LSA08

<u>**Connection:**</u>
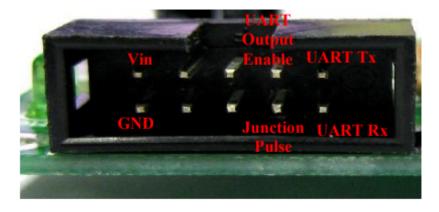
1. Analog Mode (We Used)

PORT A

2. Digital Mode

PORT B

3. UART Mode

PORT A

Code for Analog Mode (PID integrated)

1. mdd10a

```cpp
#include "CytronMotorDriver.h"

const byte analogPin = 0;    // Connect AN output of LSA08 to analog pin
0
const byte junctionPulse = 4;    // Connect JPULSE of LSA08 to pin 4

// Configure the motor driver.
CytronMD motor_Left(PWM_DIR, 10, 11);   // PWM 1 = Pin 10, DIR 1 = Pin
11.
CytronMD motor_Right(PWM_DIR, 12, 13);   // PWM 2 = Pin 12, DIR 2 = Pin
13.

int readVal,positionVal;     // Variables to store analog and line
position value
unsigned int junctionCount = 0;    // Variable to store junction count
value

//setting PID and Speed Constant
int MAX_SPEED= 255;
//PID Constants to be set by trial and error while testing
float Kp = 0.0;
float Kd = 0.0;
float Ki = 0.0;

long prevT = 0;
float prev_err = 0;
float eintegral = 0;

void setup()
{

  pinMode(junctionPulse,INPUT);
  Serial.begin(115200);

}

void loop()
{
```

```cpp
  // Checking for junction crossing; if juction detected, then keep
moving forward
  if(digitalRead(junctionPulse))
  {
    while(digitalRead(junctionPulse))
    {
      motor_Left.setSpeed(MAX_SPEED);
      motor_Right.setSpeed(MAX_SPEED);
    }

    // Increment junction count by 1 after the junction
    junctionCount++;
  }

  // Read value from analog pin
  readVal = analogRead(analogPin);

  // Convert voltage level into line position value
  positionVal = ((float)readVal/921)*70;

  long currT = micros(); //get the current time
  float deltaT = ((float) (currT - prevT))/( 1.0e6 );
  prevT = currT;

  int set_pos= 36;
  int err = positionVal - set_pos; // propotional
  float dert = (err-prev_err)/(deltaT); //derivative
  float eintegral = eintegral + err*deltaT; // integral
  prev_err = err;

  //PID-Controller applied on Model
  int w = (int) (Kp*err + Kd*dert + Ki*eintegral);

  //Converting to Differential Drive
  int w_l = MAX_SPEED - w;
  int w_r = MAX_SPEED + w;

  //Caping the speeds
  //left
  if (w_l > 255)
  {
    w_l = 255;
  }
```

```cpp
    if (w_l < -255)
    {
      w_l = -255;
    }
    //right
    if (w_l > 255)
    {
      w_l = 255;
    }
    if (w_r < -255)
    {
      w_r = -255;
    }

    //Print the direction with the speed
    if (w_l > w_r)
    {
      Serial.println("Moving Right");
    }
    else if (w_r > w_l)
    {
      Serial.println("Moving Left");
    }
    else if ((w_r == w_l) && (w_r != 0))
    {
      Serial.println("Moving Forward");
    }
    else
    {
      Serial.println("Waiting");
    }
    Serial.print("Left Speed: ");
    Serial.println(w_l);
    Serial.print("Right Speed:");
    Serial.println(w_r);

    //Printing the junction count
    Serial.print("Junction Count: ");
    Serial.println(junctionCount);

    //Running the motors
    motor_Left.setSpeed(w_l);
    motor_Right.setSpeed(w_r);
```

```
}
```

Reference:

1. https://github.com/CytronTechnologies/Arduino-LSA08/tree/master
2. Cryton Motor Driver Library