Application Note, V1.0, Oct 2006

AP08026

XC866

Brushless DC Motor Control with Hall Sensors
Using Infineon 8-bit XC866 Microcontroller

Microcontrollers

Never stop thinking

Edition 2006-10-20

Published by
Infineon Technologies AG
81726 München, Germany

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. INFINEON TECHNOLOGIES HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND (INCLUDING WITHOUT LIMITATION WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

AP08026		
Revision History:	2006-10	V1.0
Previous Version:	none	

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com

Table of Contents **Page**

1 Introduction

The common DC motor has a permanent magnet, 3 coil windings and rotating mechanical switches called commutator for each coil (Figure 1a). Current flows through two of the coils whenever a voltage is applied at the brushes. This current interacts with the magnetic field of the permanent magnet and produces a torque and this torque moves the rotor. When the motor rotates, the brushes will automatically come in contact with the commutator of a different coil causing the motor to continue its rotation. It will turn faster if the voltage is increased and it will produce more torque if the magnetic field is also increased.

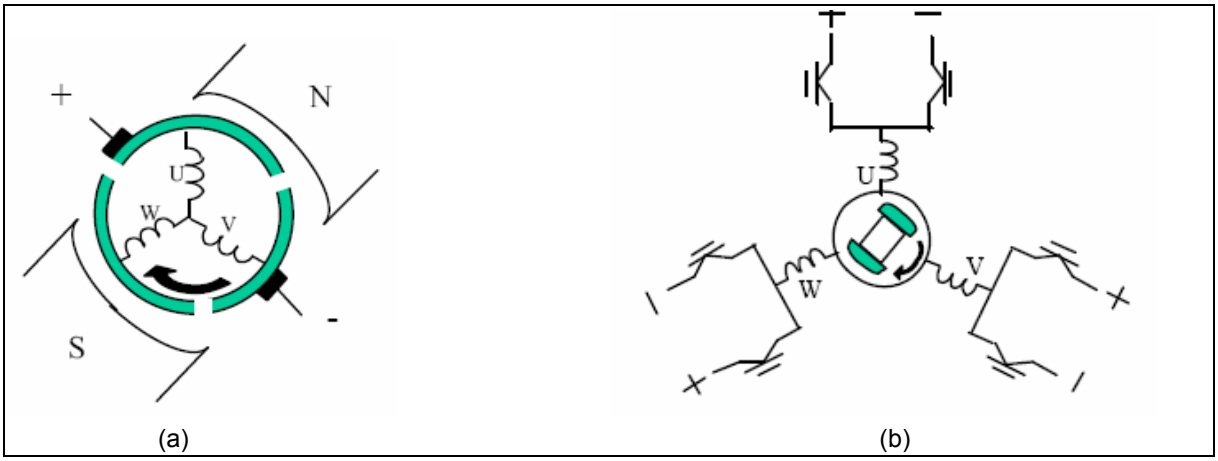

Figure 1. a) Brushed DC motor, b) Brushless DC motor

On the other hand, in a brushless DC motor (Figure 1b), the brushes and commutators are replaced with electronic switches (with MOSFET or IGBT semiconductor). Furthermore, the rotor is now the magnet and the stator is now made up of the coils. One of the main advantages of brushless DC motors is their robustness against obsolescence (due to the absence of brushes), another major benefit is the high efficiency and high torque of a brushless DC motor. Also, brushed DC motors require maintenance, e.g. to service carbon brushes and commutator. Another major problem with a brushed DC machine is the possibility of brush burnout in the event of an overload or stall condition. Moreover, the use of electronic switches allows voltage and speed control through the use of frequency modulation, namely pulse width modulation. The PWM can be easily generated by a microcontroller.

Also, for a brushed DC motor, the coil energized is dictated by the brushes and commutators. However in a brushless DC motor, without the brushes and commutators, the switching of the electronic switches has to be guided by the rotor position to ensure that proper commutation occurs and the motor rotates. Thus, the detection of the rotor position is crucial in order for the BLDC motor to work properly. Generally, rotor position sensors (usually Hall-Effect devices) are used to determine the rotor position. These sensors are placed 120° apart and outputs a ‘1’ if it senses the North Pole. Figure 2 shows a simplified representation of a single pole pair BLDC motor with hall sensors (H0, H1, H2). The arrow represents the magnetic field of the rotor while the coil windings (A,B,C) represents the stator.

Published by Infineon Technologies AG

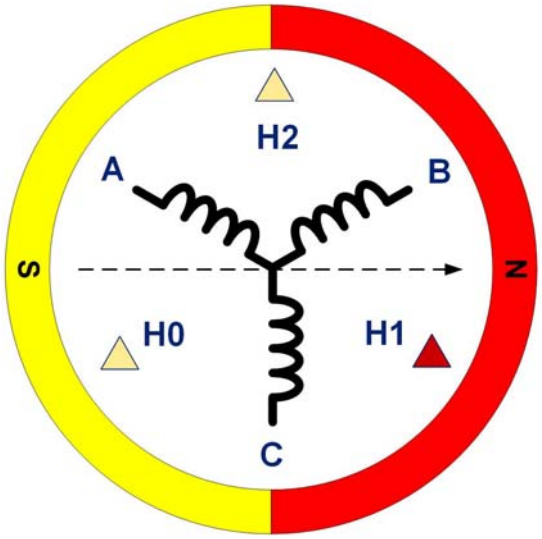

Figure 2. Single pole pair BLDC motor with hall sensors

At every 60° of rotation, the Hall sensors change its state and each combination of hall sensors state represents a specific rotor position. Shown in Figure 3 are the signals generated by the hall sensors and the corresponding rotor position for each combination of the hall sensors.

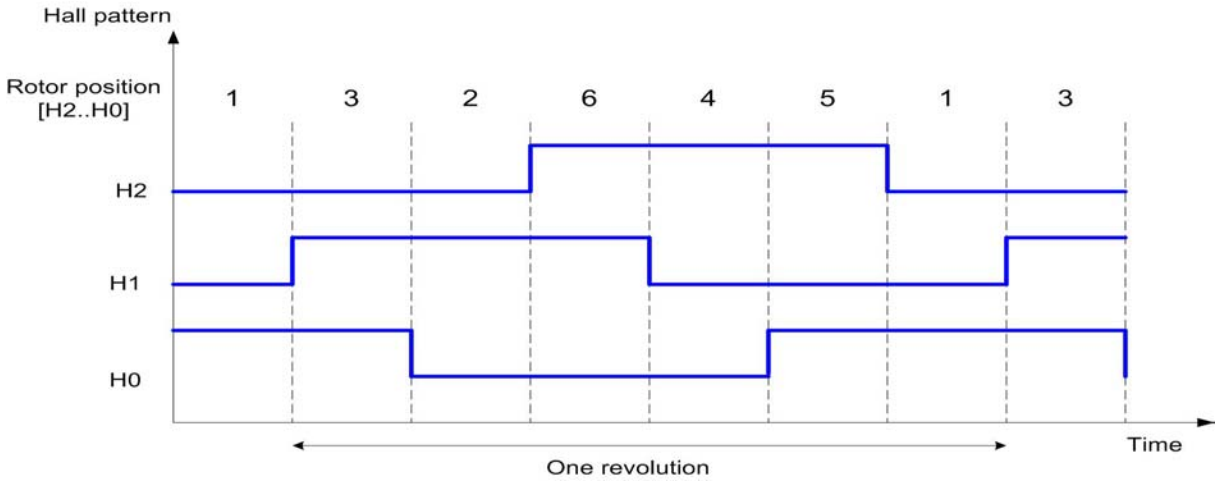

Figure 3. Hall Pattern

The generated Hall signals are fed back to the XC866 microcontroller for further processing. The next part will be on the discussion of the hardware and software implementation of BLDC motor control using the XC866 microcontroller.

2 Hardware Implementation with XC866 Microcontroller

Figure 4 shows the structure of a three phase BLDC motor driver using hall sensors. The XC866 microcontroller has a Capture/Compare Unit 6 (CCU6) that takes the hall signals as input and generates the switching patterns for the stator field according to the rotor position obtained from the hall signals. The CCU6 (Figure 5) also generates the PWM which is used for speed or torque control. The 3-phase driver IC takes the CCU6 switching patterns as input and provides the output signals for the 3-phase inverter. It is also capable of implementing short-circuit current protection, over-/under-voltage protection and over-temperature protection by making use of the CTRAP functionality of the CCU6. The CTRAP will force the CCU6 outputs into a passive state and no active modulation is possible, immediately stopping motor operation.

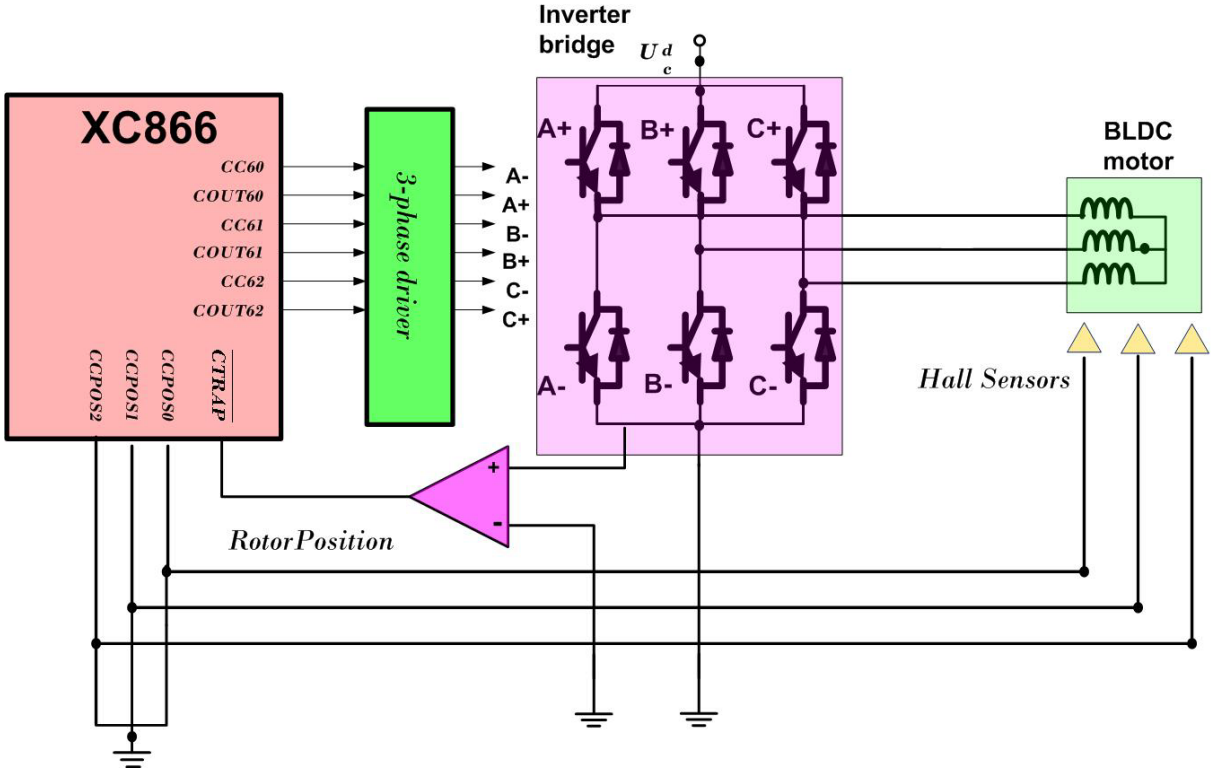

Figure 4. BLDC motor driver with hall sensors

Infineon OptiMOS® N-channel power transistors make up the inverter switches which powers the corresponding phases of the BLDC motor. There is always a pair of switches which controls one motor phase (e.g. A+ and A- control phase A) – this is called half bridge configuration. There is a high side switch which is connected to the V+ DC-rail voltage and the opposite one which is called low side switch which connects the coil to the GND. Current through the motor will flow if one high side switch is on and another low side switch is on. The current will flow from V+ through one coil in positive direction and through another coil in negative direction to GND. This is called block commutation mode where one phase is always inactive whereas current will flow to the other two phases. Also, a bridge shortcut can occur if the high side and low side switches are on at the same time. This will burn the inverter immediately because very high current will flow very fast (within a few microseconds) and this situation have to be avoided (through hardware protection). Thus timer 12 of the CCU6 has a dead time control module which will delay the switching of either a high side or a low side switch so that a short circuit does not occur.

2.1 Capture/Compare Unit 6 (CCU6)

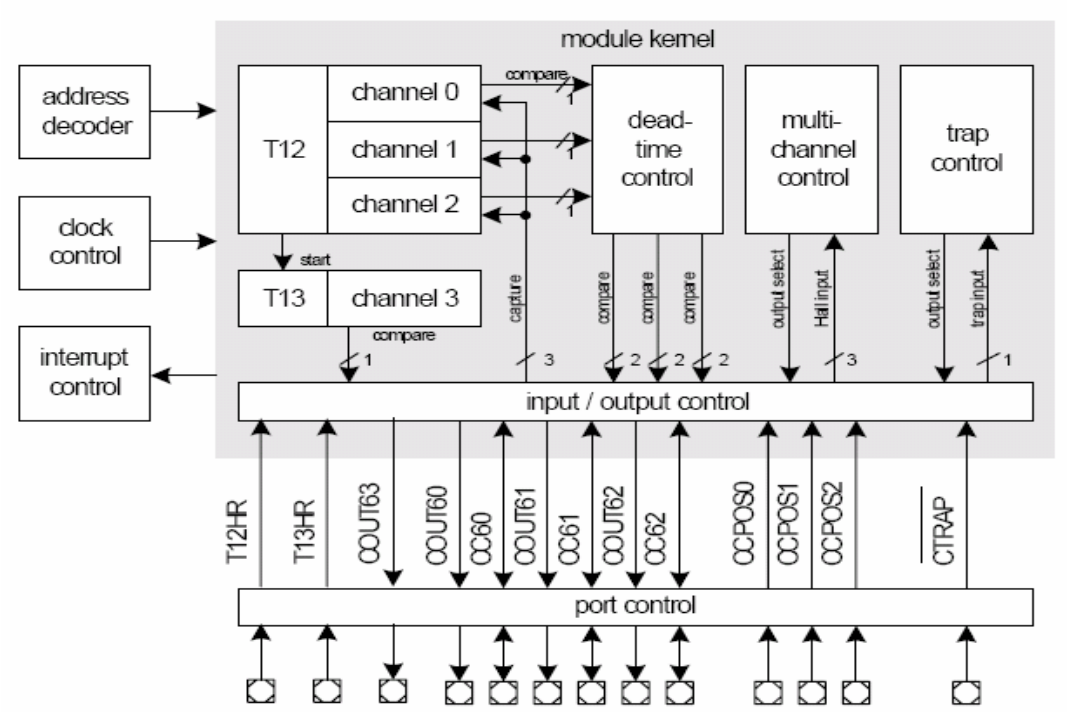

Figure 5. CCU6 Functional Block Diagram

In BLDC motor control, the switching patterns which define stator conducting status depend on the pattern of the Hall signal inputs (an indicator of rotor position). There is a strong correlation between them. Because of different machine types, the modulation pattern for driving the motor can vary. Therefore, it is beneficial to have wide flexibility in defining the correlation between the Hall pattern and the corresponding modulation pattern. The CCU6 offers this by having two registers. One of them (MCMOUTSH) contains the actual Hall pattern (CURHS) and the next expected Hall pattern (EXPHS), and the other register (MCMOUTSL) defines the corresponding output pattern (MCMPS).

The control can be implemented via three basic steps: sampling of Hall pattern, Hall pattern and output pattern software update, and shadow transfer of updated patterns upon pre-defined event (new patterns take effect).

Sampling of Hall pattern

The Hall pattern (on CCPOSx input pins as in Figure 5) is sampled with the module clock f_{CCU6}. In the hall sensor mode (mode MSEL6x = 1000B), the dead-time counter DTC0 is used (if bit DBYP is 0) as a hardware noise filter to suppress spikes on the Hall inputs. In case of a Hall event, the DTC0 is automatically reloaded and started by the hardware and generates a delay between the detected event (change of Hall input signals) and the sampling point. After the counter value of 1 is reached, the CCPOSx inputs are sampled (without noise and spikes) and are compared to the current Hall pattern (CURH) and to the expected Hall pattern (EXPH). If the sampled pattern equals to the current pattern, it means that the edge on CCPOSx was due to a noise spike and no action will be triggered (implicit noise filter by delay). If the sampled pattern equals to the next expected pattern, the edge on CCPOSx was a correct Hall event, and the bit CHE is set which can induce an interrupt.

Hall pattern and output pattern software update

After the actual Hall pattern is sampled at input pins CCPOSx, the corresponding output pattern of CCU6 should be generated in order to control BLDC motor. The CCU6 generates the switching pattern thru its output channels CC6x and COUT6x (x=0,1,2). It provides two independent 16-bit timers (T12, T13); timer 13 generates the PWM which modulates DC rail voltage and hence motor speed. This PWM is delivered by the COUT6x output channels. Looking back to Figure 4, the COUT6x control the high side switches while the CC6x control the low side switches. Table 1 shows the rotor position and the corresponding CCU6 output channels and inverter switches, and Figure 6 shows the CCU6 output signals together with the hall pattern.

Published by Infineon Technologies AG

Rotor Position [H2...H0]	Next Position	COUT62 C+	CC62 C-	COUT61 B+	CC61 B-	COUT60 A+	CC60 A-	Hex Equivalent
1	3	0	1	0	0	1	0	0x12
2	6	1	0	0	1	0	0	0x24
3	2	0	0	0	1	1	0	0x06
4	5	0	0	1	0	0	1	0x09
5	1	0	1	1	0	0	0	0x18
6	4	1	0	0	0	0	1	0x21

Table 1 Commutation Table

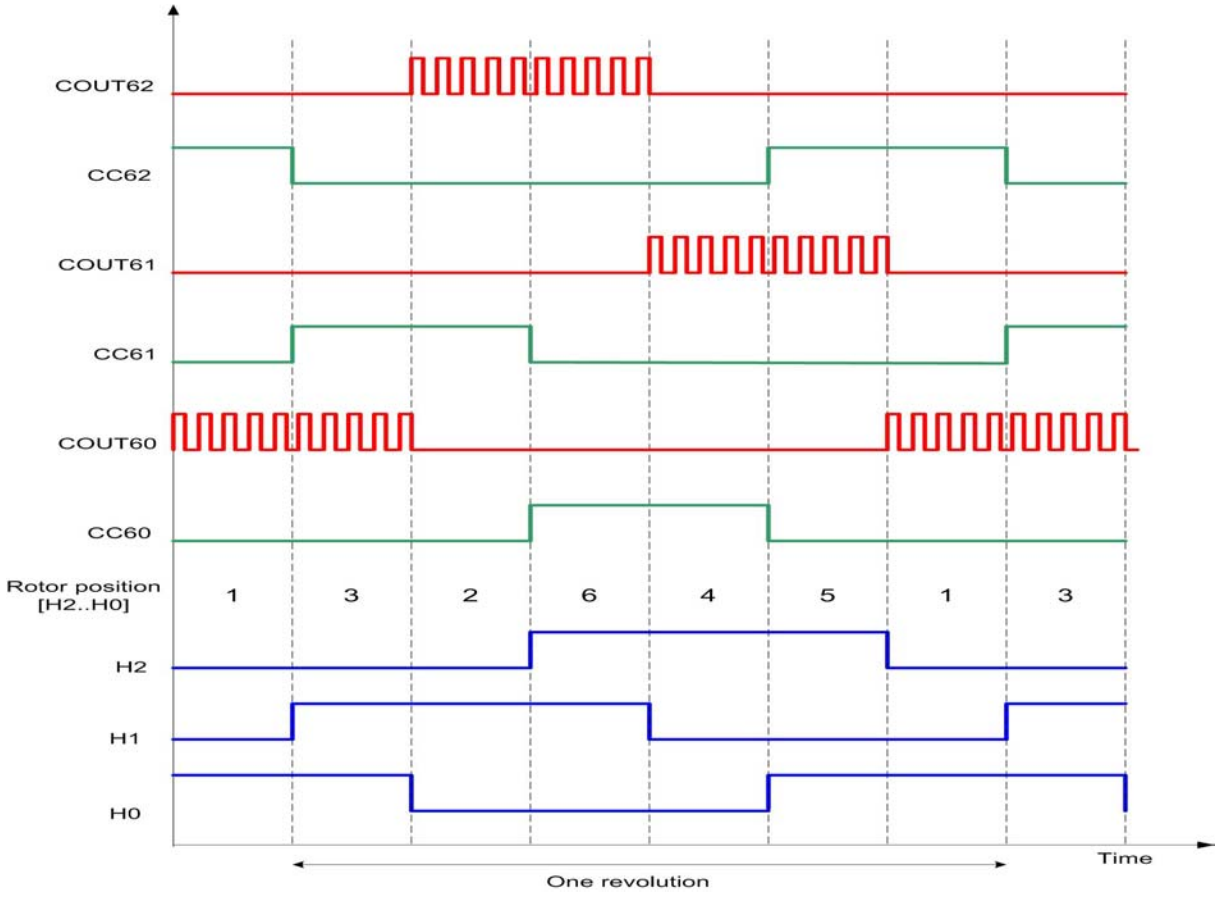

Figure 6. CCU6 output signals and Hall pattern

These output patterns and rotor position sequence are realized in the program by storing them in a table as shown below[1].

```
unsigned char HallPatt[] = { 000,  013,  026,  032,  045,  051,  064};

unsigned char revHallPatt[] = {  000,  015,  023,  031,  046,  054,  062};

unsigned char OutputPatt[] = {0x00, 0x12, 0x24, 0x06, 0x09, 0x18, 0x21};
```

Note 1: Hall pattern and output pattern shown in the table are defined according to the characteristics of the BLDC motor used in this application. Different BLDC motors may have various patterns and should be tuned case by case.

The HallPatt[] and revHallPatt[] holds the rotor position pairs. The 2nd digit represents the current rotor position while the 3rd digit represents the next rotor position. The rotor position pairs are loaded in the MCMOUTSH (Figure 7a) register. The pair that is loaded in the register depends on the current Hall signals[H2…H0]. For example, if the hall signals are [010] = 2, then the expected rotor position is 6, thus the 6th rotor position pair in HallPatt[] which is 064 is loaded into MCMOUTSH. On the other hand, the MCMOUTSL (Figure 7b) will store the expected CCU6 output pattern from the OutputPatt[] table. Again, the value to be loaded into MCMOUTSL depends on the hall signal. Using earlier example with the hall signal indicating rotor position 2, then the expected rotor position is 6 thus the 6th entry in the OutputPatt[] which is 0x21 is loaded into MCMOUTSL.

Shadow transfer of updated patterns

Although Hall pattern and output pattern are updated via software, they don't take effect immediately. A mechanism named *Shadow Transfer* will synchronize them with pre-defined event. This is because usually PWM outputs are used to drive high-voltage or high-current applications, and should be synchronized with certain hardware event due to safety reason. To implement this mechanism, some special function registers come in pairs: a shadow register and an actual one. Writing operation targets shadow registers and not directly to the actual registers, while the read access targets the registers actually used.

In this application, MCMOUTSH and MCMOUTSL are the shadow registers of MCMOUTH and MCMOUTL respectively. The values in the MCMOUTS registers are transferred to the actual register, MCMOUT, when there is a correct hall event or other events such as:

- a T12 period-match while counting up (T12pm)
- a T12 one-match while counting down (T12om)
- a T13 period-match (T13pm)
- a T12 compare-match of channel 1 (T12c1cm)

The transfer can also be requested by software by setting the corresponding shadow transfer request bit: bit STRMCM for register MCMOUTSL, and bit STRHP for register MCMOUTSH. By using this, the update takes place completely under software control.

Once the updated values in shadow registers are transferred to the actual registers, the next set of rotor position pairs and CCU6 output pattern are loaded into the MCMOUTL/H registers. After that, the MCMOUT controls the CCU6 output channels and these output channels decide which of the inverter switches are activated.

MCMOUTSL
Multi-Channel Mode Output Shadow Register ,Low Byte **[Reset value: 00_H]**

7	6	5	4	3	2	1	0
STRMCM	-	CC62	COUT62	COUT61	CC61	COUT60	CC60
w	r	rw					

(a)

MCMOUTSH
Multi-Channel Mode Output Shadow Register ,High Byte **[Reset value: 00_H]**

7	6	5	4	3	2	1	0
STRHP	-	CCPOS2	CCPOS1	CCPOS0	Expected Rotor Position		
w	r	rw			rw		

(b)

Figure 7 (a) MCMOUTSL register, (b) MCMOUTSH register

3 Evaluate Your Own BLDC Motor Control with Hall Sensor

Consistent motor operation can be achieved with the BLDC motor using hall sensors. This application note has presented some key features of Infineon's 8-bit XC866 microcontroller and how the CCU6's unique features are able to achieve high performance motor control. The BLDC motor control (with Hall Sensor) techniques presented in this application note can be implemented using the following hardware available from Infineon now:

1. Infineon XC866 Microcontroller Starterkit (Innovator Kit for XC800 Family)
2. Infineon Low Voltage Motor Driver Board (MDB LV45G v1.1)
3. BLDC motor (BL3056-18-028)
4. Power Supply - I/P: AC 100-240V 50/60 HZ 0.3 A, O/P: DC 24V 2.1A (Meanwell S-50-24)

Published by Infineon Technologies AG

Additionally, the folder **HOT2_BLDC_Hall_Sensor_ Advanced** contains all the DAvE and Keil reference source files needed for the operation of the BLDC motor using Infineon's 8-bit XC866 microcontroller. The software provided was written in C for the Keil Compiler using DAvE and comes with step-by-step guides on setting up your own BLDC motor control. All references codes and guides can be easily migrated to other Infineon microcontrollers in the XC800 family for immediate evaluation too.

Additionally, Infineon's Autocode generator, DAvE configures all the peripherals of the XC866 microcontroller with ease, saving many hours of work. Although no speed or torque control loop was implemented in this application note, however a simple PI controller can be easily implemented and should provide robust regulation.

Kindly refer to the software attached with this application note for further evaluation on how Infineon's low-cost high performance 8-bit microcontrollers allows you to achieve your demanding design requirements. Please contact your local distributor should you wish to obtain your own motor control training kit complete with hardware and reference algorithm to perform an evaluation or allow our experts to contact you on how we may assist you in your designs needs.

4 Other Relating References

You may wish to refer to the following application notes to learn more about Infineon's 8-bit XC866 microcontroller and Brushless DC motor control.

1. Application note **AP08019** - Sensorless Brushless DC Motor Control
2. Application note **AP08018** - Start-up Control Algorithm for Sensorless and Variable Load BLDC Control Using Variable Inductance Sensing Method