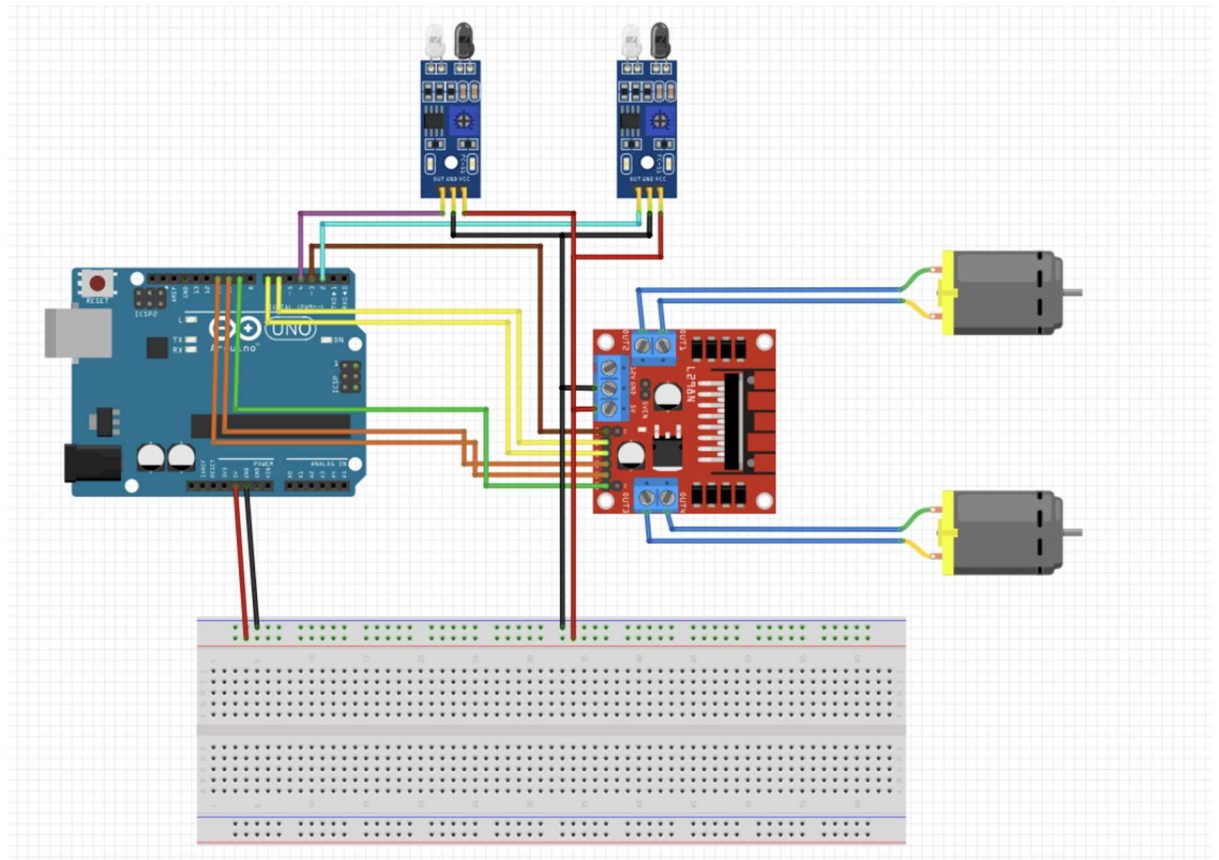


Nama : MUHAMMAD NOVIAN
NIM : 20210120004

TUGAS ROBOTIKA LINE FOLLOWER PID

1. Gambar rangkaian



2. Algoritma / Skenario Robot

Inisialisasi: Setel nilai awal untuk parameter PID (K_p , K_i , K_d),

Kalibrasi sensor untuk mendeteksi garis hitam.

Baca Sensor: Baca data dari sensor inframerah untuk mendeteksi posisi garis,

Konversi nilai sensor ke dalam format yang dapat digunakan oleh kontrol PID.

Hitung Posisi Error: Hitung posisi error berdasarkan perbedaan antara posisi aktual robot dan posisi tengah garis.

Hitung PID Output: Hitung output PID menggunakan rumus:

$$\text{PID_Output} = (K_p * P) + (K_i * I) + (K_d * D)$$

di mana P, I, dan D adalah nilai proporsional, integral, dan derivatif, masing-masing.

Kontrol Motor: Aplikasikan output PID untuk mengatur kecepatan motor kiri dan kanan, Pastikan bahwa perubahan kecepatan sesuai dengan arah dan tingkat kesalahan.

Looping: Ulangi langkah 2 hingga 5 dalam loop untuk mengatur pergerakan robot secara terus-menerus.

Deteksi Silang: Jika robot mendeteksi persilangan atau bercabang, terapkan logika khusus untuk memutuskan arah mana yang harus diambil.

Berhenti atau Melanjutkan: Tentukan kondisi berhenti atau melanjutkan berdasarkan tujuan robot atau kondisi lingkungan.

Selesai:

bahwa nilai K_p , K_i , dan K_d serta parameter lainnya harus disesuaikan dan dikalibrasi sesuai dengan kondisi lingkungan.

3. Program

```
#include <PID_v1.h>

// Pin definisi
#define R_S 4 // Sensor inframerah kanan
#define L_S 2 // Sensor inframerah kiri
#define enA 3 // Enable1 L293 Pin enA
#define in1 6 // Motor1 L293 Pin in1
#define in2 7 // Motor1 L293 Pin in1
#define in3 10 // Motor2 L293 Pin in1
#define in4 11 // Motor2 L293 Pin in1
#define enB 9 // Enable2 L293 Pin enB

// Variabel PID
double position = 0; // Posisi robot terhadap garis lintasan
double setpoint = 0.5; // Nilai tengah untuk posisi garis lintasan
double output = 0; // Output PID

// Konstanta PID
double Kp = 2; // Konstanta Proporsional
double Ki = 0.1; // Konstanta Integral
double Kd = 1; // Konstanta Derivatif

// Inisialisasi PID
PID pid(&position, &output, &setpoint, Kp, Ki, Kd, DIRECT);

// Deklarasi fungsi
double read_position();
void control_motors(double output);

void setup() {
    Serial.begin(9600);

    // Pin mode untuk sensor dan motor
    pinMode(R_S, INPUT);
    pinMode(L_S, INPUT);
    pinMode(enA, OUTPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    pinMode(enB, OUTPUT);

    // Aktifkan motor
    digitalWrite(enA, HIGH);
```

```

digitalWrite(enB, HIGH);

// Mulai PID
pid.SetMode(AUTOMATIC);
}

void loop() {
    // Baca posisi dari sensor
    position = read_position();

    // Hitung output PID
    pid.Compute();

    // Kendalikan motor berdasarkan output PID
    control_motors(output);

    // Waktu singkat sebelum membaca sensor berikutnya
    delay(10);
}

double read_position() {
    // Baca nilai sensor inframerah dan hitung posisi relatif terhadap garis
    int sensorRight = analogRead(R_S);
    int sensorLeft = analogRead(L_S);

    double position = (double)(sensorRight - sensorLeft) / (sensorRight + sensorLeft);
    return position;
}

void control_motors(double output) {
    // Kendalikan motor berdasarkan output PID
    int speed = 50; // Kecepatan motor (dapat disesuaikan)

    // Hitung kecepatan roda kiri dan kanan berdasarkan output PID
    int leftSpeed = constrain(speed - output, 0, 255);
    int rightSpeed = constrain(speed + output, 0, 255);

    // Atur kecepatan motor
    analogWrite(enA, leftSpeed);
    analogWrite(enB, rightSpeed);

    // Logika kontrol motor (sesuaikan dengan konfigurasi motor dan robot Anda)
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
}

```

