AN2502
Application note

Enhanced sensorless startup control of BLDC motors using ST7FMC

Introduction

The ST7FMC microcontroller is capable of controlling 3 phase brushless DC (BLDC) motors in position sensorless mode because of its powerful motor control macro cell. It is capable of monitoring zero crossings of all 3 back EMF (BEMF) voltages using a direct BEMF detection feature, where the entire motor terminal voltage is directly applied to an onchip zero crossing detector during the zero crossing period without any attenuation. Hence it is able to track the BEMF zero crossings even at very low motor speeds.

This direct BEMF detection feature helps to start the motor in auto commutated mode right from zero speed. In traditional methods, where the motor voltage signals are scaled down at the comparator input, it is necessary to step the rotor through a few rotations initially to let the comparator see sufficient levels of BEMF signals to detect zero crossings. The rate at which the rotor can be stepped through these initial rotations is highly influenced by load inertia, and hence a step timing table has to be created on the basis of load inertia. This can be a cumbersome task if the load inertia is not constant. With ST7FMC, because of the direct BEMF detection feature, this procedure can be avoided completely, thereby saving on development time as well as code memory space.

In any case, an alignment of the rotor to a known position is mandatory. At the end of alignment, if a commutation is performed by switching the next set of windings into conduction, the rotor is forced to turn in the direction of new step. If the winding current is set close to rated motor current at the end of alignment, then this step change will apply a rated torque on the rotor. Generally, this torque should be sufficient to kick the rotor to turn in the right direction at a reasonable rate (or speed), inducing a reasonable BEMF in the floating winding good enough for the zero crossing detector to identify a zero crossing as the full BEMF is available at the comparator input without any attenuation. Hence from the very first step, the device can be set in auto commutated mode like in a Hall position sensor based control.

Contents

1 Implementation

Alignment is done by passing current through motor windings corresponding to one of 6 steps of inverter. Let us call it STEP 0. At the end of alignment, the rotor comes to a parking position that will be a magnetically neutral position on the floating winding.

During alignment, bits DAC(MCRA) and SWA(MCRA) are set to zero and MCOMP is also set to zero. This has two implications. If DAC bit is zero, then writing to MPHST will only update the preload registers and become active only during a commutation (or C) event. With SWA bit being zero, a zero value on MCOMP will retain MTIM at zero and will create a C event for every clock effectively transferring associated preload registers to active registers with one clock latency. Hence, at the end of alignment, if the MPHST register is written with the next step (STEP 1), the inverter switching states will change to the new step (STEP1) during the next clock cycle and the rotor will turn in the direction of new step. The CPB(MCRB) bit must also be set appropriately corresponding to the new step to identify zero crossing for the new step when it happens. To ensure a good starting torque, the dutycycle may be selected to ensure rated current in the motor windings during this time.

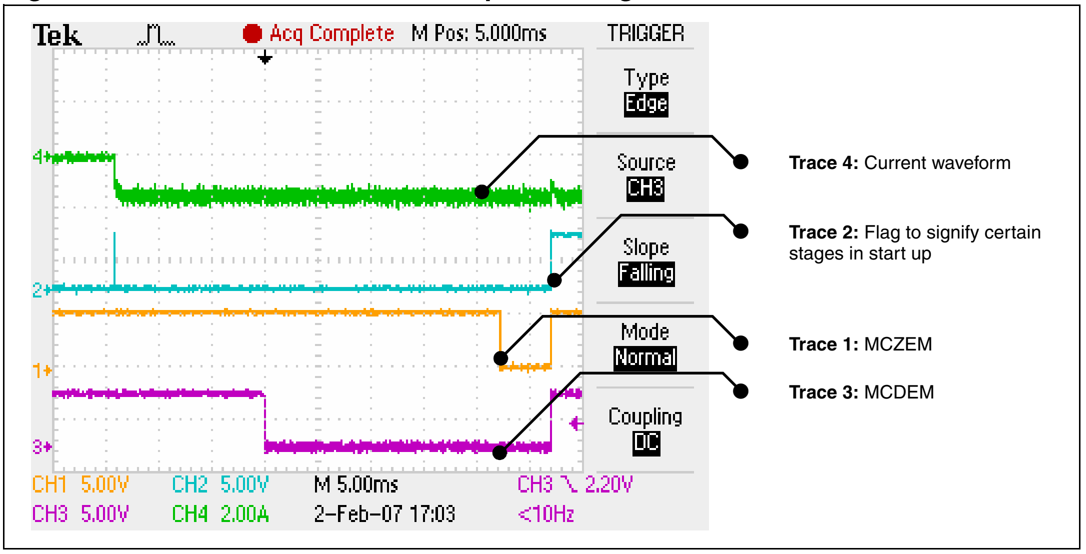
At this instant, the SWA(MCRA) bit should be set HIGH to put the device into auto-commutated mode. In this mode, commutation instances are estimated by hardware and its associated time count is loaded into MCOMP at the time of zero crossing (Z) events, based on zero crossing periods MZREG/ MZPRV and a coefficient MWGHT. Before setting SWA, since MCOMP was zero, MTIM was forced to remain at zero generating C events for every clock. Once SWA is set, the hardware watches out for a D event and the MTIM timer starts counting from zero in pursuit of D and Z events. After setting up SWA, the CIM(MIMR) bit should be set to enable C interrupts. At this point, a C interrupt takes place immediately because its flag was previously set repeatedly when SWA was zero, as explained above.

Inside the C interrupt routine, the MPHST and MCRB registers are updated for the next step (STEP2). These are only written into preload registers and are not active yet. If simulated demag option (SDM(MCRB)=1) is chosen, then MDREG must be updated inside the C interrupt routine. After the generation of a D event, the comparator hardware is set up to identify zero crossing of the floating winding. The active step on the inverter is STEP 1 yet, and if the motor current is good enough to produce a good starting torque to turn the rotor at a good rate, it will generate a reasonable BEMF on the floating winding that will pass through a natural zero crossing during this inverter step. This event is captured by the hardware and a Z event is generated. This event buffers MZREG into MZPRV and captures MTIM into MZREG and also loads MCOMP with a new time value, based on MZREG/MZPRV and MWGHT, to setup the next commutation event. When a match between MTIM and MCOMP happens, a C event is generated transferring contents from preload MPHST to active state setting up next inverter step (STEP 2). Updating of all relevant registers inside the C routine is repeated as before, and this C event is followed by D event and subsequently by Z event and thus the cycle repeats continuously. The sequence of events during starting is captured in *Figure 1*.

2 Experimental results

This concept is implemented on a starter kit to run an onboard 24V Ametek BLDC motor. The captured waveforms are given in *Figure 1* and *Figure 2*.

Figure 1. First two commutation steps after alignment



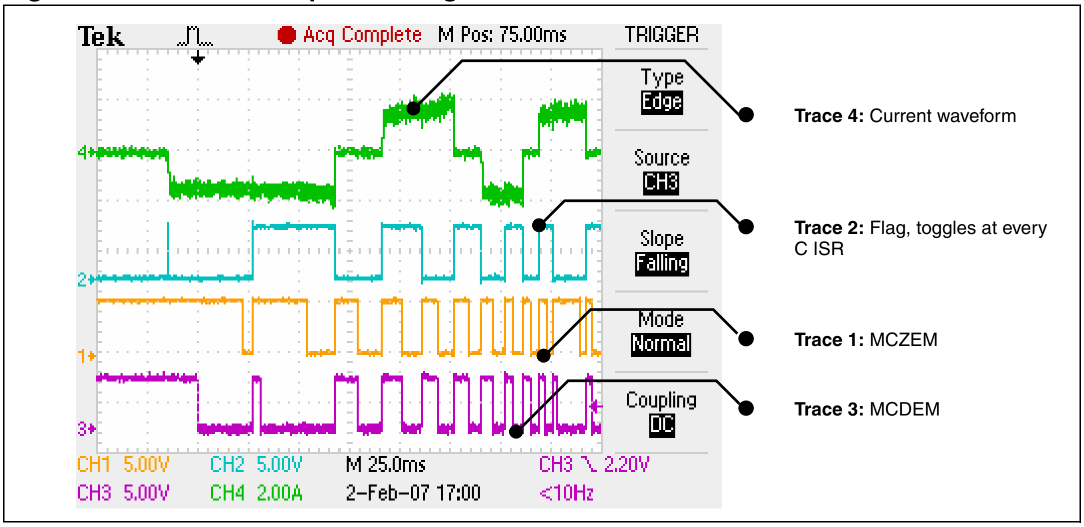
The edges of trace 2 have the following significances:

- Before first rising edge
  - end of alignement (at STEP 0)
  - DAC(MCRA) = 0, SWA(MCRA) = 0, MCOMP = 0
  - active step is STEP 0
- During first level HIGH
  - write MPHST corresponding to next step (STEP 1)
  - MCRB updated for STEP 1
  - SWA(MCRA) set
  - CIM(MIMR) set
  - active step is STEP 1
- first falling edge
  - first C interrupt execution
  - Inside C ISR, preloads of MPHST and MCRB updated for next step (STEP 2) and MDREG is updated (if simulated demag option chosen)
  - Notice Trace 3 (MCDEM) falling edge (D event instance) happening after this edge
  - Notice Trace 1 (MCZEM) falling edge (Z event instance) happening after D instance
  - active step is STEP 1

- second rising edge
  - second C interrupt execution
  - active registers get copied from preloads, first auto commutated step visible outside
  - active step is STEP 2
  - preloads of MPHST and MCRB updated for next step (STEP 3)

The pictorial details of the first few steps are shown in *Figure 2*

Figure 2. First few steps after alignment



3 Conclusion

The need to generate a forced start up stepping sequence and to tune its stepping rate is completely avoided. The waveforms shown in *Figure 2* reveal the excellent starting performance obtained with a low voltage (24V) BLDC motor that generates a very weak BEMF at start up. For higher voltage motors, such as 300V DC, this method will be even more reliable. This was verified by running a blower motor and a washing machine motor and the performance was found to be satisfactory. Most low cost sensorless control methods requires initial forced rotation of the rotor. But ST7FMC, because of its direct BEMF detection method, differentiates itself from the rest by this superior start up performance.

Appendix A Associated software

To verify this method on the starter kit, download and use the attached software *sup_sensrls.zip* file. All motor control modules are written in *motcon.c* and program start module is written in *main.c*. Control board hardware definitions are given in *pcboard.h*. To see the waveforms given in this application note, open *parameters.h* file and set macro TEST to 1, removing jumpers J15(13:14), J15(11:12) and W10. The evaluation details are as follows:

For test signals:

- J15 pin 14 --> MCDEM represented by trace 3
- J15 pin 12 --> MCZEM represented by trace 1
- W10 mid pin --> flag signal represented by trace 2

Control set up:

- W12 --> in Fixed mode
- POT1 --> sets current limit reference
- RV2 --> sets dutycycle
- SW1 --> On/Off command to start / stop motor

4 Revision history

Table 1. Document revision history

Date	Revision	Changes
02-Mar-2007	1	Initial release.