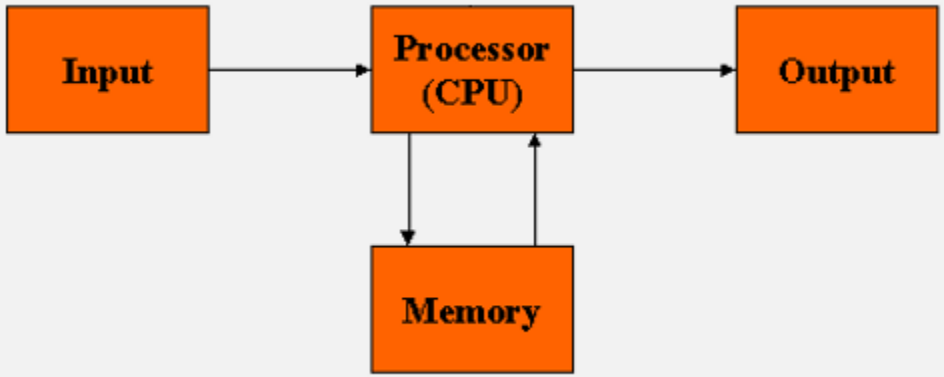
INTRODUCTION TO ARDUINO PROGRAMMING

EEE Association

MICROPROCESSOR

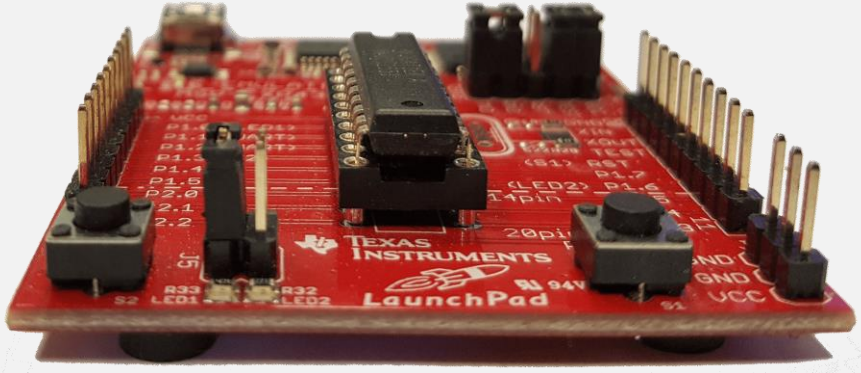
- Controlling unit of a micro-computer, fabricated on a small chip
- Capable of performing Arithmetic Logical Unit (ALU) operations and communicating with the other devices connected to it.



MICROCONTROLLER

A self-contained system with a microprocessor, memory and peripherals

Arduino Uno



Texas Instruments MSP430

DIFFERENCES BETWEEN A MICROPROCESSOR AND A MICROCONTROLLER

Microprocessor

- No memory is present(RAM, ROM, etc)
- Peripherals are not present.
- High processing speed
- Cannot be used independently without adding peripherals.

Microcontoller

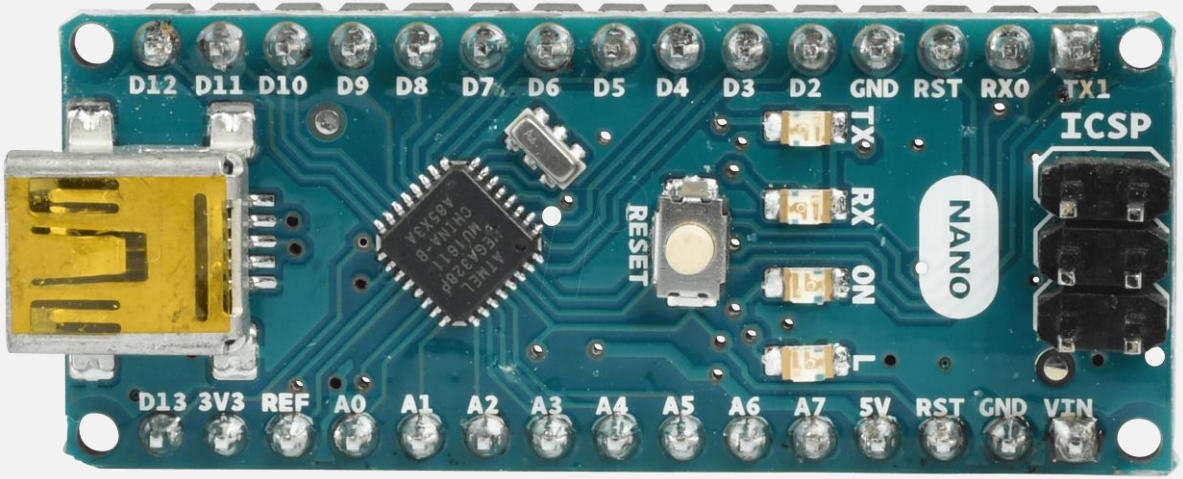
- Memory is Present
- Fixed peripherals are present
- Lower processing speed
- Can be used independently

ARDUINO

- Open-source electronic prototyping platform
- Have sets of digital and analog input/output (I/O) pins
- Serial communications interfaces including Universal Serial Bus (USB) which is also used to write programs from a computer
- NOTE: Arduino output pins have a limited current capacity

ARDUINO NANO

- Based on the ATmega328 microcontroller
- Works with a Mini-B USB cable
- 6 Analog input pins and 14 Digital pins



PROGRAMMING AN ARDUINO MICROCONTROLLER

- Programmed through the Arduino IDE
- Can be programmed in the Arduino language or Embedded-C
- Arduino language: Very similar to C/C++

Every Arduino program – two main parts:

I. setup

II. loop

- setup():
  - This block of code is executed when the program starts to execute
  - Contains initialising commands like setting up the state of the digital pins

- loop():
  - This block gets executed repeatedly
  - Contains the codes of processes which are to be run repeatedly

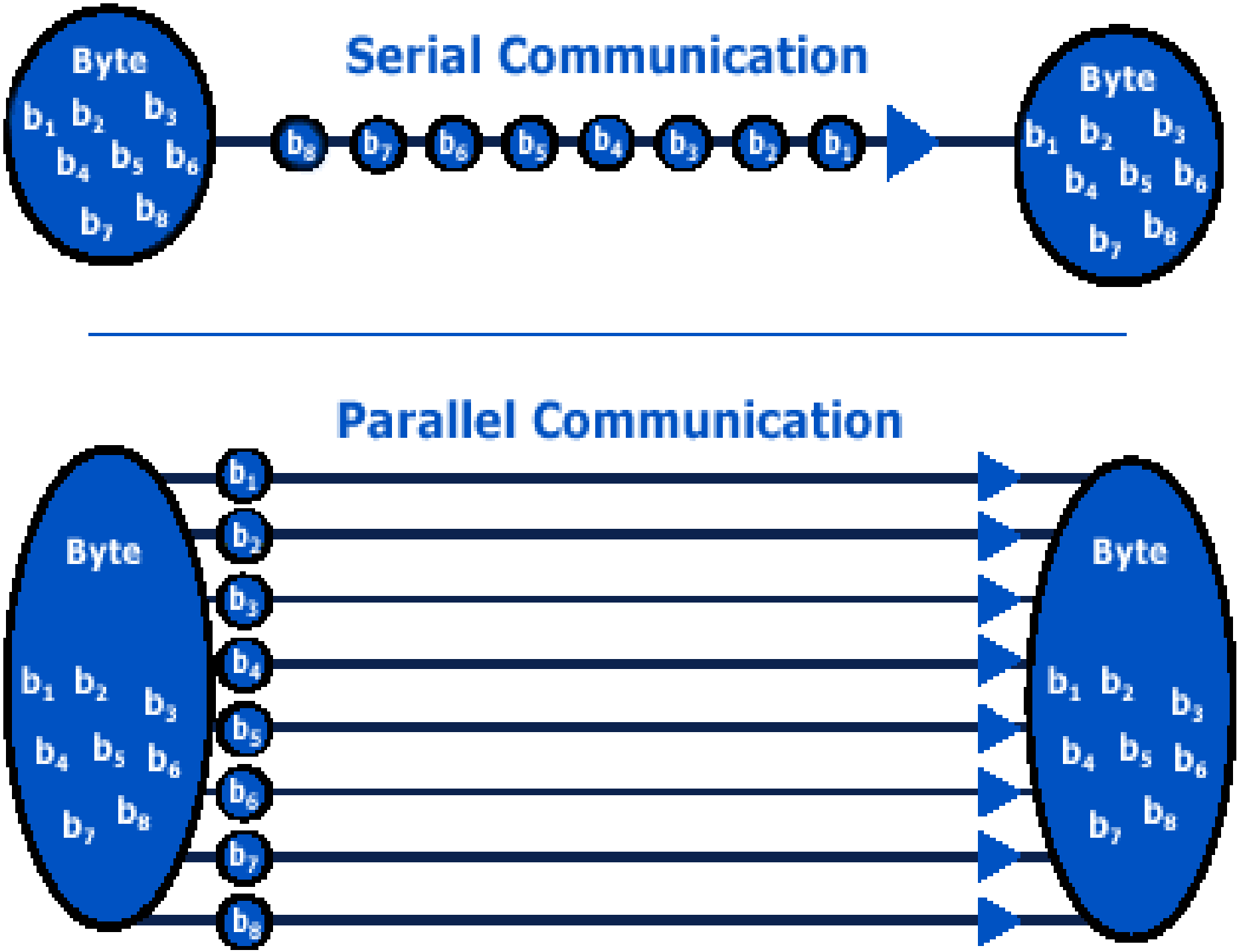
BLINKING AN LED

- On-board LED: Digital Pin 13
- Steps
  1. Initialise the pins
  2. Glow the LED
  3. Wait for a particular duration
  4. Turn off the LED
  5. Wait for a particular duration
  6. Repeat steps 2 – 5

```
void setup()            //Executed once
{
            //corresponding to On-Board LED as OUTPUT pin
            pinMode(13, OUTPUT); //Sets the pin
}

void loop()             //Repeatedly executed
{
            digitalWrite(13, HIGH);         //Turns ON the LED
            delay(1000);                    // Wait for 1000 milliseconds
            digitalWrite(13, LOW);          //Turns OFF the LED
            delay(1000);                    // Wait for 1000 milliseconds
}
```

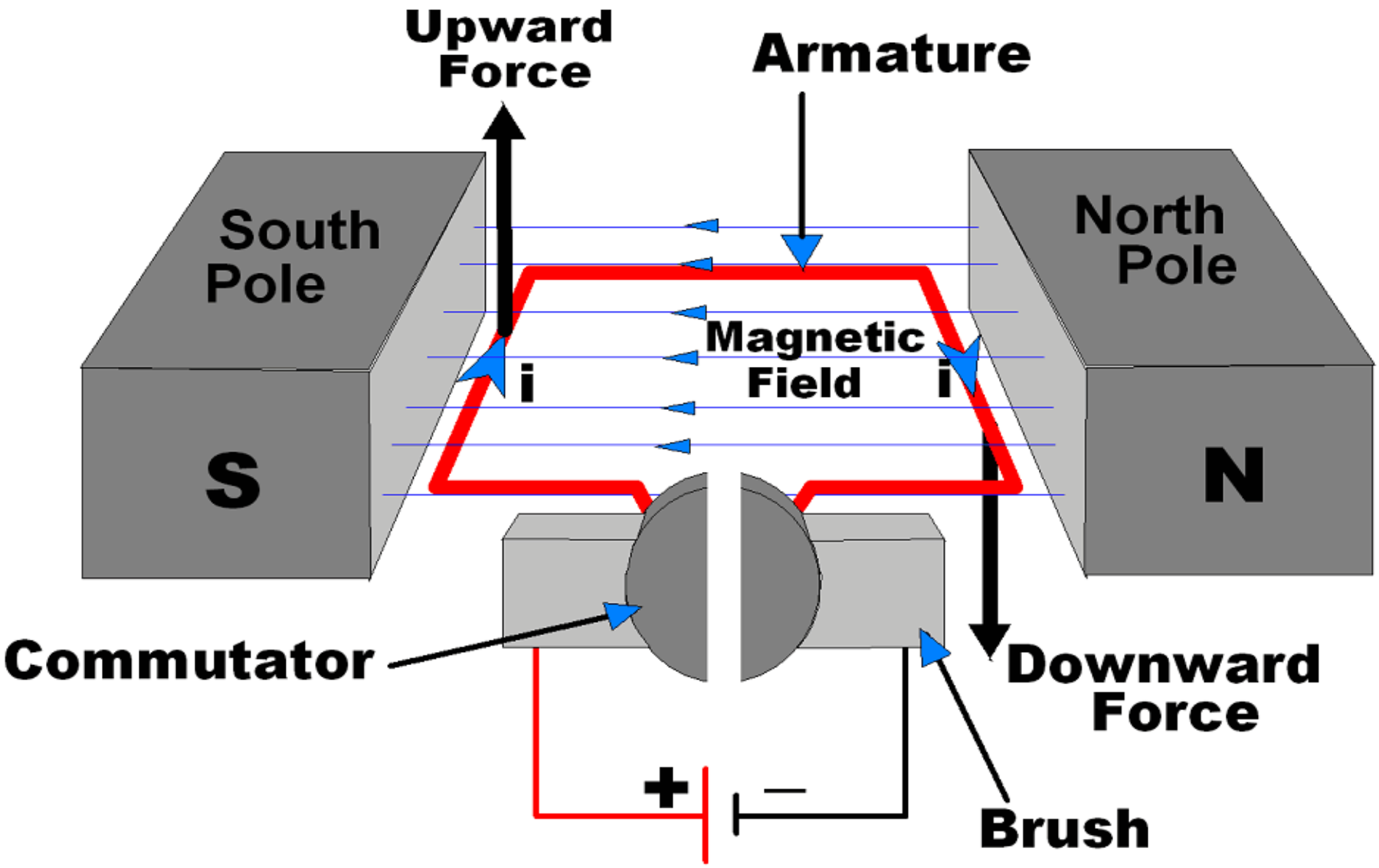
SERIAL COMMUNICATION

- Sending data one bit at a time, sequentially, over a communication channel
- Baud rate: The rate at which information is transferred in a communication channel.
- Arduino Serial library: Used for communication between the Arduino board and a computer or other devices
- Arduino IDE: Serial monitor can be used to send and view received data


Serial Communication
Byte
b1 b2 b3
b4 b5 b6
b7 b8
b8 b7 b6 b5 b4 b3 b2 b1
Byte
b1 b2 b3
b4 b5 b6
b7 b8
Parallel Communication
Byte
b1 b2 b3
b4 b5 b6
b7 b8
b1
b2
b3
b4
b5
b6
b7
b8
Byte
b1 b2 b3
b4 b5 b6
b7 b8

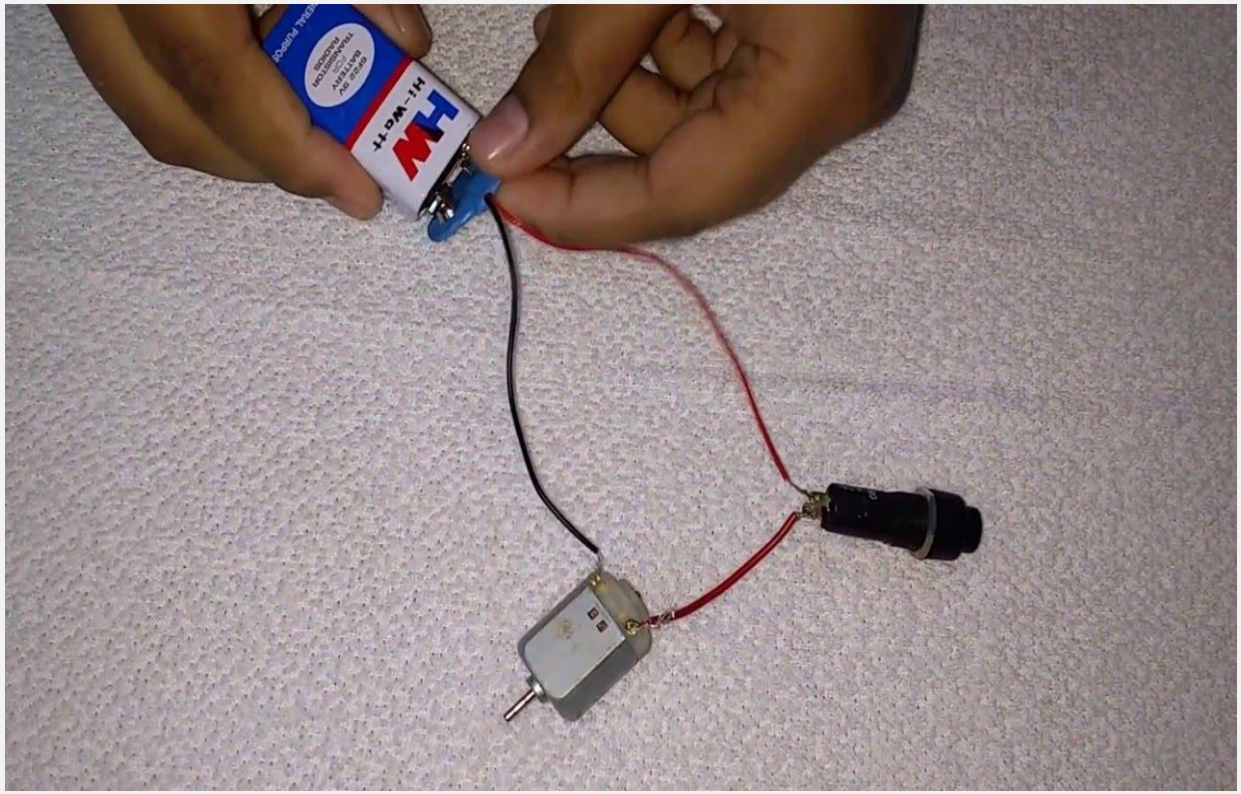
```
int i = 25;                  // intializing integer i = 25
void setup()
{
        Serial.begin(9600);      // opens the serial port at 9600 bps
        }

void loop()
{
        Serial.print("Hello");       // prints string "hello"
        Serial.print("\t");           // prints a tab
        Serial.println('a');          // prints char 'a' in a new line
        Serial.println(i);            // print as an ASCII-encoded decimal
        Serial.println(i, DEC);    // print as an ASCII-encoded decimal
        Serial.println(i, HEX);    // print as an ASCII-encoded hexadecimal
        Serial.println(i, OCT);   // print as an ASCII-encoded octal
        Serial.println(i, BIN);     // print as an ASCII-encoded binary
}
```

MOTOR



DC Motor Conceptual Diagram

- The basic idea of an electric motor is really simple: you put electricity into it at one end and an axle (metal rod) rotates at the other end giving you the power to drive a machine of some kind.



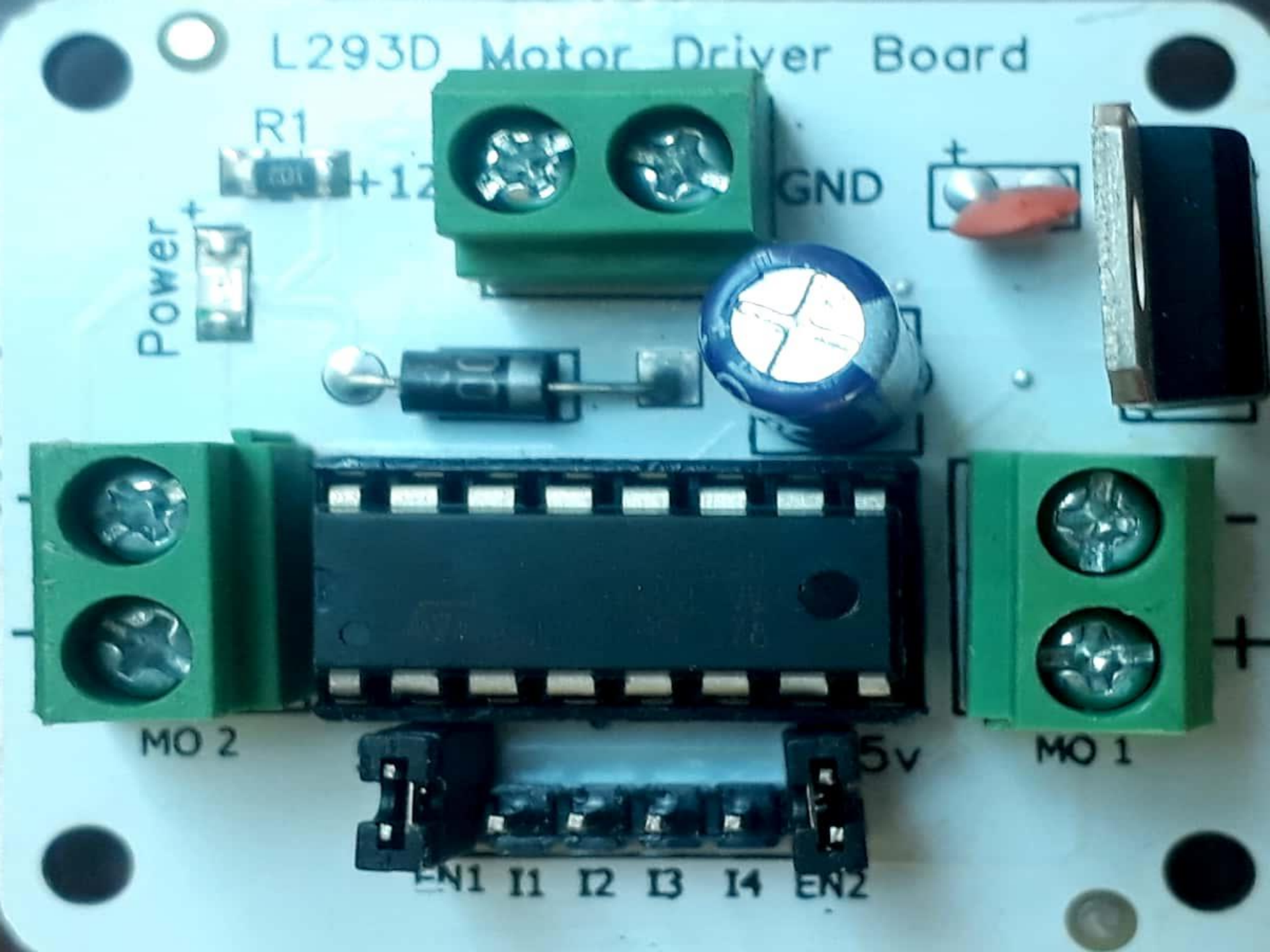
POWERING OF A DC MOTOR

POWERING A DC MOTOR

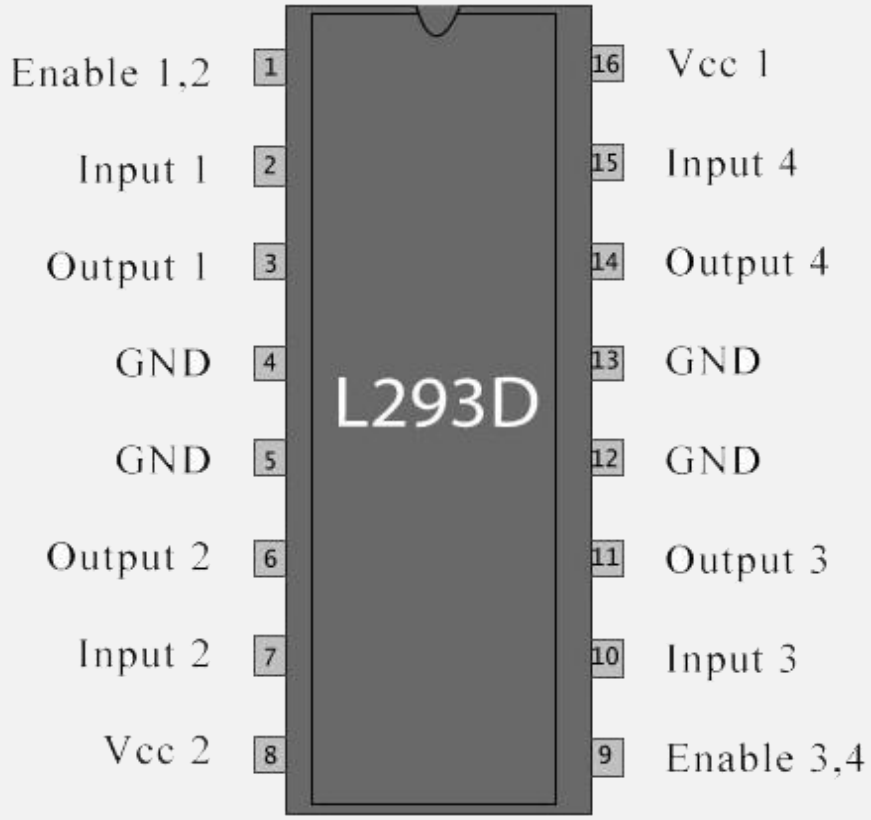
Current Rating of GPIO pins of the Arduino Nano is less than <50 mA Insufficient to match the DC Motor's power requirements.

MOTOR DRIVER

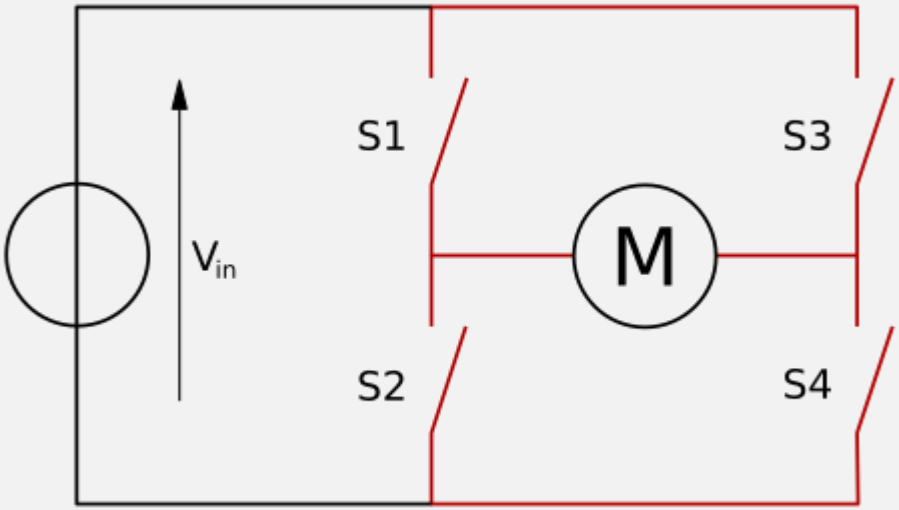
A motor driver is a little current amplifier, the function of motor drivers is to take a low-current control signal and then turn it into a higher-current signal that can drive a motor.


L293D Motor Driver Board
R1
+12
GND
Power
MO 2
MO 1
5v
EN1 I1 I2 I3 I4 EN2

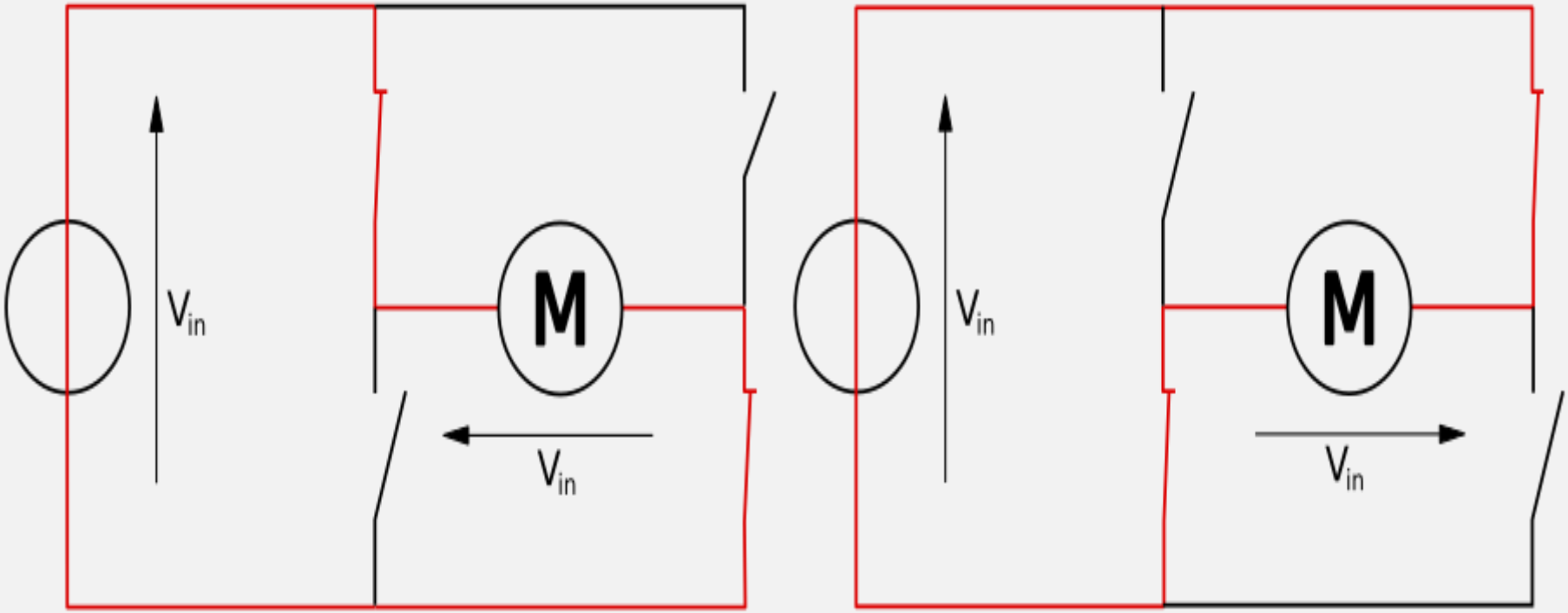
PIN CONFIGURATION OF L293D



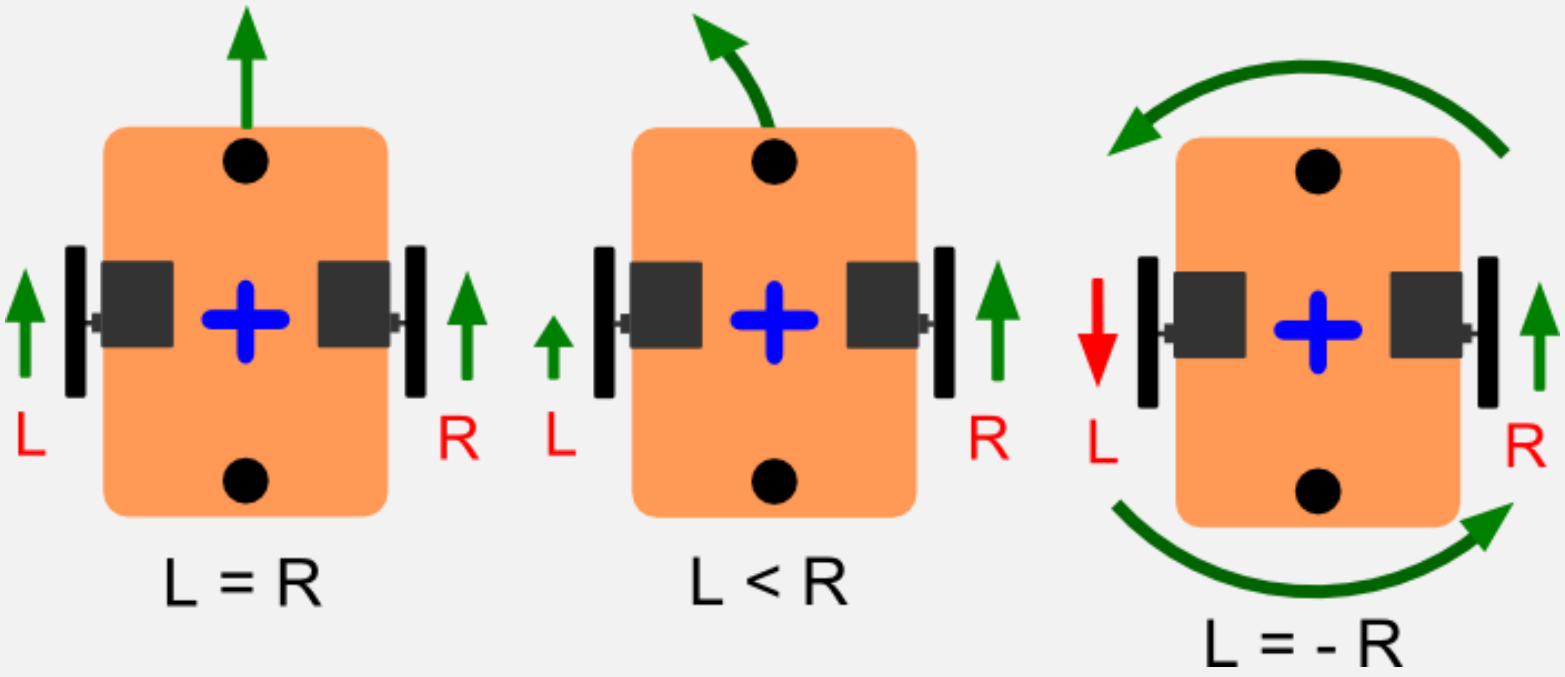
H - BRIDGE



A H-bridge is an electronic circuit that switches the polarity of a voltage applied to a load. These circuits are often used in robotics and other applications to allow DC motors to run forwards or backwards.


V_{in}
M
V_{in}
V_{in}
M
V_{in}

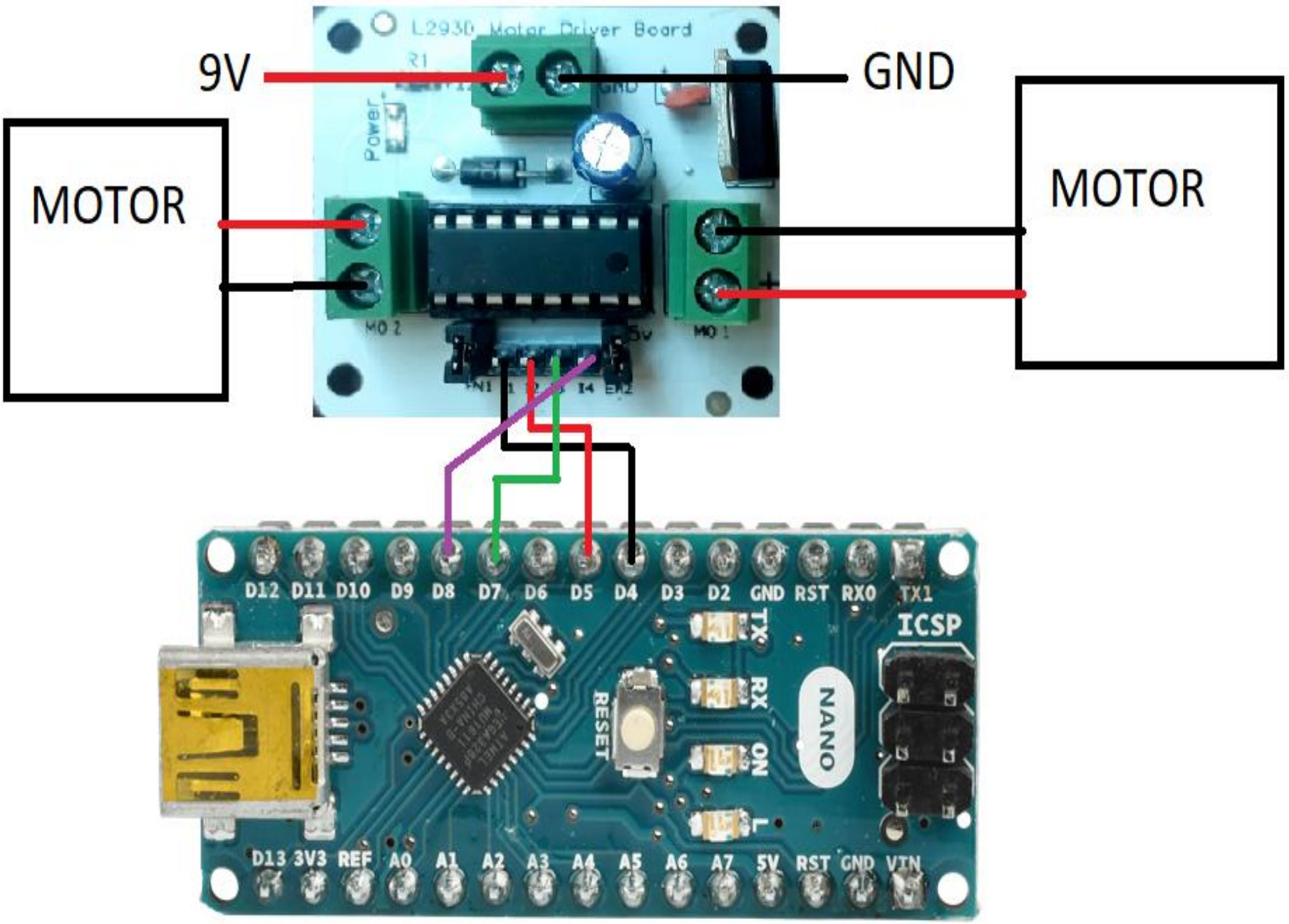
DIFFERENTIAL DRIVE



BOT NAVIGATION TEST

Procedure used in the code:

1. Start
2. Obtain the rotation of the motors corresponding to forward direction of motion of the bot
3. Send the corresponding control signals to the motor driver
4. Wait for a particular time
5. Repeat steps 2 – 4 for left, right and backward directions
6. Stop


9V
GND
MOTOR
MOTOR
L293D Motor Driver Board
Power
MO 2
MO 1
5v
D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
ICSP
TX
RX
ON
L
RESET
NANO
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN

```
// Upload this code in arduino IDE
#define in1 4                    //Connect the in1 to pin 4 of arduino nano
#define in2 5                    // Connect the in2 to pin 5 of arduino nano
#define in3 7                    // Connect the in3 to pin 7 of arduino nano
#define in4 8                    // Connect the in4 to pin 8 of arduino nano
int c;
void setup() {
  Serial.begin(9600);
  pinMode(in1,OUTPUT);
  pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT);
  pinMode(in4,OUTPUT);
}
```

```
void loop() {
  c=Serial.parseInt();                  //Reads user input (an integer) from the serial stream
  Serial.println(c);                    //Prints the read user input on the serial monitor
  switch(c)
 {
    case 1:
      digitalWrite(in1,HIGH);           //FORWARD
      digitalWrite(in2,LOW);
      digitalWrite(in3,HIGH);
      digitalWrite(in4,LOW);
      delay(5000);                                          //Continue for 5 seconds
    case 2:
      digitalWrite(in1,HIGH);           //LEFT
      digitalWrite(in2,LOW);
      digitalWrite(in3,LOW);
      digitalWrite(in4,LOW);
      delay(5000);                                          //Continue for 5 seconds
    case 3:
      digitalWrite(in1,LOW);            //RIGHT
      digitalWrite(in2,LOW);
      digitalWrite(in3,HIGH);
      digitalWrite(in4,LOW);
      delay(5000);                                          //Continue for 5 seconds
```

```
  case 4:
     digitalWrite(in1,LOW);  //BACK
     digitalWrite(in1,HIGH);
     digitalWrite(in3,LOW);
     digitalWrite(in3,HIGH);
     delay(5000);                                  //Continue for 5 seconds
    case 5:
     digitalWrite(in1,LOW);  //STOP
     digitalWrite(in2,LOW);
     digitalWrite(in3,LOW);
     digitalWrite(in4,LOW);
     delay(5000);                                  //Wait for 5 seconds
   }
}
```

CONTROL MOTION OF BOT THROUGH USER INPUT

Brief procedure

1. Read user choice from serial monitor
2. Obtain the direction of rotation of the motors
3. Send the corresponding control signals to the motor driver
4. Repeat steps 1 - 3

```
#define in1 4                                   //Connect the in1 to pin 4 of arduino nano
#define in2 5                                   // Connect the in2 to pin 5 of arduino nano
#define in3 7                                   // Connect the in3 to pin 7 of arduino nano
#define in4 8                                   // Connect the in4 to pin 8 of arduino nano

int c;

void setup() {
  Serial.begin(9600);          //Starts serial communication with a baud rate of 9600
  pinMode(in1,OUTPUT);
  pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT);
  pinMode(in4,OUTPUT);
}
```

```
void loop() {

  c=Serial.parseInt();            //Reads user input (an integer) from the serial stream
  Serial.println(c);              //Prints the read user input on the serial monitor

  switch(c)
  {
    case 1:
      digitalWrite(in1,HIGH);   //FORWARD
      digitalWrite(in2,LOW);
      digitalWrite(in3,HIGH);
      digitalWrite(in4,LOW);
      break;
    case 2:
      digitalWrite(in1,HIGH);    //LEFT
      digitalWrite(in2,LOW);
      digitalWrite(in3,LOW);
      digitalWrite(in4,LOW);
      break;
```

```
  case 3:
    digitalWrite(in1,LOW);    //RIGHT
    digitalWrite(in2,LOW);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    break;
   case 4:
    digitalWrite(in1,LOW);    //BACK
    digitalWrite(in1,HIGH);
    digitalWrite(in3,LOW);
    digitalWrite(in3,HIGH);
    break;
   case 5:
    digitalWrite(in1,LOW);    //STOP
    digitalWrite(in2,LOW);
    digitalWrite(in3,LOW);
    digitalWrite(in4,LOW);
    break;
 }
}
```

SERIAL COMMUNICATION THROUGH PYTHON

```
import serial
import time
ser = serial.Serial('com4', 9600)
while(1):
    a=raw_input ("Enter a")
    if a =="F":
        print "Forward"
        time.sleep(1)
        ser.write('1')
    elif a =="S":
        print "Stop"
        time.sleep(1)
        ser.write('5')

    else:
        print "Stop"
        time.sleep(1)
        ser.write('5')
```