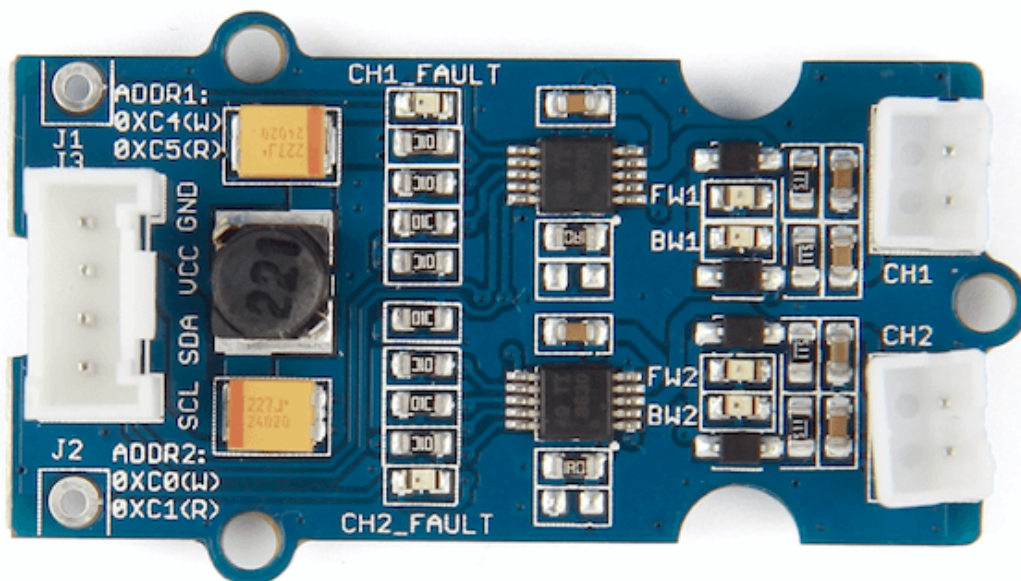


Grove - Mini I2C Motor Driver v1.0



This Grove - Mini I2C motor driver includes two DRV8830. The DRV8830 provides an integrated motor driver solution for battery-powered toys, printers, and other low-voltage or battery-powered motion control applications. The module has two H-bridge drivers, and can drive two DC motors or two winding of stepper motors, as well as other loads like solenoids. It requires an onboard 5V voltage regulator which can power the I2C bus. All driver lines are diode protected from back EMF. It features two LEDs for fault indicator and four LEDs to indicate which direction each motor is running.

GROVE system plug and I2C interface enables you to daisy-chain the driver with many other devices.

[Get One Now !\[\]\(99f58673407353e96a019fbca558fd72_img.jpg\)](#)

[<https://www.seeedstudio.com/Grove%C2%A0-%C2%A0I2C%C2%A0Mini%C2%A0Motor%C2%A0Driver-p-2508.html>]

Features

- Without external power supply
- Two leds for fault indicator
- Default maximum drive current 200 mA
- Grove compatible
- I2C interface
- Motor's speed and direction can control
- Number of channels: 2
- Easy to use



Tip

More details about Grove modules please refer to [Grove System](#)
[https://wiki.seeedstudio.com/Grove_System/]

Application Ideas

This motor driver can be used to drive any brushed electronic motor as long as it doesn't consume more than 1A at 5v. Two motors can

be driven simultaneously while set to a different speed and direction. The speed can be set fully proportional and is controlled by I2C command.

- Battery-Powered:
 - Printers
 - Toys
 - Robotics
 - Cameras
 - Phones
- Small Actuators, Pumps, etc.

Here are some projects for your reference.

Make a Mini Toy Car

Make a Steampunk S



Make it NOW!

[https://community.seeedstudio.com/project_detail.html?id=392]

Make it NOW!



[https://community.seeedstudio.com/project_detail.html?id=1131]

Specifications

Item	Min	Typical	Max	Unit
Working Voltage	2.75	5	6.8	VDC
Max Output Current per channel	0.2(default)	-	1	A
Input/output voltage on I2C bus	3.3/5			V
Communication protocol	I2C			/
Default I2C Address	0xC0, 0xC4			/



Platforms Supported

Arduino	Raspberry Pi		
			

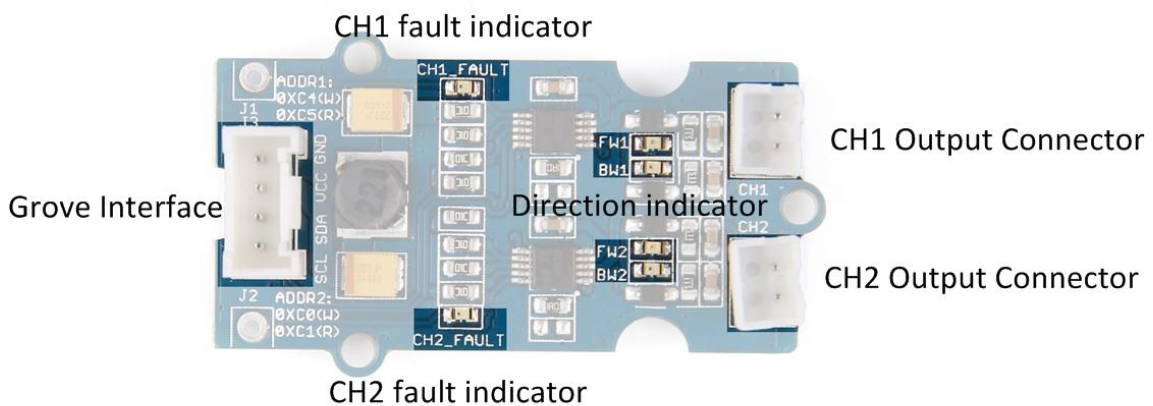




Caution

The platforms mentioned above as supported is/are an indication of the module's software or theoretical compatibility. We only provide software library or code examples for Arduino platform in most cases. It is not possible to provide software library / demo code for all possible MCU platforms. Hence, users have to write their own software library.

Hardware Overview



- Grove Interface** - Grove products have a eco system and all have a same connector which can plug onto the **Base Shield**. Connect this module to the I²C port of Base Shield, and then it can work well with Arduino. However, you can also connect Grove - Mini I2C Motor Driver to Arduino without Base Shield by jumper wires.

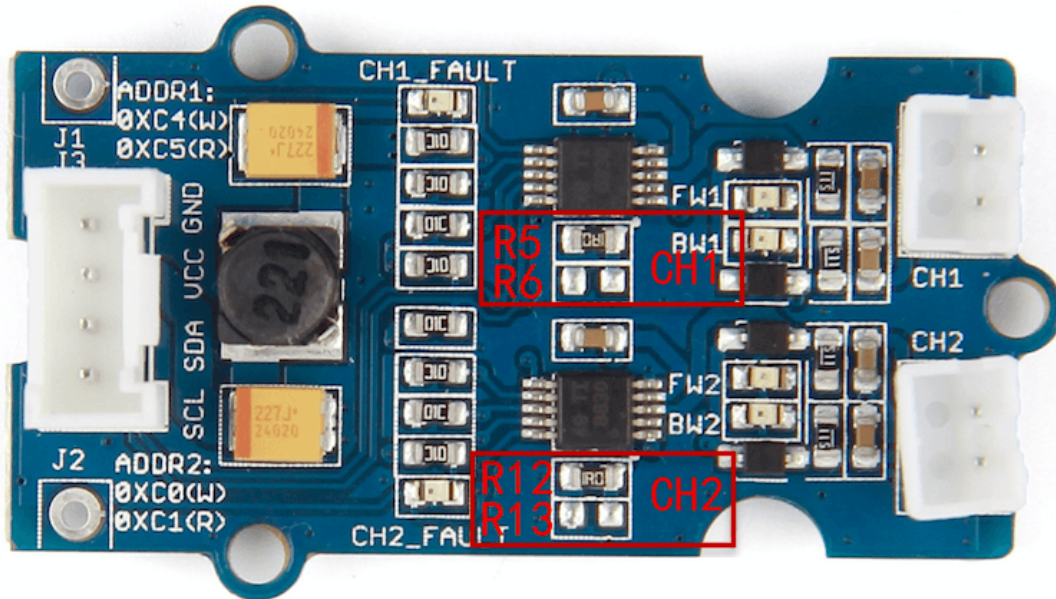
Arduino UNO	Base Shield	Grove - Mini I2C Motor Driver
5V	I2C port	VCC
GND		GND
SDA		SDA
SCL		SCL

- **CH1 fault indicator** - Channel 1 fault indicator.
- **CH2 fault indicator** - Channel 2 fault indicator.
- **Direction indicator** - Motor direction indicator.
- **CH1 Output Connector** - Motor 1 connector.
- **CH2 Output Connector** - Motor 2 connector.

Hardware function

Change Default maximum drive current

The default maximum drive current of each channel is 200mA, see the front picture of the board



Each channel (CH1,CH2) has been added a resistor, and each value of resistor (R5,R12) is 1 Ω , so the maximum drive current is 200mA according to the following equation

$$I_{max} = \frac{200mV}{R}$$

Meantime, each channel provides a reserved solderable pad (R6 for CH1, R13 for CH2), so you can solder a resistor onto the board to change the resistor value of each channel. Following is the new equation if adding resistor to the board

$$CH1 \left\{ \begin{array}{l} R_{CH1} = \frac{R_5 \times R_6}{R_5 + R_6} \\ I_{CH1_max} = \frac{200mV}{R_{CH1}} \end{array} \right.$$

$$CH2 \left\{ \begin{array}{l} R_{CH2} = \frac{R_{12} \times R_{13}}{R_{12} + R_{13}} \\ I_{CH2_max} = \frac{200mV}{R_{CH2}} \end{array} \right.$$

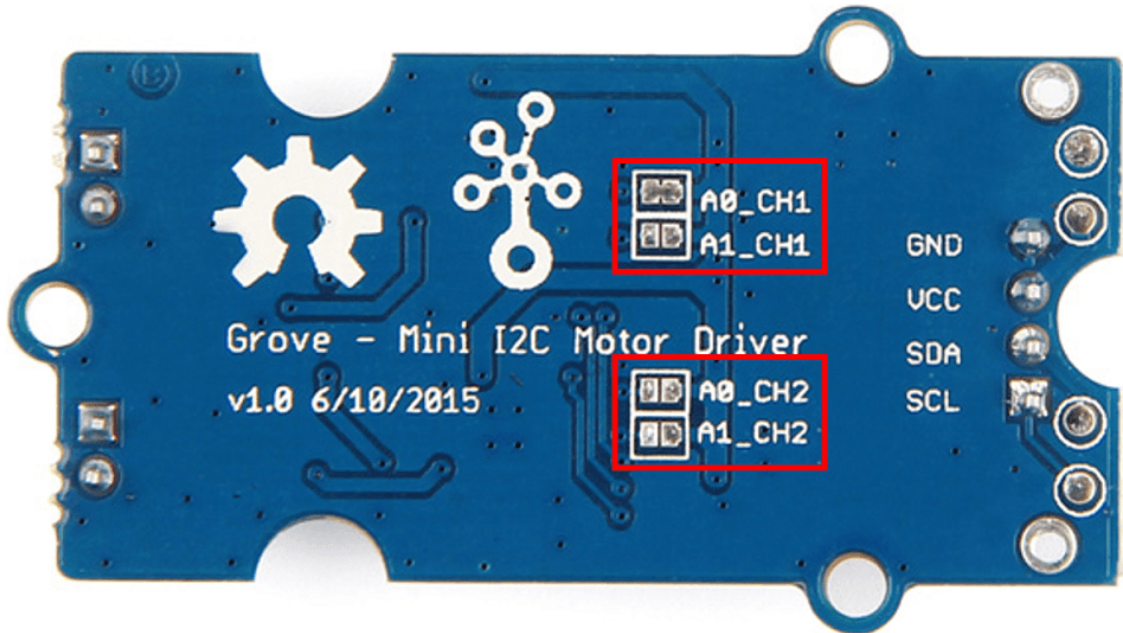


Caution

Maximum working current of each channel must be less than 1A. So the minimum value of resistor soldered to the reserved pad should not less than 0.2 Ω.

Change Default I²C Address

The I²C address of each channel is changeable. Please take a look at the back side of the board, you will find there are 4 jumper pads; A0_CH1 and A1_CH1 are for channel 1, A0_CH2 and A1_CH2 are for channel 2, as shown below:



You can solder or unsolder each jumper to change the I2C address:

- 1 - You need a solder iron, just solder two sides of the jumper together
- 0 - You need a solder iron, just unsolder two sides of the jumper.

A1	A0	Address Write	Address Read	Note
0	0	0xC0h	0xC1h	Default addr of CH2
0	1	0xC4h	0xC5h	Default addr of CH1
1	0	0xCCh	0xCDh	
1	1	0xD0h	0xD1h	



Note

The library of Grove - Mini I2C Motor driver is dependent on the default address.

Getting Started

Now, let us begin to use the Grove - Mini I2C Motor Driver.

Preparations

Now we are making a demo for Grove - Mini I2C Motor Driver v1.0 which require following modules.

- 2 * DC Motor 2V-6V

- [Seeeduino Lite](#)

[<https://www.seeedstudio.com/depot/Seeeduino-Lite-p-1487.html>]

Seeeduino Lite is compatible with Arduino.

If you are using an Arduino UNO or any others Arduino compatible boards that with out a Grove connect,

You will need a [Grove Base Shield](#)

[https://www.seeedstudio.com/depot/base-shield-v13-p-1378.html?cPath=132_134] to connect the Grove easily.

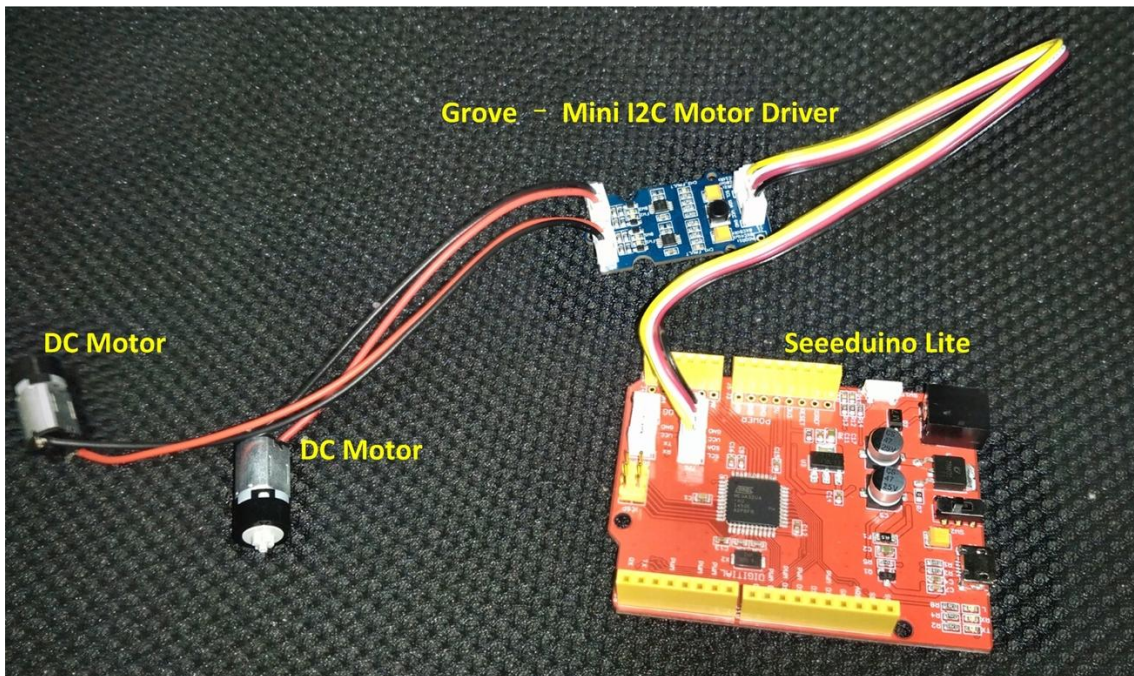
If this is your first time using Arduino or Seeeduino, Please put hand on [here](#) [/Getting_Started_with_Seeduino] to start your Arduino journey.

Hardware Installation

Grove - Mini I2C Motor Driver got one Grove socket for connecting two modules above. They are:

- 2 * DC Motor 2V-6V - connect to CH1 & CH2 Output connector.
- Seeeduino Lite

Connect Seeeduino's Grove I2C Interface to Mini Motor Driver's Grove Interface as shown below:



Software Work

Because the module has 2 H-bridges, you can not only make a robot go forwards and backwards, but also turn around by having each wheel spin in a different direction.

Connect Seeeduino to computer use a micro USB cable.

Now, let us use the Grove - Mini I2C Motor Driver to control two DC motors rotating in the positive or opposite direction.

Given below is an example program to be used with an Arduino. The code for this is very basic, but you can also change it and do it your own way.

```

1  /*****
2  Example code demonstrating the use of the Arduino Libra
3  the SparkFun MiniMoto board, which uses the TI DRV8830
4  Low-voltage DC motor control.
5
6  This code is beerware; if you use it, please buy me (or
7  SparkFun employee) a cold beverage next time you run in
8  us at the local.
9
10 17 Sep 2013- Mike Hord, SparkFun Electronics
11
12 Code developed in Arduino 1.0.5, on a Fio classic board
13
14 **Updated for Arduino 1.6.4 5/2015**
15 ****
16
17 #include <SparkFunMiniMoto.h> // Include the MiniMoto
18
19 // Create two MiniMoto instances, with different addresses
20 MiniMoto motor0(0xC4); // A1 = 1, A0 = clear
21 MiniMoto motor1(0xC0); // A1 = 1, A0 = 1 (default)
22
23 #define FAULTn 16 // Pin used for fault detection.
24
25 // Nothing terribly special in the setup() function- pr

```

```
26 // serial port, print a little greeting, and set up ou
27 // pin as an input.
28 void setup()
29 {
30     Serial.begin(9600);
31     Serial.println("Hello, world!");
32     pinMode(FAULTn, INPUT);
33 }
34
35 // The loop() function just spins the motors one way, t
36 // other, while constantly monitoring for any fault co
37 // to occur. If a fault does occur, it will be reporte
38 // the serial port, and then operation continues.
39 void loop()
40 {
41     Serial.println("Forward!");
42     motor0.drive(100);
43     motor1.drive(100);
44     delayUntil(1000);
45     Serial.println("Stop!");
46     motor0.stop();
47     motor1.stop();
48     delay(1000);
49     Serial.println("Reverse!");
50     motor0.drive(-100);
51     motor1.drive(-100);
52     delayUntil(1000);
53     Serial.println("Brake!");
54     motor0.brake();
55     motor1.brake();
56     delay(1000);
57 }
58
59 // delayUntil() is a little function to run the motor e
60 // a designated time OR until a fault occurs. Note tha
61 // a very simple demonstration; ideally, an interrupt
62 // used to service faults rather than blocking the app
63 // during motion and polling for faults.
64 void delayUntil(unsigned long elapsedTime)
65 {
66     // See the "BlinkWithoutDelay" example for more det
```

```

67 // and why this loop works the way it does.
68 unsigned long startTime = millis();
69 while (startTime + elapsedTime > millis())
70 {
71     // If FAULTn goes low, a fault condition *may*
72     // sure, we'll need to check the FAULT bit.
73     if (digitalRead(FAULTn) == LOW)
74     {
75         // We're going to check both motors; the loop
76         // for each...
77         byte result = motor0.getFault();
78         // If result masked by FAULT is non-zero, w
79         // condition, and we should report it.
80         if (result & FAULT)
81         {
82             Serial.print("Motor 0 fault: ");
83             if (result & OCP) Serial.println("Chip
84             if (result & ILIMIT) Serial.println("Lo
85             if (result & UVLO) Serial.println("Unde
86             if (result & OTS) Serial.println("Over
87             break; // We want to break out of the m
88             // so we can stop motion in response t
89         }
90         result = motor1.getFault();
91         if (result & FAULT)
92         {
93             Serial.print("Motor 1 fault: ");
94             if (result & OCP) Serial.println("Chip
95             if (result & ILIMIT) Serial.println("Lo
96             if (result & UVLO) Serial.println("Unde
97             if (result & OTS) Serial.println("Over
98             break;
99         }
100     }
101 }
102 }

```

Now click Upload(CTRL+U) to burn testing code. Please refer to [here](#) [/Arduino_Common_Error] for any error prompt and you can

also add comment on [community](#)
[<https://community.seeedstudio.com/>].

Review Results

After upload is complete, the motors will rotate in the positive or opposite direction in cycle.

Schematic Online Viewer



Resources

- [DRV8830 Datasheet](https://files.seeedstudio.com/wiki/Grove-Mini_I2C_Motor_Driver_v1.0/res/DRV8830.pdf) [https://files.seeedstudio.com/wiki/Grove-Mini_I2C_Motor_Driver_v1.0/res/DRV8830.pdf]
- [Grove - Mini I2C Motor Driver_Eagle_File](https://files.seeedstudio.com/wiki/Grove-Mini_I2C_Motor_Driver_v1.0/res/Grove-Mini_I2C_Motor_Driver_v1.0_SCH_PCB.zip) [https://files.seeedstudio.com/wiki/Grove-Mini_I2C_Motor_Driver_v1.0/res/Grove-Mini_I2C_Motor_Driver_v1.0_SCH_PCB.zip]
- [Grove - Mini I2C Motor Driver Schematic Document](https://files.seeedstudio.com/wiki/Grove-Mini_I2C_Motor_Driver_v1.0/res/Grove-Mini_I2C_Motor_Driver_v1.0_SCH.pdf) [https://files.seeedstudio.com/wiki/Grove-Mini_I2C_Motor_Driver_v1.0/res/Grove-Mini_I2C_Motor_Driver_v1.0_SCH.pdf]
- [Grove - Mini I2C Motor Driver Source Library](https://github.com/Seeed-Studio/Drv8830_Motor_Driver) [https://github.com/Seeed-Studio/Drv8830_Motor_Driver]

Tech Support

Please submit any technical issue into our [forum](https://forum.seeedstudio.com/) [https://forum.seeedstudio.com/].



[https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts]

