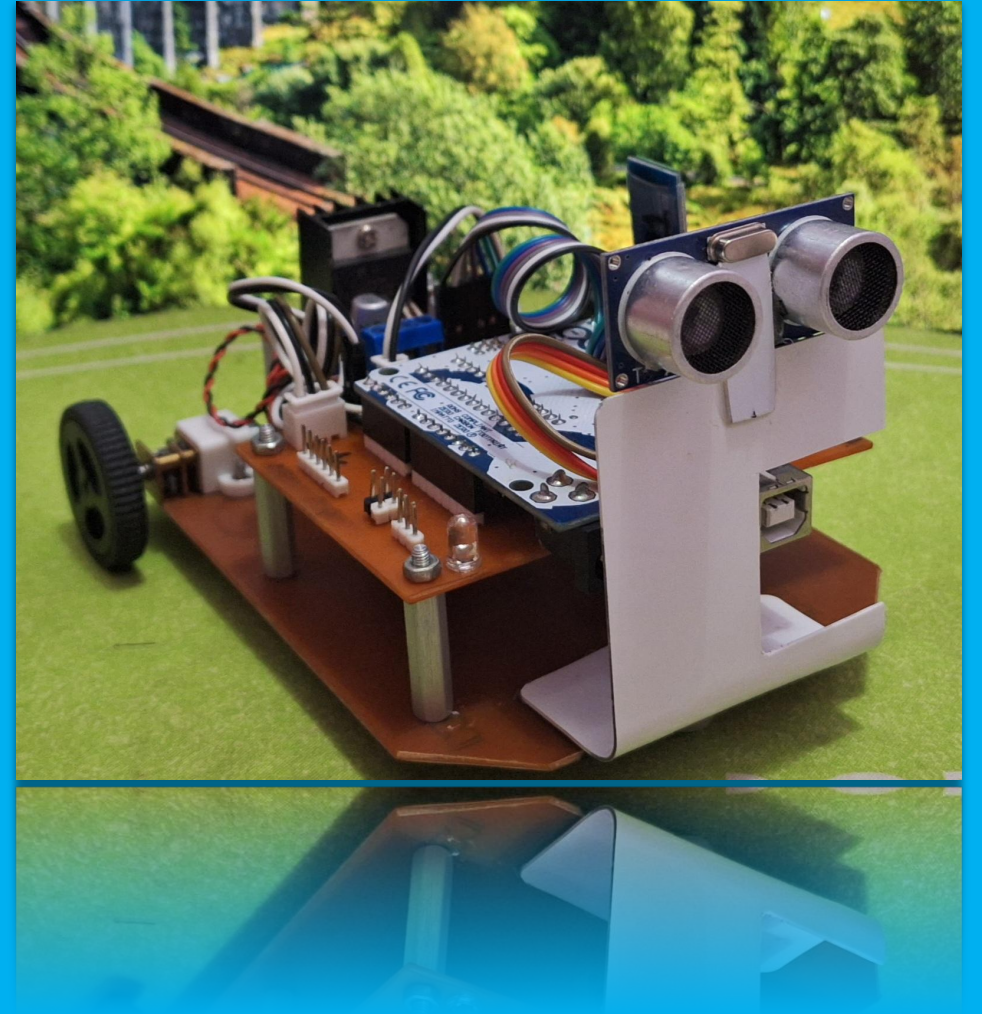


MikonkuBot

Education robot



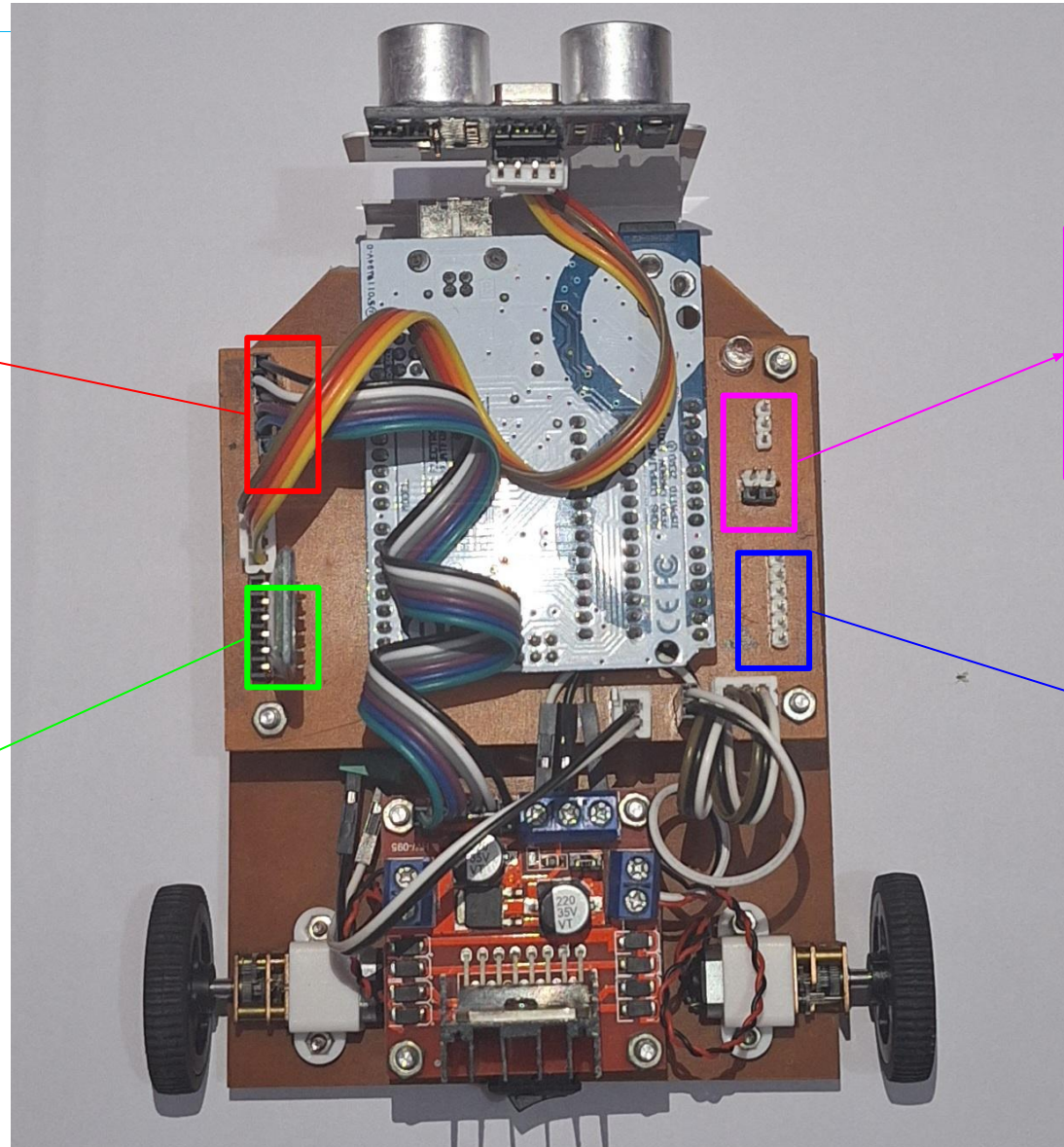
#1 Layout/Schematic

Digital Pin:
D11
to
D6

Bluetooth Module:
Rx to D2
Tx to D3

PowerPin:
White = 5v/3v3
Black = GND

Analog Pin:
A0
to
A5



**Jangan memutar langsung roda nya
dengan tangan!**

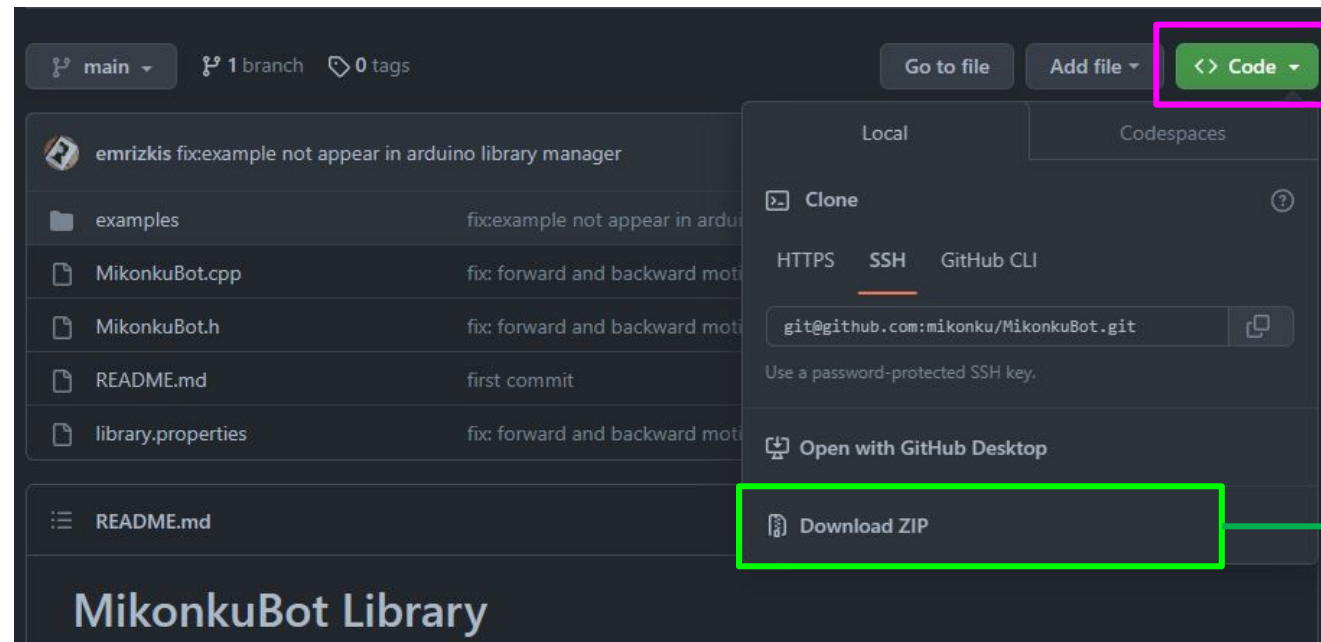
Bagaimana jika tidak punya MikonkuBot?

#2 MikonkuBot Library

MikonkuBot Library berisi perintah-perintah untuk menjalankan robot serta untuk menggunakan beberapa sensor atau perangkat lain yang terhubung pada MikonkuBot

Download MikonkuBot Library

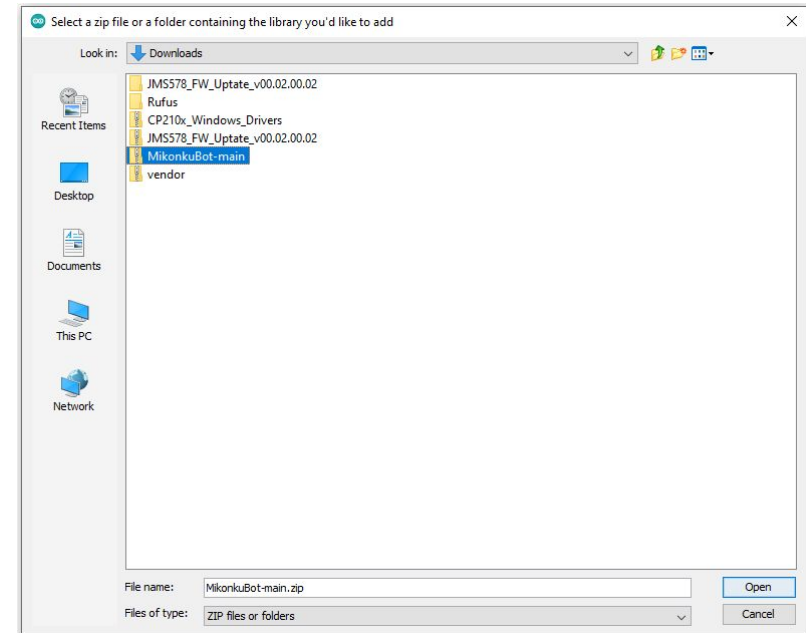
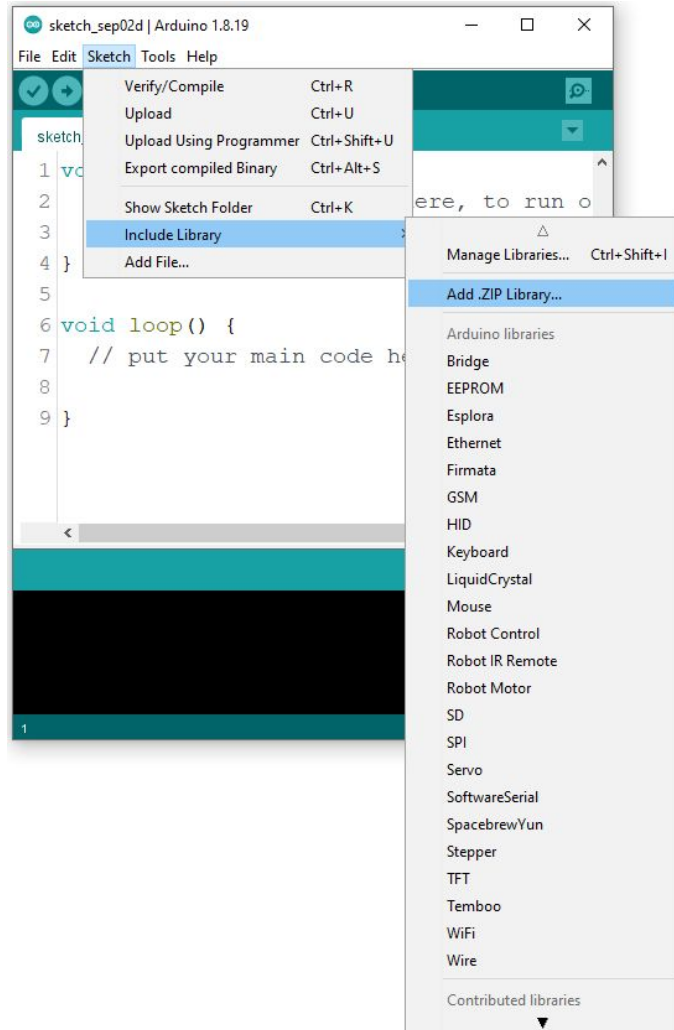
<https://github.com/mikonku/MikonkuBot>



Klik ini menampilkan
pilihan download

Klik ini untuk download file
dalam bentuk zip

Add MikonkuBot Library To Arduino IDE

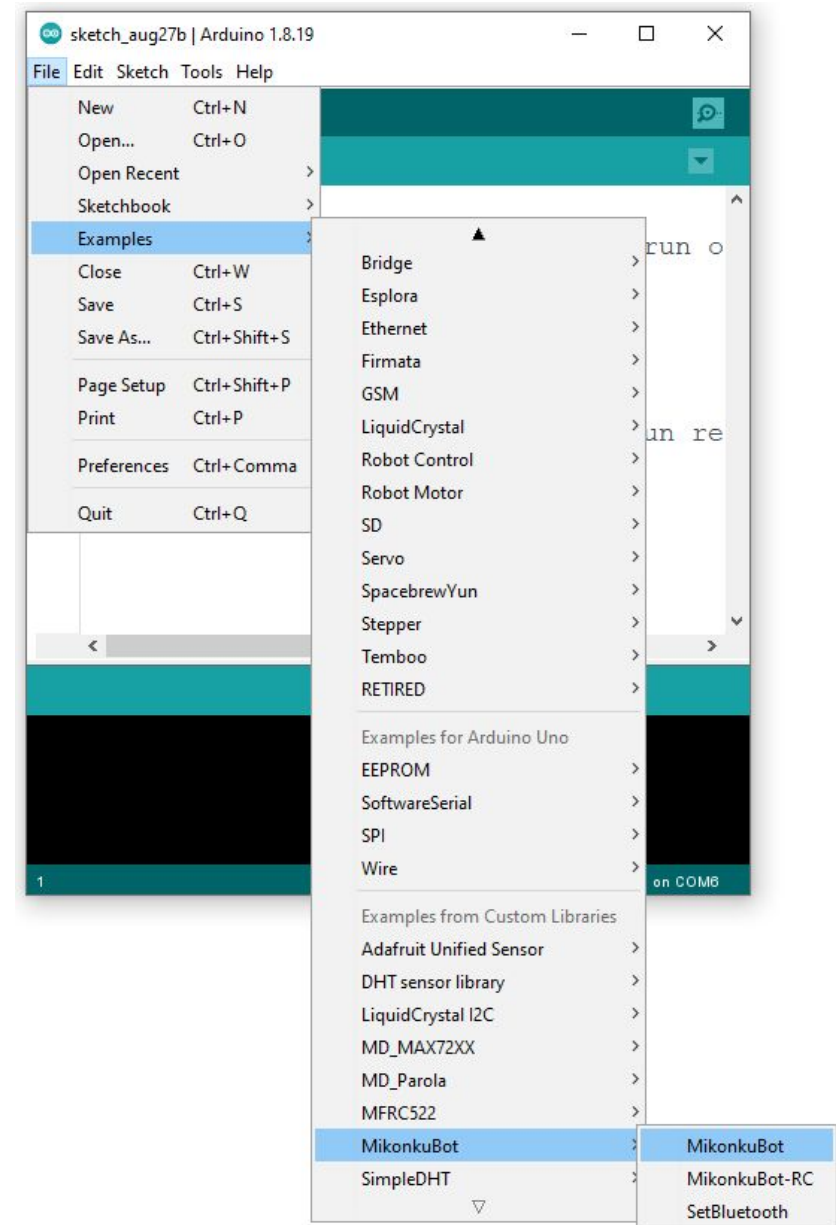


#3 MikonkuBot Library Examples

Examples

Berisi contoh kode untuk menjalankan MikonkuBot.

Go to
File>Examples>MikonkuBot>MikonkuBot



```
#include <MikonkuBot.h>
```

```
// Define the pins for your MikonkuBot instance
```

```
const int ENA_PIN = 11;
```

```
const int IN1_PIN = 10;
```

```
const int IN2_PIN = 9;
```

```
const int IN3_PIN = 8;
```

```
const int IN4_PIN = 7;
```

```
const int ENB_PIN = 6;
```

```
// Create an instance of MikonkuBot
```

```
MikonkuBot myBot(ENA_PIN, IN1_PIN, IN2_PIN, IN3_PIN, IN4_PIN, ENB_PIN);
```

```
void setup() {
```

```
    // Initialize the MikonkuBot
```

```
    myBot.begin();
```

```
    myBot.setSpeed(200,200);
```

```
}
```

```
void loop() {
```

```
    // Put your code here
```

```
}
```

```
#include <MikonkuBot.h>
```

mengimport library MikonkuBot agar dapat digunakan

```
// Define the pins for your MikonkuBot instance
```

```
const int ENA_PIN = 11;  
const int IN1_PIN = 10;  
const int IN2_PIN = 9;  
const int IN3_PIN = 8;  
const int IN4_PIN = 7;  
const int ENB_PIN = 6;
```

menentukan Pin ENA, IN1, IN2, IN3, IN4, dan ENB

```
// Create an instance of MikonkuBot
```

```
MikonkuBot myBot(ENA_PIN, IN1_PIN, IN2_PIN, IN3_PIN, IN4_PIN, ENB_PIN);
```

membuat object dengan nama myBot diisi dengan pin-pin yang sebelumnya telah ditentukan. **myBot** hanyalah nama object artinya kita dapat mengubah dengan nama lainnya

```
void setup() {
```

```
    // Initialize the MikonkuBot
```

```
    myBot.begin();
```

```
    myBot.setSpeed(200,200);
```

```
}
```

memulai robot, ini wajib ada jika robot ingin bekerja

mengatur speed dari masing-masing roda

```
void loop() {
```

```
    // Put your code here
```

```
}
```

#4 Perintah-perintah Pada MikonkuBot

Method	Keterangan
begin();	Untuk mulai menyalakan MikonkuBot, diletakkan sekali pada void setup(). Contoh: myBot.begin()
setSpeed(left_speed,right_speed);	Untuk mengatur kecepatan roda kiri dan roda kanan. <i>left_speed</i> dan <i>right_speed</i> dapat diisi dengan nilai interger dari 0 hingga 255. Contoh: myBot.setSpeed(200,200)
goForward(delay);	Untuk membuat robot bergerak maju lurus ke depan, <i>delay</i> merupakan waktu fungsi ini berjalan dalam satuan mili sekon. Contoh jika ingin membuat robot bergerak maju kedepan selama 2 detik: myBot.goForward(2000)
goBackward(delay);	Untuk membuat robot bergerak mundur ke belakang, <i>delay</i> merupakan waktu fungsi ini berjalan dalam satuan mili sekon. Contoh jika ingin membuat robot bergerak mundur selama 2 detik: myBot.goBackward(2000)
goTurnRight(delay);	Untuk membuat robot belok ke kanan, <i>delay</i> merupakan waktu fungsi ini berjalan dalam satuan mili sekon. Contoh jika ingin membuat robot bergerak belok ke kanan selama 1 detik: myBot.goTurnRight(1000)
goTurnLeft(delay);	Untuk membuat robot belok ke kiri, <i>delay</i> merupakan waktu fungsi ini berjalan dalam satuan mili sekon. Contoh jika ingin membuat robot bergerak belok ke kiri selama 1 detik: myBot.goTurnLeft(1000)
stop();	Untuk membuat robot berhenti. Contoh: myBot.stop()
light(state)	Untuk menyalakan LED pada robot, <i>state</i> dapat diisi dengan HIGH atau LOW . Contoh: myBot.light(HIGH) atau myBot.light(LOW)

Challenge!

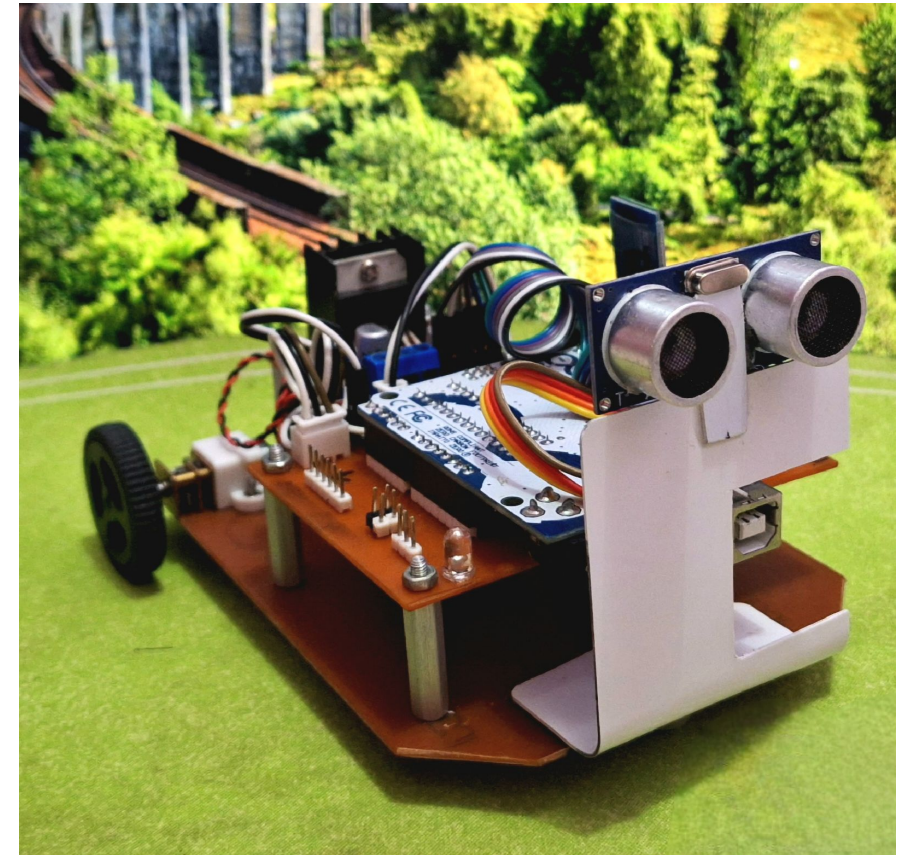
Buat robot agar dapat bergerak dengan bentuk lintasan berbentuk kotak!

#5 Project 1: RC (Remote Control) Car Robot

Definisi

Mobil Robot Pengendali Jarak Jauh (Remote Control Car Robot atau RC Car Robot) adalah jenis robot yang dirancang untuk **dikendalikan** oleh seorang **operator** manusia **melalui perangkat remote control**.

RC Car Robot biasanya memiliki kemampuan bergerak, seperti mobil mainan, dan dapat digunakan untuk berbagai tujuan, termasuk hiburan, kompetisi, atau bahkan penelitian dan pengembangan.

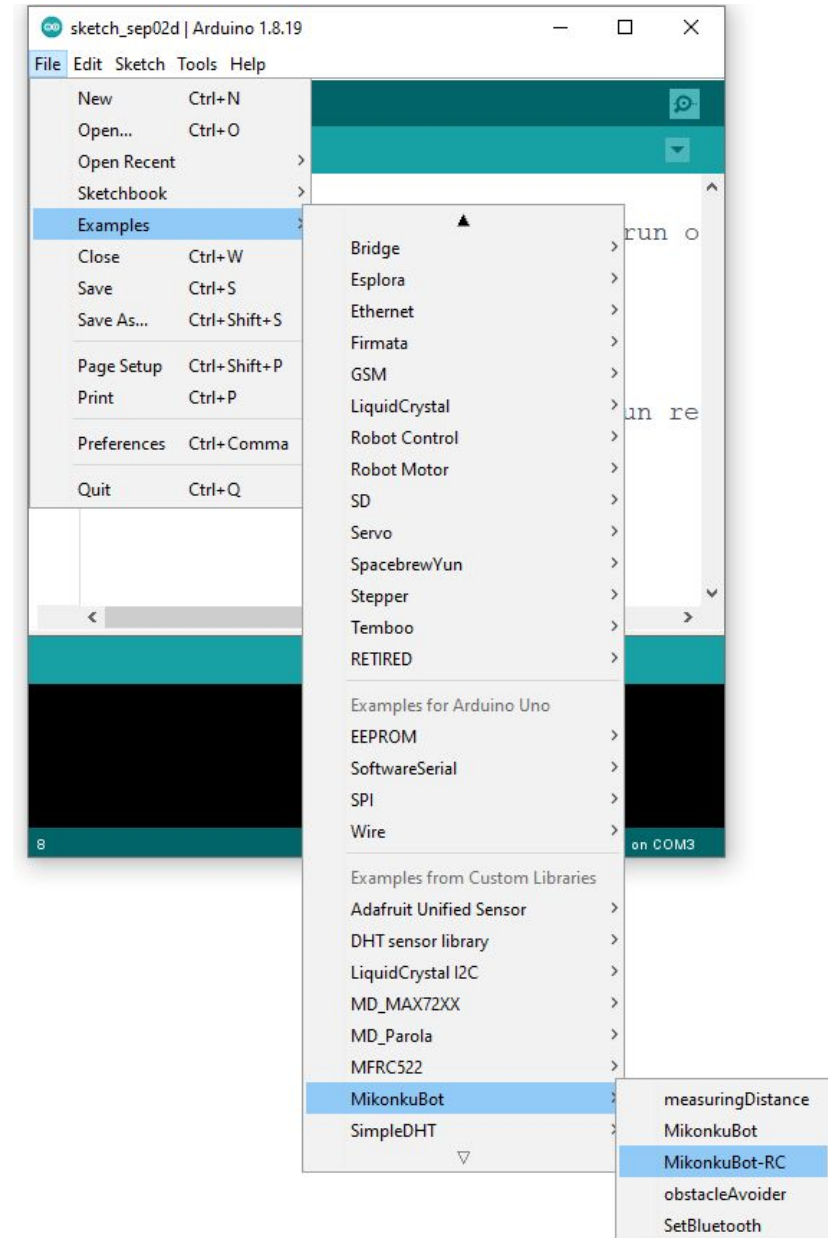


Example Code

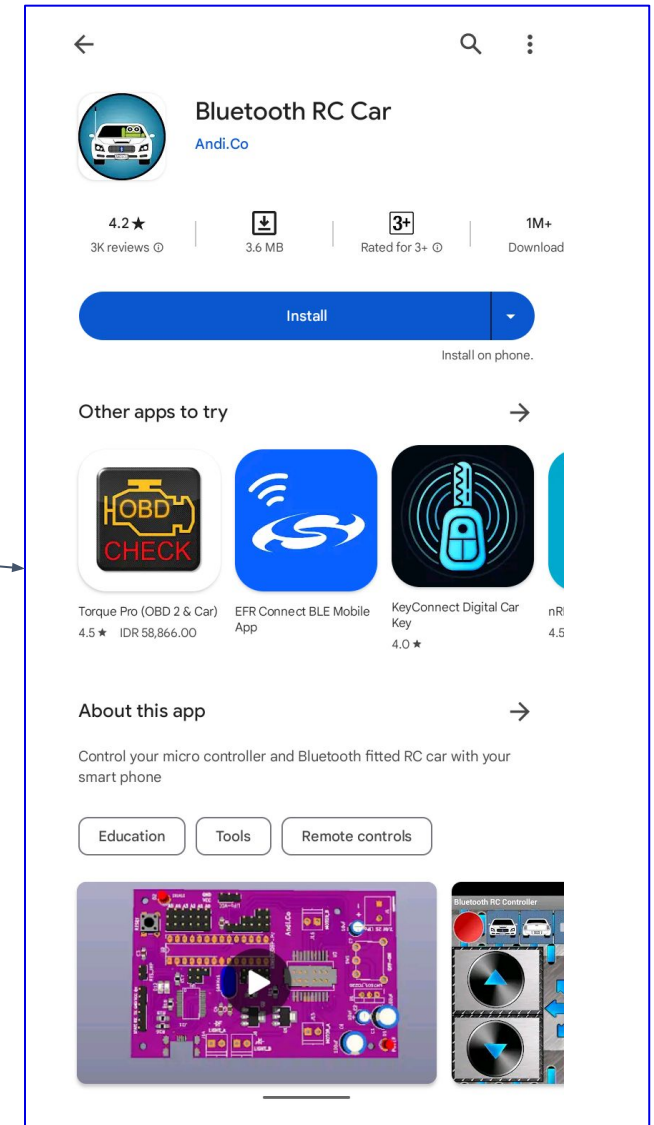
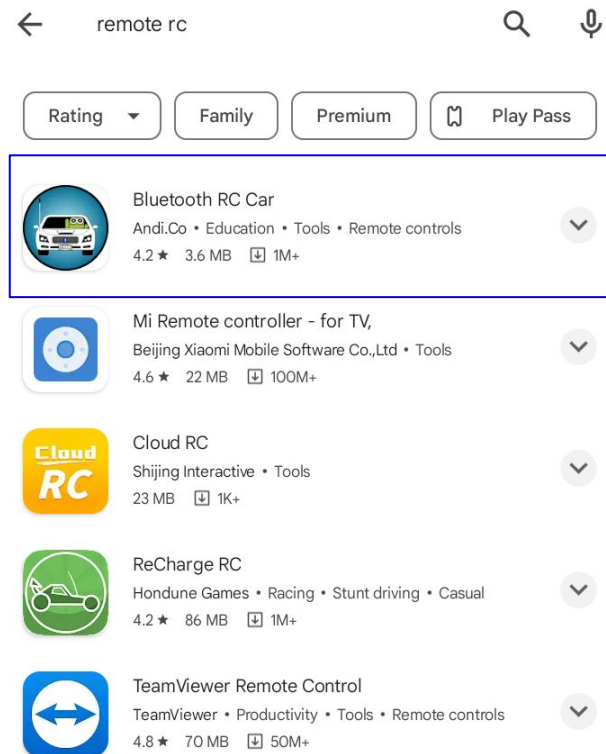
Berisi contoh kode untuk menjalankan sensor ultrasonic MikonkuBot.

Go to

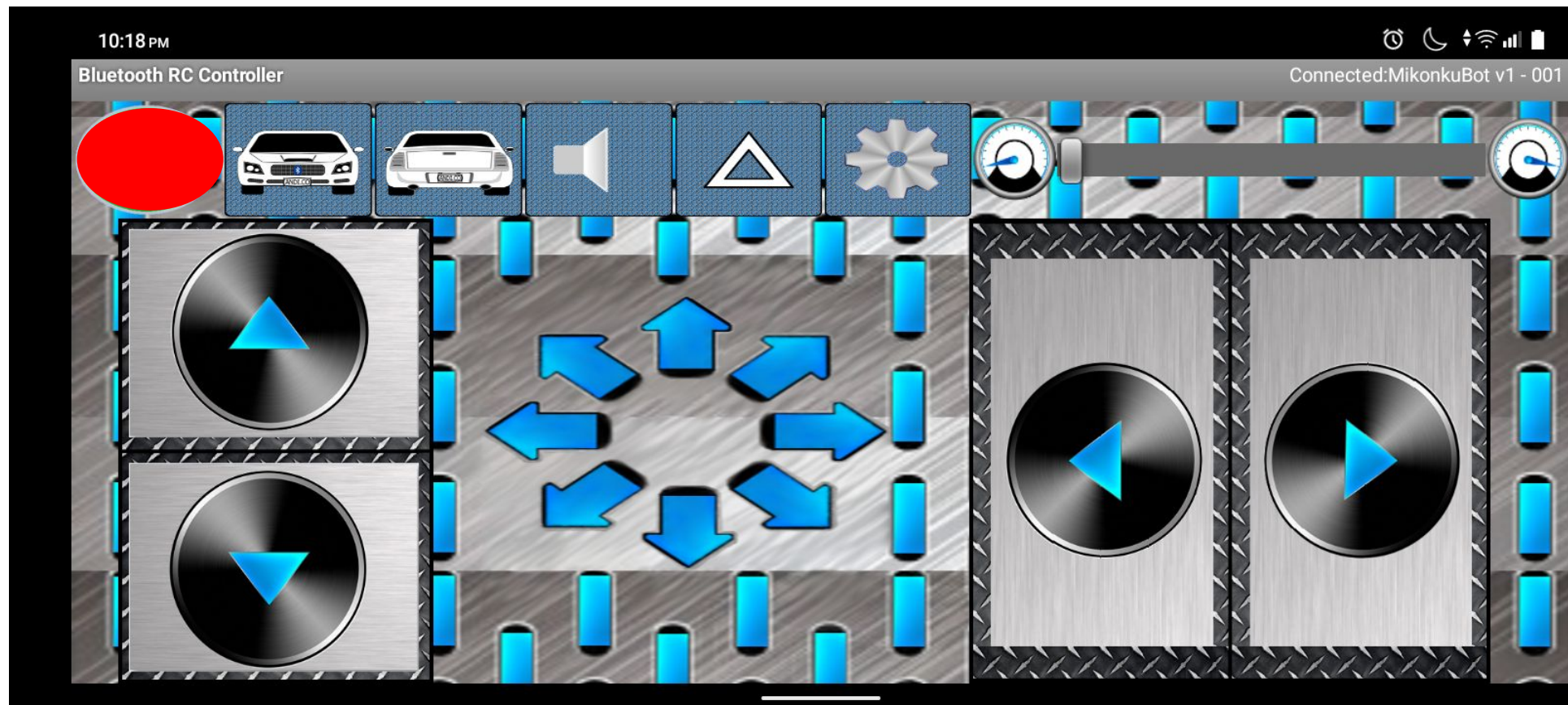
File>Examples>MikonkuBot>MikonkuBot-RC



Remote Control (Android App)



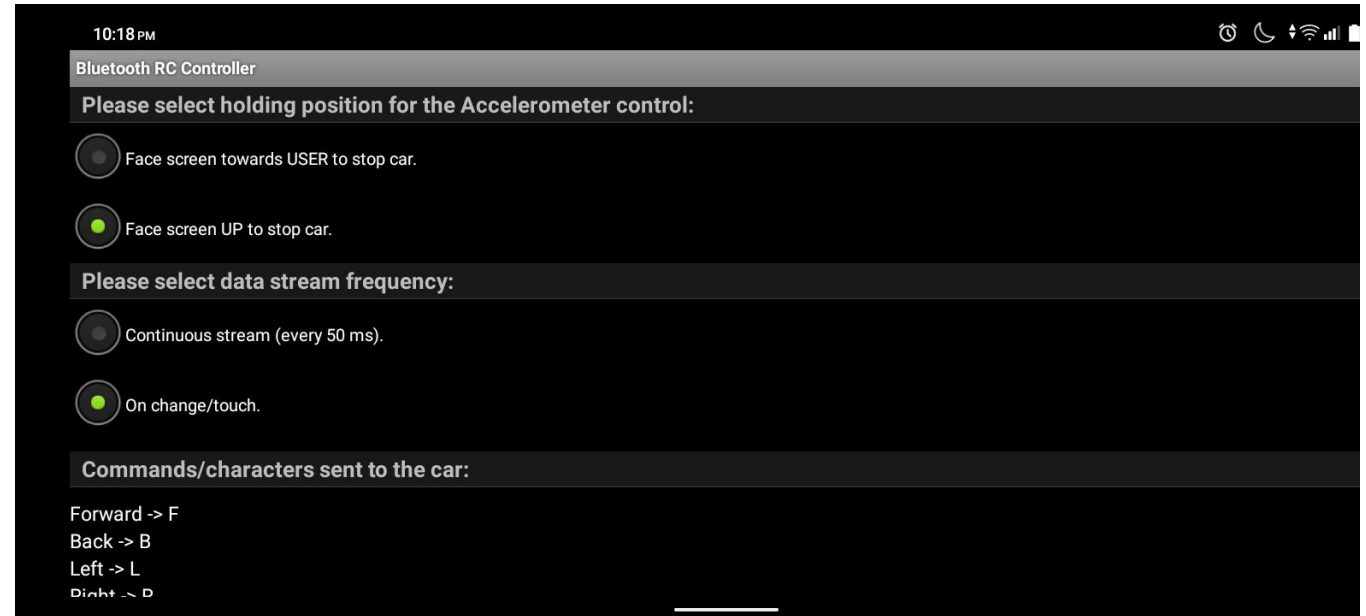
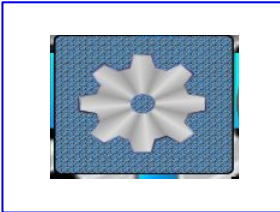
Open Remote RC Car App



Setting Type Stream Data

Go to:

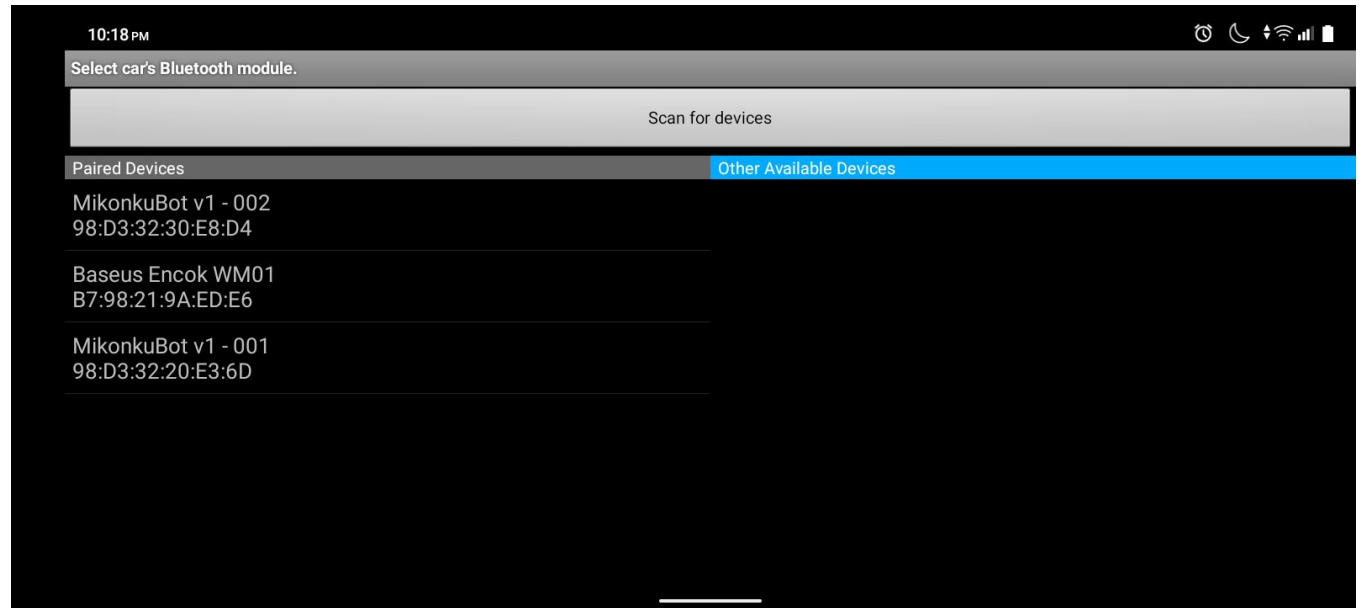
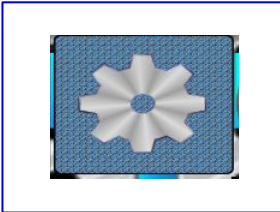
Options Menu > Settings



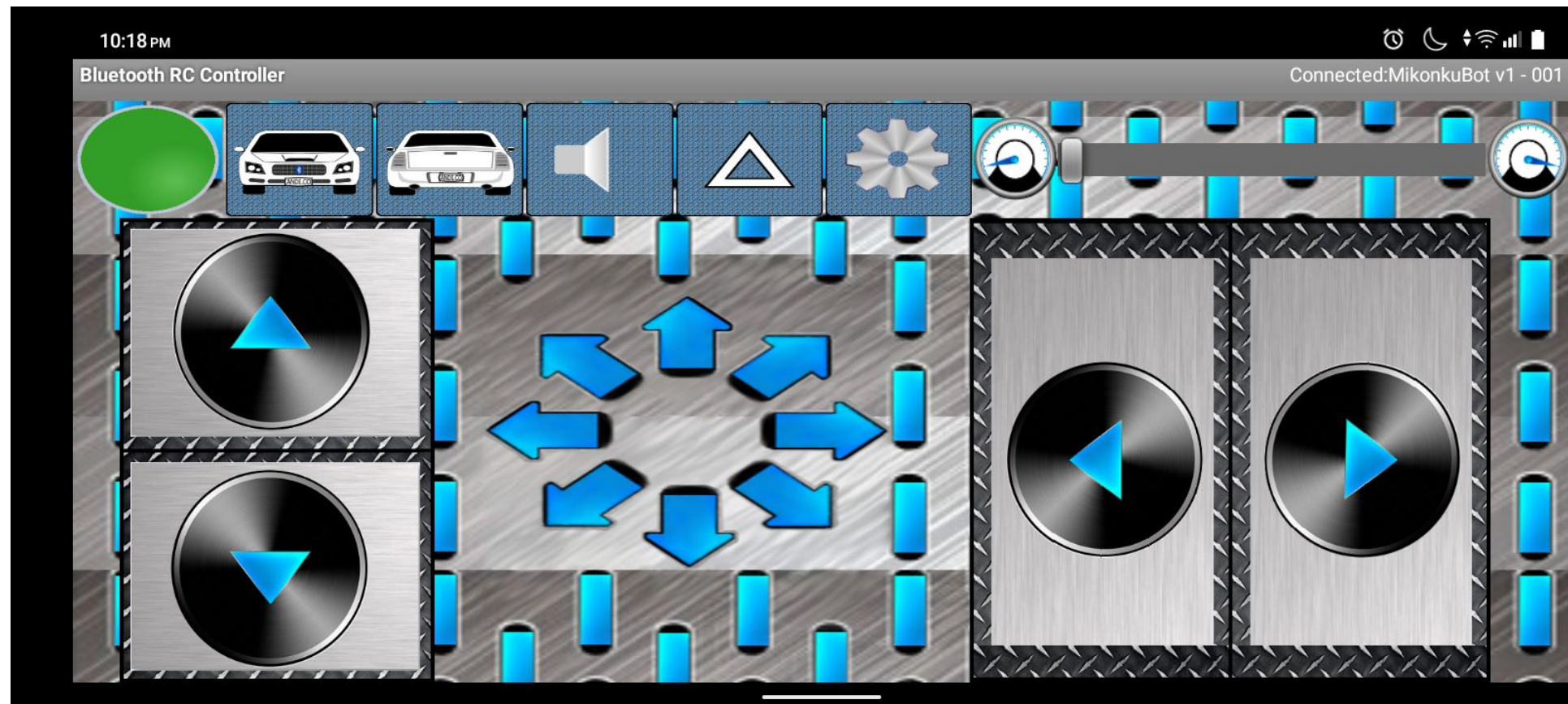
Connect To Car (Robot)

Go to:

Options Menu > Connect to Car



Play With Robot!



Bagaimana jika tidak memiliki smartphone android?

Anda dapat membuat remote nya sendiri, namun butuh beberapa konfigurasi. Berikut adalah link tutorialnya (modul dan video):

modul : <https://github.com/mikonku/MikonkuBot/blob/main/docs/Remote%20Control%20Car.pdf>

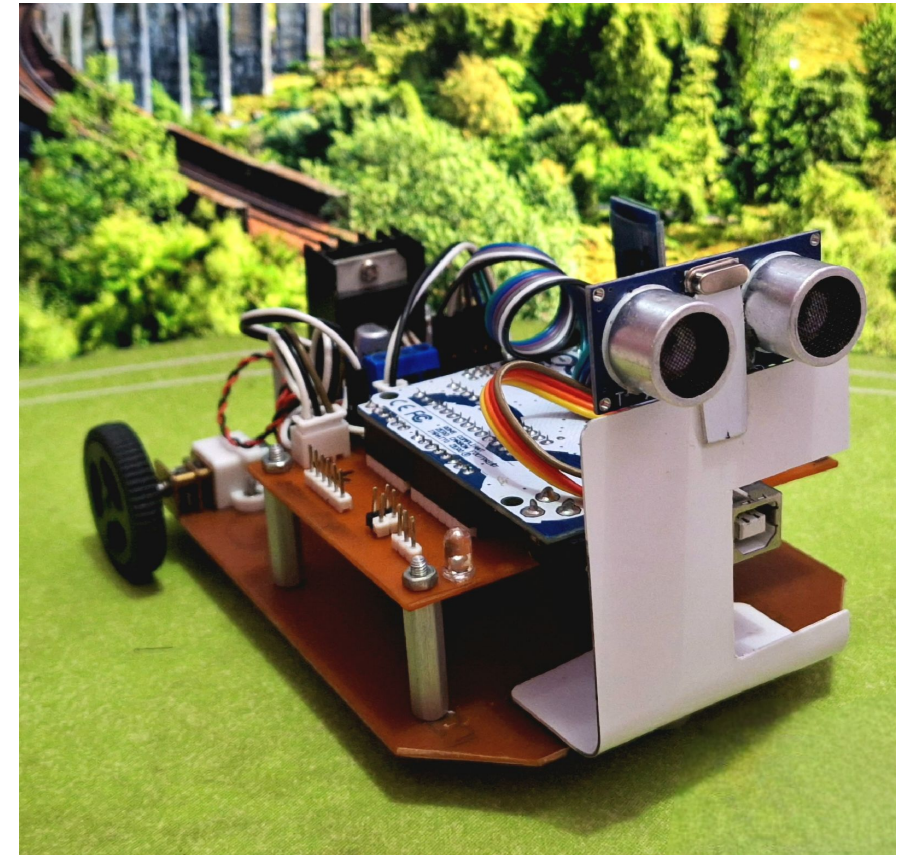
video : <https://youtu.be/VnAEoAgxl2k?si=BHeVZ-ZA9Sk76lYY>

#6 Project 2: Obstacle Avoider Robot

Definisi

Robot Penghindar Rintangan (Obstacle Avider robot) adalah jenis robot yang dirancang untuk dapat **menghindari** atau **mengelak** dari **rintangan** di sekitarnya secara otomatis.

Robot semacam ini biasanya dilengkapi dengan sensor-sensor seperti sensor **ultrasonik**, sensor inframerah, atau kamera untuk mendeteksi objek atau rintangan di sekitarnya.



MikonkuUltrasonicSensor Library

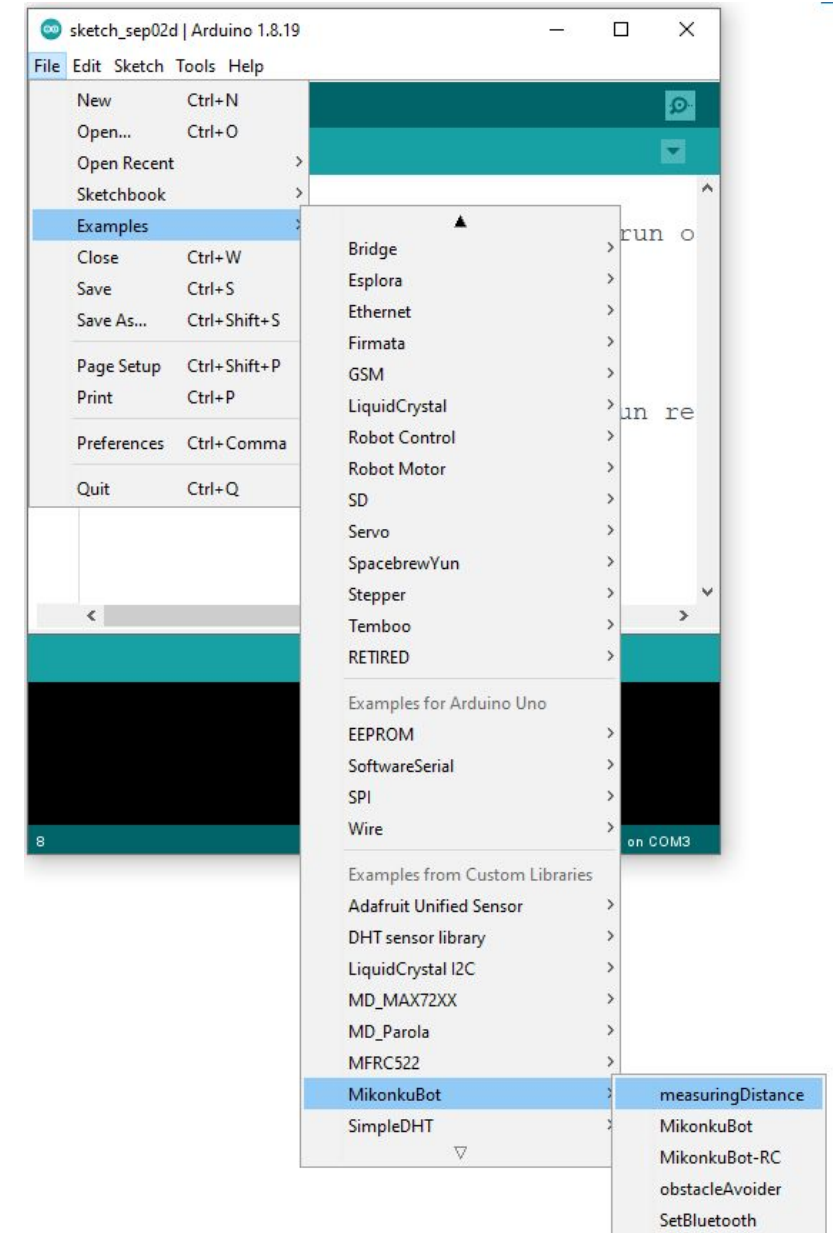
MikonkuUltrasonicSensor Library berisi perintah-perintah untuk mengakses sensor ultrasonic pada MikonkuBot.

Sudah *include* saat mendownload **MikonkuBot Library**

Examples

Berisi contoh kode untuk menjalankan sensor ultrasonic MikonkuBot.

Go to
File>Examples>MikonkuBot>measuringDistance



```
#include <MikonkuUltrasonicSensor.h>
```

```
// Pin Trigger ke D5, Pin Echo ke D4
```

```
MikonkuUltrasonicSensor ultrasonic(5, 4);
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  ultrasonic.begin();
```

```
}
```

```
void loop() {
```

```
  int distance = ultrasonic.getDistance();
```

```
  float distanceInCm = ultrasonic.toCm(distance);
```

```
  float distanceInMm = ultrasonic.toMm(distanceInCm);
```

```
  Serial.print("Jarak: ");
```

```
  Serial.print(distanceInCm);
```

```
  Serial.print(" cm atau ");
```

```
  Serial.print(distanceInMm);
```

```
  Serial.println(" mm");
```

```
  delay(1000);
```

```
}
```

```
#include <MikonkuUltrasonicSensor.h>
```

mengimport library
MikonkuUltrasonicSensor agar dapat
digunakan

```
// Pin Trigger ke D5, Pin Echo ke D4
```

```
MikonkuUltrasonicSensor ultrasonic(5, 4);
```

membuat object dengan nama ultrasonic
diisi dengan pin-pin yang sebelumnya
telah ditentukan. **ultrasonic** hanyalah
nama object artinya kita dapat mengubah
dengan nama lainnya

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  ultrasonic.begin();
```

```
}
```

menyalakan sensor ultrasonic

```
void loop() {
```

```
  int distance = ultrasonic.getDistance();
```

mendapatkan pembacaan jarak by default
dalam centimeter

```
  float distanceInCm = ultrasonic.toCm(distance);
```

konversi pembacaan jarak ke dalam
centimeter

```
  float distanceInMm = ultrasonic.toMm(distanceInCm);
```

konversi pembacaan jarak ke dalam
milimeter

```
  Serial.print("Jarak: ");
```

```
  Serial.print(distanceInCm);
```

```
  Serial.print(" cm atau ");
```

```
  Serial.print(distanceInMm);
```

```
  Serial.println(" mm");
```

```
  delay(1000);
```

```
}
```

mencetak semua nilainya di Serial
Monitor

Barebone Code For Obstacle Avider Robot

```
#include <MikonkuBot.h>
```

```
#include <MikonkuUltrasonicSensor.h>
```

```
const int ENA_PIN = 11;
```

```
const int IN1_PIN = 10;
```

```
const int IN2_PIN = 9;
```

```
const int IN3_PIN = 8;
```

```
const int IN4_PIN = 7;
```

```
const int ENB_PIN = 6;
```

```
// Inisialisasi objek robot dan sensor ultrasonik
```

```
MikonkuBot myBot(ENA_PIN, IN1_PIN, IN2_PIN, IN3_PIN, IN4_PIN, ENB_PIN);
```

```
MikonkuUltrasonicSensor ultrasonic(5, 4); // Pin Trigger ke D5, Pin Echo ke D4
```

```
void setup() {
```

```
    myBot.begin(); // Memulai kontrol motor
```

```
    myBot.setSpeed(100, 100); // Atur kecepatan motor kiri dan kanan (misalnya, 100)
```

```
    Serial.begin(9600);
```

```
    ultrasonic.begin();
```

```
}
```

```
void loop() {
```

```
    // Put your code here
```

```
}
```

Semua logic akan di taruh
disini

Simple Algorithm For Obstacle Avoider Robot

1. Tentukan ***obstacle threshold*** untuk jarak minimum dari sebuah penghalang
2. Jika sensor mendeteksi penghalang yang berjarak kurang dari ***obstacle threshold*** yang ditentukan, maka robot akan berhenti.
3. Setelah berhenti robot berputar arah (belok kiri atau belok kanan)

```
#include <MikonkuBot.h>
#include <MikonkuUltrasonicSensor.h>
```

```
const int ENA_PIN = 11;
const int IN1_PIN = 10;
const int IN2_PIN = 9;
const int IN3_PIN = 8;
const int IN4_PIN = 7;
const int ENB_PIN = 6;
```

```
// Inisialisasi objek robot dan sensor ultrasonik
```

```
MikonkuBot myRobot(ENA_PIN, IN1_PIN, IN2_PIN, IN3_PIN, IN4_PIN, ENB_PIN);
```

```
MikonkuUltrasonicSensor ultrasonicSensor(5, 4); // Pin Trigger ke D5, Pin Echo ke D4
```

```
const int obstacleThreshold = 10; // Jarak batasan untuk menghindari rintangan (misalnya, 10 cm)
```

```
void setup() {
```

```
    myRobot.begin(); // Memulai kontrol motor
```

```
    myRobot.setSpeed(100, 100); // Atur kecepatan motor kiri dan kanan (misalnya, 100)
```

```
    Serial.begin(9600);
```

```
    ultrasonicSensor.begin();
```

```
}
```

```
void loop() {
```

```
    // Baca jarak dari sensor ultrasonik
```

```
    int distance = ultrasonicSensor.getDistance();
```

```
    // Cek apakah jarak kurang dari obstacleThreshold
```

```
    if (distance < obstacleThreshold) {
```

```
        // Jika ya, berhenti robot
```

```
        myRobot.stop(0); // Memberhentikan robot tanpa delay
```

```
        // Kemudian, berputar ke kiri (Anda dapat mengganti arah belok sesuai kebutuhan)
```

```
        myRobot.goTurnLeft(1000); // Putar ke kiri selama 1 detik
```

```
    } else {
```

```
        // Jika tidak, lanjutkan maju
```

```
        myRobot.goForward(1000); // Maju selama 1 detik (Anda dapat menyesuaikan durasi)
```

```
    }
```

```
}
```

Materi Selanjutnya

Line Follower Robot