Exp. 06 – Line Follower Robot with IR Sensor Array - Continued

Version 1 – Using PWM pins for enabling the Motor Driver L293D

Equipment List:

1. 1x Arduino Nano
2. 1x USB 2.0 to USB type A connecting cable
3. 1x Breadboard
4. 2x/4x small DC Motor for tires
5. 1x L293D motor driver
6. 1x 3/4 wheel motor chassis
7. 1x 5 IR sensor array

Schematic Diagram:

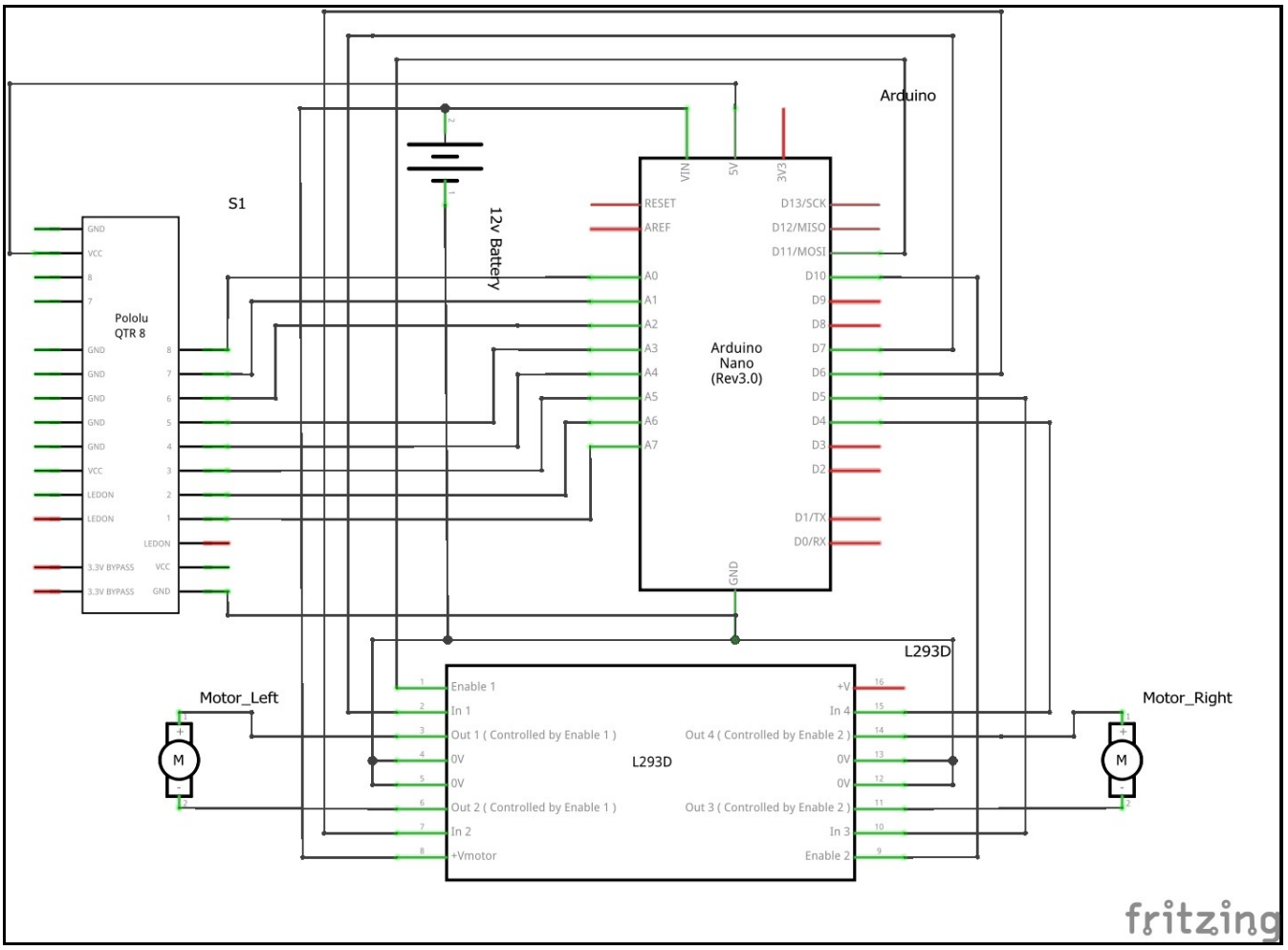

Figure 1 - Exp. 06 - Version 1 – Using PWM pins for enabling the Motor Driver L293D

Experiment Setup: Connect the PINS as per the following –

Connection of L293D motor driver

- PIN 9 with 5V output of Arduino
- PIN 1 with PWM PIN 11 and PIN 2 with PWM 10 of Arduino
- PIN 4,5 and 12,13 with GROUND PIN of battery

- PIN 3,6 with LEFT_MOTOR
- PIN 11,14 with RIGHT_MOTOR
- PIN 2,7 with Arduino Digital PIN 7,6
- PIN 10,15 with Arduino Digital PIN 5,4
- PIN 8 to 12V of battery.

Connection of Arduino

- PIN 2,7 of L293D with Arduino Digital PIN 7,6
- PIN 10,15 of L293D with Arduino Digital PIN 5,4
- PIN 1, 9 and 16 of L293D with 5V output of Arduino.
- VIN of Arduino to 12V Battery and GROUND of Arduino to GROUND of Battery.
- Connect Data OUT PINS of Sensor to ANALOG PINS of Arduino
- Upload the code and check the Output

Expected Output:

- The car should follow any black line on its way in a moderate speed.

Code:

```
#define NUM_SENSORS   5    // number of sensors used
#define enableA 11
#define enableB 10
#define direction1motorA 2
#define direction2motorA 3
#define direction1motorB 7
#define direction2motorB 8
unsigned int sensorValues[NUM_SENSORS];
int SumLeft = 0;
int SumRight = 0;
int SumDifference = 0;
int Last = 0;
void setup()
{
   delay(500);
   pinMode(enableA, OUTPUT);
   pinMode(enableB, OUTPUT);
   pinMode(direction1motorA, OUTPUT);
   pinMode(direction2motorA, OUTPUT);
   pinMode(direction1motorB, OUTPUT);
   pinMode(direction2motorB, OUTPUT);
}
void loop()
{
  sensorValues[0] = analogRead(0);
  sensorValues[1] = analogRead(1);
  sensorValues[2] = analogRead(2);
  sensorValues[3] = analogRead(3);
  sensorValues[4] = analogRead(4);
  for (unsigned char i = 0; i < NUM_SENSORS; i++)
  {
```

```
    Serial.print(sensorValues[i]);
    Serial.print('\t');
  }
  SumLeft =(sensorValues[0] + sensorValues[1] + sensorValues[2]);
  SumRight = (sensorValues[2] + sensorValues[3] + sensorValues[4]);
  SumDifference = (SumLeft - SumRight);
  if(abs(SumDifference) < 700){
    Forward();
    delay(1);
  }
  if((SumDifference > 700) && (SumDifference < 1000)){
    SlightRight();
    Last = 1;
    delay(1);
  }
  if(SumDifference >= 1000){
    HardRight();
    Last = 1;
    delay(1);
  }
  if((SumDifference < -700) && (SumDifference > -1000)){
    SlightLeft();
    Last = 2;
    delay(1);
  }
  if((SumDifference) <= -1000){
    HardLeft();
    Last = 2;
    delay(1);
  }
  if ((SumLeft < 100) && (SumRight < 100)){
    if (Last == 1){
      HardRight();
      delay(1);
    }
    if (Last == 2){
      HardLeft();
      delay(1);
    }
  }
}
void Forward(){
    digitalWrite(direction1motorA, HIGH);
    digitalWrite(direction2motorA, LOW);
    digitalWrite(direction1motorB, HIGH);
    digitalWrite(direction2motorB, LOW);
    analogWrite(enableA,175);
    analogWrite(enableB,175);
}
void SlightRight(){
    digitalWrite(direction1motorA, HIGH);
    digitalWrite(direction2motorA, LOW);
```

```
    digitalWrite(direction1motorB, LOW);
    digitalWrite(direction2motorB, HIGH);
    analogWrite(enableA,125);
    analogWrite(enableB,175);
}

 void HardRight(){
    digitalWrite(direction1motorA, HIGH);
    digitalWrite(direction2motorA, LOW);
    digitalWrite(direction1motorB, LOW);
    digitalWrite(direction2motorB, HIGH);
    analogWrite(enableA,0);
    analogWrite(enableB,175);
 }

void SlightLeft(){
    digitalWrite(direction1motorA, LOW);
    digitalWrite(direction2motorA, HIGH);
    digitalWrite(direction1motorB, HIGH);
    digitalWrite(direction2motorB, LOW);
    analogWrite(enableA,175);
    analogWrite(enableB,125);
}
void HardLeft(){
    digitalWrite(direction1motorA, LOW);
    digitalWrite(direction2motorA, HIGH);
    digitalWrite(direction1motorB, HIGH);
    digitalWrite(direction2motorB, LOW);
    analogWrite(enableA,175);
    analogWrite(enableB,0);
}
void Stop(){
    digitalWrite(direction1motorA, LOW);
    digitalWrite(direction2motorA, LOW);
    digitalWrite(direction1motorB, LOW);
    digitalWrite(direction2motorB, LOW);
    analogWrite(enableA,0);
    analogWrite(enableB,0);
}
```

Code Listing 1

Version 2 – Using PWM pins for controlling the motors

Equipment List:

1. 1x Arduino Nano
2. 1x USB 2.0 to USB type A connecting cable
3. 1x Breadboard
4. 2x/4x small DC Motor for tires
5. 1x L293D motor driver
6. 1x 3/4 wheel motor chassis
7. 1x 5 IR sensor array

Schematic Diagram:

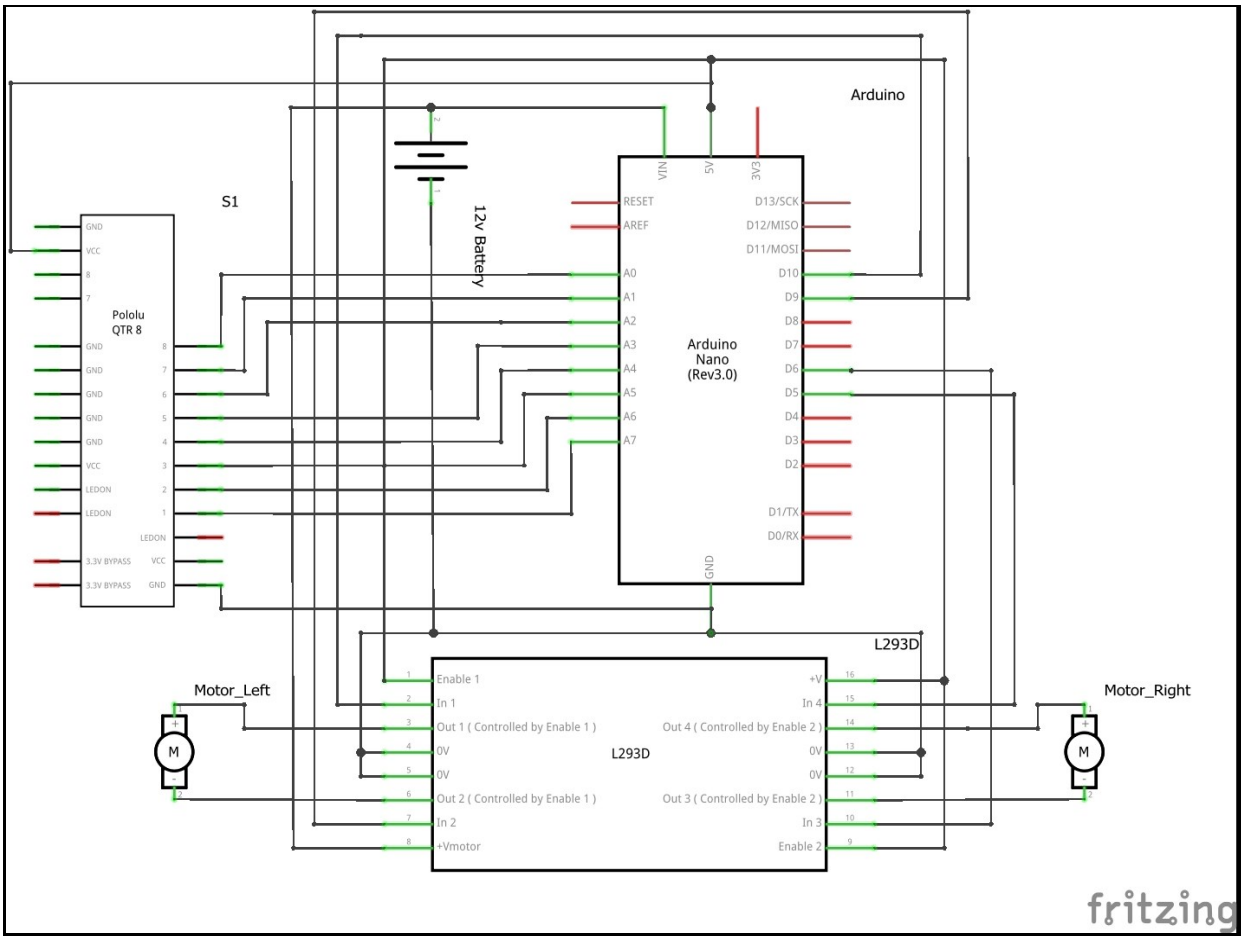

Figure 2 - Exp. 06 – Version 2 – Using PWM pins for controlling the motors

Experiment Setup: Connect the PINS as per the following –

Connection of L293D motor driver

- PIN 1,9 and 16 with 5V output of Arduino
- PIN 4,5 and 12,13 with GROUND PIN of battery
- PIN 3,6 with LEFT_MOTOR
- PIN 11,14 with RIGHT_MOTOR
- PIN 2,7 with Arduino PWM Digital PIN 10,9

- PIN 10,15 with Arduino PWM Digital PIN 6,5
- PIN 8 to 12V of battery.

Connection of Arduino

- PIN 2,7 of L293D with Arduino PWM Digital PIN 10,9
- PIN 10,15 of L293D with Arduino PWM Digital PIN 6,5
- PIN 1, 9 and 16 of L293D with 5V output of Arduino.
- VIN of Arduino to 12V Battery and GROUND of Arduino to GROUND of Battery.
- Connect Data OUT PINS of Sensor to ANALOG PINS of Arduino
- Upload the code and check the Output

Expected Output:

- The car should follow any black line on its way in a moderate speed.

Code:

```
#define NUM_SENSORS   5    // number of sensors used
#define direction1motorA 5
#define direction2motorA 6
#define direction1motorB 9
#define direction2motorB 10
int sensorValues[NUM_SENSORS];
void setup()
{
   Serial.begin(9600);
   pinMode(enableA, OUTPUT);
   pinMode(enableB, OUTPUT);
   pinMode(direction1motorA, OUTPUT);
   pinMode(direction2motorA, OUTPUT);
   pinMode(direction1motorB, OUTPUT);
   pinMode(direction2motorB, OUTPUT);
}
void loop()
{
  sensorValues[0]=analogRead(0);
  sensorValues[1]=analogRead(1);
  sensorValues[2]=analogRead(2);
  sensorValues[3]=analogRead(3);
  sensorValues[4]=analogRead(4);
  for(int i=0;i<5;i++){
    Serial.print(sensorValues[i]);
    Serial.print("\t");
  }
  Serial.println();
  if(sensorValues[2] > 920){
    Forward();
    delay(1);
  }
  if(sensorValues[3]>920){
    SlightRight();
```

```
    delay(1);
  }
  if(sensorValues[4]>920){
    HardRight();
    delay(1);
  }
  if(sensorValues[1]>880){
    SlightLeft();
    delay(1);
  }
  if(sensorValues[0]>880){
    HardLeft();
    delay(1);
  }
}
void Forward(){
    analogWrite(direction1motorA, 120);
    analogWrite(direction2motorA, 0);
    analogWrite(direction1motorB, 120);
    analogWrite(direction2motorB, 0);
}
void SlightRight(){
    analogWrite(direction1motorA, 120);
    analogWrite(direction2motorA, 0);
    analogWrite(direction1motorB, 0);
    analogWrite(direction2motorB, 120);
}
void HardRight(){
    analogWrite(direction1motorA, 255);
    analogWrite(direction2motorA, 0);
    analogWrite(direction1motorB, 0);
    analogWrite(direction2motorB, 255);
}
void SlightLeft(){
    analogWrite(direction1motorA, 0);
    analogWrite(direction2motorA, 120);
    analogWrite(direction1motorB, 120);
    analogWrite(direction2motorB, 0);
}
void HardLeft(){
    analogWrite(direction1motorA, 0);
    analogWrite(direction2motorA, 255);
    analogWrite(direction1motorB, 255);
    analogWrite(direction2motorB, 0);
}
void Stop(){
    analogWrite(direction1motorA, 0);
    analogWrite(direction2motorA, 0);
    analogWrite(direction1motorB, 0);
    analogWrite(direction2motorB, 0);
}
```

Code Listing 2