AN205199

F²MC-8FX/16LX/16FX/FR, All Series, BLDC Drive With The PPG

Brushless motors are meanwhile used in a wide variety of applications. Their durability, dynamic and silent run as well as their high torque and power per weight/volume make them a favorable motor for many fields formerly dominated by brushed DC or universal motors. Brushless permanent magnet motors are used in high-end servo drives as well as in low-cost fans. Also the demands for the used microcontroller and its software vary with the application, from high-performance control loops in servo drives to simple open-loop control in fans or simple electro-mechanical systems.

Contents

1 Introduction....1
2 BLDC basics....2
2.1 BLDC basics....2
2.2 BLDC with block commutation....3
3 Software examples....5
3.1 Example for 8FX....5
3.2 Example for 16FX....6
3.3 Example for FR81S....7
4 Appendix....8
4.1 Related Documents....8
4.2 Related software examples....8
Document History....9
Worldwide Sales and Design Support....10
Products....10
PSoC® Solutions....10
Cypress Developer Community....10
Technical Support....10

1 Introduction

Brushless motors are meanwhile used in a wide variety of applications. Their durability, dynamic and silent run as well as their high torque and power per weight/volume make them a favorable motor for many fields formerly dominated by brushed DC or universal motors. Brushless permanent magnet motors are used in high-end servo drives as well as in low-cost fans. Also the demands for the used microcontroller and its software vary with the application, from high-performance control loops in servo drives to simple open-loop control in fans or simple electro-mechanical systems.

In many applications, the dynamic requirements for the motor are not too high, and a brushless motor is chosen because of its silent run, durability and robustness. In some of these cases, the used microcontroller does not necessarily need special hardware resources to control such a motor.

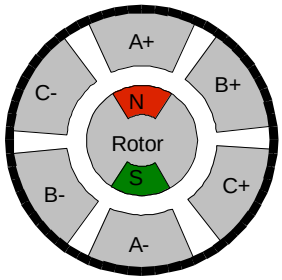
This application note will give a short introduction to the principles of the BLDC (Brushless DC) motor using Hall sensors and how it can be controlled using virtually any microcontroller with at least three or four external interrupt (or input capture) pins and one PPG channel. Also, in some applications a single MCU can control multiple motors, with different demands on dynamics. As an example, the drum of a modern direct-drive washing machine will work best with a sophisticated vector drive for efficiency and noise reduction, while the water pump might be equipped with a small BLDC running open-loop with block commutation. By using some hints shown in this application note, both can be controlled by a single microcontroller, which can also handle other control tasks.

2 BLDC basics

2.1 BLDC basics

The basic construction of a brushless DC motor bears some analogy to a usual brushed DC motor turned inside out. Now that the moving rotor contains the permanent magnets and the stator carries the phase windings, the mechanical commutator is replaced by an electronic one. This saves space in the motor and eliminates the problems caused by the commutator's brushes, like brush wear, abrasion and arcs. The lifetime of a BLDC motor is basically only limited by the bearings. The BLDC offers a better torque-to-volume ratio than its brushed ancestor, because of the saved room usually needed for the commutator. Additionally, the BLDC can dissipate heat much better because of the lower thermal resistance of the windings to the housing. Also, the complete motor can be sealed (e.g. for pumps) or even run directly in liquids. The absence of arc sources makes it ideal for explosive environments. With modern microcontrollers and power electronics, the electronic commutation of BLDC motors is easily possible. Most BLDC motors are equipped with three Hall sensors for rotor position feedback, displaced 120 el. degrees to each other, so that the combination of the Hall sensor signals yields the rotor position within 60° el. For more precise position measurement, e.g. optical encoders can be used.

Figure 1. Three-phase BLDC motor



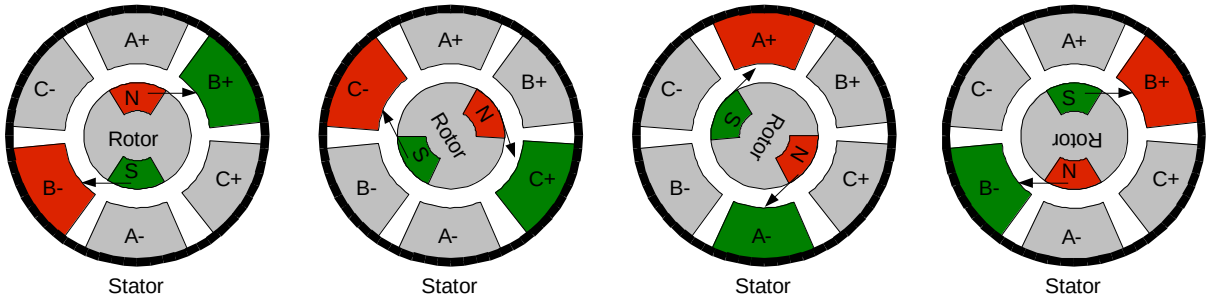
As the motor rotates, the permanent magnets induce a voltage in the motor phase windings, the so called back-EMF (Electro-Magnetic Force). In many BLDC motors, the back-EMF has a nearly trapezoidal shape as a result of their optimization for six-step commutation, while a 'pure' synchronous motor has a nearly sine-wave back-EMF shape. These motors preferably are driven by a sine wave inverter, e.g. using space vector modulation.

Most BLDC motors are three-phase motors, but also two-phase versions exist. These mainly are used for small pumps or blowers, e.g. in personal computers. Because of its close relationship to other permanent magnet motors, the BLDC is also known by other names, such as *Electronically Commutated* (EC) motor, PMSM (*Permanent Magnet Synchronous Motor*) or *Brushless AC* motor. Often, 'EC' or 'BLDC motor' means the block-commutated variant with trapezoidal back-EMF voltages, because its behavior is very similar to a brushed DC motor. PMSM or PMAC mostly refer to a motor with sinusoidal back-EMF which is driven by sine-modulated currents. This results in nearly no torque ripple and a very high performance capability.

2.2 BLDC with block commutation

The easiest way to drive a BLDC motor is called block commutation (also known as six-step or trapezoidal drive). It is easy to implement on many microcontrollers, especially when the motor has Hall sensors. In this case, every slope on a Hall sensor signal triggers the next commutation. A part of this sequence is shown below:

Figure 2. Rotating BLDC motor



Block commutation is often used in simple applications with moderate demands on dynamic behavior and acceptable torque ripple / noise. However, it is possible to optimize the motor construction for trapezoidal drive, reducing the noise and torque ripple.

Setting up a three-phase PWM for block commutation is quite simple and does not necessarily need special waveform generator functions. All active channels can use the same duty cycle, or the PWM is applied to the active high- or low-side transistor only. Complementary switching (and with it dead-time insertion) is not necessary in this application. The main task is to generate a PWM signal and distribute it to two of the six inverter transistors according to a commutation lookup table. This can be done in several ways:

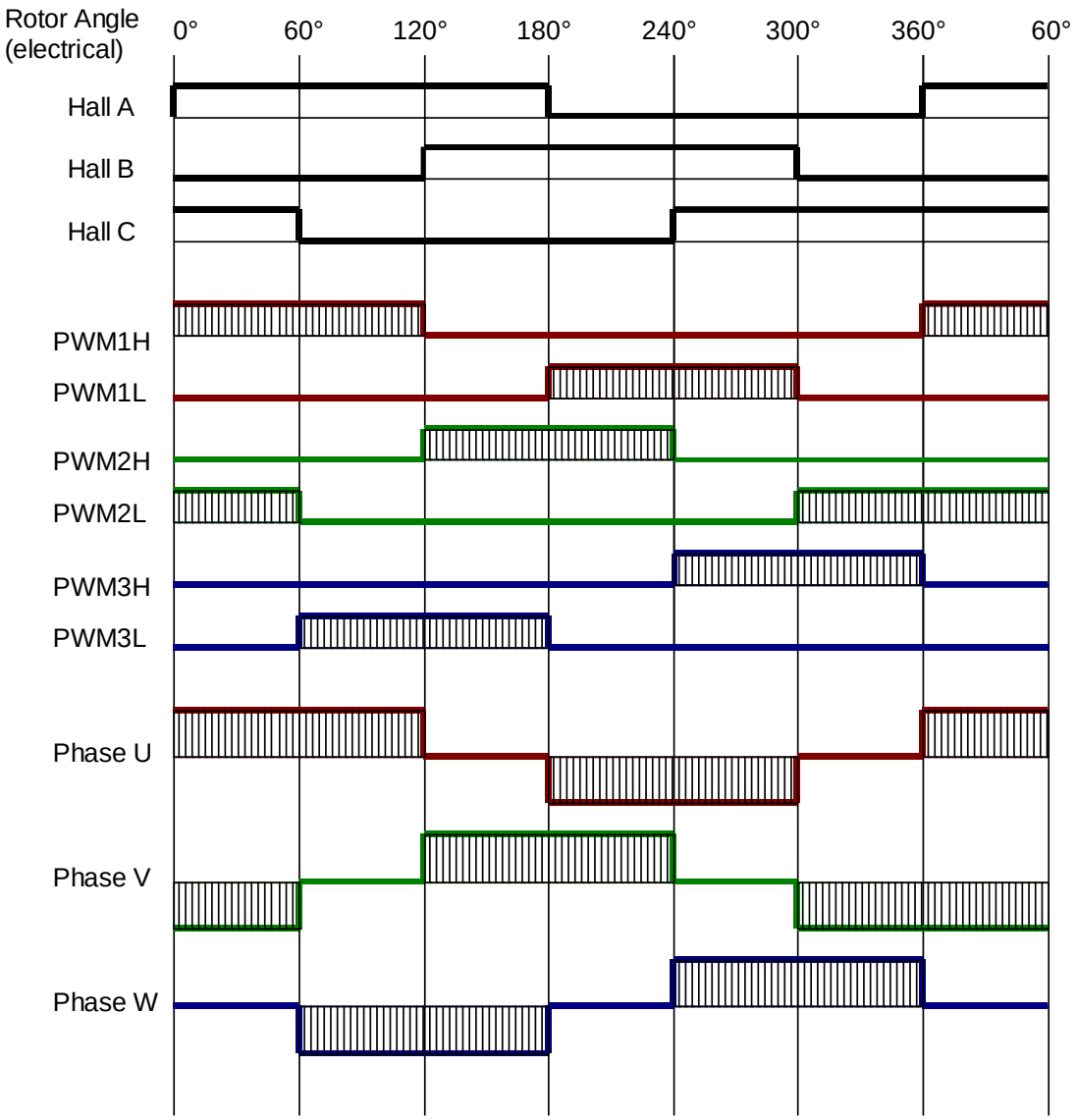
1. Usage of six PPGs or OCUs and enabling / disabling their output pins at every Hall sensor change
2. Usage of one PPG or OCU; In the corresponding interrupt service routine, the drive pattern for the inverter according to the current sector is applied (rising PPG slope) or cleared (falling PPG slope) on arbitrary I/O pins.
3. Usage of one PPG or OCU, six I/O-Pins and an external 6ch AND-Gate

The third possibility works well, but requires additional hardware and connections.

If the PPG unit of the MCU has the possibility to issue interrupt requests on duty match as well as on counter zero (e.g. Cypress 16FX), the second possibility is very easy to implement. The PPG output pin is then not needed. In case the PPG only offers an interrupt on counter zero or even no interrupt at all, the PPG output pin can be connected to an external Interrupt or Input Capture pin in order to generate the Interrupts for the PWM.

Because of the interrupt latency, there are some restrictions on minimum and maximum duty cycle, especially when using the variant with the external Interrupt. The minimum duty cycle is no limitation, because most BLDC motors will need about 20-40% duty to keep running anyway. Also the limited maximum duty is no real problem, since it can be raised to 100% by simply not clearing the pattern in the ISR, leaving only the ‘pure’ commutation.

Figure 3. Block commutation pattern



3 Software examples

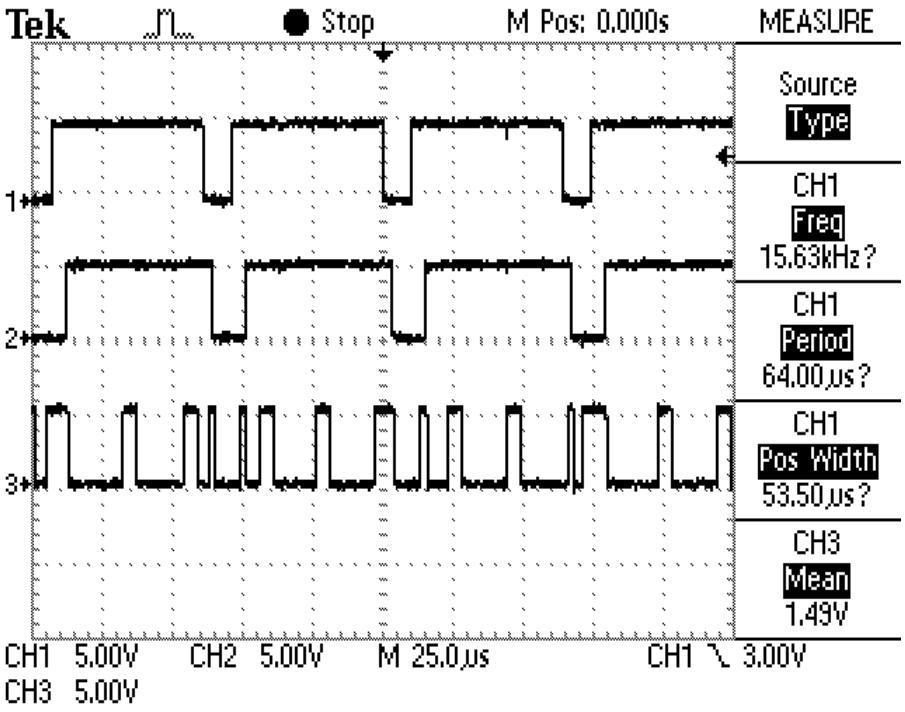
3.1 Example for 8FX

In the software examples packed with this document, a configuration example for the 8FX Series MB95F108 ('Bassoon') Flash MCU used on the CONCERTO-Kit is included. The example uses a single 8-/16-bit PPG channel to generate a right-aligned PWM. The PPG zero interrupt service routine is used to turn all inverter transistors off. The ISR of an external interrupt connected to the PPG output is used to detect the rising edge of the PPG and turn on the correct transistors for each sector using a lookup table. For a left aligned PWM, the PPG output could be inverted and the external interrupt would be configured for falling edge. Then, the PPG zero interrupt would switch on the outputs, while the match interrupts would clear them. For this application, the PWM type is arbitrary.

The PPG itself is configured as 16-bit PPG with a period setting of 0x100. Since the duty cycle in the example only reaches 0xff, it is assured that edges always occur to trigger the external Interrupt.

As soon as the duty cycle becomes too high (relative to the IRQ latency), the PPG ISR does no longer switch off the outputs, resulting in 100% output duty cycle, independent of the PPG duty cycle. The following figure shows the PPG output, one phase PWM output and the overall IRQ load, which remains nearly constant independent of the motor speed.

Figure 4. 8FX CPU usage during BLDC drive



It can be seen that the MCU has about 30-40% IRQ load (including the ADC interrupts), so that there is still time for other tasks.

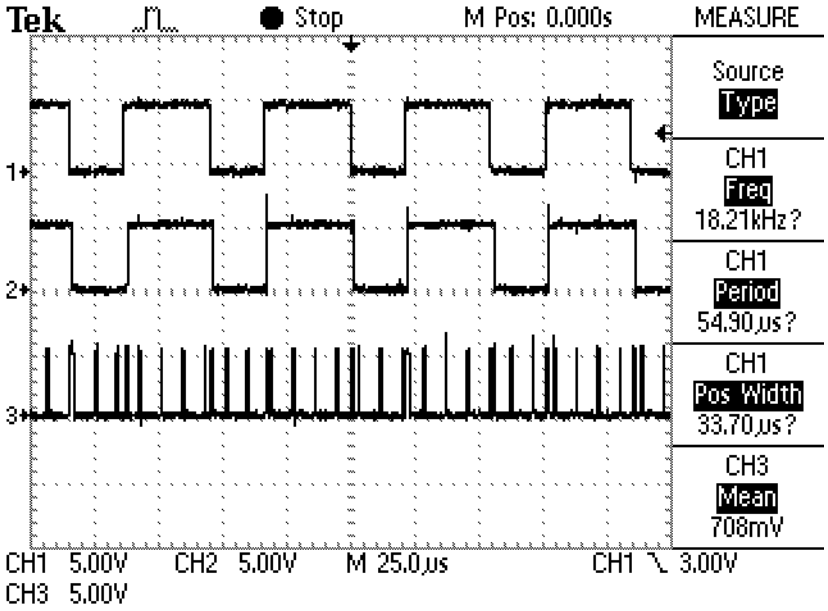
3.2 Example for 16FX

Since the PPG unit of the 16FX Series can issue interrupt requests for both PPG zero and duty match, no external connection to an interrupt or input capture is needed. Instead, to distinguish whether the PWM output shall be turned on or off, the actual timer value is compared to the duty cycle of the PPG. Since the PDUT registers are write-only, the duty cycle has to be stored in a variable to be used in the ISR:

```
__interrupt void ppg1_isr (void){
  if ((ppg_duty<=PPG_MAX_DUTY)&&(PTMR1>ppg_duty)) DRIVE_PORT = ALL_OFF; // off
  else DRIVE_PORT = active_drivers;              // on
  PCNL1_IRQF=0; // clear interrupt flag
}
```

Due to the higher clock speed of the 16FX Series, also the overall MCU load is reduced, as well as the latency between PPG slope and PWM output and thereby the available duty cycle range. Note that most interrupts in the figure below are ADC interrupts.

Figure 5. 16FX CPU usage incl. ADC



See the software example packed with this document for the complete example source code.

3.3 Example for FR81S

The FR81S Series is a MCU with special motor control functions. The Waveform Generator together with the PPG can control a three-phase motor using various methods, from block commutation to space-vector modulation. It can handle the necessary dead-time insertion for complementary PWM as well as fault protection and the synchronization between the ADC units and the PWM. Even though for example the MB91520 Series has only one Waveform Generator unit to drive a single motor, but the MB91580 Series supports two Waveform Generator units.

Since the FR81S Series a highly efficient loopback control is achieved by using a high-speed, dedicated 12-bit A/D converter and Resolver-to-Digital Converter (RDC). Dedicated hardware generates the required motor control parameters (such as the electrical angle's sine and cosine values) to accurately detect motor current and position at high speeds, enabling precise motor control waveform generation. An IEEE-754-compliant Floating-Point Unit enables the MCU to process vector conversion and PID control calculations rapidly, using data from the 12-bit A/D converter and RDC.

The MB91580 Series use a dedicated embedded resolver sensor interface to control the three-phase inverter, resulting in optimal motor controllability. This enables the system designer to achieve higher performance while lowering the system cost.

4 Appendix

4.1 Related Documents

- MB95100 Series Hardware Manual
- MB96300 Series Hardware Manual
- MB96600 Series Hardware Manual
- MB91520 Series Hardware Manual
- MB91580 Series Hardware Manual

4.2 Related software examples

- The newest version of these software examples can be found at:
  Software and Drivers
- Additional information to other Cypress products can be found at:
  http://www.cypress.com/cypress-microcontrollers

Document History

Document Title: AN205199 - F²MC-8FX/16LX/16FX/FR, All Series, BLDC Drive with the PPG

Document Number: 002-05199

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	MKEA	03/02/2007	CHa First release
*A	5078010	MKEA	01/08/2016	Converted Spansion Application Note “MCU-AN-300020-E-V10” to Cypress format
*B	5868867	AESATMP9	08/30/2017	Updated logo and copyright.
*C	6052824	NOFL	02/02/2018	Updated links. Updated Sales page.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2007-2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.