# Effect of PWM Duty Cycle on I2C Motor Driver Stability

## Problem

When new code is uploaded containing the Grove_I2C_Motor_Driver_v1_3 and a speed call is made to the I2C Motor Driver board the outputs are not correct. They were previously corrected with a manual reset to the I2C Motor Driver via the reset button or 6-AA battery holder cycling.

Grove_I2C_Motor...v1_3-master.zip

## Solution

The correct can now be done in software via the voidfrequence(unsignedchar _frequence) function. The fix should be implemented as follows.

**Grove I2C Motor Driver Fix (12/11/2018)**

```
#include "Grove_I2C_Motor_Driver.h"

// default I2C address is 0x0f
#define I2C_ADDRESS 0x0f

void setup() {
  Motor.begin(I2C_ADDRESS);
  Motor.frequence(F_490Hz);
}
```

## Hypothesis

## Experiment

Code below was upload. Voltage readings were taken at the M1 and M2 terminal blocks. One reading was taken just after uploading while the other was taken after a reset to the I2C Motor Driver.

## Procedure

### Testing Code

```
/*
 * motor_test.ino
 * Example sketch for Grove - I2C Motor Driver v1.3
 *
 * Copyright (c) 2012 seeed technology inc.
 * Website    : www.seeed.cc
 * Author     : Jerry Yip
 * Create Time: 2017-02
 * Change Log :
 *
 * The MIT License (MIT)
 *
 * Permission is hereby granted, free of charge, to any person obtaining a copy
 * of this software and associated documentation files (the "Software"), to deal
 * in the Software without restriction, including without limitation the rights
 * to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
 * copies of the Software, and to permit persons to whom the Software is
 * furnished to do so, subject to the following conditions:
 *
 * The above copyright notice and this permission notice shall be included in
 * all copies or substantial portions of the Software.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN
 * THE SOFTWARE.
 */

#include "Grove_I2C_Motor_Driver.h"

// default I2C address is 0x0f
#define I2C_ADDRESS 0x0f

void setup() {
  Motor.begin(I2C_ADDRESS);
  //Motor.frequence(F_490Hz);
}

void loop() {
  // Set speed of MOTOR1, Clockwise, speed: -100~100
  Motor.speed(MOTOR1, 50);
  // Set speed of MOTOR2, Anticlockwise
  Motor.speed(MOTOR2, -50);
  delay(5000);

  Motor.stop(MOTOR1);
  Motor.stop(MOTOR2);
  delay(2000);
}

// End of file
```

## Data

All measurements taken with:

- Arduino on USB +5 Volt power
- Arduino running at 16 MHz
- Voltage to I2C Motor Driver at 8.1 Volts
- M2 set at -50% speed
- M1 set at 50% speed

| Frequency (Hz) | Upload M2 (Volts) | Upload M1 (Volts) | Reset M2 (Volts) | Reset M1 (Volts) |
|---|---|---|---|---|
| 31372 | -.365 | .348 | -3.62 | 2.84 |
| 3921 | -1.7 | 2.85 | -3.63 | 2.88 |
| 490 | -3.62 | 2.85 | -3.63 | 2.84 |
| 122 | -3.93 | 3.35 | -3.61 | 2.79 |
| 30 | -4 | 3.35 | -3.62 | 2.84 |

## Analysis

The higher the frequency the lower the duty cycle is for the motor controller.

## Conclusion

490 Hertz is the correct value when using a Seeeduino v4.2 or thimbleduino v1.0.