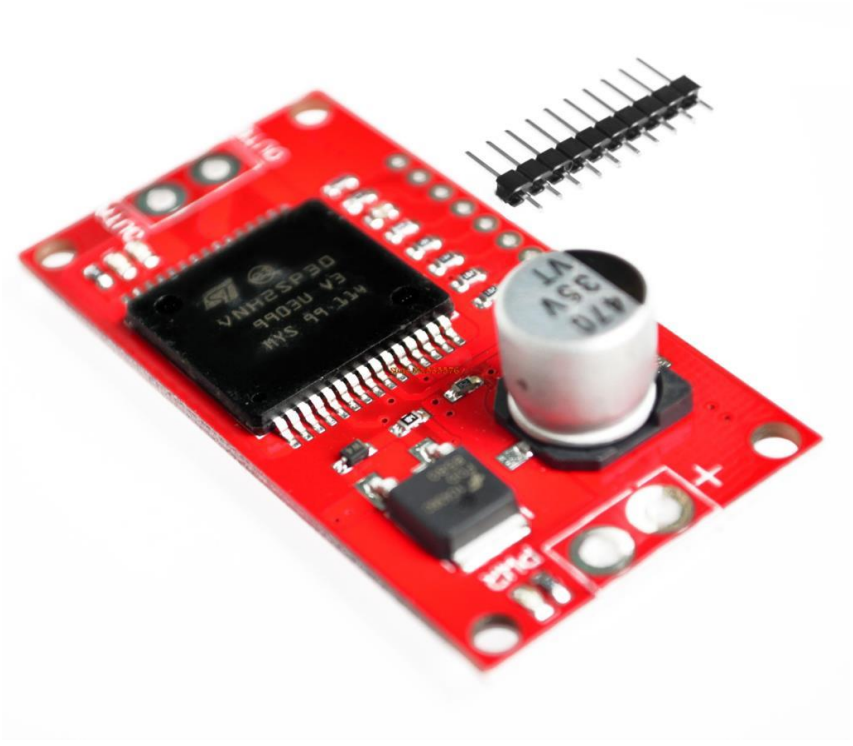
SYNACORP TECHNOLOGIES SDN. BHD. (1310487-K)
No.25 Lorong 1/SS3, Bandar Tasek Mutiara,
14120 Simpang Ampat, Penang, Malaysia.
T: +604.586.0026 F: +604.586.0026
www.synacorp.my | Email: sales@synacorp.com.my

Arduino VNH2SP30 Mini 30A Stepper Motor Driver Monster Moto Shield Module

Introduction:

VNH2SP30 is a full bridge motor driver intended for a wide range of automotive applications. The device incorporates a dual monolithic high side driver and two low side switches. The high side driver switch is designed using the STMicroelectronic's well known and proven proprietary VIPower M0 technology which permits efficient integration on the same die of a true Power MOSFET with an intelligent signal/protection circuitry. The VIN and motor out are pitched for 5mm screw terminals, making it easy to connect larger gauge wires. INA and INB control the direction of each motor, and the PWM pins turns the motors on or off. For the VNH2SP30, the current sense (CS) pins will output approximately 0.13 volts per amp of output current.



Components:

- Arduino Uno Board
- Monster Motor Shield VNH2SP30
- 3V-6V Miniature DC Brush Motor W/O Gear
- Li-Ion Rechargeable Battery 7.4V 1200mAh
- USB Cable
- Jump Wires / Wire with Crocodile End Clip

SYNACORP TECHNOLOGIES SDN. BHD. (1310487-K)
No.25 Lorong 1/SS3, Bandar Tasek Mutiara,
14120 Simpang Ampat, Penang, Malaysia.
T: +604.586.0026 F: +604.586.0026
www.synacorp.my | Email: sales@synacorp.com.my

Objective:

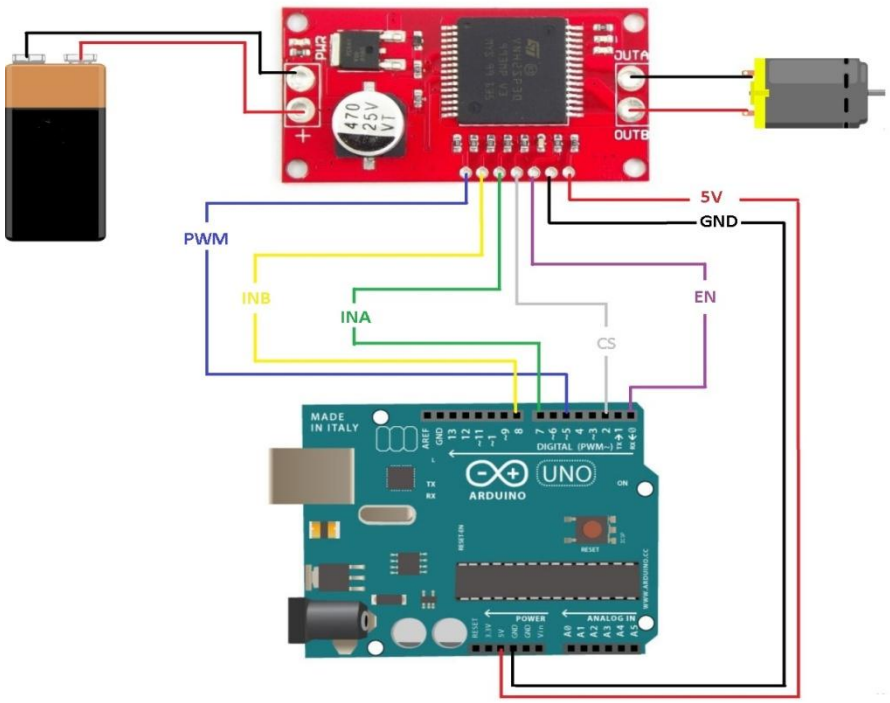
Allows to easily control motor direction and speed using an Arduino.

Procedures:

Step 1: Build a circuit.

The connection between the Arduino VNH2SP30 Mini 30A Stepper Motor Driver Monster Moto Shield module and the Arduino Uno Board:

Arduino VNH2SP30 Mini 30A Stepper Motor Driver Monster Moto Shield module	Arduino Uno Board
5v	5v
GND	GND
EN	RX
CS	PIN 2
IN A	PIN 7
IN B	PIN 8
PMW	PIN 5



Step 2: Insert the sample programming provided below by copy and paste it into Arduino IDE.

```
#define BRAKE 0
#define CW    1
#define CCW   2
#define CS_THRESHOLD 15   // Definition of safety current (Check: "1.3 Monster Shield Example").

//MOTOR 1
#define MOTOR_A1_PIN 7
#define MOTOR_B1_PIN 8

#define PWM_MOTOR_1 5

#define CURRENT_SEN_1 A2

#define EN_PIN_1 A0

#define MOTOR_1 0

short usSpeed = 150;  //default motor speed
unsigned short usMotor_Status = BRAKE;

void setup()
{
  pinMode(MOTOR_A1_PIN, OUTPUT);
  pinMode(MOTOR_B1_PIN, OUTPUT);

  pinMode(PWM_MOTOR_1, OUTPUT);

  pinMode(CURRENT_SEN_1, OUTPUT);

  pinMode(EN_PIN_1, OUTPUT);

  Serial.begin(9600);             // Initiates the serial to do the monitoring
  Serial.println("Begin motor control");
  Serial.println(); //Print function list for user selection
  Serial.println("Enter number for control option:");
  Serial.println("1. STOP");
  Serial.println("2. FORWARD");
  Serial.println("3. REVERSE");
  Serial.println("+. INCREASE SPEED");
```

```
  Serial.println("-. DECREASE SPEED");
  Serial.println();

}

void loop()
{
  char user_input;



  while(Serial.available())
  {
    user_input = Serial.read(); //Read user input and trigger appropriate function
    digitalWrite(EN_PIN_1, HIGH);


    if (user_input =='1')
    {
       Stop();
    }
    else if(user_input =='2')
    {
      Forward();
    }
    else if(user_input =='3')
    {
      Reverse();
    }
    else if(user_input =='+')
    {
      IncreaseSpeed();
    }
    else if(user_input =='-')
    {
      DecreaseSpeed();
    }
    else
    {
      Serial.println("Invalid option entered.");
    }
```

```
  }
}

void Stop()
{
  Serial.println("Stop");
  usMotor_Status = BRAKE;
  motorGo(MOTOR_1, usMotor_Status, 0);
}

void Forward()
{
  Serial.println("Forward");
  usMotor_Status = CW;
  motorGo(MOTOR_1, usMotor_Status, usSpeed);
}

void Reverse()
{
  Serial.println("Reverse");
  usMotor_Status = CCW;
  motorGo(MOTOR_1, usMotor_Status, usSpeed);
}

void IncreaseSpeed()
{
  usSpeed = usSpeed + 10;
  if(usSpeed > 255)
  {
    usSpeed = 255;
  }

  Serial.print("Speed +: ");
  Serial.println(usSpeed);

  motorGo(MOTOR_1, usMotor_Status, usSpeed);
}

void DecreaseSpeed()
{
  usSpeed = usSpeed - 10;
  if(usSpeed < 0)
```

```
  {
    usSpeed = 0;
  }

  Serial.print("Speed -: ");
  Serial.println(usSpeed);

  motorGo(MOTOR_1, usMotor_Status, usSpeed);
}

void motorGo(uint8_t motor, uint8_t direct, uint8_t pwm)         //Function that controls the variables: motor(0 ou 1), direction (cw ou ccw) e pwm (entra 0 e 255);
{
  if(motor == MOTOR_1)
  {
    if(direct == CW)
    {
      digitalWrite(MOTOR_A1_PIN, LOW);
      digitalWrite(MOTOR_B1_PIN, HIGH);
    }
    else if(direct == CCW)
    {
      digitalWrite(MOTOR_A1_PIN, HIGH);
      digitalWrite(MOTOR_B1_PIN, LOW);
    }
    else
    {
      digitalWrite(MOTOR_A1_PIN, LOW);
      digitalWrite(MOTOR_B1_PIN, LOW);
    }

    analogWrite(PWM_MOTOR_1, pwm);
  }
}
```

Step 3: Open the serial monitor to observe the result as shown below.