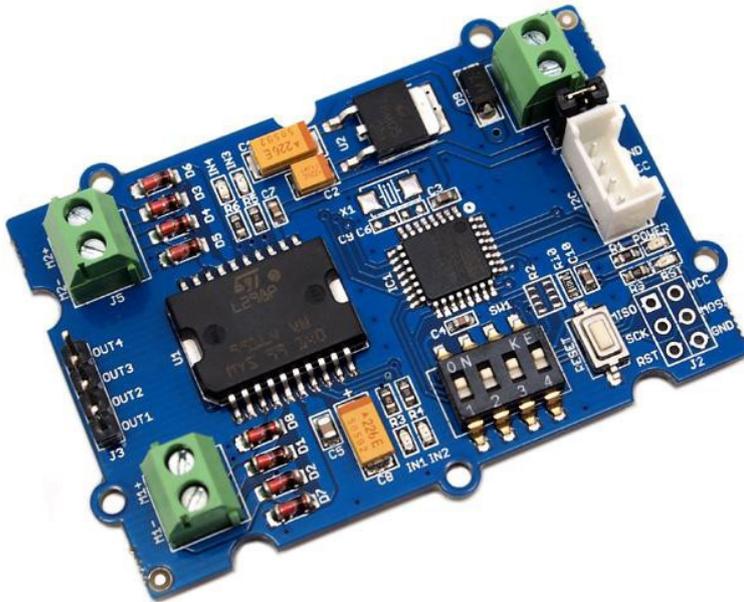


Grove - I2C Motor Driver V1.3

Introduction

3.3V 5.0V I2C



The Grove - I2C Motor Driver V1.3 (latest version) can directly control Stepper Motor or DC Motor. Its heart is a dual channel H-bridge driver chip (L298P) that can handle current up to 2A per channel, controlled by an Atmel ATmega8L which handles the I2C communication with for example an Arduino. Both motors can be driven simultaneously while set to a different speed and direction. It can power two brushed DC motors or one 4-wire two-phase stepper motor. It requires a 6V to 15V power supply to power the motor and has an onboard 5V voltage regulator which can power the I2C bus and the Arduino(selectable by jumper). All driver lines are protected by diodes from back-EMF.

Contrast to the [Grove - I2C motor driver V1.2](#), the V1.3 enables users to control the stepper more easily. You do not need to control the steppers all the time anymore, simply send a command to I2C motor driver V1.3 to drive a stepper, and it will act as your command, which would save your Arduino resource and simplify your code.

Version Tracker

Revision	Descriptions	Release
v1.0	Initial public release	May 17th, 2012
v1.2	Modify the I2C address set by hardware	July 2nd, 2012
v1.3	Modify the firmware to support off-line Stepper	Feb 18th, 2013

Features

- Grove Compatible
- I2C Interface
- Adjustable motor speed and rotation direction
- Changeable slave address by hardware

Tip

More details about Grove modules please refer to [Grove System](#)

Specifications

Item	Min	Typical	Max	Unit
Working Voltage	6	-	15	VDC
Max Output Current per channel	2			A
Maximum Total current	1.0			A
Input/output voltage on I2C bus	5			V
Communication protocol	I2C			/

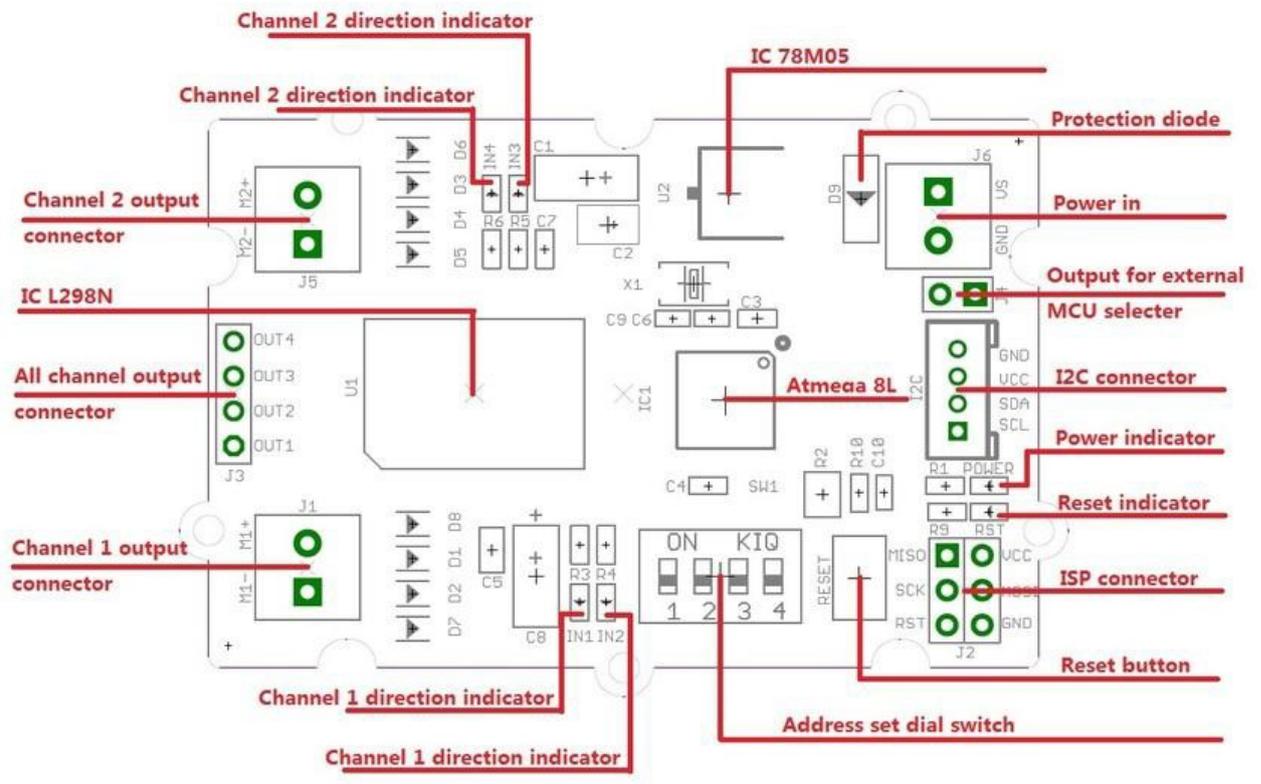
Platforms Supported

Arduino	Wio	BeagleBone	Raspberry Pi	LinkIt ONE
				

Caution

The platforms mentioned above as supported is/are an indication of the module's hardware 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



78M05 IC: 5V voltage regulator

L298 IC: Dual full bridge driver

ATmega8 IC: Control Motor Rotate.

Note

Input voltage on screw terminals is regulated to 5V and connected to I2C +5V via a jumper (J4). Remove jumper if both external power via the screw terminals and power via the I2C header are used. Use jumper if 5V should be supplied to the I2C bus.

Application Ideas

- Robots
- Homebuilt RC cars
- Case fans
- High power LED illumination

Caution

The board will be very hot while operating over 1Amp. Do keep your hands off!

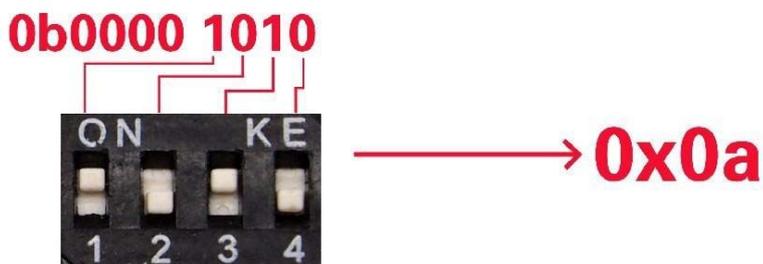
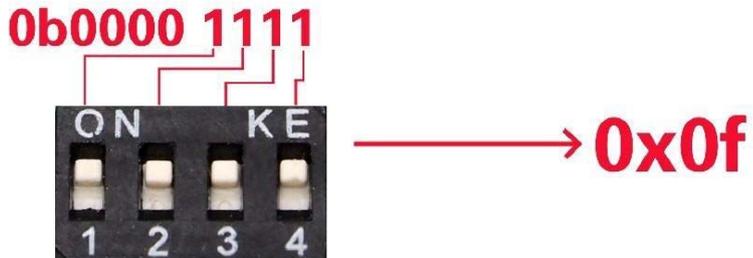
Usage

The I2C Motor Driver can control motor which is based on the chip L298. The L298 isn't just a dual motor driver, it is a dual H-bridge. An h-bridge is basically a specific setup of transistors that allow you to switch direction of current. Hooking up to a motor means you can have it spin in both directions; and with PWM input, you can use your Arduino to make them spin at any speed. Because the L298 has 2 H-bridges, you can make a robot turn around by spinning each wheel in different directions, and of course go forwards and backwards.

Now, let us use the I2C Motor Driver to control two DC motors or a stepper rotating clockwise and anticlockwise.

Set the address of the I2C Motor Driver

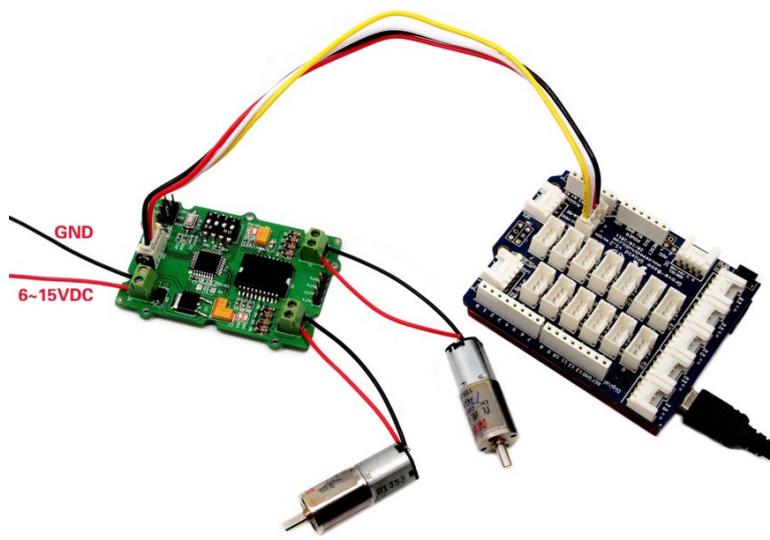
- Set the address by dial switch is a new function added to the new I2C Motor Driver.



- Then keep the address setup in the program the same as the address setup on the I2C motor driver. The default address setup in the program is 0x0f.

```
#define I2CMotorDriverAdd 0x0f>>1 // Set the address of the I2CMotorDriver
```

How to drive 2 DC motors



The first thing to notice however, is that you need an external power source for your DC motors. The 5V pin on the Arduino cannot provide enough power to drive 2 motors, you may damage your Arduino if you do so.

REGISTER SET AND PROGRAMMER'S MODEL

TABLE 1 : COMMAND BYTE DESCRIPTION (SMBUS/I²C READ_BYTE AND WRITE_BYTE)

Command name	Code	Function
MSS	0x82	Set PWM value to fix speed rotation
PWMFS	0x84	Set PWM Frequency
DS	0xAA	Set Direction rotation motor
MSA	0xA1	Set high logical level on output ENA
MSB	0xA5	Set high logical level on output ENB
NOT	0x01	Terminate transaction
ENST	0x1A	Enable Stepper mode
UNST	0x1B	Unable Stepper mode
SRNU	0x1C	Define the step of motor stepper

TABLE 2 : CONFIGURATION BYTE to SET DIRECTION - 8 BITWRITE

Byte State	Motor A	Motor B	Byte State	Motor A	Motor B
0000 0000b	Brake	Brake	0000 1000b	Counterclockwise Rotation	Brake
0000 0001b	Brake	Clockwise Rotation	0000 1001b	Counterclockwise Rotation	Clockwise Rotation
0000 0010b	Brake	Counterclockwise Rotation	0000 1010b	Counterclockwise Rotation	Counterclockwise Rotation
0000 0011b	Brake	Brake	0000 1011b	Counterclockwise Rotation	Brake
0000 0100b	Clockwise Rotation	Brake	0000 1100b	Brake	Brake
0000 0101b	Clockwise Rotation	Clockwise Rotation	0000 1101b	Brake	Clockwise Rotation
0000 0110b	Clockwise Rotation	Counterclockwise Rotation	0000 1110b	Brake	Counterclockwise Rotation
0000 0111b	Clockwise Rotation	Brake	0000 1111b	Brake	Brake

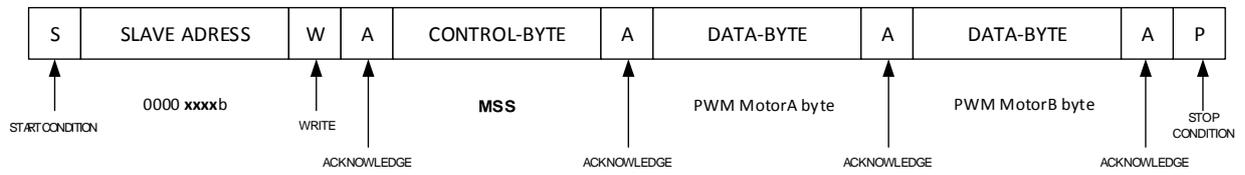


FIGURE 2 : Example set PWM speed Motors DC

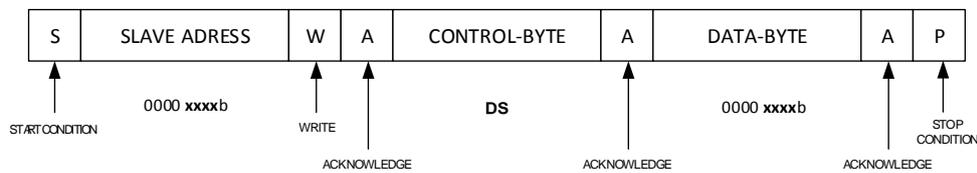


FIGURE 3 : Example set Direction Motors DC

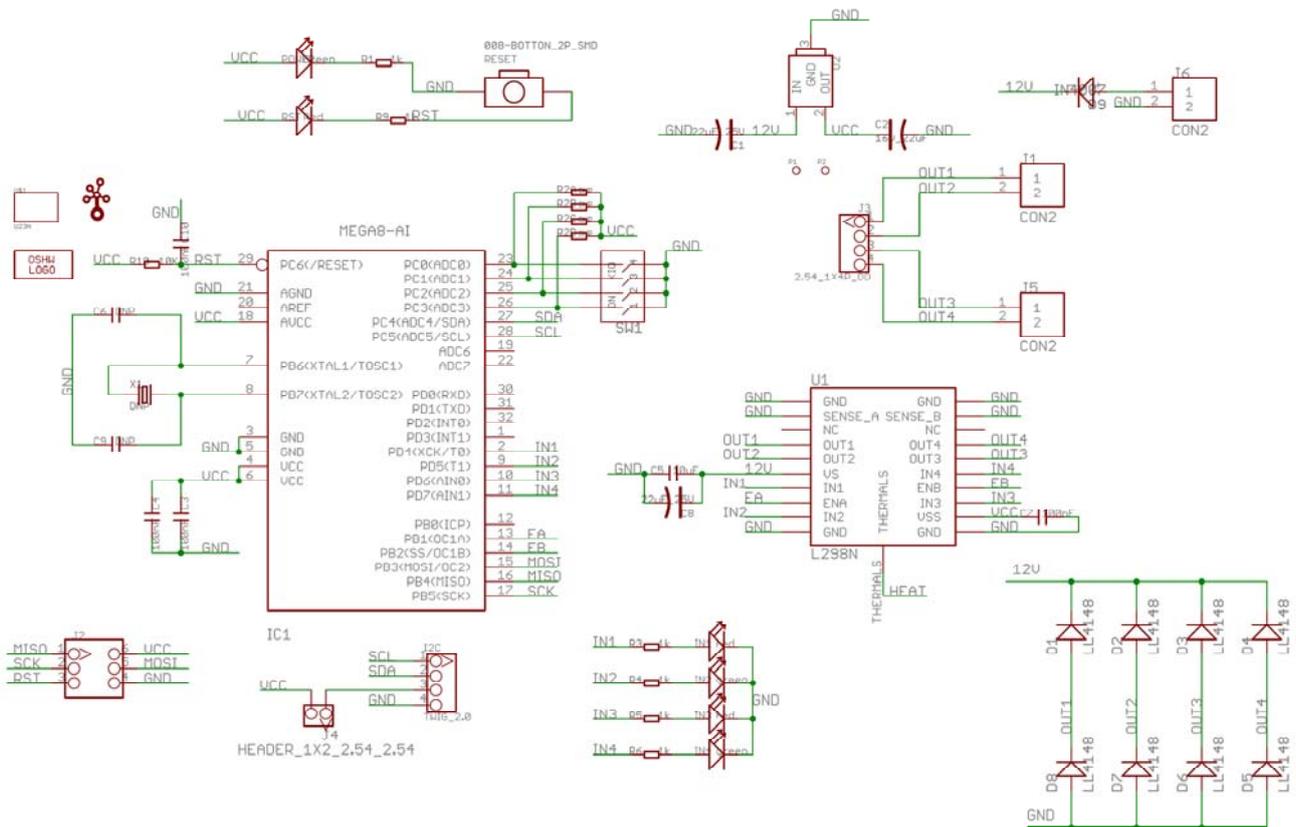


FIGURE 4 : I2C Motor Driver- Electronic schema