Application Note

MCF83xx and MCT83xx EEPROM Read and Write Procedure

Sivabalan Mohan

ABSTRACT

The MCF83xx and MCT83xx devices are code-free, easy-to-use, highly configurable three-phase BLDC drivers with integrated motor control. These devices have an internal EEPROM to store the motor tuning and system level configurations like current limit, motor acceleration time, maximum speed, PWM switching frequency, supply overvoltage or undervoltage threshold, and so on. These configurations stored in the internal EEPROM are used to initialize the device on every power-up or wake-up. This application note explains the procedure for reading from and writing to this internal EEPROM.

Note

- MCF83xx refers to TI's sensorless FOC based 3-phase BLDC drivers including MCF8316D, MCF8315D, MCF8316C-Q1, MCF8315C-Q1, MCF8315C, MCF8315A, MCF8316A and MCF8329A.
- MCT83xx refers to TI's sensorless trapezoidal commutation based 3-phase BLDC drivers including MCT8316A, MCT8316A-Q1, MCT8315A and MCT8329A.
- MCF83xx and MCT83xx family of devices are referred to as MCx devices in this application note.

Table of Contents

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The MCx devices are highly configurable in terms of motor and system level parameters like acceleration time, current limit, fault response and retry time, fault (overcurrent/overvoltage/undervoltage) thresholds, GPIO configurations, control loop Kp, Ki values etc., These configurations are stored permanently in the internal EEPROM and used to initialize the device on every power-up/wake-up cycle.

Figure 1-1 shows the internal memory architecture of the MCx devices. Every EEPROM register has a corresponding mirror register in the RAM (shadow) region. On every power-up/wake-up, the device reads the EEPROM registers in to the corresponding shadow registers to initialize the device variables. The user can modify the shadow register values when the device is in idle state using I^{2}C write to specific shadow registers. These modifications in the shadow registers are lost on device power reset or sleep entry. In case the modifications need to be stored permanently, the user has to issue an EEPROM write command to copy the modified values from shadow registers to the EEPROM. The EEPROM contents can be read in to the corresponding shadow registers by issuing an EEPROM read command.

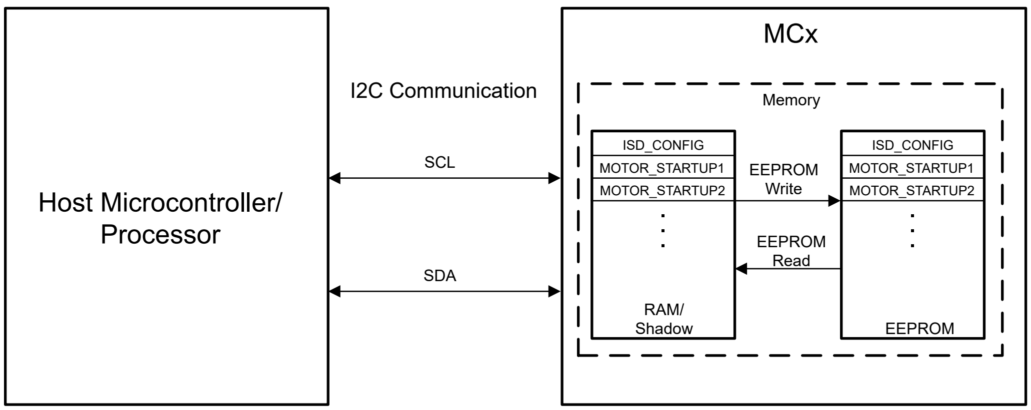

Figure 1-1. MCx Interface Circuit With Host Microcontroller or Processor

The EEPROM read and write commands can be issued by a host microcontroller over I^{2}C wherein the host acts as the primary device and MCx acts as the secondary device in the I^{2}C communication as shown in Figure 1-1. Refer to device data sheet for details on the I^{2}C protocol.

2 EEPROM Read and Write Procedure

2.1 EEPROM Read

In MCx devices, the EEPROM read procedure (also shown in Figure 2-1) is as follows,

1. Stop driving the motor to place the device in idle/standby state. In MCF devices, motor can be stopped by writing 0x8000000 to ALGO_DEBUG1 register located at 0xEC. In MCT devices, motor can be stopped by writing 0x00008000 to DEVICE_CTRL register located at 0xE8.
2. Issue a clear fault command to clear faults, in case the device encountered a fault during motor stop operation. In MCF devices, faults can be cleared by writing 0x30000000 to ALGO_CTRL1 register located at 0xEA. In MCT devices, faults can be cleared by writing 0x30000000 to ALGO_CTRL1 register located at 0xE6.
3. Issue an EEPROM read command by writing 0x40000000 into ALGO_CTRL1 register (located at 0xEA in MCF and at 0xE6 in MCT devices) to read the EEPROM data into the shadow/RAM registers located between 0x000080-0x0000AE.
4. Wait for 200ms after issuing an EEPROM read command.
5. After 200ms, read the ALGO_CTRL1 register value; a read-back value of 0x00000000 indicates that the EEPROM read is successful.

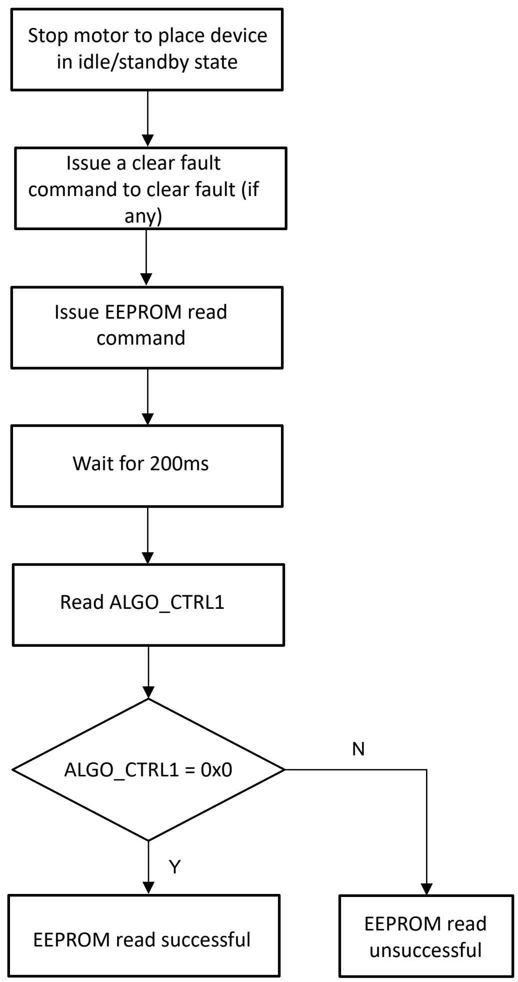

Figure 2-1. EEPROM Read Procedure

2.2 EEPROM Write

In MCx devices, the EEPROM write procedure (also shown in Figure 2-2) is as follows,

1. Stop driving the motor to place the device in idle/standby state. In MCF devices, motor can be stopped by writing 0x8000000 to ALGO_DEBUG1 register located at 0xEC. In MCT devices, motor can be stopped by writing 0x00008000 to DEVICE_CTRL register located at 0xE8.
2. Issue a clear fault command to clear faults, in case the device encountered a fault during motor stop operation. In MCF devices, faults can be cleared by writing 0x30000000 to ALGO_CTRL1 register located at 0xEA. In MCT devices, faults can be cleared by writing 0x30000000 to ALGO_CTRL1 register located at 0xE6.
3. Write the required values of the EEPROM registers to the corresponding shadow/RAM locations between 0x80-0xAE
4. Issue an EEPROM write command by writing 0x8A500000 into ALGO_CTRL1 register (at 0xEA in MCF and 0xE6 in MCT devices) to write the shadow/RAM registers' (located between 0x000080-0x0000AE) contents into corresponding EEPROM registers.
5. Wait for 750ms after issuing an EEPROM write command.
6. After 750ms, read the ALGO_CTRL1 register value; a read-back value of 0x00000000 indicates that the EEPROM write is successful.

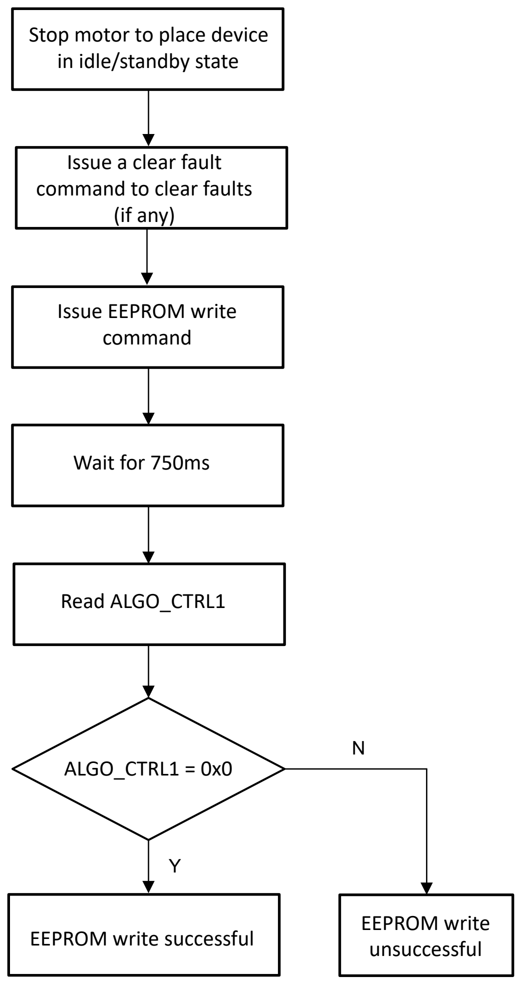

Figure 2-2. EEPROM Write Procedure

Note

EEPROM Write Do's and Dont's

- EEPROM write needs to be issued only when device is in idle state (not driving/powering the motor)
- Device supply (VM) needs to be ≥ 6V and all power rails (AVDD, FB_BK and DVDD) needs to stay within the data sheet specified operating limits throughout the EEPROM write process. An undervoltage fault on any power rail during the EEPROM write process can result in incomplete EEPROM write that can cause unexpected device behavior on subsequent power-up/wake-up.
- TI does not recommend writing to the EEPROM upon every power-up/wake-up due to aging/write cycle limitations on number of EEPROM writes. Refer to the device data sheet for maximum allowed EEPROM write cycles. Repetitive register setting changes can be done at shadow/RAM locations (and not written to EEPROM); only default configurations need to be written to EEPROM (at first power-up only).
- After a successful EEPROM write, a power reset is needed for all the EEPROM setting changes to take effect.

2.3 CRC

MCx devices support an optional CRC to verify the I^2C packet data integrity. An 8-bit CCIT polynomial (($x^8 + x^2 + x + 1$) with an initial value of 0xFF is used to calculate the CRC value. The CRC feature helps identify any data corruption in the I^2C packet. When CRC is enabled for a read operation, MCx calculates the 8-bit CRC for the entire packet (Target ID + Write bit, Control Word (3 bytes), Target ID + Read bit, Data Bytes (2/4/8 bytes)) and appends the CRC at the end of packet; the primary I^2C device (host) reading the data from MCx device can verify the CRC by comparing the received CRC (from MCx) against the calculated (by host) CRC. When CRC is enabled for a write operation, primary I^2C device (host) writing data into MCx device needs to append the calculated 8-bit CRC for the entire packet (from Target ID + Write bit, Control Word (3 bytes), Data Bytes (2/4/8 bytes)) at the end of I^2C data packet; I^2C packet can be ignored in case of wrong or missing CRC (when enabled).

Refer device data sheet for detailed explanation on I^2C packet communication using CRC.

3 Summary

TI recommends following the EEPROM read or write sequence as detailed in this application note. MCx devices read the EEPROM content for configuration or initialization of the motor control parameters - hence this is critical to follow the appropriate EEPROM read or write procedure for reliable device operation.

4 References

- Texas Instruments, *MCF8316D Sensorless Field Oriented Control (FOC) Integrated FET BLDC Driver*, data sheet.
- Texas Instruments, *MCT8315A High Speed Sensorless Trapezoidal Control Integrated FET BLDC Driver*, data sheet.