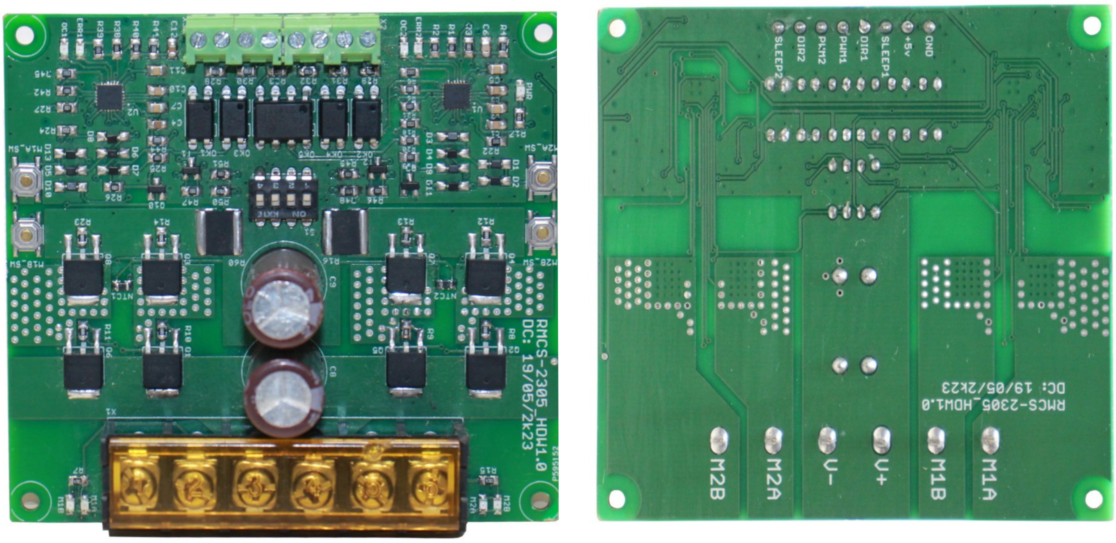
Dual Channel Optically isolated DC Motor Driver 6V-30V 20Amp



RMCS – 2305

Operating Manual v1.0

Contents

Introduction :

The Rhino Dual Driver 20A enables bidirectional control of two high-power brushed DC motors from 6V to 30V. With full discrete NMOS H-Bridge design, this motor driver is able to support 20Amp continuously for each motor without any heat-sink

The onboard test buttons and motors output LEDs allow functional test of the motor driver in a quick and convenient way without hooking up the host controller. This motor driver can be controlled with PWM and DIR inputs.

it’s compatible with wide variety of host controller (e.g. Arduino, Raspberry Pi, PLC).

The advantage of this drive is that all inputs are electrically and optically isolated.

Various protection features are also incorporated in each channel of the Rhino 20A. Over-current protection prevents the motor driver from damage when the motor stalls or an over-sized motor is hooked up. When the motor is trying to draw current more than what the motor driver can support, the motor current will be limited at the maximum threshold.

Assisted by temperature protection, the maximum current limiting threshold is determined by the board temperature. The higher the board temperature, the lower the current limiting threshold. This way, Rhino Dual Driver 20A able to deliver its full potential depending on the actual condition without damaging the MOSFETs.

Note: Power input does not have reverse-voltage protection. Connecting the battery in reverse polarity will damage the motor driver instantaneously.

Technical specifications :

Supply Voltage and Current :

Specification	Min	Max	Units	Comments
Supply voltage	6	30	Volts DC	Between +VCC and GND
Phase Current (continuous)	-	20	Amps	Peak 20 Amps per phase
Phase current (peak)	-	60	Amps	Max 60amps capacity for fluctuation.
Logic Input Voltage (PWM & DIR)	0	5	Volts	Can be given through pulse generator or some microcontroller.
PWM Frequency	0	20	KHz	Output frequency will be same as input frequency.

Note : Peak current is limited by the over-current protection circuit. Actual current limit is depending on board temperature. Value shown is at room temperature (25 - 30 degree Celsius).

Pin Configuration of the Drive:

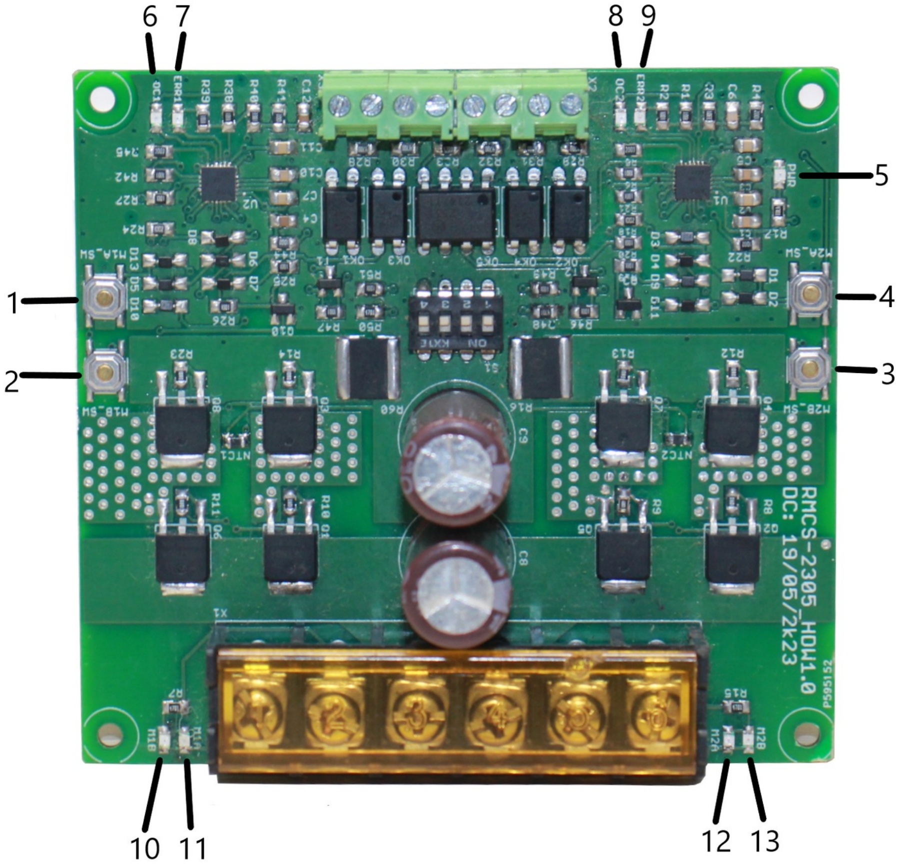


Fig 1:

No	Description
1	M1A_SW
2	M1B_SW
3	M2A_SW
4	M2B_SW
5	POWER LED
6	OC1 LED
7	ERR1 LED

No	Description
8	OC2 LED
9	ERR2 LED
10	M1B LED
11	M1A LED
12	M2A LED
13	M2B LED

NAME	FUNCTION
M1A_SW	Switch to run motor M1 clockwise. (Direction of motor will depend on connection of motor wires)
M1B_SW	Switch to run motor M1 counter clockwise.
M2A_SW	Switch to run motor M2 clockwise. (Direction of motor will depend on connection of motor wires)
M2B_SW	Switch to run motor M2 counter clockwise.
POWER LED	Indicates power is on. If its not on then check your connections.
OC1 LED	For motor M1 : Turn on when current limiting is in action. Current limit threshold is depending on the board temperature.
ERR1 LED	For motor M1 : Turn on during Under voltage, Shutdown or Hardware fault
OC2 LED	For motor M2 : Turn on when current limiting is in action. Current limit threshold is depending on the board temperature.
ERR2 LED	For motor M2 : Turn on during Under voltage, Shutdown or Hardware fault
M1A LED	Indicates clockwise direction for motor M1
M1B LED	Indicates counter clockwise direction for motor M1.
M2A LED	Indicates clockwise direction for motor M2.
M2B LED	Indicates counter clockwise direction for motor M2.

Connection diagram of the Drive:

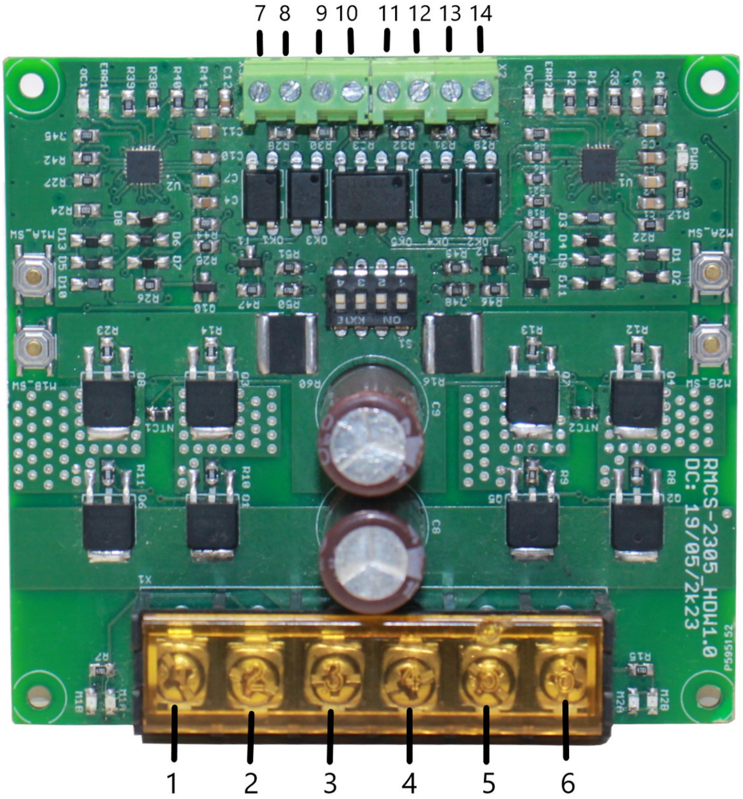


Fig 2 :

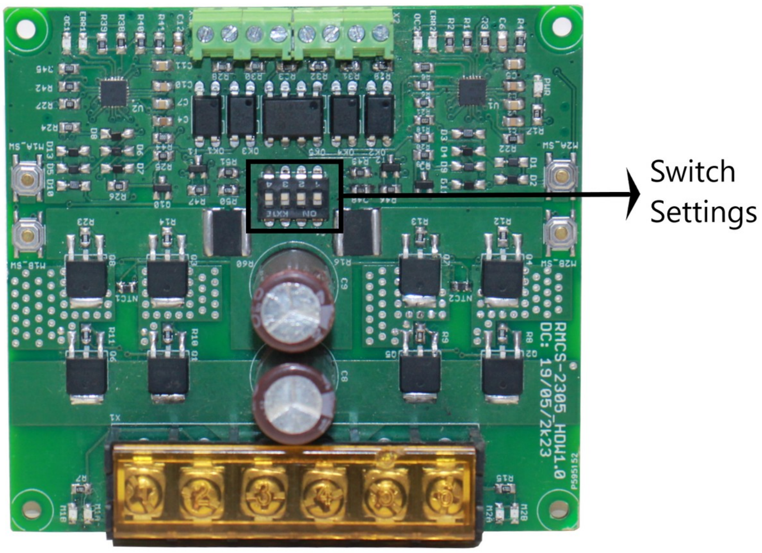
No	Description
1	M1A
2	M1B
3	VCC (6 – 30 V) OF POWER SUPPLY
4	GND OF POWER SUPPLY
5	M2A
6	M2B
7	GND

No	Description
8	+5V
9	SLEEP1
10	DIR1
11	PWM1
12	PWM2
13	DIR2
14	SLEEP2

NAME	FUNCTION
M1A	Connect motor M1.
M1B	Connect motor M1. (Connect in reverse to change direction.)
VCC	Connect positive of power supply. (6 – 30 V)
GND	Connect GND of power supply.
M2A	Connect motor M2.
M2B	Connect motor M2. (Connect in reverse to change direction.)
GND	Connect to Gnd of microcontroller
+5V	+5V output.
SLEEP 1	Motor M1 Enable (Motor Free). Active high pin (+5V). (Can give through switch from +5V pin or microcontroller pins.)
DIR1	Active high pin. Use to change the direction of motor M1. (Can give through switch from +5V pin or microcontroller pins.)
PWM1	PWM input for motor M1.
PWM2	PWM input for motor M2.
DIR2	Active high pin. Use to change the direction of motor M2. (Can give through switch from +5V pin or microcontroller pins.)
SLEEP 2	Motor M2 Enable (Motor Free). Active high pin (+5V). (Can give through switch from +5V pin or microcontroller pins.)

Switch Settings of the Drive:

Fig 3:



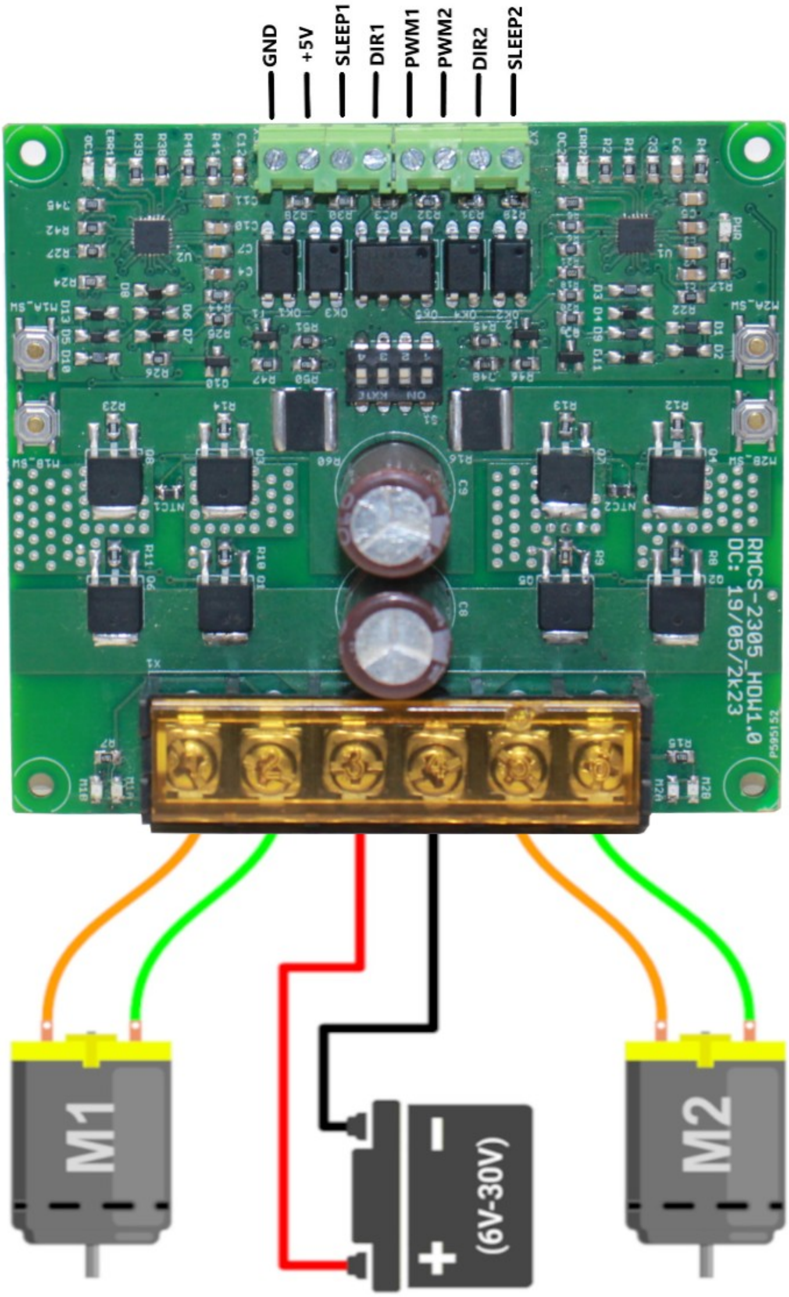
Peak current Motor M1	Switch S3	Switch s4
60 Amps	OFF	OFF
40 Amps	ON	OFF
16 Amps	ON	ON

Peak current Motor M2	Switch S1	Switch S2
60 Amps	OFF	OFF
40 Amps	ON	OFF
16 Amps	ON	ON

Connection Diagram :

Hardware Connection :

Fig 4:



Controlling :

As depicted in fig 4 , there is a provision to connect two motors to the driver, with a power supply positioned in between.
To initiate a test run of the motors, you can conveniently utilize the push buttons integrated into the driver, specifically M1A_SW and M1B_SW for motor 1, and M2A_SW and M2B_SW for motor 2.
It is important to note that the direction of each motor is determined by its individual connection.
Reversing the motor's connection will result in a reversal of its direction
of rotation. Reversing the connection of motor will reverse its direction.

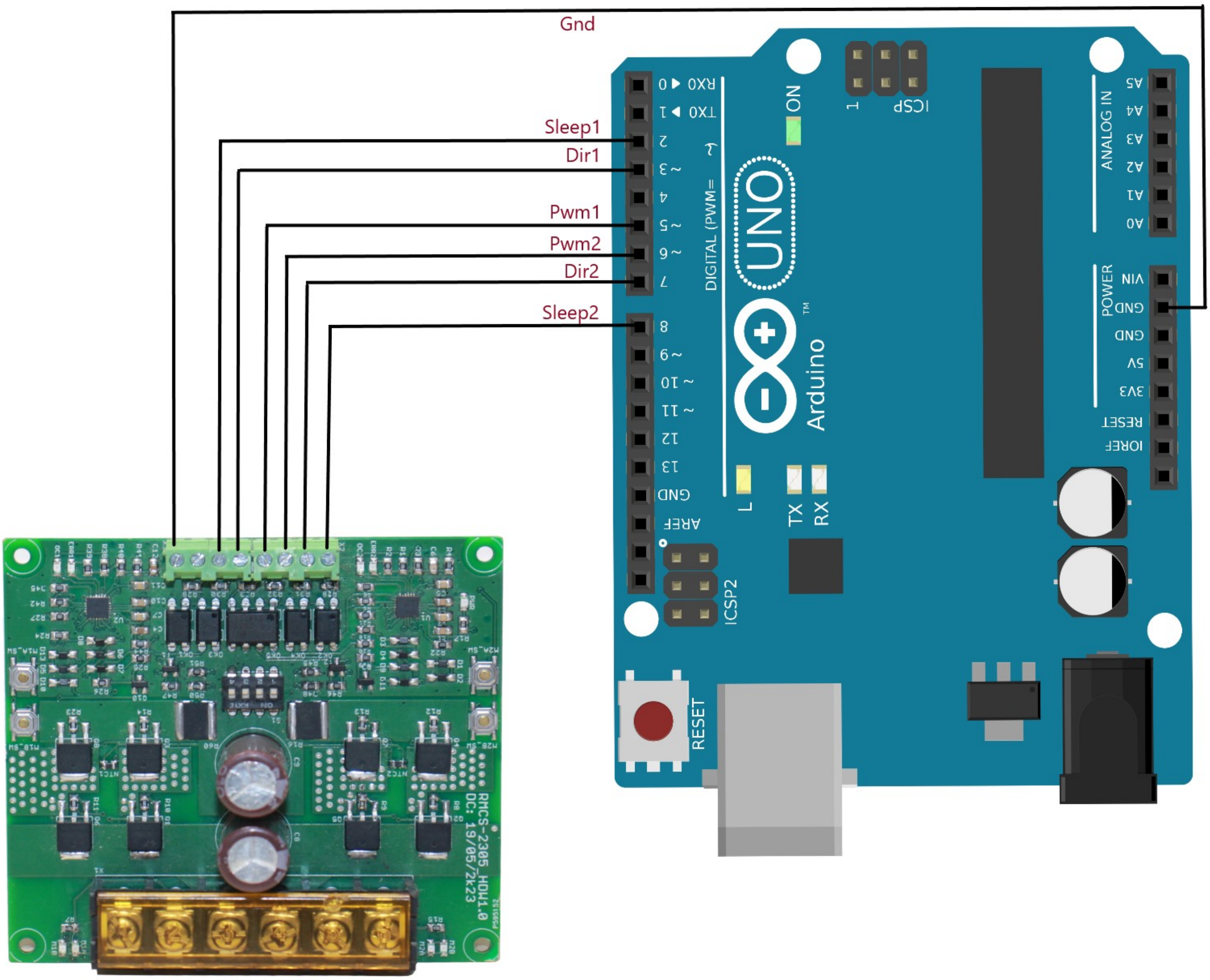
Motor control using Microcontroller :

The motors can be easily controlled using any microcontroller equipped with 5V pins. By providing PWM signals to the PWM pins of the drive (PWM1 and PWM2), the motor speed can be adjusted accordingly.
To change the direction of the motors, a signal from any digital 5V pin can be directed to the corresponding direction pins (DIR1 and DIR2). Furthermore, to stop the motor motion, a 5V signal from any digital pin can be applied to the sleep pins of the drive (SLEEP1 and SLEEP2). For a visual representation of these connections, please refer to fig 5 on the subsequent page.

Motor control using potentiometer :

If you intend to control the motor's speed using a potentiometer, you will need to employ a microcontroller such as Arduino.
This microcontroller can receive input from the potentiometer and subsequently provide output to the PWM pins of the drive.
For a visual reference regarding this setup, please refer to fig 6 on page 13.

Fig 5:

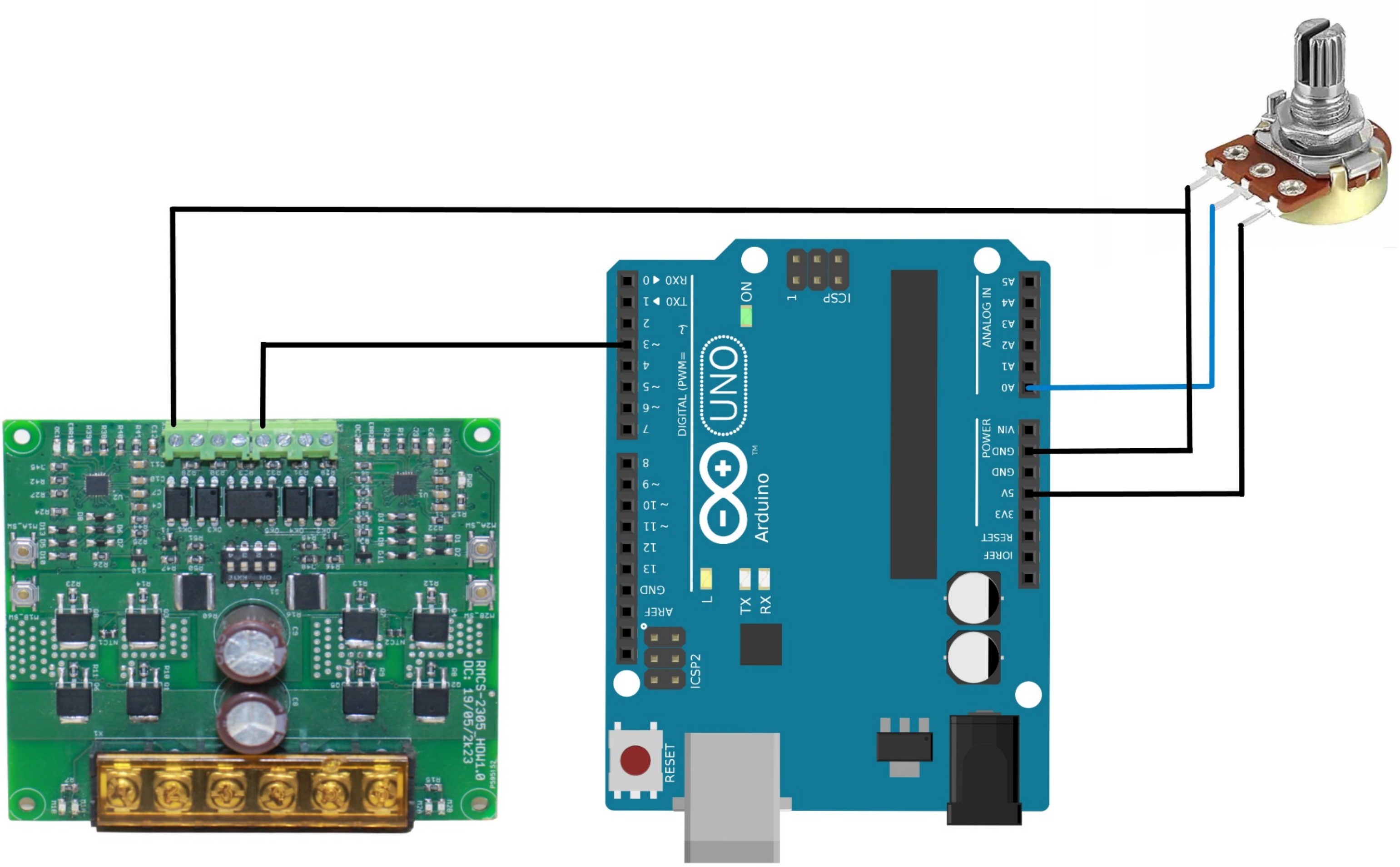

Gnd
Sleep1
Dir1
Pwm1
Pwm2
Dir2
Sleep2
RMCS-2305_HDW1.0
DC: 19/05/2k23
Arduino
UNO
DIGITAL (PWM= ~)
ANALOG IN
POWER
RESET
ICSP
ICSP2
ON
TX
RX
L
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX0 ▶ 1
RX0 ◀ 0
A0
A1
A2
A3
A4
A5
VIN
GND
GND
5V
3V3
RESET
IOREF

Sample code for motor control using Arduino :

Here you can check this sample code of Arduino to run motor with arduino :

```
/* Connections of RMCS-2305 with arduino :
 *  RMCS2305 : Arduino pins
 *  Sleep1 : 2
 *  Gnd : gnd
 *  Dir1 : 3
 *  pwm1 : 5
 *  pwm2 : 6
 *  Dir2 : 7
 *  Sleep2 : 8

 */




const int Sleep1 = 2; //For the motor M1
const int Dir1 = 3; //pin to change direction for motor M1
const int pwm1 = 5; //Pwm pin for motor M1
const int pwm2 = 6;// Pwm pin for motor M2
const int Dir2 = 7; // pin to change direction for motor M2
const int Sleep2 = 8; //For the motor M2


void setup()
{

  pinMode(Sleep1,OUTPUT);
  pinMode(Dir1,OUTPUT);
  pinMode(pwm1,OUTPUT);
  pinMode(pwm2,OUTPUT);
  pinMode(Dir2,OUTPUT);
  pinMode(Sleep2,OUTPUT);


}

void loop() {


  digitalWrite(Sleep1,LOW); // To stop motor M1 change Signal to HIGH
  digitalWrite(Dir1,HIGH);  // To change the direction of motor M1, change signal to LOW
  digitalWrite(Sleep2,LOW); // To stop motor M2 change Signal to HIGH
  digitalWrite(Dir2,HIGH); // To change the direction of motor M2, change signal to LOW

  analogWrite(pwm1,255); // you can give value in between 0 - 255
  analogWrite(pwm2,255); // you can give value in between 0 - 255


}
```

Fig 6 :


RMCS-2305 HDW1.0
DC: 19/05/2k23
RX0 ◂ 0
TX0 ▸ 1
2
~ 3
4
~ 5
~ 6
7
8
~ 9
~ 10
~ 11
12
13
GND
AREF
DIGITAL (PWM= ~)
ON
ICSP
1
UNO
Arduino
L
TX
RX
ICSP2
RESET
ANALOG IN
A5
A4
A3
A2
A1
A0
POWER
VIN
GND
GND
5V
3V3
RESET
IOREF

Sample code for speed control using potentiomer :

Here you can check this code of Arduino to run motor with potentiomer and Arduino :

```
//Constants:
const int PwmPin = 9;  //pin 9 has PWM funtion
const int potPin = A0; //pin A0 to read analog input

//Variables:
int value; //save analog value

void setup(){
  //Input or output?
  pinMode(PwmPin, OUTPUT);
  pinMode(potPin, INPUT);
}

void loop(){
  value = analogRead(potPin);          //Read and save analog value from potentiometer
  value = map(value, 0, 1023, 0, 255); //Map value 0-1023 to 0-255 (PWM)
  analogWrite(PwmPin, value);          //Send PWM value to drive
  delay(100);                          //Small delay
}
```