DC Motor Driver VNH2SP30

(30A Peak – 15A Continuous)

This is a high power dc motor driver based on VNH2SP30 chip from ST. It is designed for high power dc motor control applications with peak current up to 30A and continuous current of 14A.

The module is easy to use with arduino or any other microcontroller. The motor driver also have thermal shutdown capability for protection of overheating, it also have current sensing and overvoltage protection. It is very similar to sparkfun Monester moto driver, the only difference is that this module drive only one motor at a time.

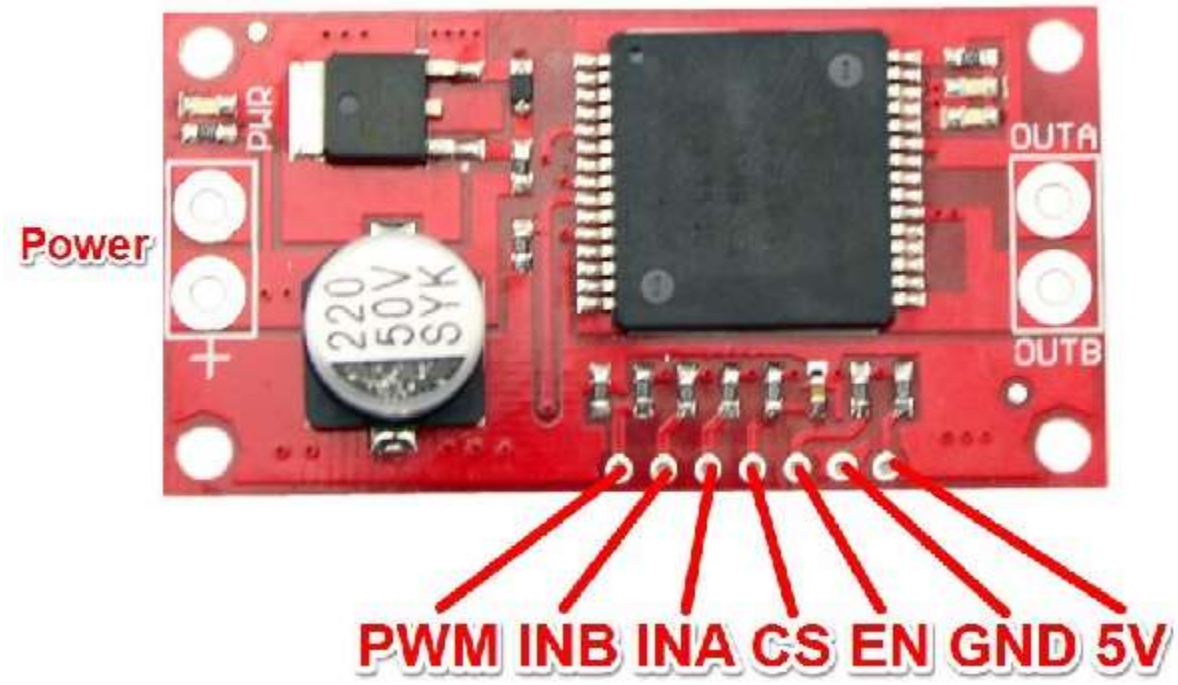
Features

- Voltage max: 16V
- Maximum current rating: 30 A
- Practical Continuous Current: 14 A
- Current sensing available to Arduino analog pin
- MOSFET on-resistance: 19 mΩ (per leg)
- Maximum PWM frequency: 20 kHz
- Thermal Shutdown
- Undervoltage and Overvoltage shutdown

Specifications

- Max. Input Voltage: 16VDC
- Max. Current Rating: 30A
- Practical Continuous Current: 14A
- MOSFET on-resistance: 19 mΩ per leg
- Max. PWM Frequency: 20 kHz
- Weight: 25g
- Size: 53 x 22mm

Pin Definition



Out A, Out B: Motor Power
PWR (+/-): Power Supply Voltage
PWM: Pulse Width Signal (control the motor speed)
INA, INB: Rotation direction and brake control
CS: Current Sensor (optional)
EN: Status of switches output (Analog pin - Optional)

Arduino Connection

```
EN A----------PIN 7
EN B----------PIN 8
PWM----------PIN 5
CS-----------PIN A2
EN-----------PIN A0
```

Arduino Code

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

#define MOTOR_1 0
#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

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("4. READ CURRENT");
  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);
  }
}
```

Results