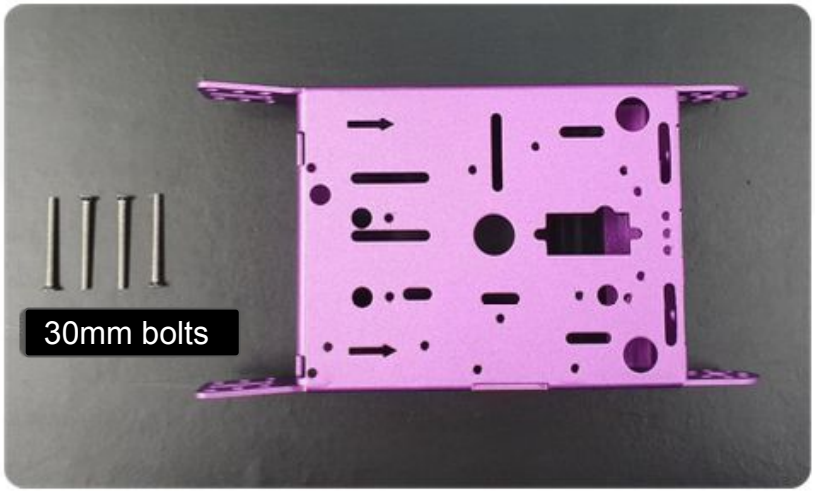
https://github.com/jxcross/self-driving-car



RC Car Assembly

Body Frame and bolts

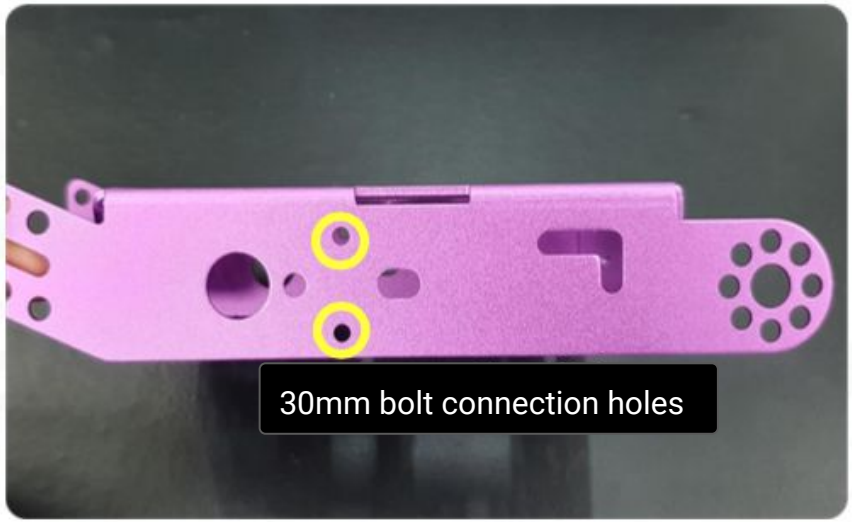
Prepare a 30mm bolt and metal body frame.

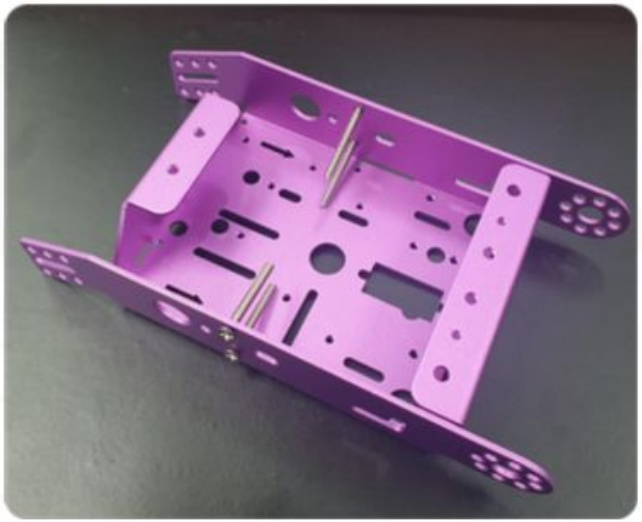


Metal body frame
Four 30mm bots

RC Car Assembly

Attach the 30mm bolts to the connection slots on the metal frame.



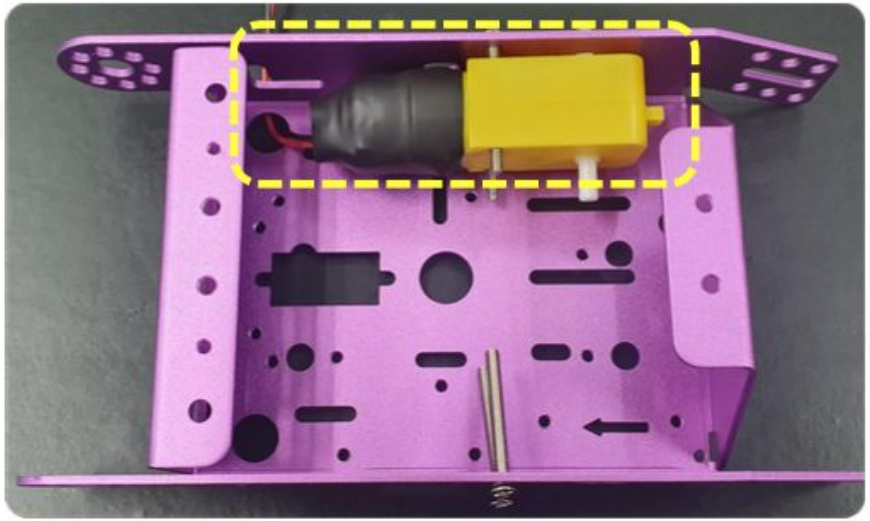


Referring to the image, attach the 30mm bolts to the connection slots on the side of the metal frame.

RC Car Assembly

Attaching the DC motor

Attach the DC motor to the 30mm bolts connected to the metal frame.



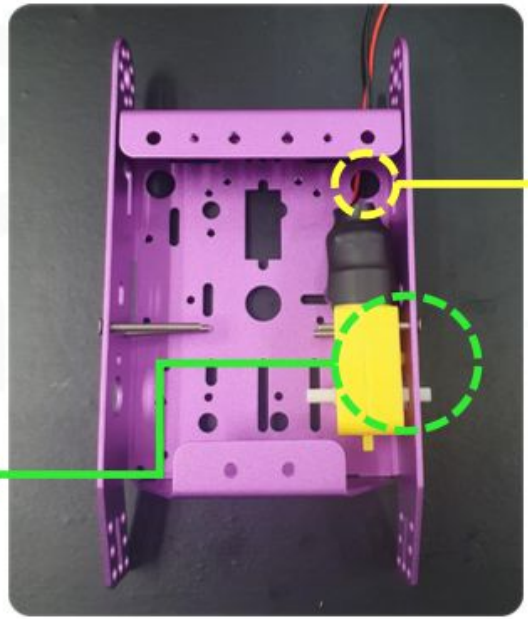
After connecting the 30mm bolts to the DC motor, remove the DC motor cable.

Attaching the DC motor

RC Car Assembly

Connecting the motor, refer to the following instructions

Pull out the motor cable through the motor wire outlet

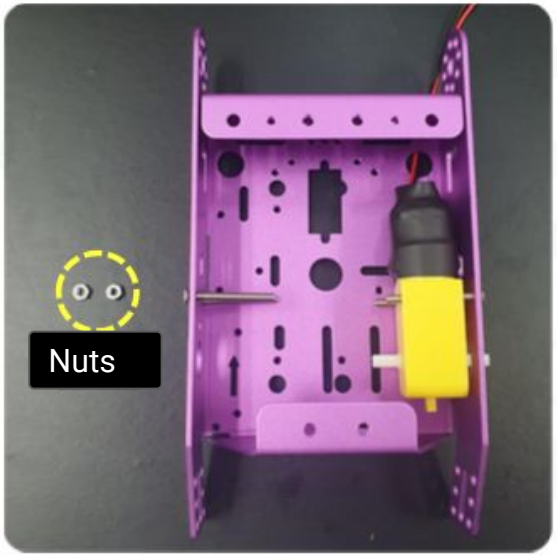


Insert the protrusion of the DC motor into the connection slot on the body frame

Attaching the DC motor

RC Car Assembly

Fix the DC motor to the metal frame with nuts

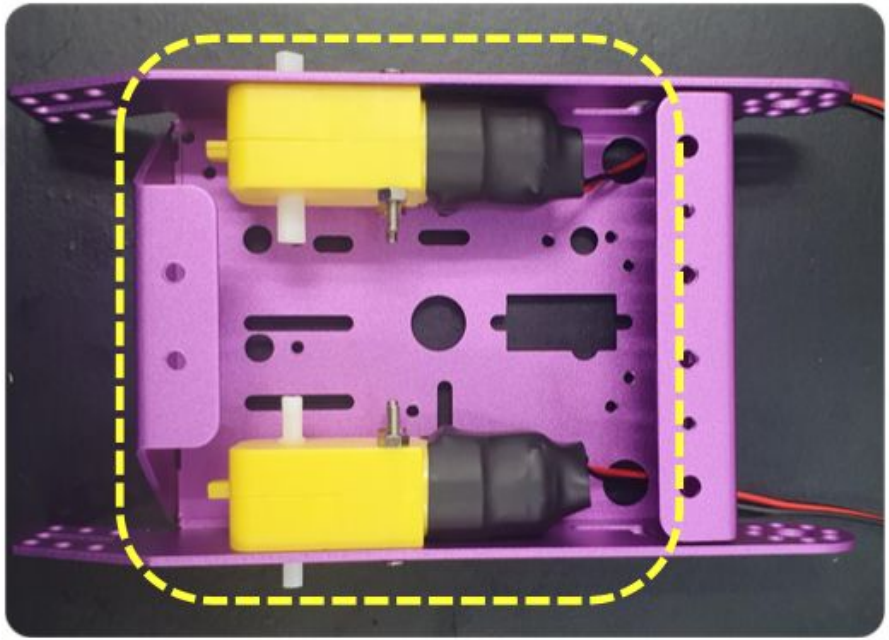


DC motor fixed to the metal frame with nuts

Prepare the nuts and attach the DC motor to the metal frame

RC Car Assembly

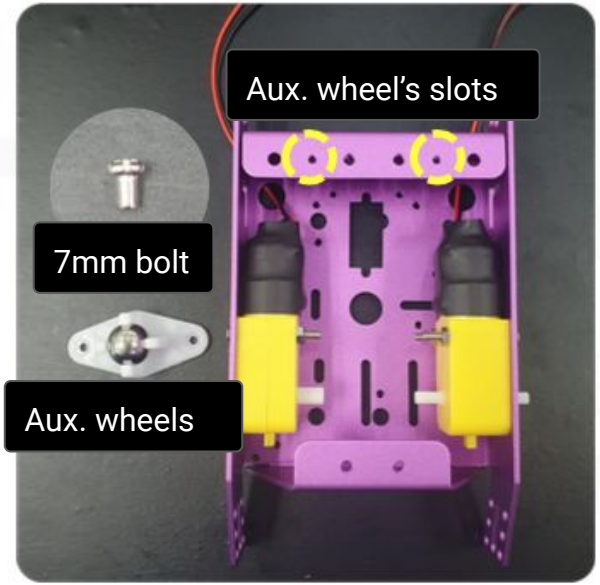
Attach the remaining DC motor in the same way as before.

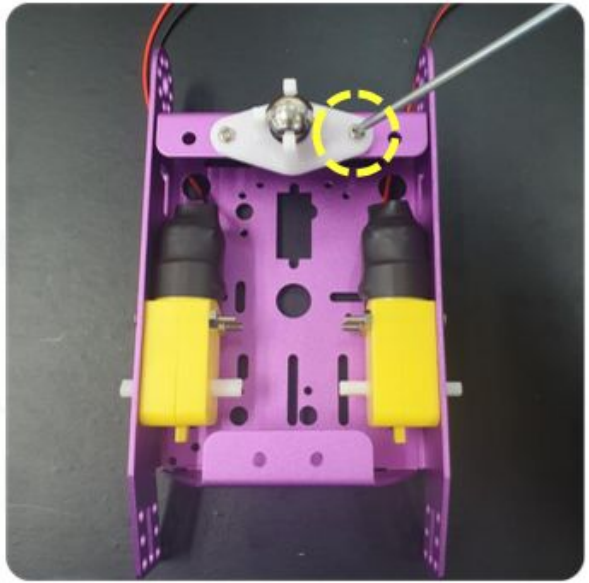


Attaching the Aux. Wheels

RC Car Assembly

Attach the auxiliary wheel to the metal frame using a 7mm bolt.





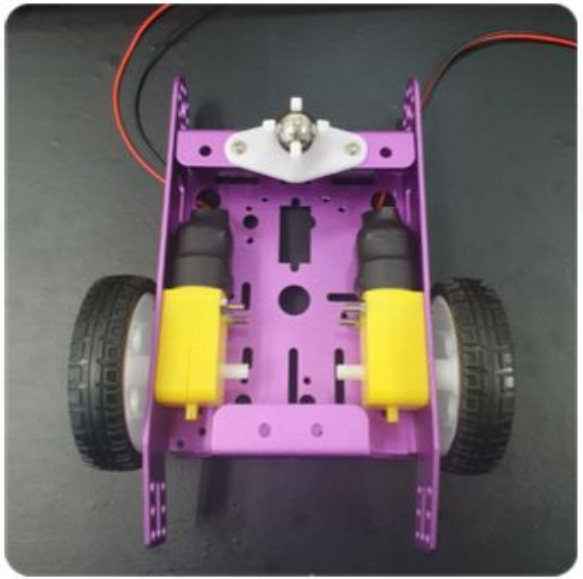
Use a 7mm bolt to attach the auxiliary wheels to the connection slots of the auxiliary wheel

RC Car Assembly

Attach the wheels to the DC motor fixed to the metal frame.

DC motor connection shape

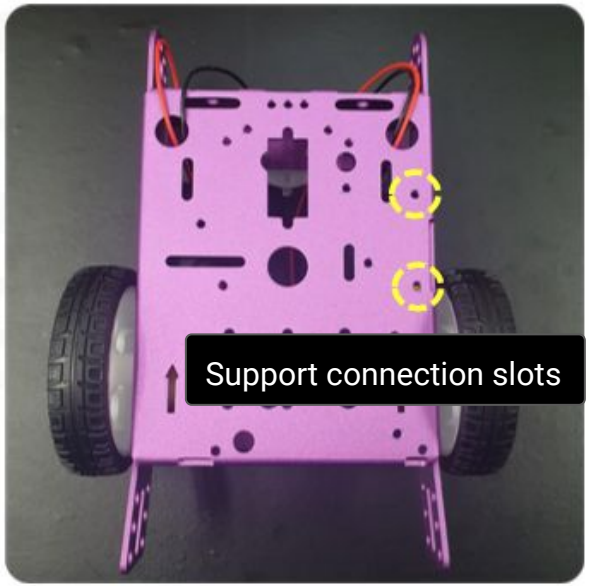
Wheel connection shape

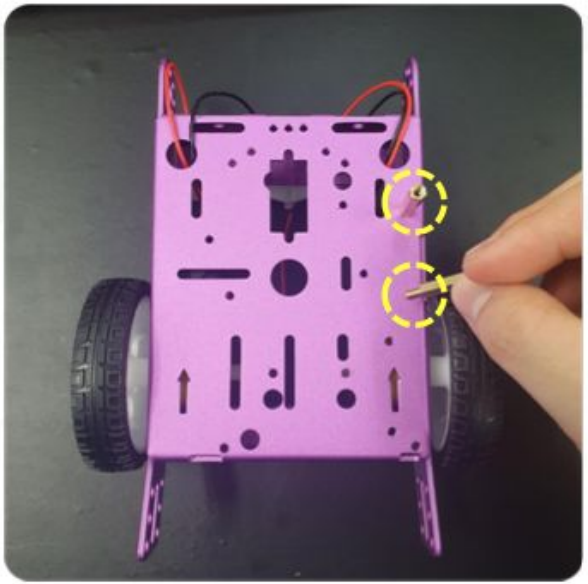


Check the connection shape of the DC motor and the bundled wheel and connect them.

RC Car Assembly

Attach the supports to the support connection slots on the metal frame.

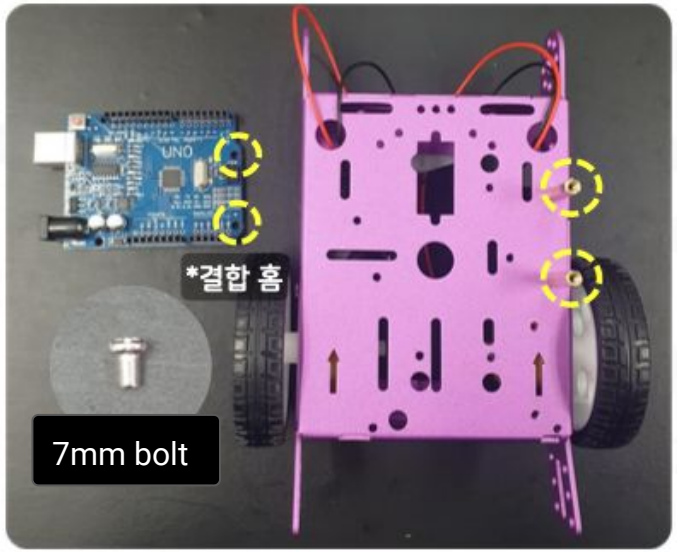


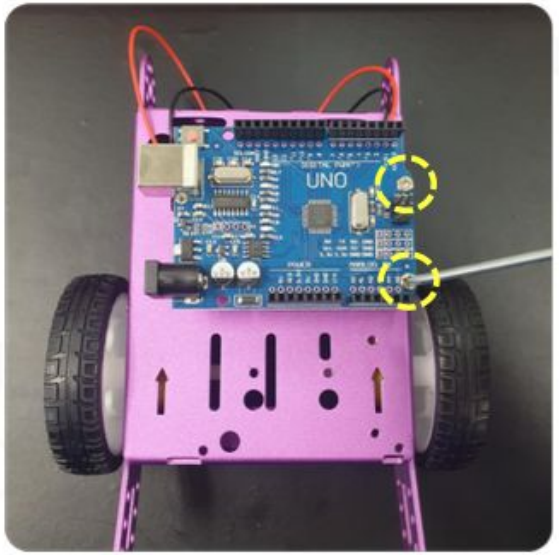


Attaching the UNO board

RC Car Assembly

Secure the Arduino UNO board onto the support using bolts

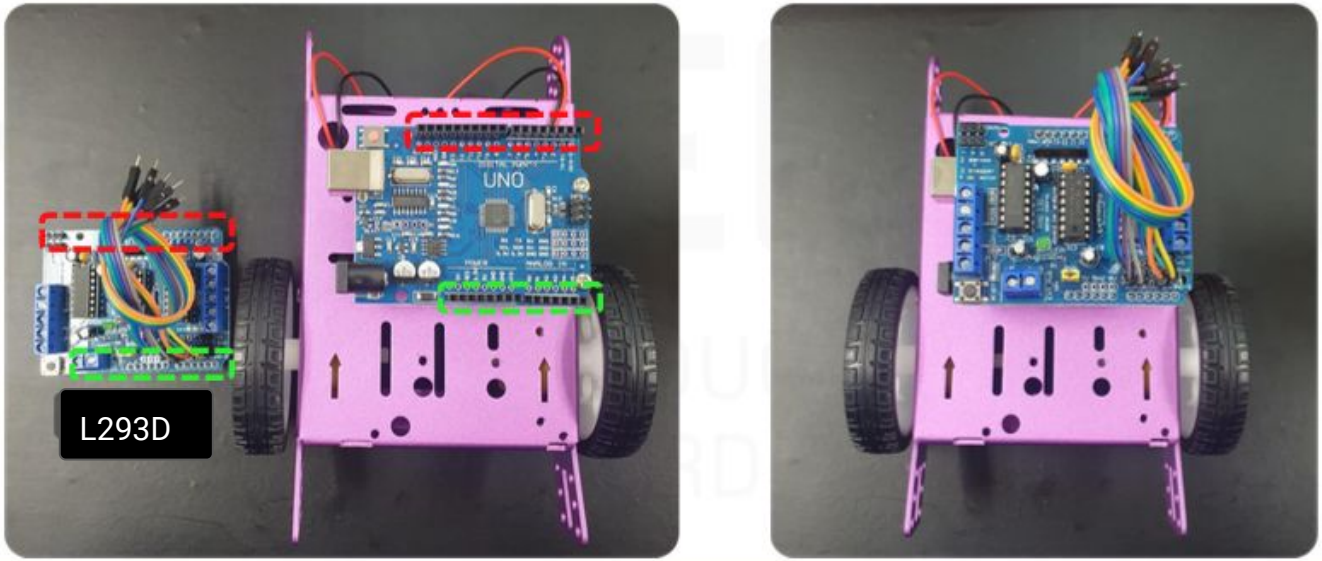




Align the Arduino UNO board on the support according to the connection slots on the board, and secure it with a 7mm bolt.

RC Car Assembly

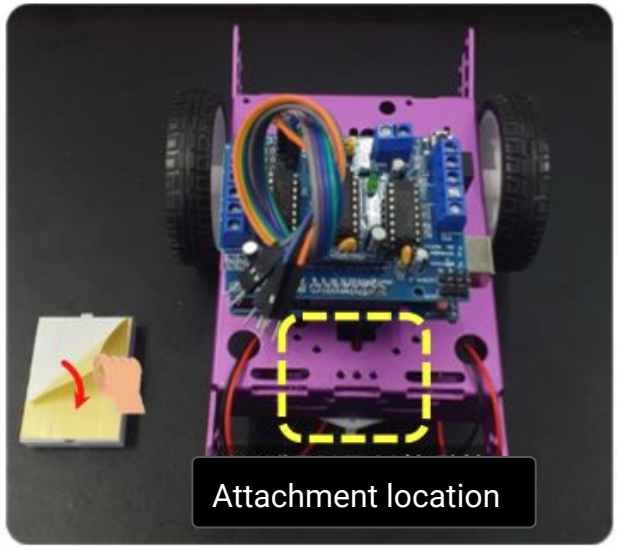
Attach the motor driver shield to the Arduino UNO board

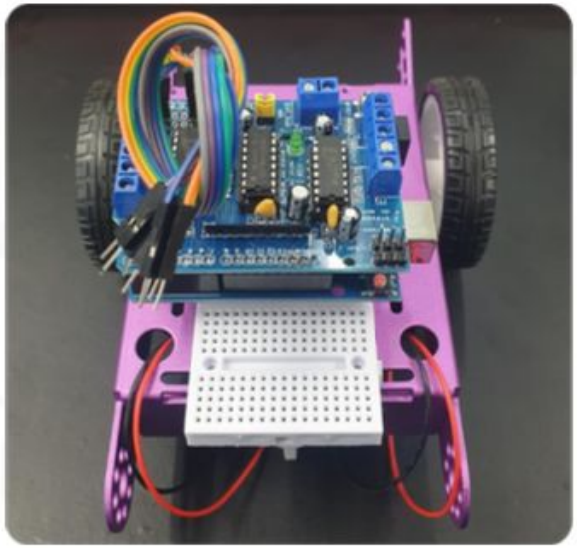


Pin-connect the motor driver shield(L293D) onto the UNO board that has been connected to the metal frame.

RC Car Assembly

Attach the breadboard in front of the metal frame.



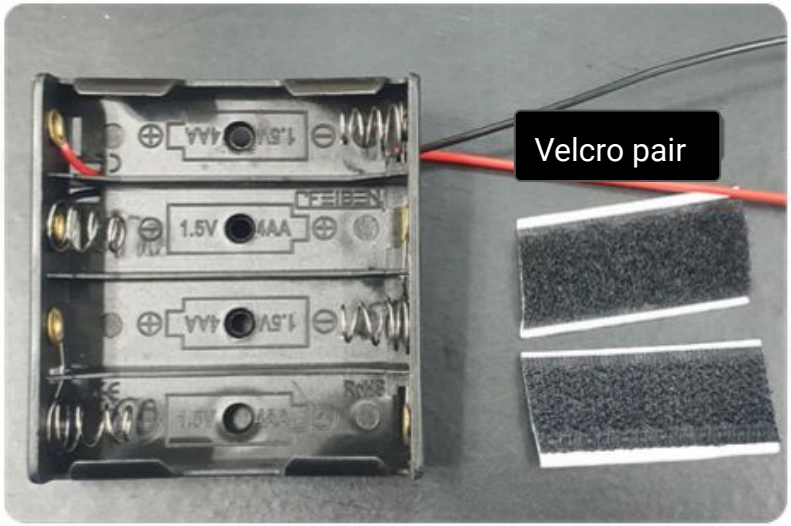


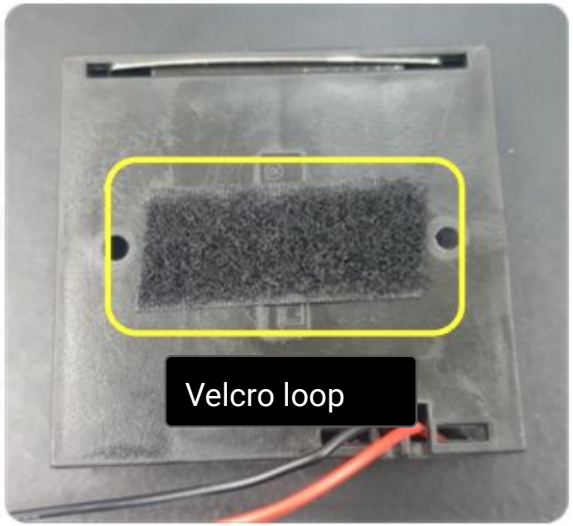
Remove the back cover of the breadboard and attach it to the front of the metal frame.

Attaching the Battery holder

RC Car Assembly

Attach Velcro to the back of the battery holder.



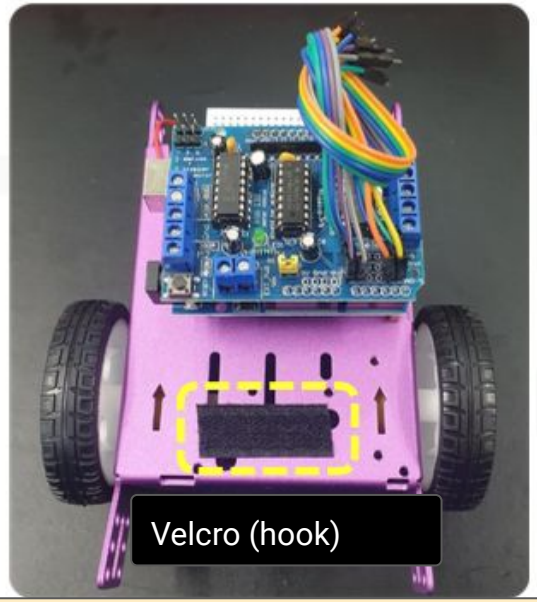


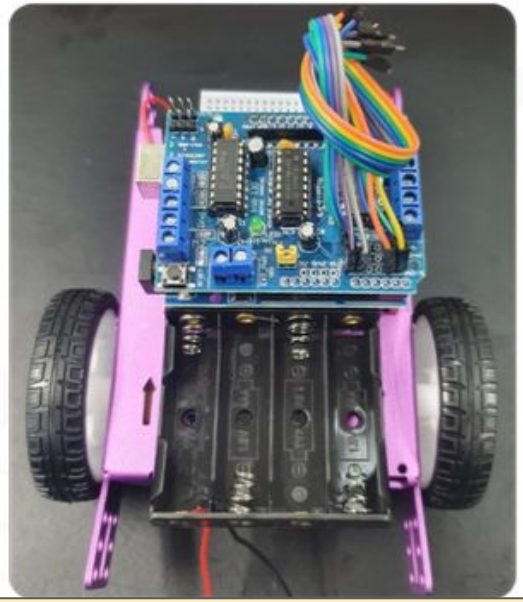
Prepare a pair of Velcro (hook and loop) for the battery holder and attach the loop Velcro to the back of the battery holder.

Attaching the Battery holder

RC Car Assembly

Attach the battery holder to the metal frame.





Attach the loop Velcro to the back of the metal frame and then attach the battery holder to the metal frame.

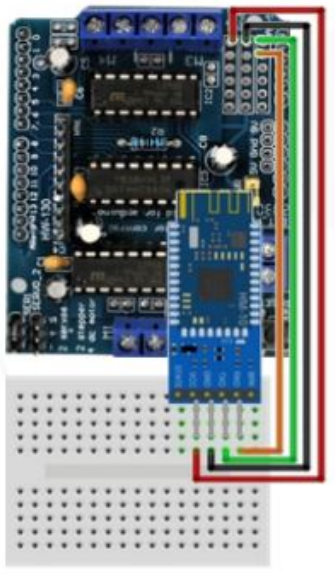
RC Car Assembly

Refer to the circuit diagram and connection table to wire the motor driver shield(L293D) and Bluetooth module.

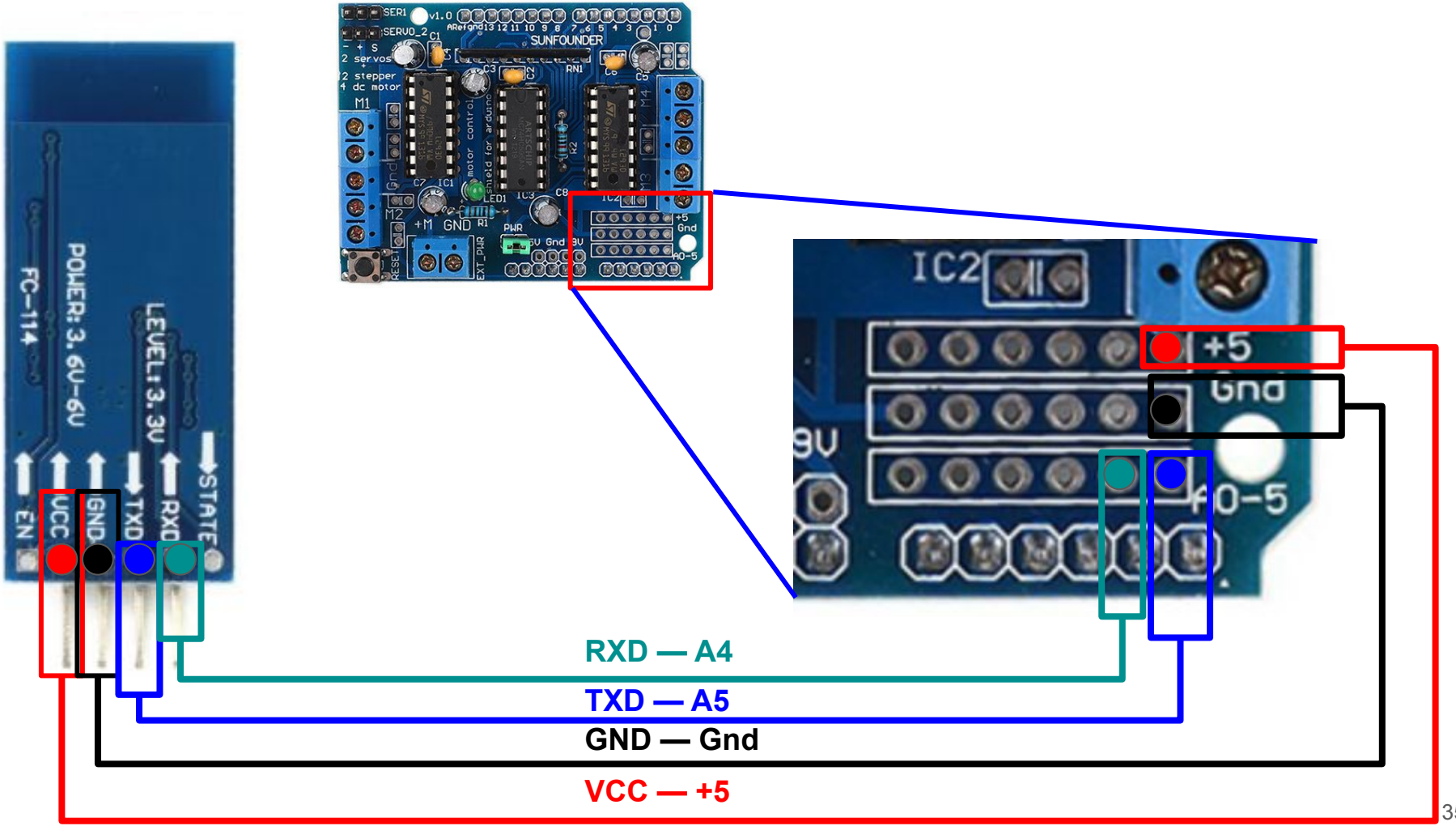
HC-06

Motor Driver Shield (L293D)	Bluetooth Module
A4	RX
A5	TX
GND	GND
VCC	VCC

Connection table



HM-10


FC-114
POWER: 3.6V-6V
LEVEL: 3.3V
EN
VCC
GND
TXD
RXD
STATE
SUNFOUNDER
IC2
+5
Gnd
A0-5
RXD — A4
TXD — A5
GND — Gnd
VCC — +5

2강

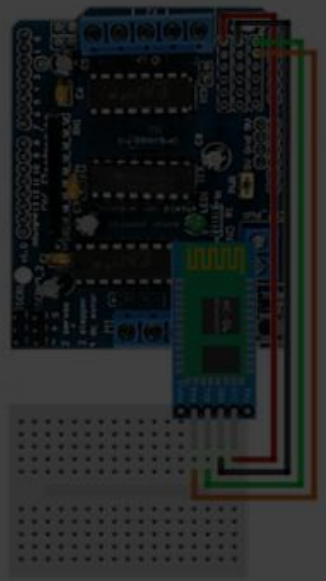
블루투스 모듈

메탈 RC카 조립하기

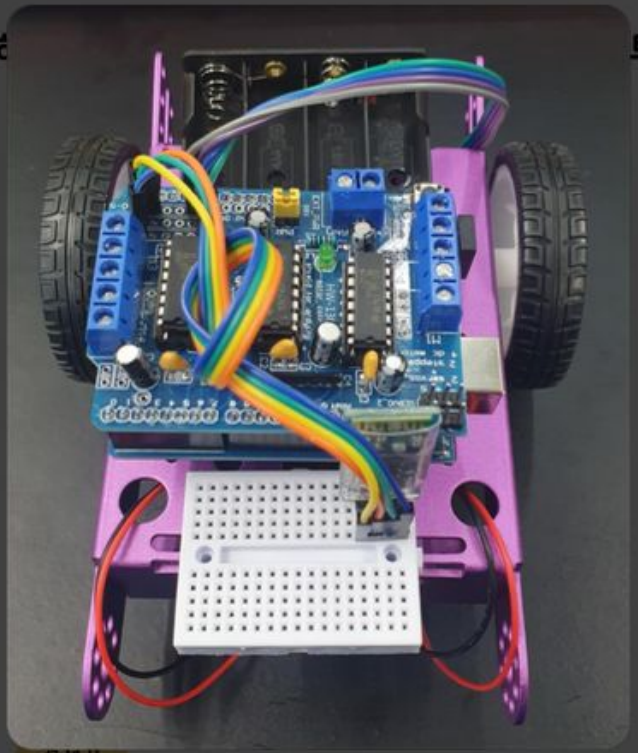
Picture of the completed wiring

▪ 회로도, 결선표를 참고하

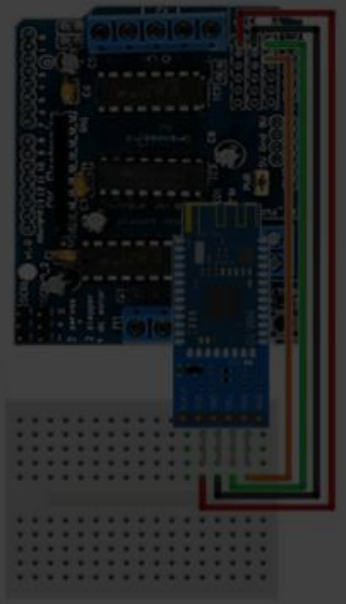
투스 모듈을 결선합니다.



HC-06



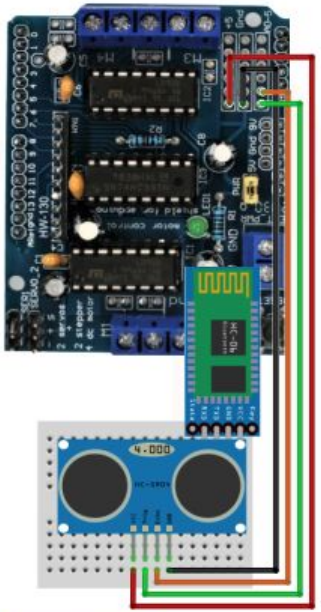
결선표



HM-10

메탈 RC카 조립하기

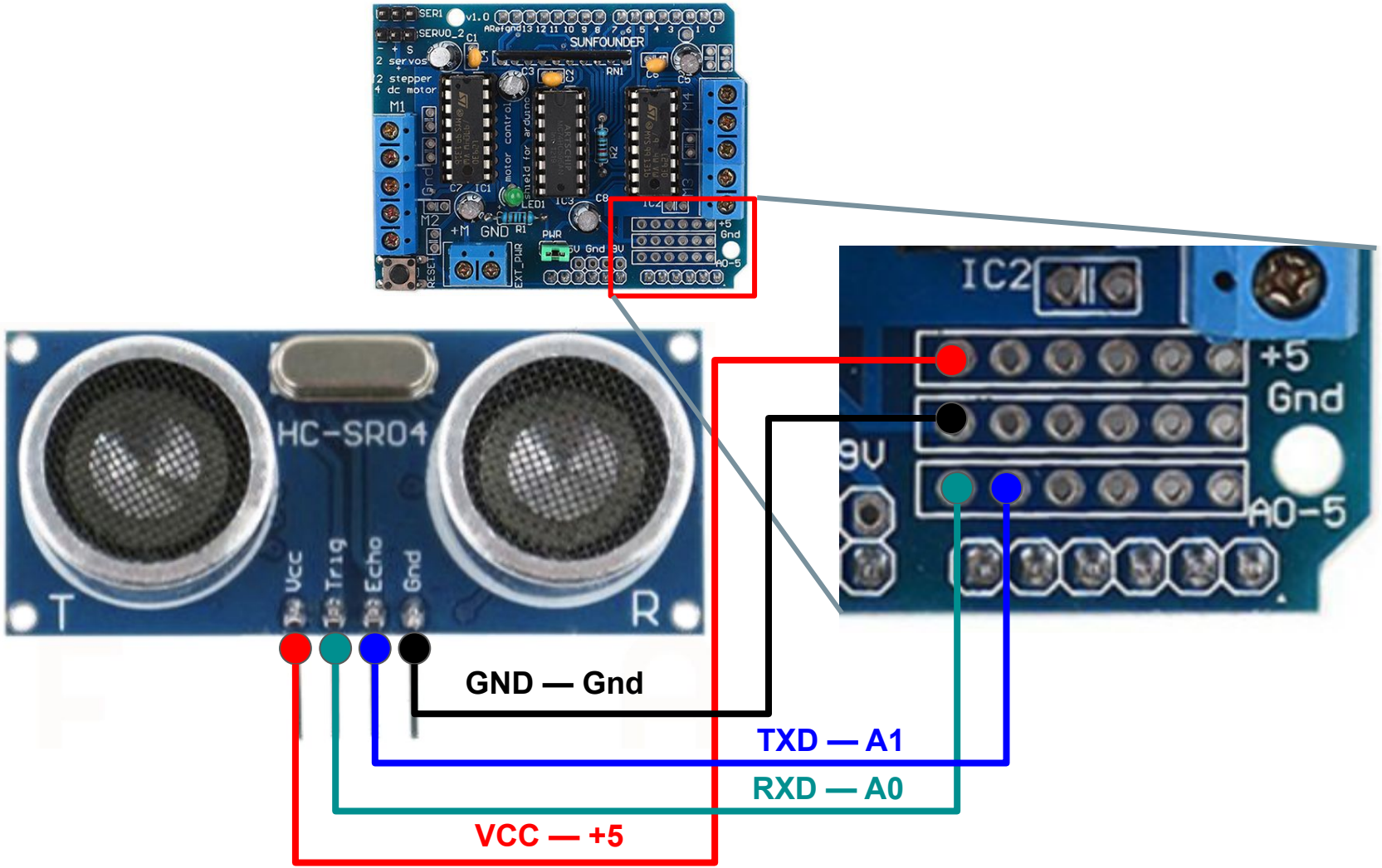
Wire the MDS(L293D) and Ultrasonic sensor



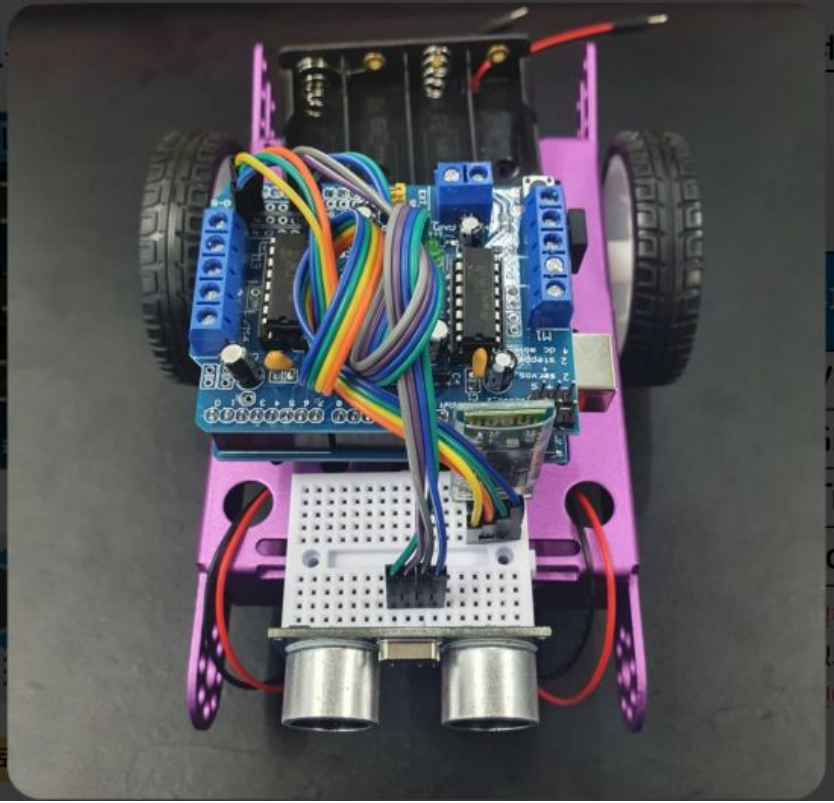
Circuit diagram

Motor Driver Shield (L293D)	Ultrasonic Sensor
5V	VCC
GND	GND
A0	Trig
A1	Echo

Connection Table

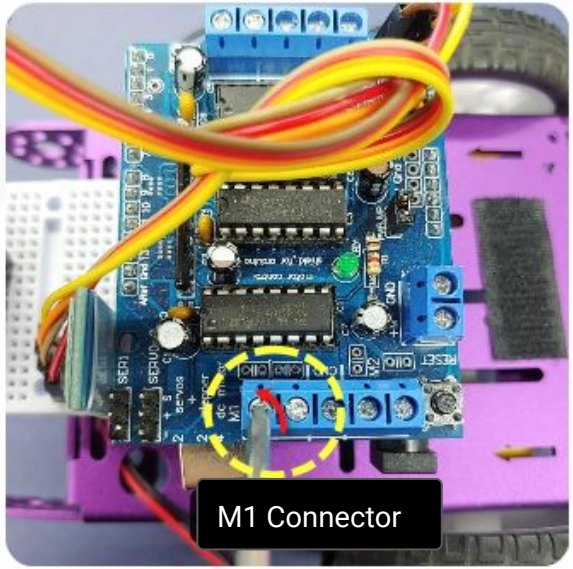

SUNFOUNDER
IC2
+5
Gnd
A0-5
HC-SR04
Vcc
Trig
Echo
Gnd
T
R
GND — Gnd
TXD — A1
RXD — A0
VCC — +5

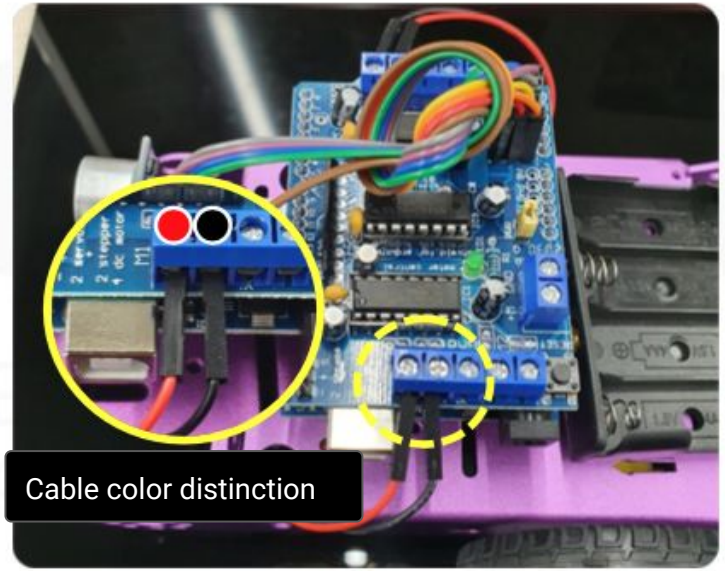
Picture of the completed wiring



RC Car Assembly

Connect the Left DC motor to the M1 connector on the MDS(L293D)

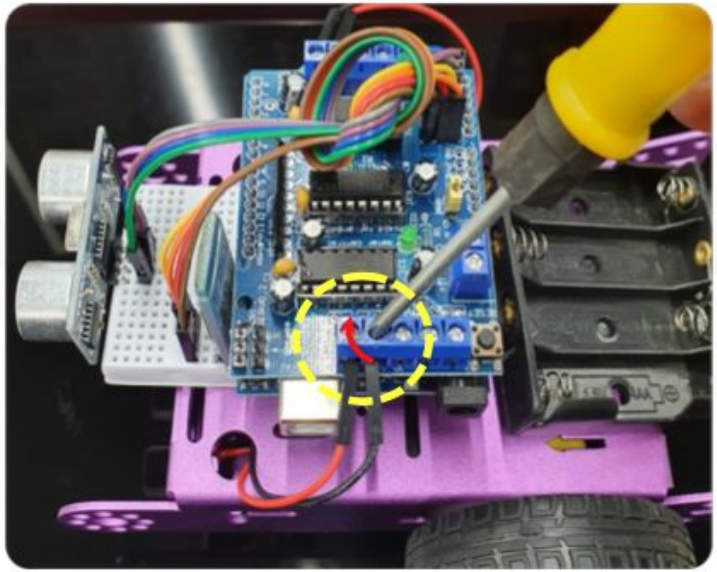




Turn the M1 connector counter-clockwise to loosen the driver and connect the Left wheel DC motor cable.

RC Car Assembly

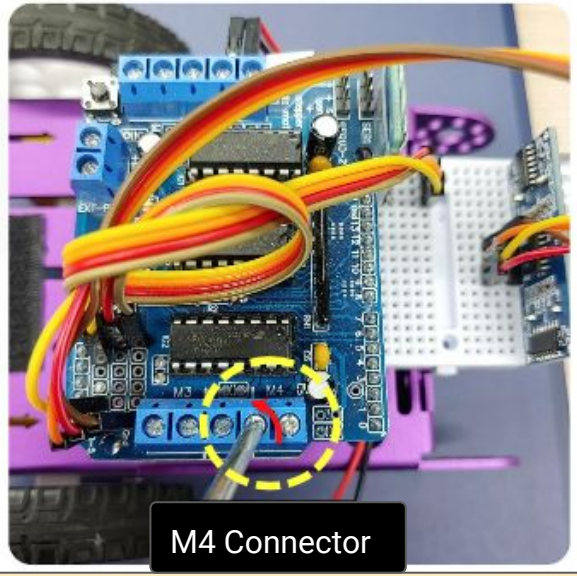
Connect the Left DC motor to the M1 connector on the MDS (L293D)

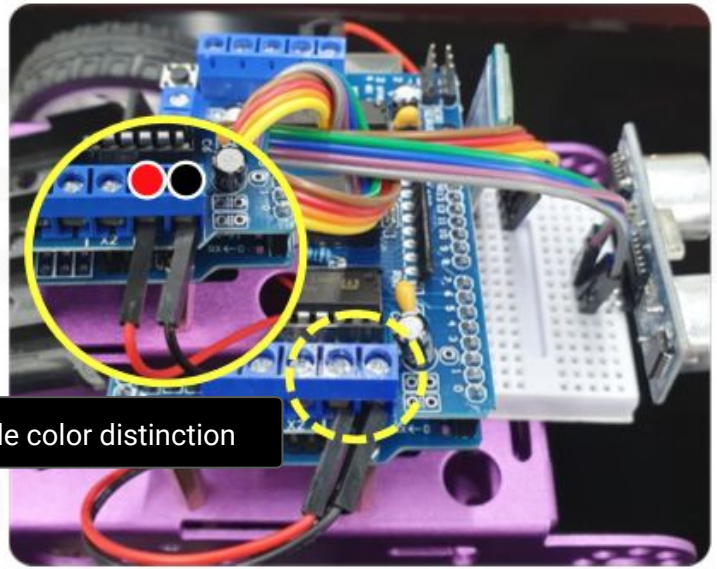


Tighten the bolt on the M1 connector with a driver in a clockwise direction

RC Car Assembly

Connect the Right DC motor to the M4 connector on the MDS(L293D).

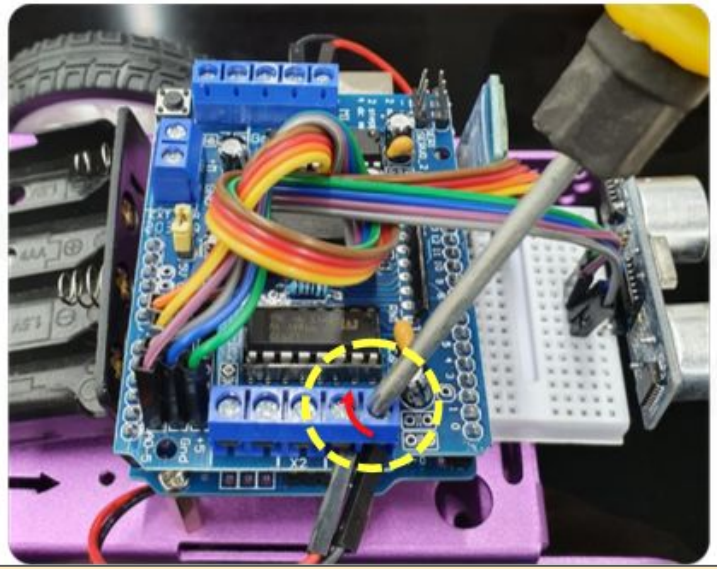




Turn the M4 connector counter-clockwise with a driver to loosen it, and then connect the Right wheel DC motor cable.

RC Car Assembly

Connect the Right DC motor to the M4 connector on the MDS(L293D).

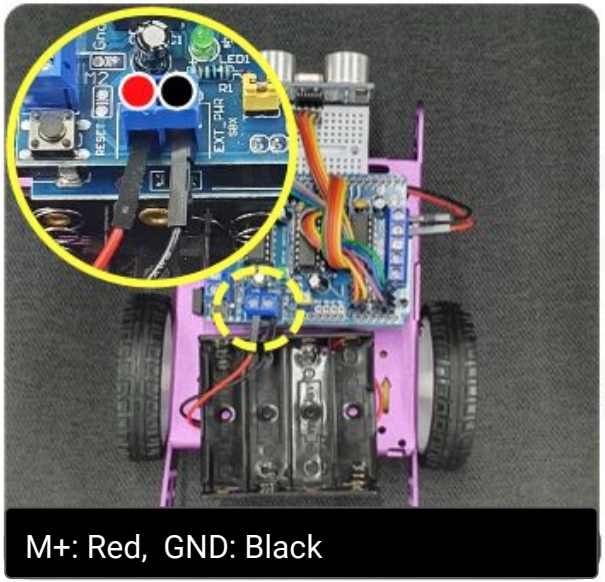


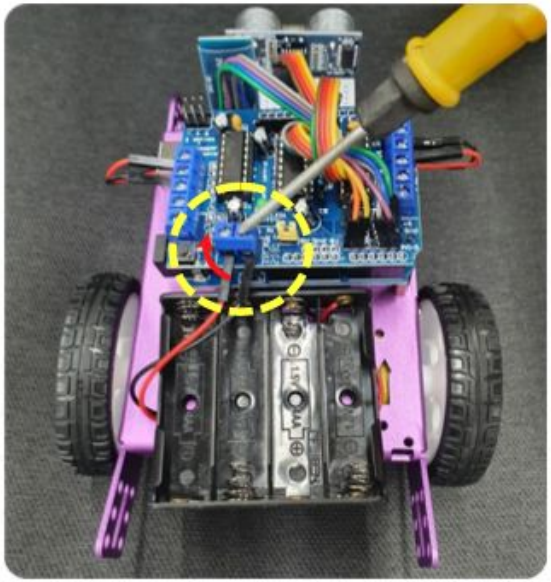
Tighten the bolt on the M4 connector with a driver in a clockwise direction.

Connecting the Battery holder wires

RC Car Assembly

Wire the battery holder to the MDS (L293D)

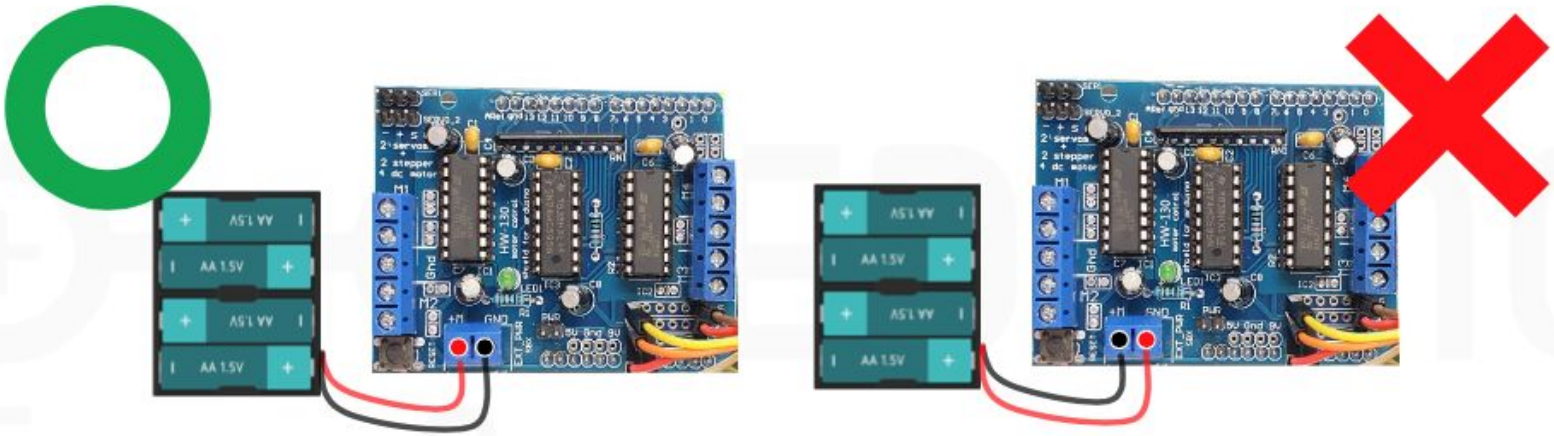




Connect the battery holder cables carefully according to the color of the cables and then tighten the bolt in a clockwise direction with a driver.

RC Car Assembly

Connecting the Battery holder wires

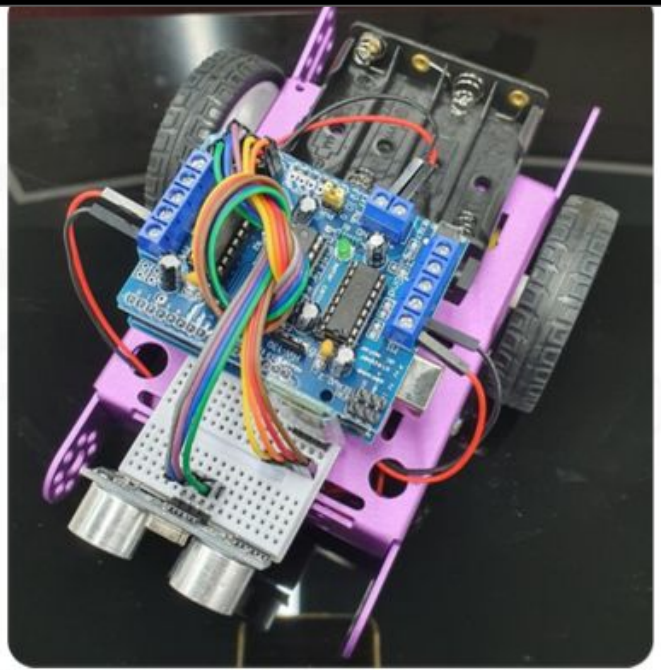


M+: **Red**, GND: **Black**

1. Please make sure to wire the ***red*** and ***black*** cables in the exact same order as the image!
2. If the cables are connected in reverse, the motor not work or heat up, ***so please be careful!***

RC Car Assembly

The autonomous RC car is now complete.
Let’s check if the assembly has been done correctly.

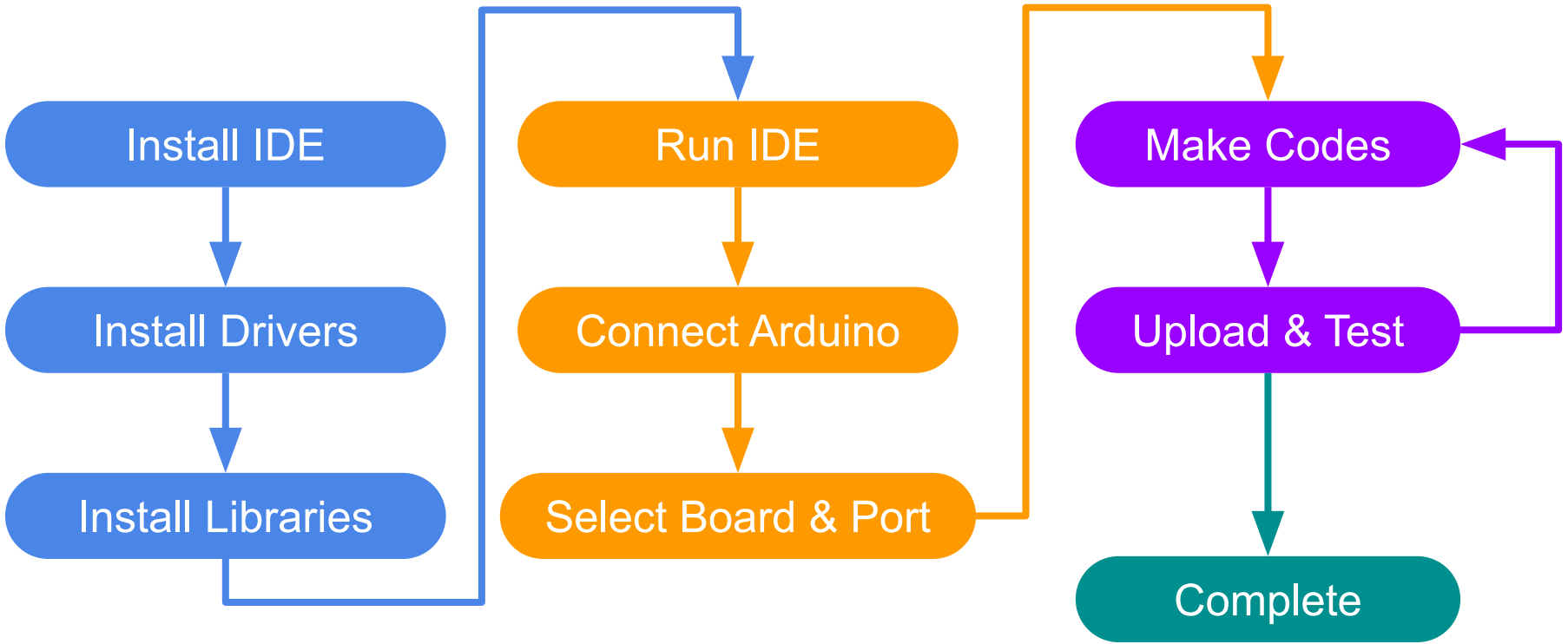


ASSE MBLY TEST

Make Parts Moving

1. Connect to PC
2. LED Blinking
3. Ultrasonic Sensor
4. Wheel Motor Control

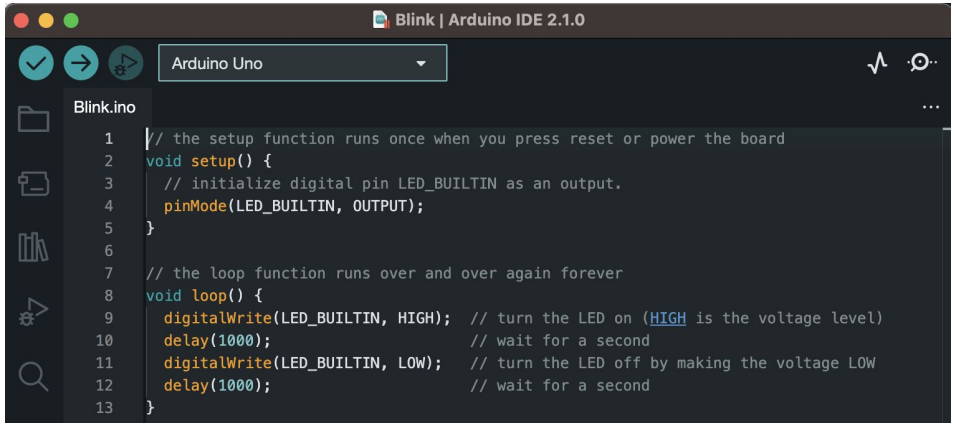
Execution Process



Connect to PC & Make LED blinking

1. Turn on the PC (Windows OS)
2. Connect the Car to PC using USB cable
3. Check the COM port

4. Start Arduino IDE
   - Select Arduino Uno & COM port
   - Select Blink code
   - Upload the code
   - Check the LED Blinking

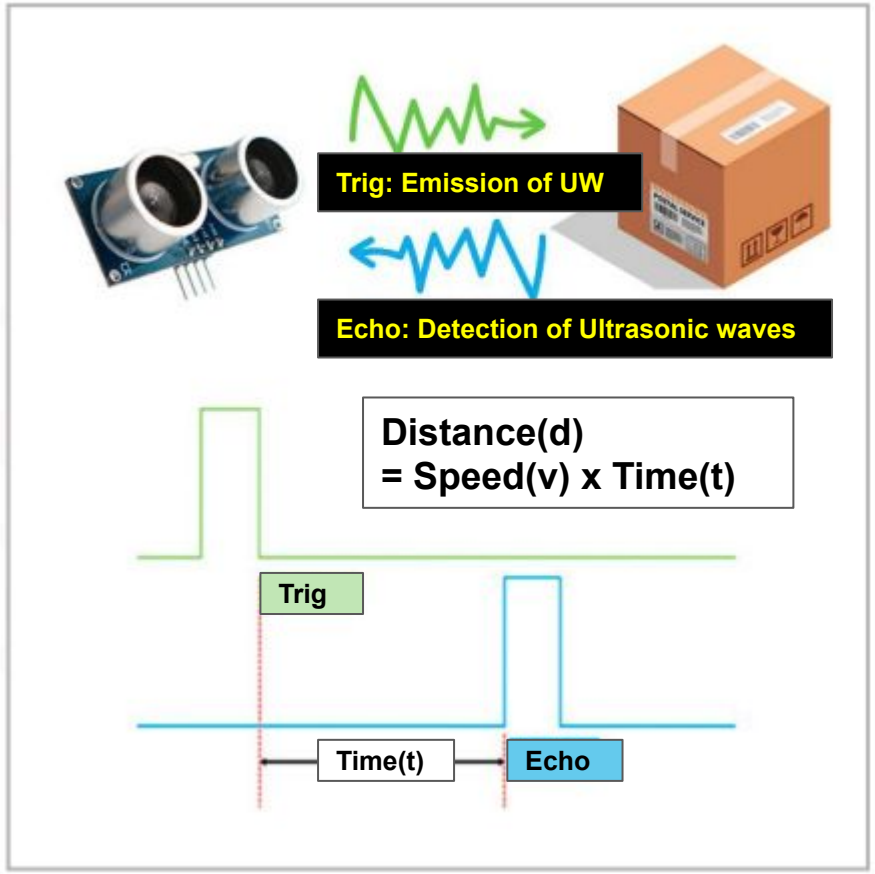


ASSEMBLY TEST

Ultrasonic Sensor

Ultrasonic Sensor

- **Ultrasound** is a sound with a frequency range higher than the **audible** range of 20-40kHz for humans.
- Its **speed** is approximately **340m/s** and it is attenuated by properties such as diffusion, absorption, and scattering.
- Distance can be measured using an ultrasonic sensor.



Example: Ultrasonic Sensor

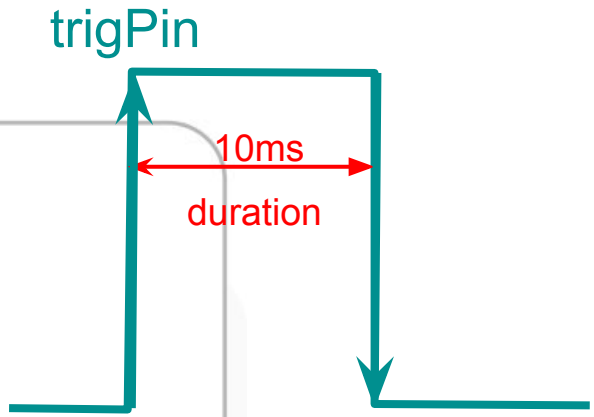
```
1   int trigPin = A0;
2   int echoPin = A1;
3
4   void setup() {
5     Serial.begin(9600);
6     pinMode(echoPin, INPUT);
7     pinMode(trigPin, OUTPUT);
8   }
9
10  void loop() {
11    long duration, distance;
12    digitalWrite(trigPin, HIGH);
13    delayMicroseconds(10);
14    digitalWrite(trigPin, LOW);
15    duration = pulseIn(echoPin, HIGH);
16    distance = ((float)(340 * duration) / 1000) / 2;
17    Serial.print("distance:");
18    Serial.print(distance);
19    Serial.println("mm");
20    delay(500);
21  }
```

[Example Code]

- source: ultrasonic_sensor.ino
- When an object is detected by the ultrasonic sensor, check the distance value(mm) on the serial monitor

Example: Ultrasonic Sensor

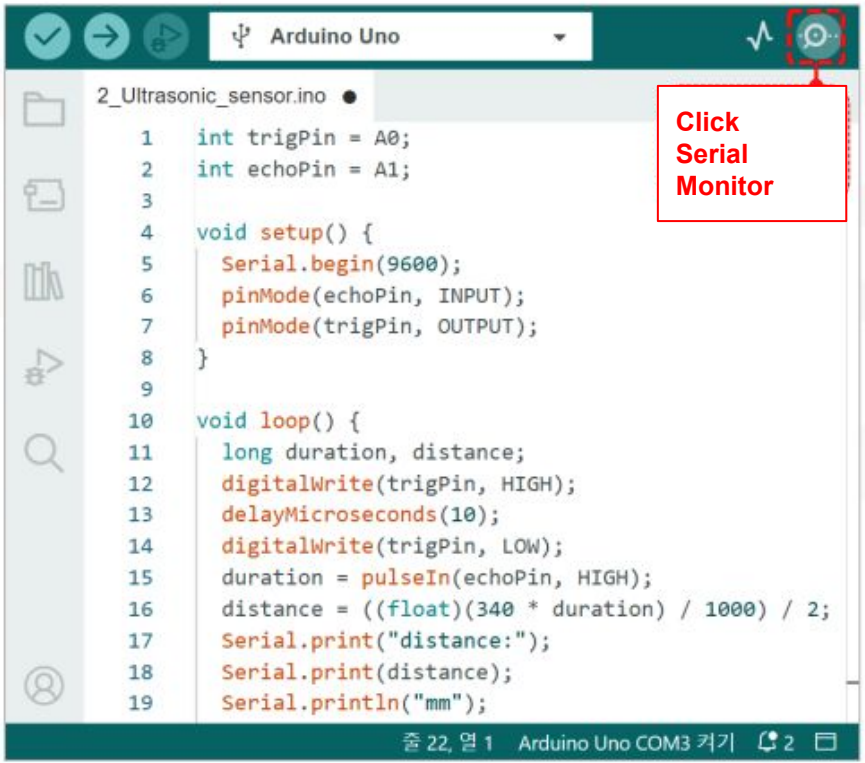
```
10  void loop() {
11    long duration, distance;
12    digitalWrite(trigPin, HIGH);
13    delayMicroseconds(10);
14    digitalWrite(trigPin, LOW);
15    duration = pulseIn(echoPin, HIGH);
16    distance = ((float)(340 * duration) / 1000) / 2;
```

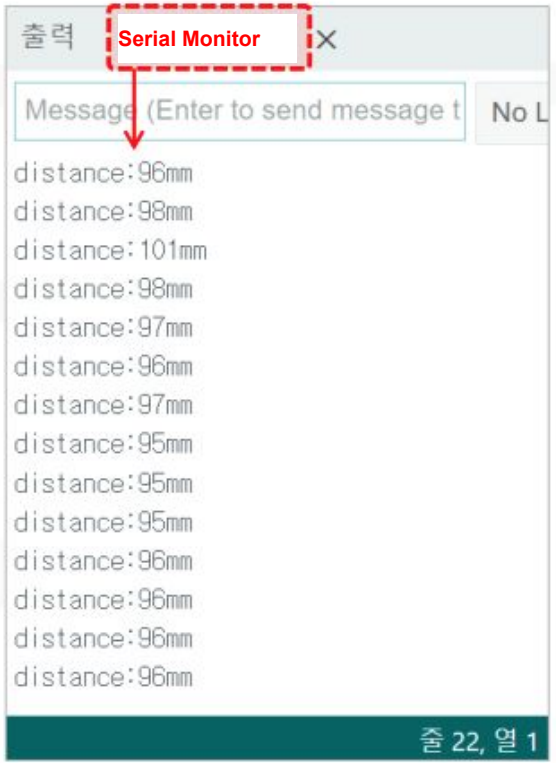


- The trigPin emits an ultrasonic wave for 10ms and then stops.
- The duration of the time that the echoPin is HIGH is stored
- The distance is calculated using the round-trip time of the ultrasonic wave

Example: Ultrasonic Sensor

You can check the distance value of the Ultrasonic sensor in numeric from using the Serial Monitor





ASSEMBLY TEST

DC Motor

Example: DC Motor

```
#include <SoftwareSerial.h>
#include <AFMotor.h>

AF_DCMotor motor_L(1);
AF_DCMotor motor_R(4);

void setup() {
  motor_L.setSpeed(175);
  motor_L.run(RELEASE);
  motor_R.setSpeed(200);
  motor_R.run(RELEASE);
}

void loop() {
 motor_L.run(FORWARD);
 motor_R.run(FORWARD);
 delay(2000);
 motor_L.run(RELEASE);
 motor_R.run(RELEASE);
 delay(1000);
 motor_L.run(BACKWARD);
 motor_R.run(BACKWARD);
 delay(2000);
 motor_L.run(RELEASE);
 motor_R.run(RELEASE);
 delay(1000);
}
```

[Example Code]

- source: front_back.ino
- Roll the wheels forward and backward

Example: DC Motor

```
4   #include <SoftwareSerial.h>
5   #include <AFMotor.h>
6
7   AF_DCMotor motor_L(1);
8   AF_DCMotor motor_R(4);
9
10  void setup() {
11    motor_L.setSpeed(175);
12    motor_L.run(RELEASE);
13    motor_R.setSpeed(200);
14    motor_R.run(RELEASE);
15  }
```

```
AF_DCMotor motor_L(1);
AF_DCMotor motor_R(4);
```

: Set the Left wheel as M1 and the Right wheels as M4

```
motor_L.setspeed(175)
```

: Set the speed of the Left wheels to 175

```
motor_R.setspeed(200)
```

: Set the speed of the Right wheels to 200

If the speed of the left and right motors is different, it can be adjusted by changing the setting values.
The range of speed that can be set is from 0 to 255.

Example: DC Motor

```
17  void loop() {
18    motor_L.run(FORWARD);
19    motor_R.run(FORWARD);
20    delay(2000);
21    motor_L.run(RELEASE);
22    motor_R.run(RELEASE);
23    delay(1000);
24    motor_L.run(BACKWARD);
25    motor_R.run(BACKWARD);
26    delay(2000);
27    motor_L.run(RELEASE);
28    motor_R.run(RELEASE);
29    delay(1000);
30  }
```

The car moves forward for 2 seconds

The car stops for 1 second

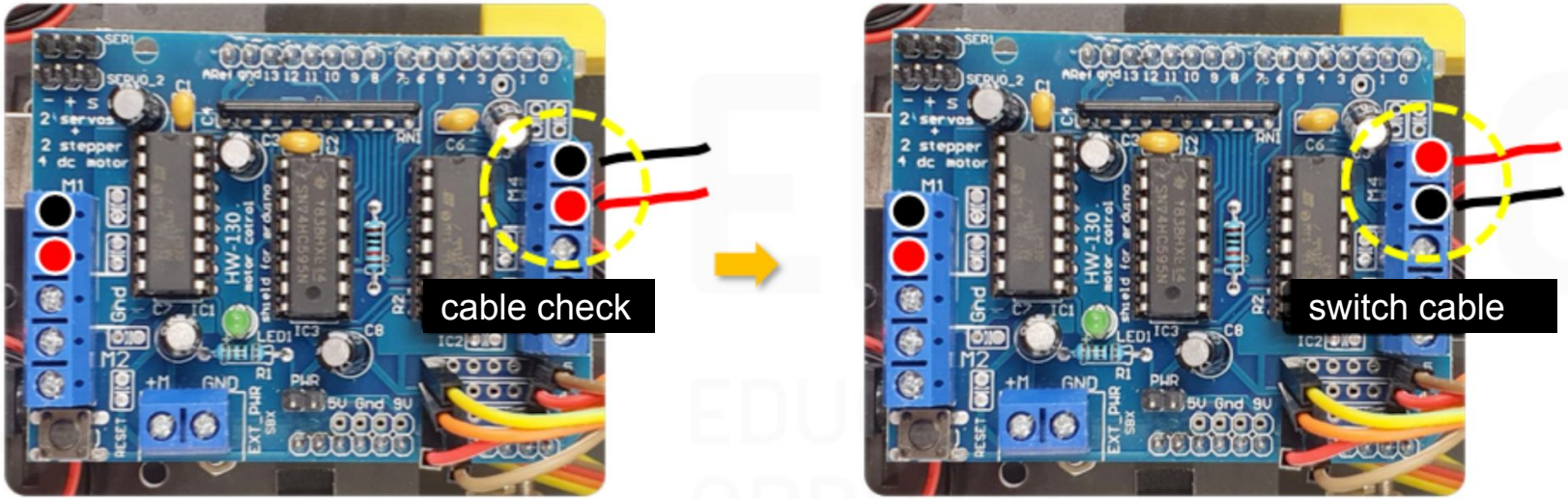
The car moves backward for 2 seconds

delay(ms) determines the operating time of the motor.

The commands inside the loop() function are repeated continuously.

Example: DC Motor

If the wheels are turning in the opposite direction, switch the position of motor lines.



Check the motor lines on the side where the wheel is turning in the opposite direction.

Switch the position of the motor lines on the opposite side.

Demo Video

https://youtu.be/mjEI760NTmo

HW: Rotate the car in place.

```
17  void loop() {
18    motor_L.run(       );
19    motor_R.run(       );
20    delay(       );
21    motor_L.run(       );
22    motor_R.run(       );
23    delay(       );
24    motor_L.run(       );
25    motor_R.run(       );
26    delay(       );
27    motor_L.run(       );
28    motor_R.run(       );
29    delay(       );
30  }
```

1. Rotate the car in place to the RIGHT for 3 seconds
2. Stop the car for 1 second
3. Rotate the car in place to the LEFT for 3 seconds

Check the Answer

```
17  void loop() {
18    motor_L.run(FORWARD);
19    motor_R.run(BACKWARD);
20    delay(3000);
21    motor_L.run(RELEASE);
22    motor_R.run(RELEASE);
23    delay(1000);
24    motor_L.run(BACKWARD);
25    motor_R.run(FORWARD);
26    delay(3000);
27    motor_L.run(RELEASE);
28    motor_R.run(RELEASE);
29    delay(1000);
30  }
```

To forcibly terminate the loop() function, use the exit() function.

exit(0);

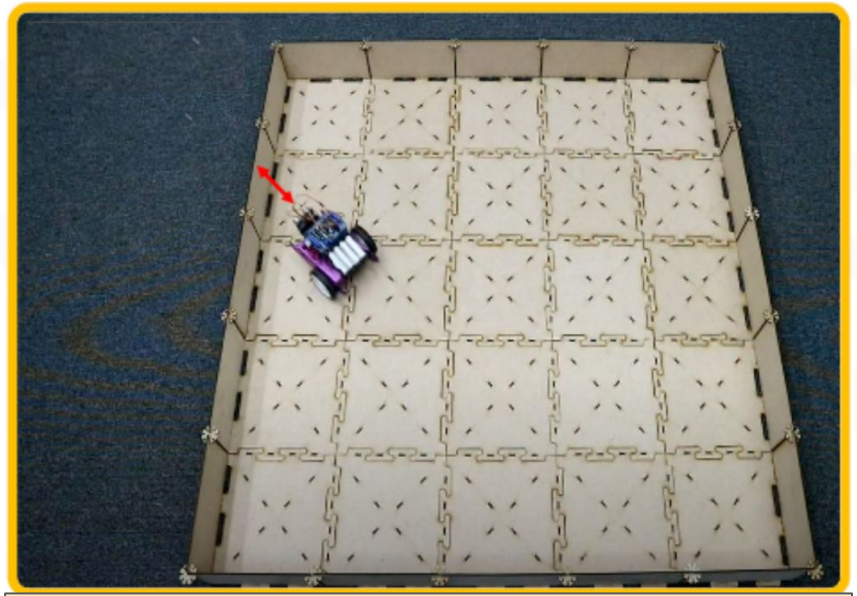
SELF-DRIVING CAR

Self-Driving Car: Obstacle Avoidance

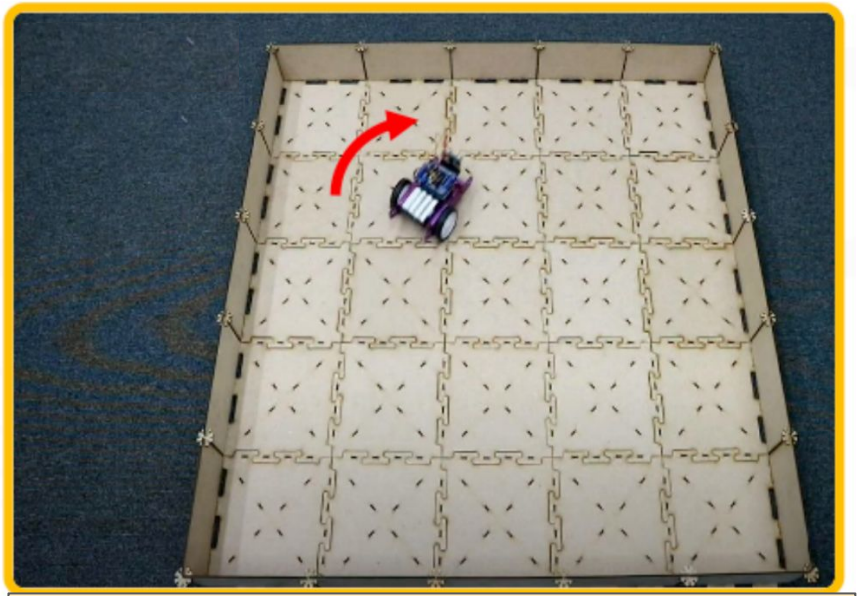
source:
obstacle_drive.ino

Self-driving Car : Obstacle Avoidance

Create a self-driving car that detects obstacles and randomly sets its direction when they are detected.



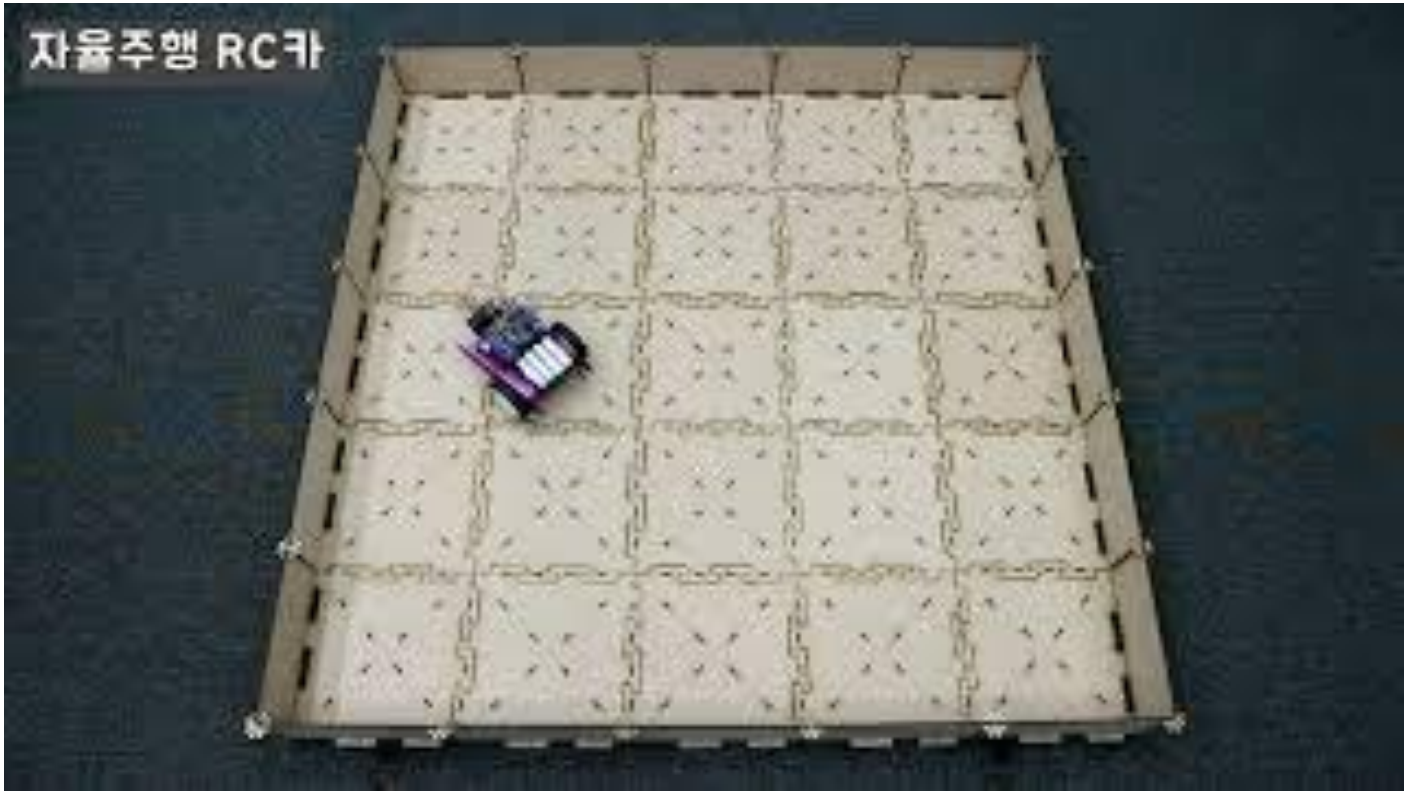
Obstacle detection



Obstacle avoidance (randomly)

Demo Video

https://youtu.be/nBzexRAIByc



Source Code

```
#include <SoftwareSerial.h>
#include <AFMotor.h>
AF_DCMotor motor_L(1);
AF_DCMotor motor_R(4);

int Lspeed = 170;
int Rspeed = 200;

int TrigPin = A0;
int EchoPin = A1;
long duration, distance;

void Obstacle_Check();
void Distance_Measurement();
void Forward();
void Backward();
void Right();
void Left();
void Stop();

void setup() {
  Serial.begin(9600);
  Serial.println("Eduino Smart Car Start!");

  pinMode(EchoPin, INPUT);
  pinMode(TrigPin, OUTPUT);

  motor_L.setSpeed(Lspeed);
  motor_L.run(RELEASE);
  motor_R.setSpeed(Rspeed);
  motor_R.run(RELEASE);
}
```

```
void loop() {
  Forward();
  delay(50);
  Obstacle_Check();
}

void Obstacle_Check() {
  int val = random(2);
  Distance_Measurement();
  delay(50);
  Serial.println(distance);
  while (distance < 300) {
    if (distance < 150) {
      Backward();
      delay(800);
      Stop();
      delay(50);
      Distance_Measurement();
      delay(100);
    }
    else {
      if (val == 0) {
        Right();
        delay(400);
      }
      else if (val == 1) {
        Left();
        delay(400);
      }
      Distance_Measurement();
      delay(100);
    }
  }
}
```

```
void Distance_Measurement() {
  digitalWrite(TrigPin, LOW);
  delay(2);
  digitalWrite(TrigPin, HIGH); delayMicroseconds(10);
  digitalWrite(TrigPin, LOW);
  duration = pulseIn(EchoPin, HIGH);
  distance = ((float)(340 * duration) / 1000) / 2;
  delay(50);
}

void Forward() {
  motor_L.run(FORWARD); motor_R.run(FORWARD);
  motor_L.setSpeed(Lspeed);
  motor_R.setSpeed(Rspeed);
}
void Backward() {
motor_L.run(BACKWARD); motor_R.run(BACKWARD);
  motor_L.setSpeed(Lspeed);
  motor_R.setSpeed(Rspeed);
}
void Right() {
  motor_L.run(FORWARD); motor_R.run(BACKWARD);
  motor_L.setSpeed(Lspeed);
  motor_R.setSpeed(Rspeed*0.5);
}
void Left() {
  motor_L.run(BACKWARD); motor_R.run(FORWARD);
  motor_L.setSpeed(Lspeed*0.5);
  motor_R.setSpeed(Rspeed);
}
void Stop() {
  motor_L.run(RELEASE); motor_R.run(RELEASE);
  motor_L.setSpeed(0); motor_R.setSpeed(0);
}
```

Self-driving Car : Obstacle Avoidance

```
13  #include <SoftwareSerial.h>
14  #include <AFMotor.h>
15  AF_DCMotor motor_L(1);
16  AF_DCMotor motor_R(4);
17
18  int Lspeed = 170;
19  int Rspeed = 200;
20
21  int TrigPin = A0;
22  int EchoPin = A1;
23  long duration, distance;
24
25  void Obstacle_Check();
26  void Distance_Measurement();
27  void Forward();
28  void Backward();
29  void Right();
30  void Left();
31  void Stop();
```

Left wheel: M1
Right wheel: M4

Set motor rotation speed

Determine obstacle detection

Measure distance using Ultrasonic sensor

Control car motion

Function** is a block of statements composed of **a set of commands

Self-driving Car : Obstacle Avoidance

The output of the motor **varies** according to the **battery level** and **motor specifications**.

Adjust the speed of the left and right motors to set the necessary speed for a **smooth driving** experience.

```
40    motor_L.setSpeed(Lspeed);
41    motor_L.run(RELEASE);
42    motor_R.setSpeed(Rspeed);
43    motor_R.run(RELEASE);
```

Left motor rotation speed

Right motor rotation speed

```
18    int Lspeed = 170;
```

Set Left motor speed

```
19    int Rspeed = 200;
```

Set Right motor speed

Self-driving Car : Obstacle Avoidance

```
53  void Obstacle_Check() {
54    int val = random(2);
55    Distance_Measurement();
56    delay(50);
57
58    Serial.println(distance);
59
60    while (distance < 300) {
61      if (distance < 150) {
62        Backward();
63        delay(800);
64        Stop();
65        delay(50);
66        Distance_Measurement();
67        delay(100);
68      }
```

Determine if an obstacle detected

If the Ultrasonic sensor distance is less than 300mm, execute a while loop

If the Ultrasonic sensor distance is less than 150mm, move back for 0.08 seconds, then stop for 0.05 seconds

Forward → distance < 300 → distance < 150 → OBSTACLE

Self-driving Car : Obstacle Avoidance

```
      else {
        if (val == 0) {
          Right();
          delay(400);
        }
        else if (val == 1) {
          Left();
          delay(400);
        }
        Distance_Measurement();
        delay(100);
      }
    }
  }
```

else: 150 <= distance < 300

Turn Right

Turn Left

Forward → distance < 300 → distance < 150 → OBSTACLE

Self-driving Car : Obstacle Avoidance

```
85  void Distance_Measurement() {
86    digitalWrite(TrigPin, LOW);
87    delay(2);
88    digitalWrite(TrigPin, HIGH);
89    delayMicroseconds(10);
90    digitalWrite(TrigPin, LOW);
91    duration = pulseIn(EchoPin, HIGH);
92    distance = ((float)(340 * duration) / 1000) / 2;
93    delay(50);
94  }
```

Function to measure distance using an Ultrasonic sensor

Self-driving Car : Obstacle Avoidance

```
97  void Forward() {
98    motor_L.run(FORWARD);  motor_R.run(FORWARD);
99    motor_L.setSpeed(Lspeed);  motor_R.setSpeed(Rspeed);
100 }
101 void Backward() {
102   motor_L.run(BACKWARD);  motor_R.run(BACKWARD);
103   motor_L.setSpeed(Lspeed);  motor_R.setSpeed(Rspeed);
104 }
105 void Right() {
106   motor_L.run(FORWARD);  motor_R.run(BACKWARD);
107   motor_L.setSpeed(Lspeed);  motor_R.setSpeed(Rspeed*0.5);
108 }
109 void Left() {
110   motor_L.run(BACKWARD);  motor_R.run(FORWARD);
111   motor_L.setSpeed(Lspeed*0.5);  motor_R.setSpeed(Rspeed);
112 }
113 void Stop() {
114   motor_L.run(RELEASE);       motor_R.run(RELEASE);
115   motor_L.setSpeed(0);  motor_R.setSpeed(0);
116 }
```

Move the car forward

Move the car backward

Turn the car right

Turn the car left

Stop the car

Run the code on the car!

HW: Change the codes => Turn Right

```
        else {
          if (val == 0) {
            Right();
            delay(400);
          }
          else if (val == 1) {
            Left();
            delay(400);
          }
          Distance_Measurement();
          delay(100);
        }
      }
    }
```

HW: Change the codes => Turn Left

```
        else {
          if (val == 0) {
            Right();
            delay(400);
          }
          else if (val == 1) {
            Left();
            delay(400);
          }
          Distance_Measurement();
          delay(100);
        }
      }
    }
```

Answer

Turn Left

```
      else {
        if (val == 0) {
          Right();
          delay(400);
        }
        else if (val == 1) {
          Left();
          delay(400);
        }
        Distance_Measurement();
        delay(100);
      }
    }
  }
```

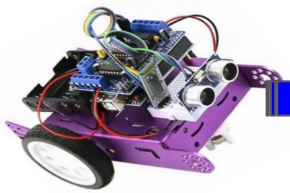
Turn Right

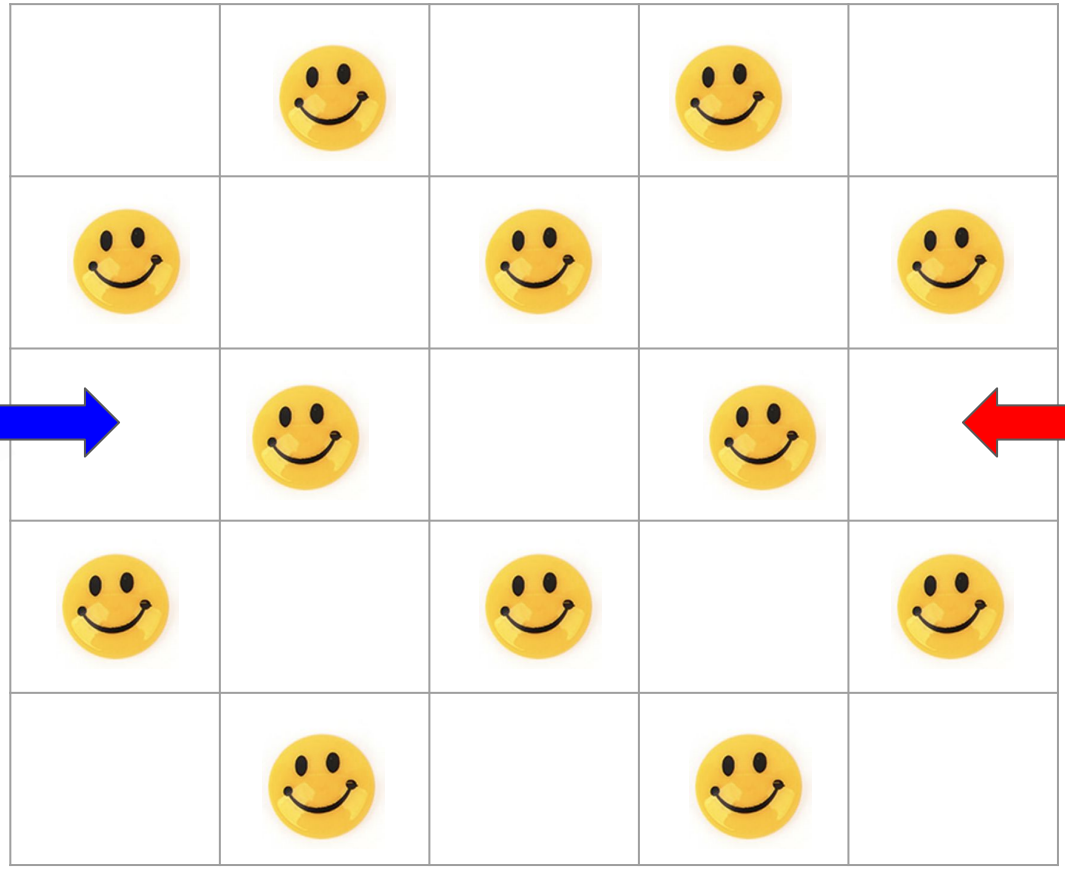
```
      else {
        if (val == 0) {
          Right();
          delay(400);
        }
        else if (val == 1) {
          Left();
          delay(400);
        }
        Distance_Measurement();
        delay(100);
      }
    }
  }
```

Let’s Start Playing the Game!

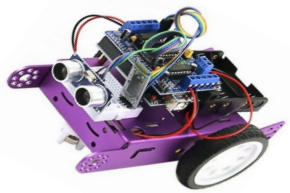
Garbage Collector!

BLUE Team





RED Team



저작권 안내

Copyright Notice

This material has been created for educational purposes with permission from Eduino, and commercial use is prohibited.
All rights in this material are reserved by Eduino.

본 자료는 에듀이노(Eduino)에 의해 수정, 번역, 작성되었고 소유권 또한 에듀이노((주)로보다인시스템)에 있습니다.

해당자료의 사용은 교육목적에 한정하며 상업적 목적의 사용을 금지합니다.

소유권자의 허가를 받지 않고 무단으로 수정, 삭제, 배포 할 수 없으며 위반시 민형사적 법적 처벌을 받을 수도 있습니다.

본 강의자료에는 네이버에서 제공한 나눔글꼴 서체가 적용되어 있습니다.

- ㈜로보다인시스템 -

Thank You.