IIUM ROBOTEAM

ROBOTIC WORKSHOP 11.0 MODULE

PART 2:

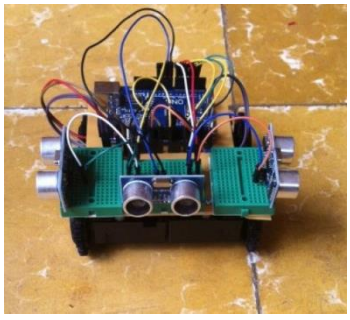
WALL FOLLOWING ROBOT

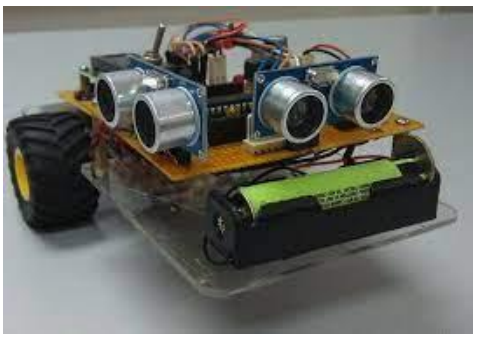
WALL FOLLOWING ROBOT

1. Introduction

1.1 What is Wall Following Robot

Wall following robot is the advance of the line following robot. This robot is designed to move along the wall without hitting it. Wall following robot has sensor mounted on the body to detect the obstacle and wall, and drive the DC motors attached to the wheels such that the robot keeps moving along the wall. For the sensor, we use ultrasonic sensor to detect and measure the distance of the wall to before the robot hit it.

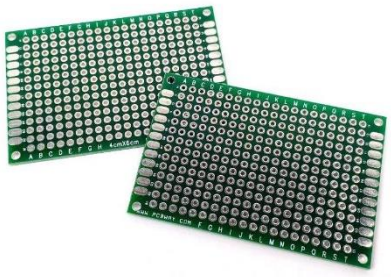




2. Assembly of the robot

2.1 Robot Parts (buat list semua parts + gambar)

- Chassis



Donut board (m3.0)

Stand-off (50mm, m3.0)

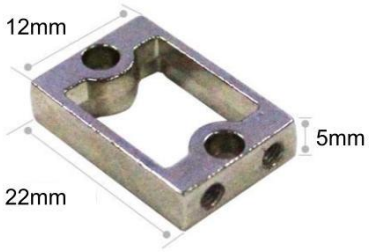
Screw (6mm, m3.0)

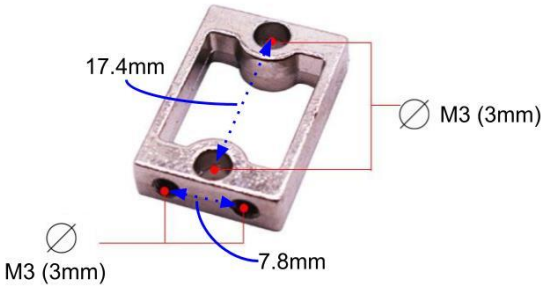
Nuts (m3.0)

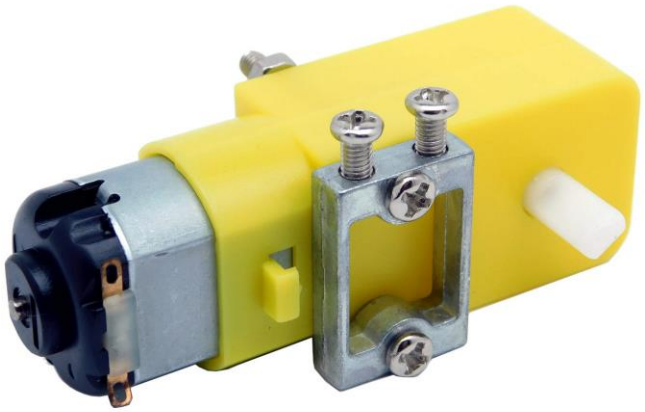
- DC motor and wheel

- Wheel Diameter: 66mm
- No-Load Speed (6V): 200 RPM (±10%) @ 0.2A
- No-Load Speed (3V): 90 RPM (±10%) @ 0.15A
- Maximum Efficiency: 2.0 kg-cm / 170 RPM / 0.6A
- Reduction Ratio: 1:48

- TT motor Bracket







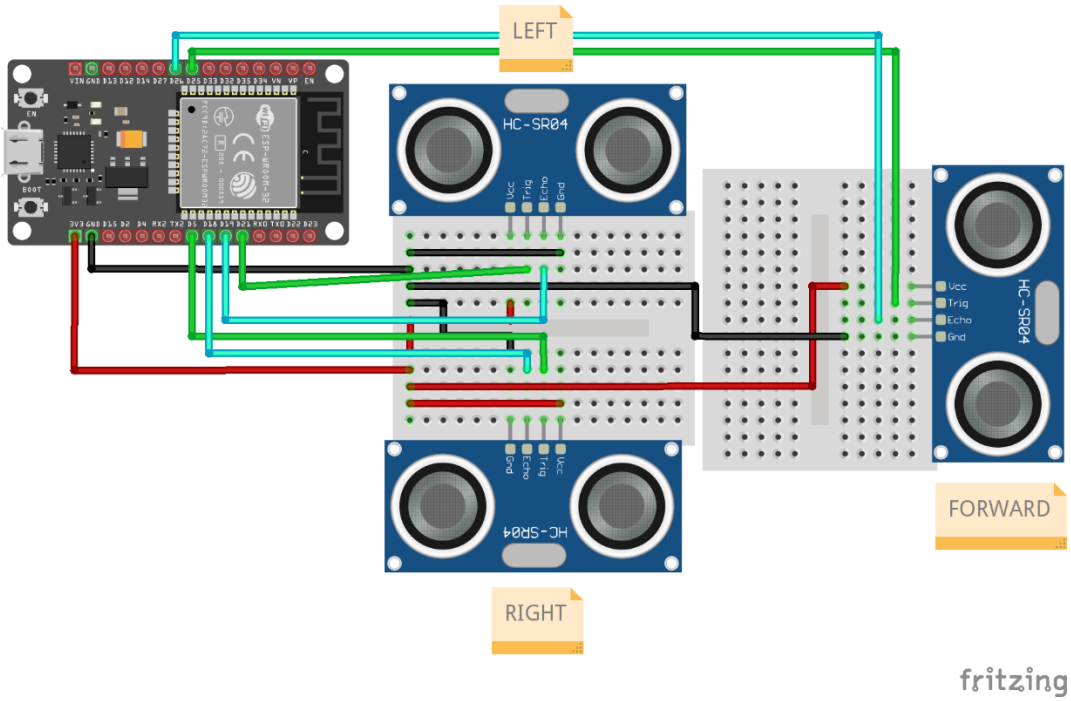
- Use to attach motor to the chassis
- Can drill the donut board to screw the bracket

- Battery and battery holder

- Li – ion battery 3.7V, 6800mAh

3. Sample Coding

3.1 3 ultrsonic sensor: *This is the example to read distance in 3 ways using 3 ultrasonic sensors*




```
#define trigPin1 5
#define echoPin1 18
#define trigPin2 21
#define echoPin2 19
#define trigPin3 25
#define echoPin3 26

long duration, distance,
RightSensor,BackSensor,FrontSensor,LeftSensor;

void setup()
{
Serial.begin (9600);
pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
pinMode(trigPin3, OUTPUT);
pinMode(echoPin3, INPUT);
}

void loop() {
SonarSensor(trigPin1, echoPin1);
RightSensor = distance;
SonarSensor(trigPin2, echoPin2);
```

```
LeftSensor = distance;
SonarSensor(trigPin3, echoPin3);
FrontSensor = distance;

Serial.print(LeftSensor);
Serial.print(" - ");
Serial.print(FrontSensor);
Serial.print(" - ");
Serial.println(RightSensor);
}

void SonarSensor(int trigPin,int echoPin)
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
}
```

3.2 Bluetooth Car Controller – Arduino RC Controller Apps

```
#include <BluetoothSerial.h>
BluetoothSerial btSerial;
#define BT_NAME "ESP32BT-Roboteam"  // Set bluetooth name

#define BUILTIN_LED 2  // LED is active low
#define M1A 33
#define M1B 25
#define M2A 26
#define M2B 27

#define PWM_FREQUENCY 1000
#define PWM_RESOLUTION 8
#define M1A_PWM_CHANNEL 0
#define M1B_PWM_CHANNEL 1
#define M2A_PWM_CHANNEL 2
#define M2B_PWM_CHANNEL 3

#define MAX_SPEED 200

boolean btConnected = false;
char key, previousKey;
int motorLeft, motorRight;
long previousMillis = 0;
int timeout = 1000;

void setup() {
  pinMode(BUILTIN_LED, OUTPUT);

  ledcSetup(M1A_PWM_CHANNEL, PWM_FREQUENCY, PWM_RESOLUTION);
  ledcSetup(M1B_PWM_CHANNEL, PWM_FREQUENCY, PWM_RESOLUTION);
  ledcSetup(M2A_PWM_CHANNEL, PWM_FREQUENCY, PWM_RESOLUTION);
  ledcSetup(M2B_PWM_CHANNEL, PWM_FREQUENCY, PWM_RESOLUTION);

  ledcAttachPin(M1A, M1A_PWM_CHANNEL);
  ledcAttachPin(M1B, M1B_PWM_CHANNEL);
  ledcAttachPin(M2A, M2A_PWM_CHANNEL);
  ledcAttachPin(M2B, M2B_PWM_CHANNEL);

  Serial.begin(115200);
  btSerial.begin(BT_NAME);

  Serial.println("ESP32 Bluetooth Mobile Robot");
  Serial.println();

  digitalWrite(BUILTIN_LED, HIGH);
}
```

```
void loop() {
  robotMove(0,0);
  if (btSerial.available()) {
    previousMillis = millis();

    char inChar = (char)btSerial.read();
    Serial.print(inChar);

    // if (btConnected == false) {
    //   btConnected = true;
    //   digitalWrite(BUILTIN_LED, LOW); // Turn on led
    //   Serial.println("Bluetooth connected.");
    // }

    if (inChar >= '0' && inChar <= '9') {
      key = inChar;

        switch (key) {
          case '0':
            Serial.println("Robot stop.");
            motorLeft = 0;
            motorRight = 0;
            break;

          case '1':
            Serial.println("Robot move forward.");
            motorLeft = 200;
            motorRight = 200;
            break;

          case '2':
            Serial.println("Robot move backward.");
            motorLeft = -200;
            motorRight = -200;
            break;

          case '3':
            Serial.println("Robot turn left.");
            motorLeft = -200;
            motorRight = 200;
            break;

          case '4':
            Serial.println("Robot turn right.");
            motorLeft = 200;
            motorRight = -200;
            break;
        }

        motorLeft = constrain(motorLeft, -MAX_SPEED, MAX_SPEED);
```

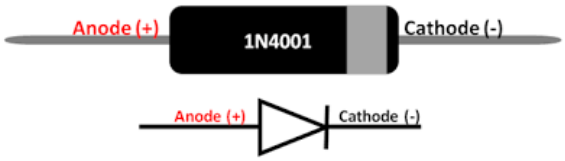
```
        motorRight = constrain(motorRight, -MAX_SPEED,
MAX_SPEED);
        robotMove(motorLeft, motorRight);

    }
  }
}

void robotMove(int speedLeft, int speedRight) {
  speedLeft = constrain(speedLeft, -255, 255);
  speedRight = constrain(speedRight, -255, 255);

  if (speedLeft > 0) {
    int speedL = map(speedLeft, 0, 255, 255, 0);
    ledcWrite(M1A_PWM_CHANNEL, speedL);
    ledcWrite(M1B_PWM_CHANNEL, 255);
  } else {
    int speedL = map(speedLeft, -255, 0, 0, 255);
    ledcWrite(M1A_PWM_CHANNEL, 255);
    ledcWrite(M1B_PWM_CHANNEL, speedL);
  }

  if (speedRight > 0) {
    int speedR = map(speedRight, 0, 255, 255, 0);
    ledcWrite(M2A_PWM_CHANNEL, speedR);
    ledcWrite(M2B_PWM_CHANNEL, 255);
  } else {
    int speedR = map(speedRight, -255, 0, 0, 255);
    ledcWrite(M2A_PWM_CHANNEL, 255);
    ledcWrite(M2B_PWM_CHANNEL, speedR);
  }
  delay(50);
}
```

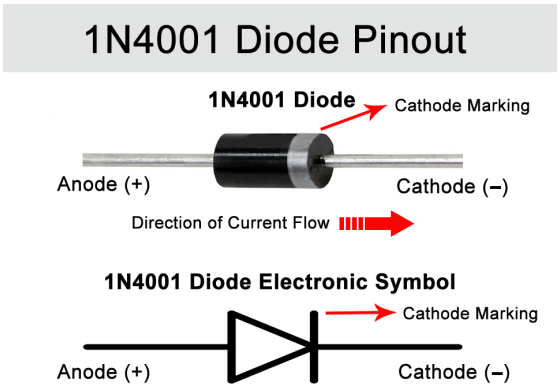
4. Additional Task

Every group will be given a DC motor fan blade to be included in the robot. A task will be given in the game need to be solve using the fan.

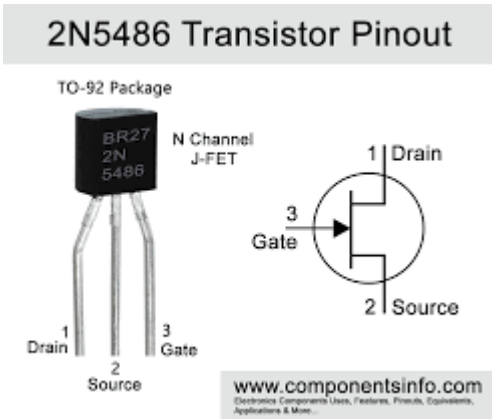
Fan Blade

Mini DC motor



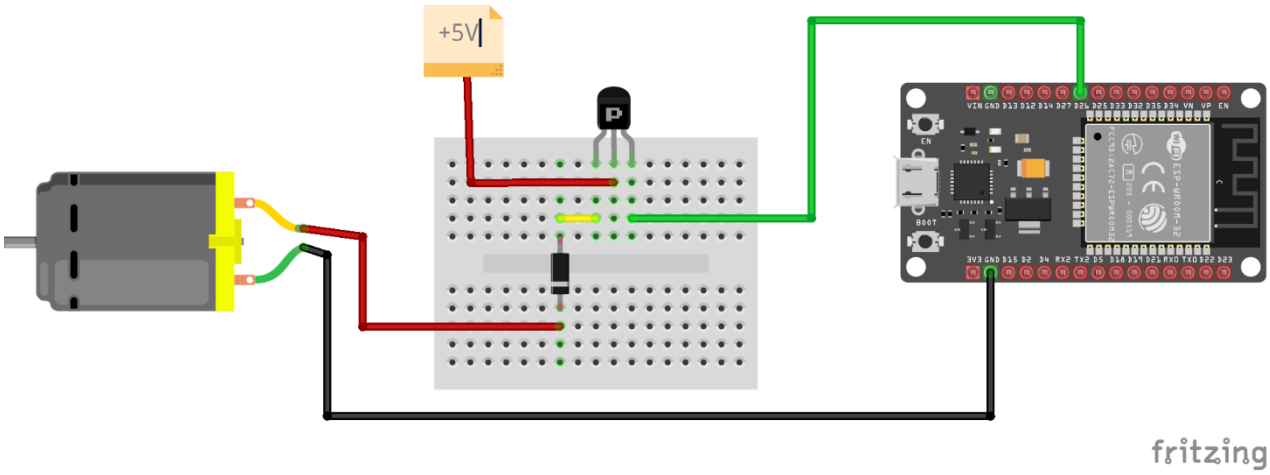


Diode



Transistor

Circuit diagram:



COMPETITION BRIEFING: WALL FOLLOWING ROBOT

Task:

On a sunny day, there was a fire burning in a location that couldn't be reached by the firefighters due to precautions. Hence, a robot firetruck is needed to put out the fire. Heads up National Safety Robotics Group! We put our hope on you.

Game Field and Robots

Game Field:

1. Field diagram is included at the end of the Rules.
2. Additional checkpoints will be added in semi-finals and semi-finals round.

Teams:

1. Every teams consists of 3 students.
2. **One (1) pilot** and **two (2) other participants** can bring and set up their robot. Once the setup of the robot is done, only the pilot will be allowed to be in the game field to pickup the robot if needed, the other two (2) members need to step out of the field.
3. Other team members were allowed to give advice without touching the controller (phone) and entering the game field.

Robot Mode:

1. Robot will be in 2 modes. Autonomous and manual controlled robot.
2. Robot need to pass the first and second gate using the autonoumous mode.
3. Then after entering the second gate, the robot need to be controlled manually to solve the given task.

Competition and Scoring

Matches:

1. Each match shall last for a maximum of **three (3) minutes.**
2. **One (1) minute** will be provided for the setting up of the robots and after one (1) minute each team has to stop their setup and leave the game field for the match to begin.

3. Once the match starts, team members are not permitted to go into the game field.
4. If the team wants to change any coding or want to reupload the code, they may do so but the robot needs to be restart from the nearest gate. Only pilot is allowed to go into the field to take the robot. Please be noted that extra time will not be given.
5. For each round, two groups will be at both starting points.
6. There will be two tasks that need to be completed by each group.
7. The first task is wall following. The robot needs to go through the field and find two gates. This task needs to be completed in autonomous mode.
8. After the second (2) gate (purple), the robot need to be controlled manually using bluetooth controller. The robot needs to go near the fire and put it out using the fan provided.
9. Time will be stop once one of the team completes all the tasks given.
10. The match may end earlier in the following cases:
    i. One of the teams win.
    ii. Game field is damaged by a team.
    iii. Referees decided that the continuity of the match is no longer possible.

11. Any attempt to cheat ,lie or any bad behavior or misconduct to the judges ,referee or the crew will lead to penalty or disqualification (depends on the judge decision).

Scoring :

1. The first robot to put out the fire while complete all the checkpoints using required modes will be the winner.
2. If the team fail to be present during the match, the opponent will be the winner. Walk over rule will be applied.
3. To determine the winners of each group, the following scoring format is going to be applied:
    i. The four (4) teams with the highest point in the group will qualify for the semi-final round.
    ii. For first, second and third round, +100 points for any team who reaches the first and second gates using autonomous mode.
    iii. There will be 10 (ten) checkpoints before arriving at gate 2. For each checkpoints reached they will gain 10 marks per checkpoint.
    iv. In the manually controlled mode, they will be two (2) checkpoints with +10 marks each.
    v. +30 points for any team that successfully put out the fire at the last checkpoint in manual mode using the fans.
    vi. Teams can skip the autonomous mode and start at the manual mode but the maximum marks they will get is only 50 marks.

vii. If both teams skip the autonomous mode, team that completed manual task with the shortest time will be the winner.

viii. If both teams can't complete the task within **three (3) minutes**, at the end of the game, the team with the highest mark will be the winner.

Walk Over Rule

The match will begin at the designated time whether both teams are present or not. The following rules apply:

i. If one of the teams are not present at the game field during their scheduled match, the match will still proceed.

ii. The team will be called for 3 times in 1 minutes.

iii. If the team still not present after the 3 times called, they are considered forfeited for the match, thus giving their opponent an uncontested victory.