FAIRCHILD
SEMICONDUCTOR®

www.fairchildsemi.com

AN-8201

FCM8201 Three-Phase Sine-Wave BLDC Motor Controller

Introduction

FCM8201 is a BLDC motor controller with a three-phase sine-wave / square-wave drive; the detection position is at 120° rotor magnetic pole. With just a small number of peripheral components, it can control BLDC motors on a stand-alone basis. In coordination with microcontroller, it can also be used for complicated motor control applications. It is suitable for motor control for various products, such as fans, water / oil pumps, tooling machines, etc.

Features

- Sine-Wave PWM Generator Uses “Space Vector Modulation” Algorithm; Super-Silent Motor Drive Properties
- Supporting both Sine-Wave and Square-Wave Motor Drive Schemes; Applicable to a Wide Range of Products
- Built-in Working Clock Generator; No Need for Crystal or Ceramic Resonator, Cost Competitive
- Built-in Error Amplifier; Supporting Product Application Controlled by Constant Torque
- Supporting Serial Port Interface (SPI); Allowing Coordination with Microcontroller to Enable Advanced Control Applications
- Programmable Current Leading Phase Correction; Enhancing Efficiency of Sine-Wave Drive
- Programmable Dead Time Adjustment; Reducing Switching Loss of Power MOSFETs
- Providing Sound Measures for Driver Circuit Protection; including Over-Voltage (OVP), Over-Current (OCP), Over-Temperature (OTP), Hall Signal Error, etc. Enhancing System Reliability

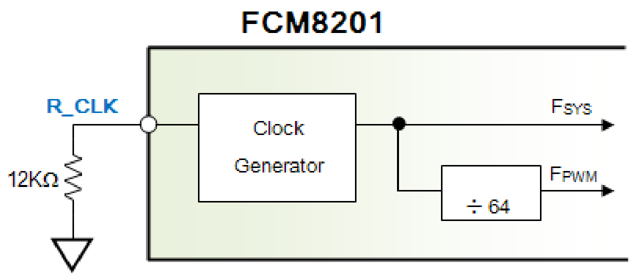


Figure 1. Clock Generator

System Working Clock and PWM Frequency

FCM8201 has a built-in clock generator, eliminating the need for additional crystal and capacitors. With a resistor (R_CLK) for selecting working frequency, it can produce working frequency for system clock and PWM frequency.

The system working clock can work at the frequency range from 960KHz to 1920KHz. The frequency of PWM is 1/64 of the system working clock. When the system working clock is set as 960KHz, the frequency of PWM equals 960KHz ÷ 64 = 15KHz; if the former is set as 1920KHz, the latter equals 30KHz. Therefore, the working range of PWM frequency falls between 15KHz and 30KHz.

If FCM8201 uses R_CLK=12KΩ, the system working clock should be 1280KHz with PWM frequency being 20KHz. Therefore, to have FCM8201 working with different system working clocks, the following equation can be used for deciding resistance of R_CLK:

$$R\_CLK = 12K\Omega \times \frac{1280KHz}{f_{SYS}} \qquad (1)$$

To use PWM frequency for determining system working frequency, the following equation can be used for selecting the resistance of R_CLK:

$$R\_CLK = 12K\Omega \times \frac{20KHz}{f_{PWM}} \qquad (2)$$

PWM Duty Control

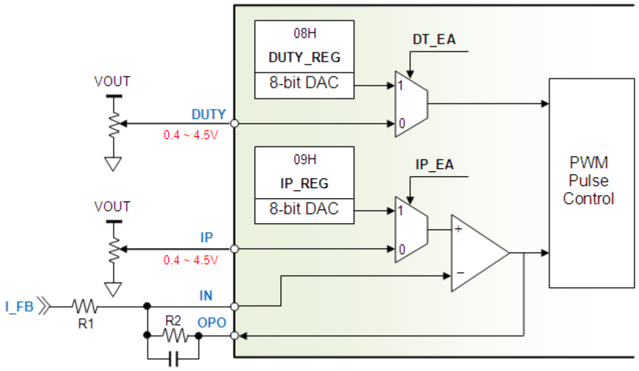


Figure 2. Block Diagram of PWM Duty Control

PWM duty can be determined by two control paths. The primary control path is determined by the DAC-converted voltage level from external DUTY pin or internal DUTY_REG register. In general application conditions, this control path is used to control setting the motor speed.

For the other control path, the DAC-converted voltage level from external IP pin or internal IP_REG register is compared with that of IN pin by error amplifier, then the amplified voltage of the difference obtained from the comparison determines the path. Generally this control path can serve as the control setting for torque circuit or the limit setting for maximum motor output. A basic application circuit is shown in Figure 2. The gain of the error amplifier equals R2 ÷ R1. Application of the error amplifier is described in more detail in the Current Feedback section.

The input voltage range of both control paths is 0.5V ~ 4.5V. They work simultaneously; the one with lower voltage level determines the actual PWM duty output. Those less than 0.5V are zero duty and 4.5V is the full duty. Therefore, if there is no need to use both PWM duty control paths, set the unused control path at 4.5V to avoid influencing the operation range of the other control path.

Square-Wave PWM Motor Drive

FCM8201 supports two square-wave PWM commutation modes: PWM-PWM and PWM-ON.

FCM8201 uses square-wave drive of PWM-PWM commutation to start the motor running. In stand-alone mode, so long as the motor running status complies with the conditions required for sine-wave drive, FCM8201 automatically uses sine-wave drive mode.

For operation in SPI control mode, after starting of motor, the microcontroller can be used for free selection: maintaining PWM-PWM commutation or changing to PWM-ON commutation based square-wave drive or using sine-wave drive.

PWM-PWM Commutation of Square-Wave Drive

With the 120° rotor magnetic pole position detection, each N-S performs 360° cycles on three Hall sensors, so only six signal changes occur. As shown in Figure 3, as long as the corresponding current directions are provided on the three-phase windings of stators based on six signal changes, rotating magnetic fields is generated to attract rotors to rotate. Each type of Hall signal corresponds with one PWM output type; there are six different PWM output types in 360°, with a commutation occurring every 60°. Therefore, it is also called six-step square-wave drive.

The output of the six-step square wave is shown in the right waveform of Figure 3. In all cases, one of these three phases outputs PWM continuously every 120° (two steps), while the other two phases get conducted at different 60° low side. Each phase outputs PWM every two steps, so it's called as PWM-PWM commutation mode.

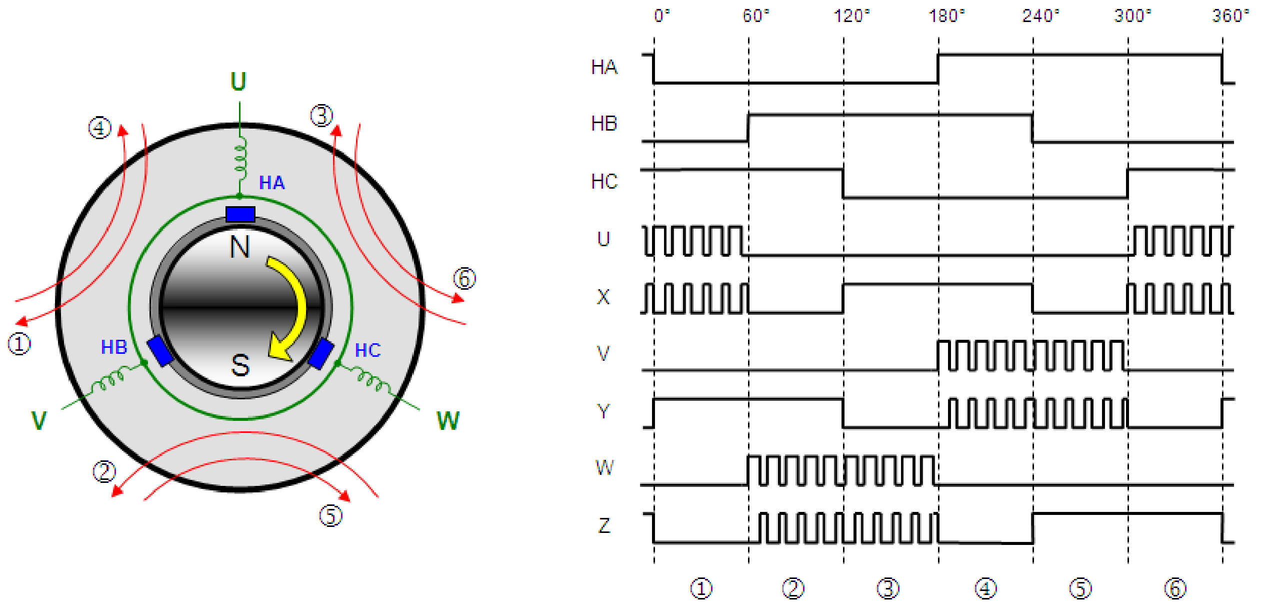


Figure 3. PWM-PWM Commutation at CW/CCW=1

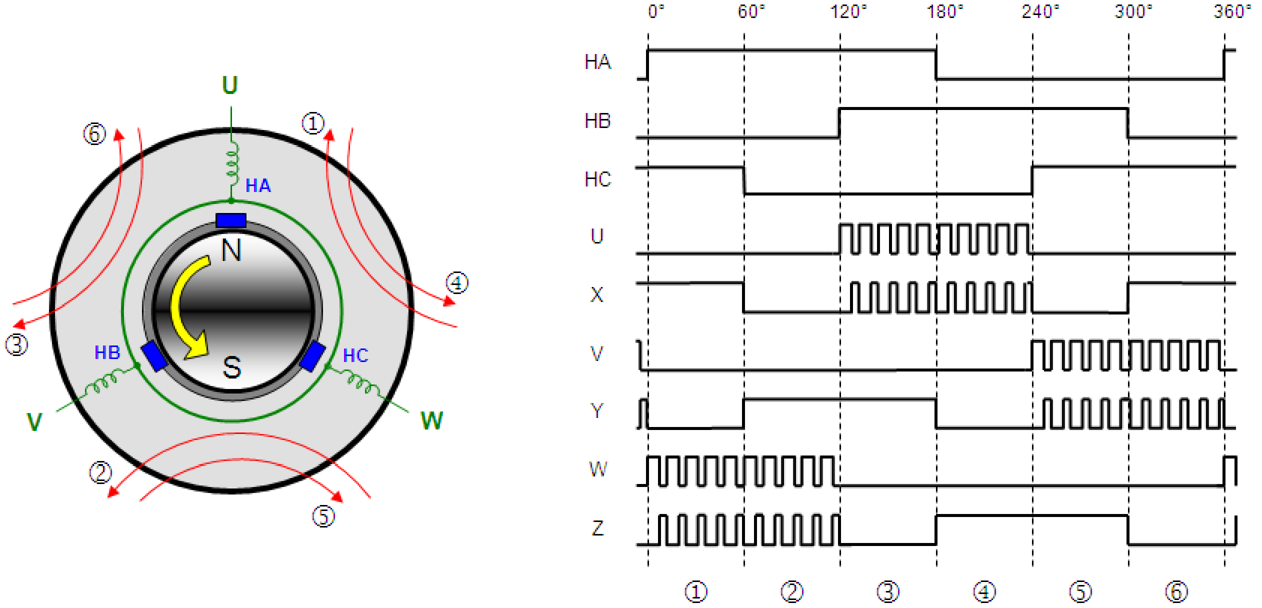


Figure 4. PWM-PWM Commutation at CW/CCW=0

PWM-PWM commutation mode greatly facilitates use of high-side driver IC in combination with driver circuit whose high and low sides both use N-channel MOSFET or IGBT as the driver circuit for the motor. This is because the high side does not keep being conducted during any commutating period and, when the high side closes, its low side with the same phase is conducted and, as a result, synchronous rectifying is enabled to improve efficiency. At this moment, the bootstrap circuit of the high-side driver IC has the chance to charge, supplementing energy for the driving MOSFET.

Although PWM output using these commutation features a somewhat simple driver circuit that doesn’t require concern about turn-on failure or partial conduction of the high-side MOSFET; between two-step continuous PWM output, negative current may return to the power source side when the low sides of the other two phases exchange conduction, as shown in Figure 5. Such negative current is one of the major noise sources for square-wave drive.

When U-phase outputs PWM, at the instant low-side conduction of W-phase switches to low-side conduction of V-phase; when both the low side of MOSFET and PWM of U-phase close, polarities of inductance of U-phase and W-phase are reverse. Thus the energy having been stored in inductance becomes negative current IW-U and returns to the power supply side via the built-in diode of high-side MOSFET of W-phase. In this way, negative current is generated.

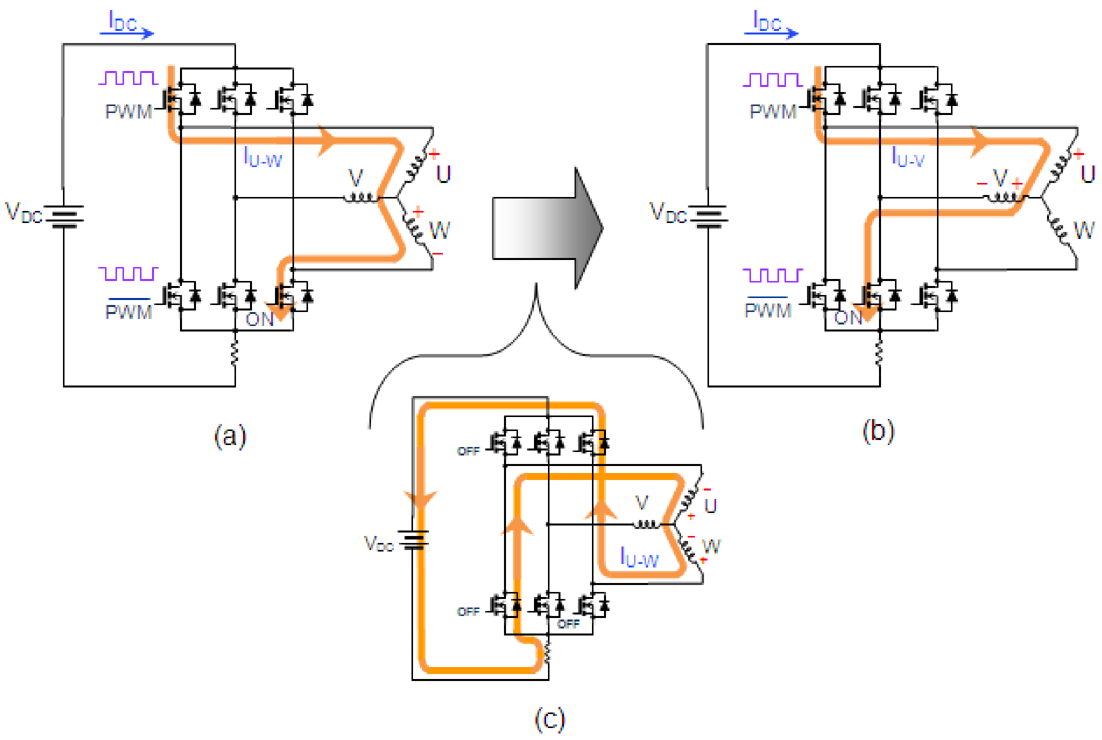


Figure 5. Commutation Between U-W and U-V

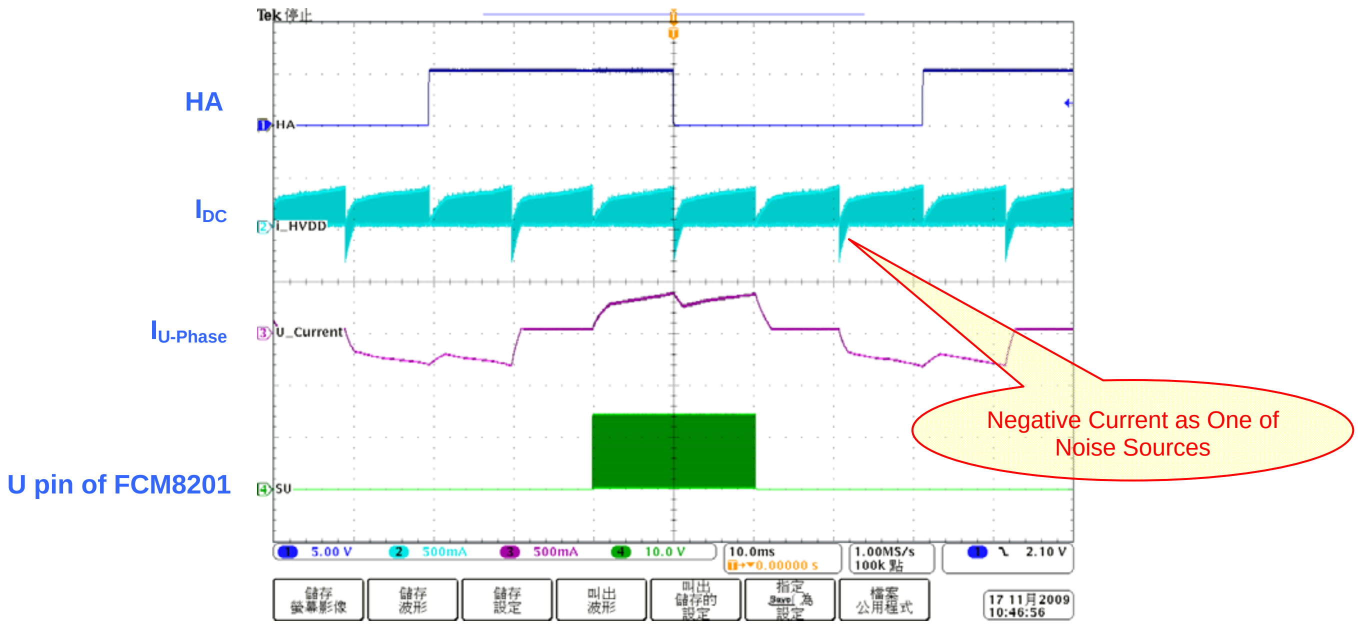


Figure 6. Current Waveform of PWM-PWM Commutation

PWM-ON Commutation of Square-Wave Drive

To avoid generation of negative current in PWM-PWM commutation and to reduce noise of square-wave drive, FCM8201 provides a PWM-ON commutation output mode. As Figure 7 shows, the difference between PWM-PWM and PWM-ON is this: when Step-2/4/6 switches to Step-3/5/1, the high-side MOSFET of the phase that has been outputting PWM directly changes to fully conducted status; meanwhile, the low side of one other phase takes it place to output PWM. In this way, inductance from the three-phase windings, which gives rise to reverse polarity and consequently generates reverse current, does not come into being. Moreover, the current direction and magnetic field consistent with PWM-PWM is maintained.

As in output of each phase in such commutation mode, the high side outputs PWM first and, after commutation to the next step, the high side changes to fully on status, it is called PWM-ON commutation mode.

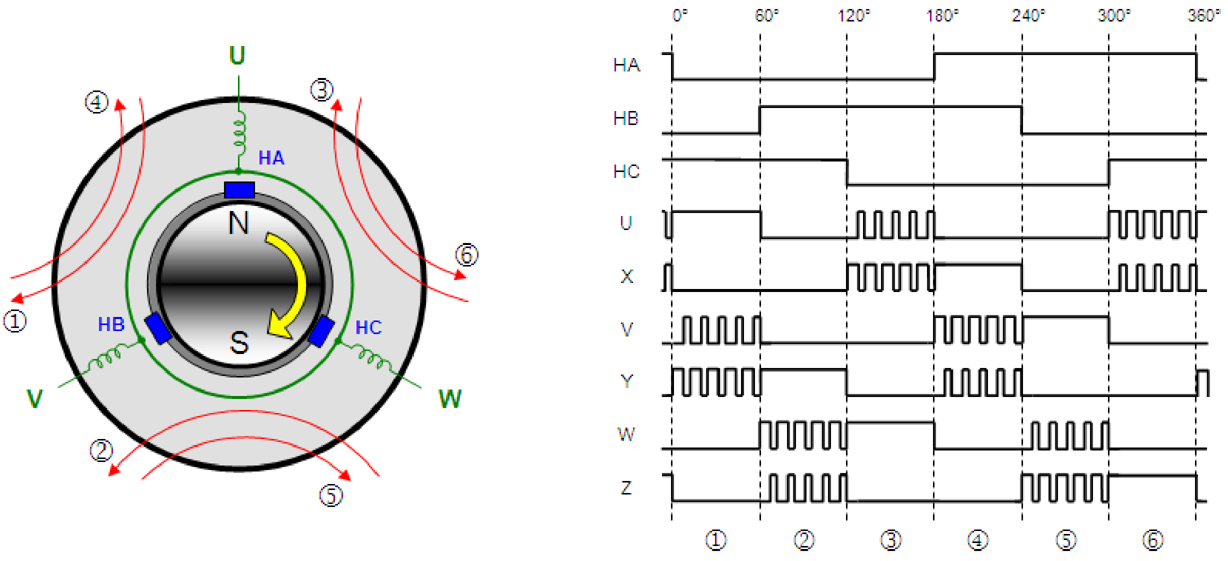


Figure 7. PWM-ON Commutation at CW/CCW=1

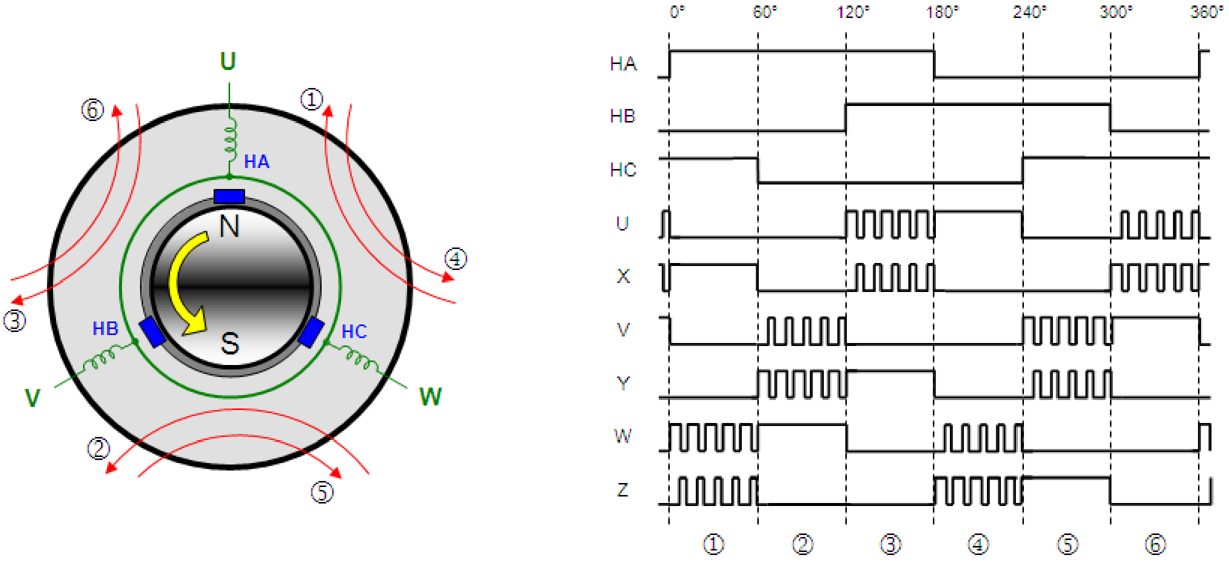


Figure 8. PWM-ON Commutation at CW/CCW=0

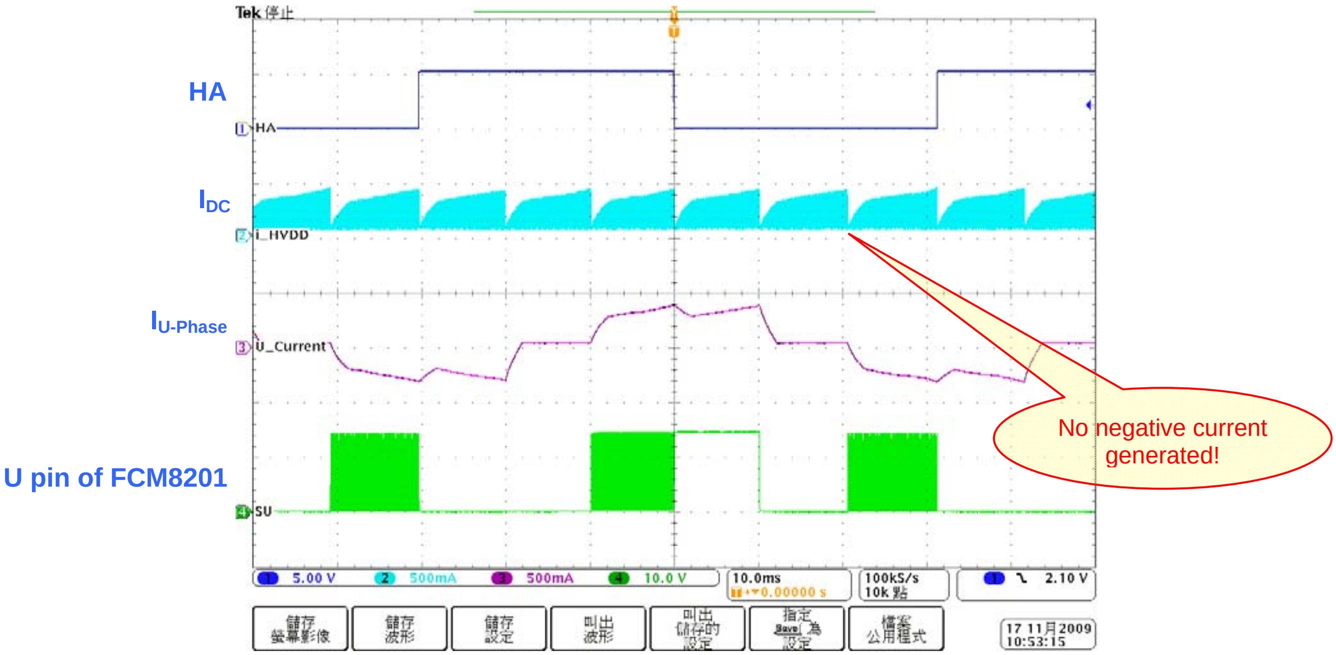


Figure 9. Current Waveform of PWM-ON Commutation

Sine-Wave PWM Motor Drive

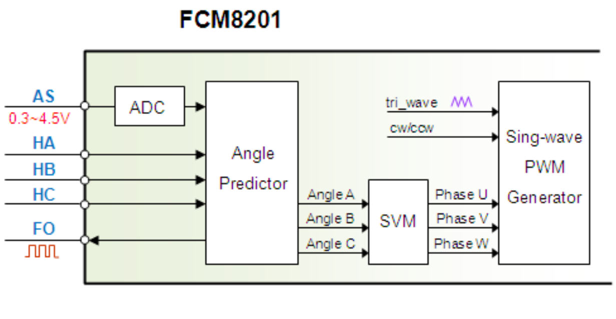


Figure 10.Block Diagram of Sine-Wave Generation

Sine-wave drive IC of FCM8201 mainly consists of three modules: angle predictor, space vector modulation (SVM), and sine-wave PWM generator.

Angle predictor uses a change of each 60° electrical angle of Hall input signal (HA/B/C) as a benchmark for prediction. When the working frequency of the system is 1.28MHz, the operable Hall signal frequency range is divided into three speed ranges that can be selected by two bits of ANG_REG register's ARNG[1:0], as shown in Table 1.

Only when the angle predicted by the angle predictor is consistent with Hall signal does FCM8201 allow switching to sine-wave drive mode. After starting, FCM8201's default frequency range is from 0.8 to 80Hz. In Stand-Alone Mode, based on a calculation of two-poled motor, the revolving speed range available to sine-wave drive falls between 48 ~ 4800 revolutions.

Table 1. Predict Range vs. Motor Revolution

ARNG[1:0]	Hall Freq.	Revolution of 2 Poles Motor
0, 0	0.8 ~ 80Hz	48 ~ 4800rpm
0, 1	0.4 ~ 40Hz	24 ~ 2400rpm
1, x	3.2 ~ 320Hz	192 ~ 19200rpm

For motors other than two-poled motors, the following equation can be used for calculating motor revolution available to sine-wave drive:

$$RPM = \frac{Hall\_Freq. \times 60}{Poles \div 2} \tag{3}$$

where poles is the number of motor poles.

There are three essential conditions for FCM8201 to enable sine-wave drive: 1. result of angle prediction; 2. voltage level of AS pin or ANG[4:0]; 3. voltage level of DUTY pin or DUTY[7:0]. Only when all the above-mentioned three conditions are met at the same time can FCM8201 automatically switch to sine-wave drive. In SPI control mode, there are three additional control bits (SIN_MAU, SIN_EA and PMOD) which are capable of forcing switching to sine-wave or square-wave drive without being subject to the above-mentioned three conditions. The detailed operation status is shown in Table 2.

Table 2. Sine-Wave Active / Inactive Conditions

Angle Predict	AS Pin / ANG[4:0]	DUTY Pin / DUTY[7:0]	SIN_MAU (b7/SVM_REG)	SIN_EA (b6/SVM_REG)	PMOD (b7/PWM_REG)	Sine-Wave
Hit	> 0.3V	> 0.75V	0	X	0	Active
Miss	> 0.3V	> 0.75V	0	X	0	Inactive
X	X	X	1	1	0	Active
X	X	X	X	X	1	Inactive
X	X	X	1	0	0	Inactive
X	X	< 0.65V	0	X	0	Inactive
X	< 0.2V	X	0	X	0	Inactive

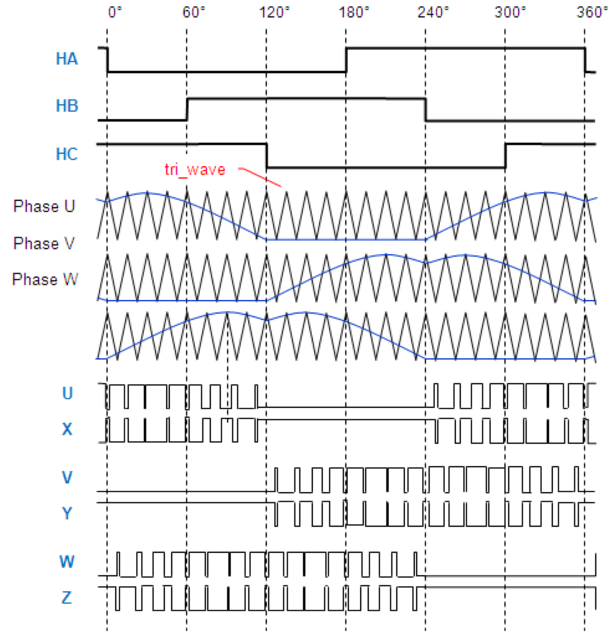


Figure 11. Sine-Wave Output at CW/CCW=1

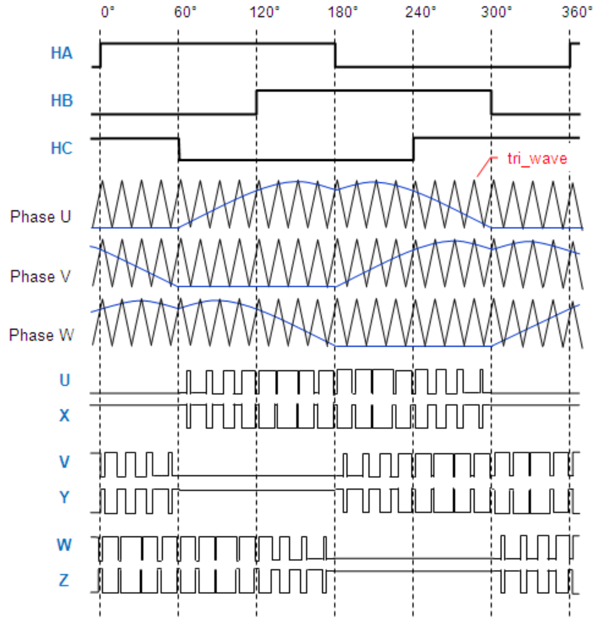


Figure 12. Sine-Wave Output at CW/CCW=0

Figure 11 and Figure 12 describe the output waveform of sine-wave PWM drive. SVM module; based on angles A, B, and C (angle output signal from angle predictor); modulates and outputs three signals of phases U, V, and W to sine-wave PWM generator. Afterward, the sine-wave PWM generator compares the modulation signal of SVM and tri-wave of PWM. Then U, X, V, Y, W, and Z pins output the PWM waveform that can generate sine-wave current drive.

Current Leading Phase Correction

In addition to angle prediction, the angle predictor can also provide angle shift function.

Due to the inductance property of stator winding, the sine-wave current actually flowing into the three-phase coil lags behind the three-phase input voltage by an angle Δθ (as shown in Figure 13). As a result, the sine-wave current fails to share the same phase with the back electromotive force and thus the output efficiency is not brought into full play.

By using voltage level of the AS pin or ANG_REG register, FCM8201 can advance the angle of output voltage by 0° ~ 58°. Once the voltage angle is advanced, the angle of current phase angle is advanced as well. In this way, the current phase does not lag behind the back electromotive force where lag leads to reduced motor output efficiency.

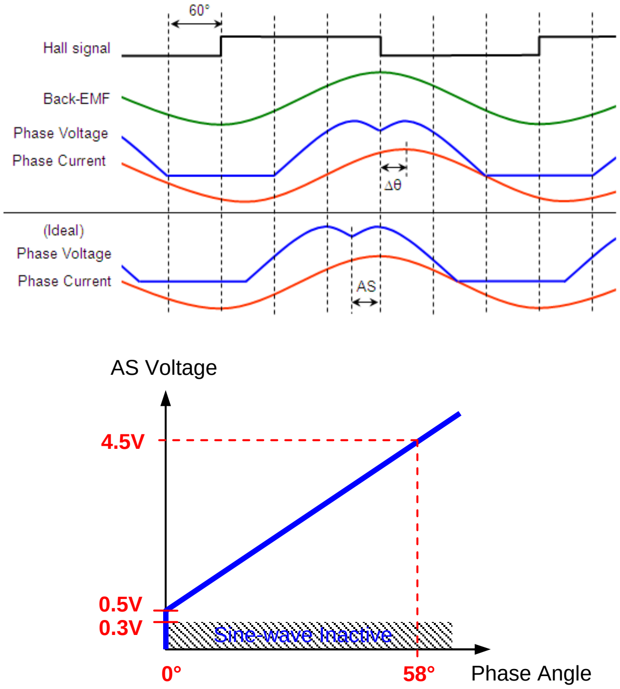


Figure 13. Current Leading Phase Correction

Circuit Design for AS Function and Current Angle Correction Method

Step 1: Design AS Application Circuit

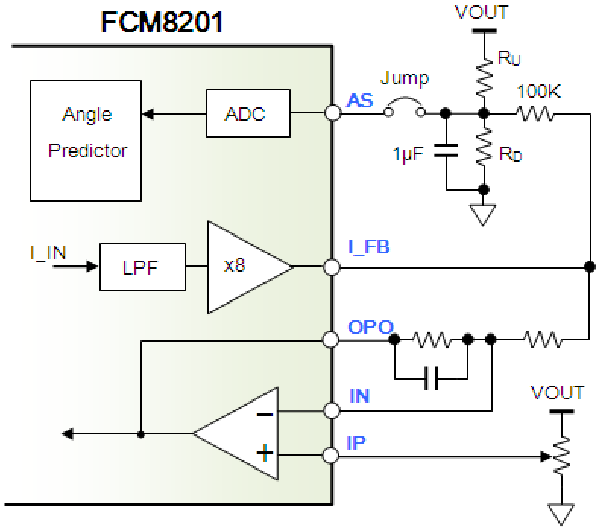


Figure 14. Stand-Alone Application of AS Function

Circuit design for Stand-Alone Mode is shown in Figure 14. The magnitude of current flowing through the motor stator winding is proportional to the lag angle of the phase current; the greater the current, the greater the lag angle. Therefore, the I_FB pin can be used to obtain the information on the current flowing through motor stator winding. After it passes the low-pass filter made up of a 100KΩ resistor and a 1µF capacitor; a voltage between 0.5V and 4.5V, which varies with current magnitude, is obtained. After it is connected to the AS pin via Jump, the current phase can be automatically adjusted in relation to magnitude of current. When the current increases, the phase angle is advanced ahead to a greater extent, and vice versa.

Resistors Jump, RU, and RD are preliminary components added for the sake of subsequent adjustment steps. The engineering prototype prepared at the early stage does not need these components welded. After motors and loads are added, the sequential adjustment steps have been performed, and the bias voltage actually needed has been identified; the value and necessity of resistance can be decided.

If FCM8201 is used in SPI Mode, the circuit between the I_FB pin and the AS pin can be omitted. Pull the AS pin up to VOUT. This is because the microcontroller can read I_FB_REG register using SPI to get the information on current of actual motor running and the AS voltage level needed for shift angle can be written by SPI into ANG-REG register control.

Step 2: Identify Position for Installing Hall Components of Motor

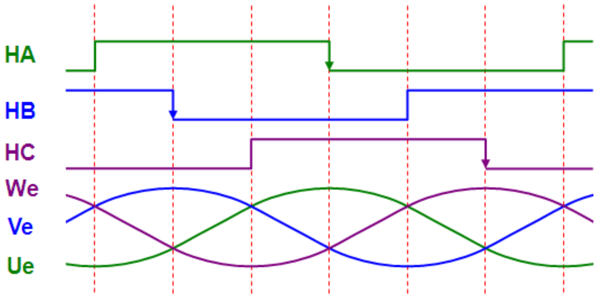


Figure 15. Hall Position

Before adjustment for phase current angle shift, it is necessary to identify and adjust position for installing Hall components. The falling edge of Hall components' output signals HA/B/C should be aligned with the upper half-cycle center of back electromotive force Ue/Ve/We of the motor's three-phase waveform.

Step 3: Configure Environment for Measuring Shift Angle Adjustment

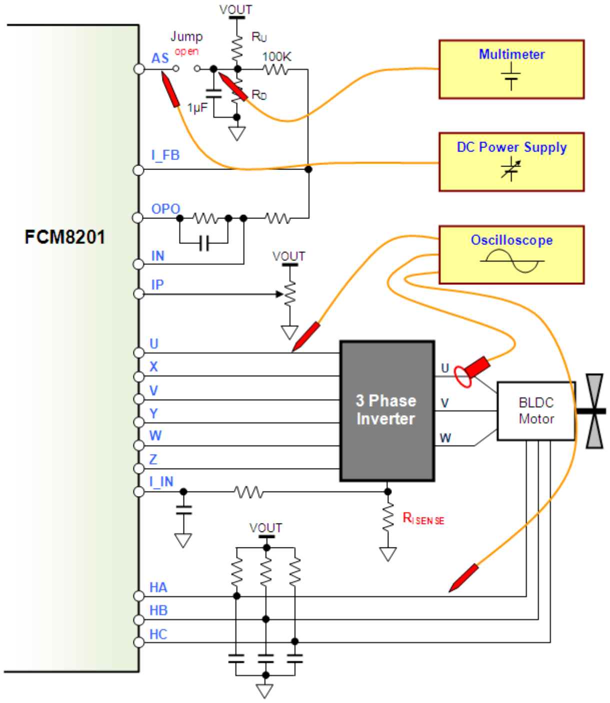


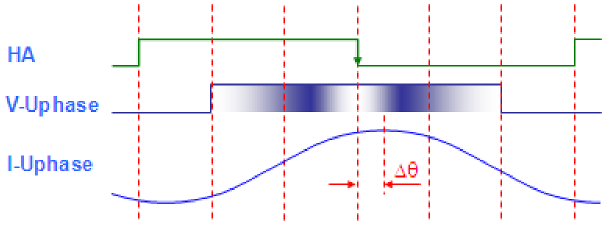
Figure 16. Measuring Configuration

For shift adjusting for phase current angle, a multimeter is needed for measuring and recording the voltage value of the I_FB pin in varied loading conditions after it has been filtered by a 100KΩ/1µF low-pass filter. A DC power supply is needed to supply the AS pin with an input voltage between 0.5V and 4.5V to adjust the advance angle of the PWM output waveform. A four-channel oscilloscope, two voltage probes, and one current probe are needed for observing HA signal, U-phase PWM output, and U-phase current waveform.

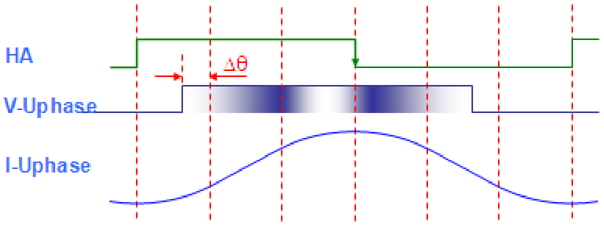
Step 4: Adjust, Measure, and Record Phase Current Angle

After the measuring environment is ready, the adjustment procedure can start.

1. Adjust the input voltage of the AS pin to 0.5V (PWM output advance angle: 0°).
2. Use an oscilloscope to observe HA, U-phase PWM output, and U-phase current waveform. It should be possible to see that the center of U-phase PWM output waveform is aligned with HA falling edge. However, the upper half cycle center of U-phase current waveform lags behind HA falling edge.



3. Add the input voltage of the AS pin slowly at a pace of 0.1V/step. Meanwhile, observe whether the U-phase PWM output and U-phase current waveform shift leftwards (forward) slowly accordingly.



4. When the upper half cycle center of U-phase current waveform shifts to HA falling edge, the U-phase current is already synchronous with the back electromotive force of U phase. At this moment, record the input voltage of the AS pin, the I_FB voltage, and the load size (motor revolving speed) measured by the multimeter.
5. Repeat the Steps 1 ~ 4 and record all the resultant AS pin and I_FB voltages after adjustment made in varied load (revolving speed) conditions.

Step 5: Analyze Measurement Records and Select Bias Resistance

Due to difference in motor properties, the measurement results recorded in Step 4 may present the following three proportion relations between AS and I_FB after these records are processed.

Type 1: Rising Rates of AS and I_FB Voltage Consistent

If the measurement records show the result in Figure 17, there is no need for welding resistors RU and RD; Jump should be shorted. After the I_FB signal passes the 100KΩ/1µF low-pass filter, directly connect it to the AS pin's input end so that the AS voltage varies with I_FB voltage. If the rising rates are consistent but the voltage levels slightly deviate, fine-tune the resistance value of the current-sensing resistor (R_{ISENSE} in Figure 16) to change the level of I_FB voltage to bring the voltage of I_FB and AS into consistency.

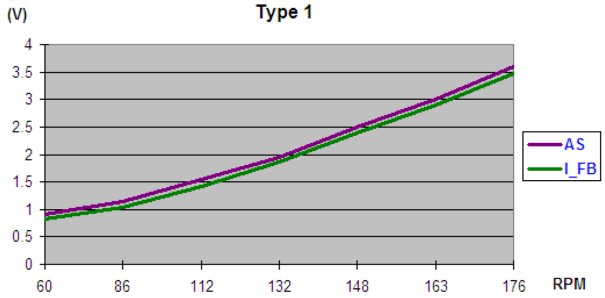


Figure 17. Curve of Type 1

Type 2: AS Voltage Greater than I_FB Voltage; Smaller Load Equals Greater Voltage Difference

If the measurement records show the result in Figure 18, there is no need to weld resistor RD; short Jump and weld resistor RU to VOUT. Using a 100KΩ resistor, a bias voltage is generated to push up the voltage level of AS. The magnitude of bias voltage is inversely proportional to voltage of I_FB; the lower the I_FB voltage, the higher the bias voltage.

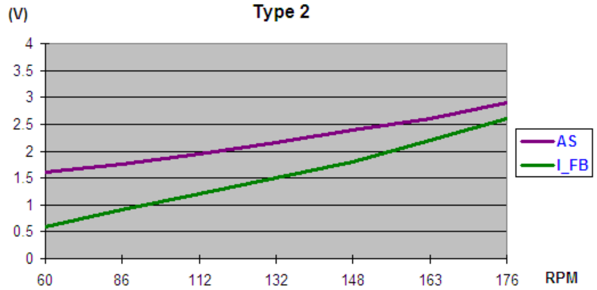


Figure 18. Curve of Type 2

Type 3: AS Voltage Lower than I_FB Voltage; Greater Load Equals Greater Voltage Difference

If the measurement records show the result in Figure 19, there is no need to weld resistor RU; short Jump and weld resistor RD. Have the input voltage of AS equal to I_FB voltage that has passed RD and 100KΩ voltage division.

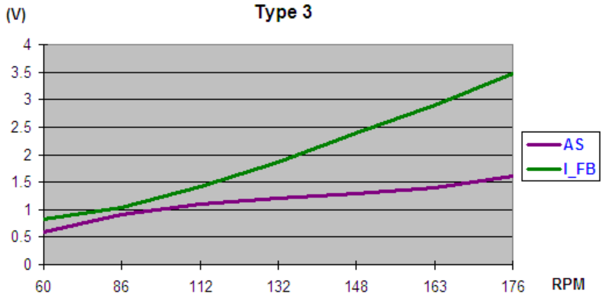


Figure 19. Curve of Type 3

Application of Close-Loop Speed Control

Although FCM8201 comes with open-loop control design, it can also realize close-loop speed control by using FO pin or reading / writing SPI register.

Use SPI Interface to Realize Application of Close-Loop Speed Control

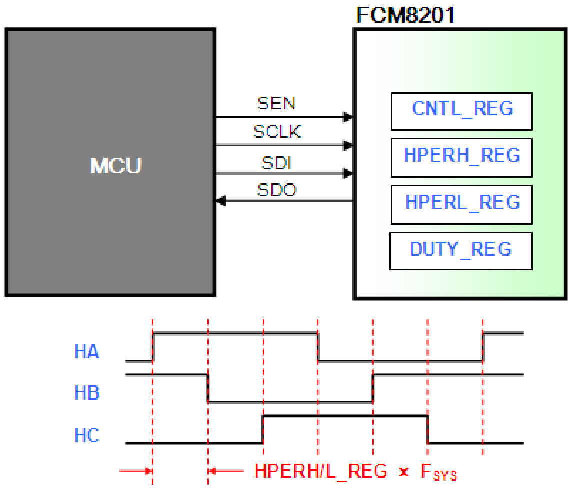


Figure 20. Close-Loop Speed Control with SPI

When the SPI interface is used for realizing close-loop speed control, four internal registers (CNTL_REG, HPERH_REG, HPERL_REG, and DUTY_REG) are used. First, have FCM8201 enter SPI Mode, then set the DT_EA bit of CNTL_REG to “1.” Then the control over PWM duty is taken over by DUTY_REG. By writing DUTY_REG, reach the goal of controlling width of PWM duty. DUTY_REG is a readable / writeable 8-bit register. When the value written into it is 0x00, that is equivalent to DUTY pin ≤ 0.5V and PWM outputs zero duty. When the value written is 0xFF, that is equivalent to DUTY pin = 4.5V and PWM outputs full duty. The equation below shows the proportional relation between written value of DUTY_REG and DUTY pin voltage:

$$Vol. = \left(\frac{4.5V - 0.5V}{255} \times DUTY\_REG \right) + 0.5V \qquad (4)$$

HPERH_REG and HPERL_REG are 16-bit read-only counter registers. The content of their count value multiplied by the reciprocal of system working frequency equals the duration of each Hall signal change. After reading these two registers, Equation (5) is used for conversion and obtaining the current motor speed. After current motor speed is obtained from HPERH/L_REG, DUTY_REG is used for adjusting duty output by PWM. In this way, control over speed feedback is realized.

$$RPM = \frac{60}{\frac{HPERH/L\_REG}{CLK_{HPER}} \times 6 \times \frac{Poles}{2}} \qquad (5)$$

where poles is the number of motor poles.

CLK_{HPER} =

ARNG[1:0] / ANG_REG	CLK_{HPER}
0, 0	$CLK_{HPER} = f_{SYS}$
0, 1	$CLK_{HPER} = f_{SYS} \times 2$
1, x	$CLK_{HPER} = f_{SYS} \div 4$

When reading these two registers, always read HPERH_REG before reading HPERL_REG. Once HPERH_REG has been read, FCM8201 immediately suspends content update of both registers. FCM8201 does not re-count the new Hall signal change duration until HPERL_REG is also read.

Use FO Signal to Realize Application of Close-Loop Speed Control

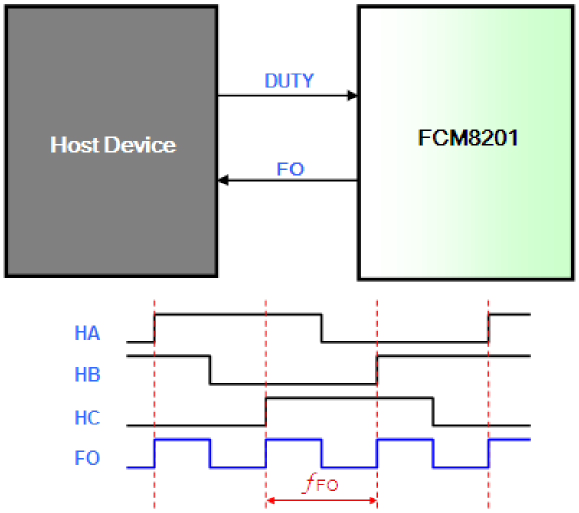


Figure 21. Close-Loop Speed Control with DUTY/FO

FO signal changes every time Hall signal changes. Therefore, the cycle time of FO signal equals one third (1/3) of Hall signal’s cycle time. In an application with a host device, the host device may use the FO pin to obtain speed information of motor, then output the appropriate voltage level to the DUTY pin to adjust duty output by PWM. This is also a way to control close-loop speed.

The equation below can be used to convert cycle time of FO signal into actual speed of the motor:

$$RPM = \frac{60}{\frac{1}{f_{FO}} \times 3 \times \frac{Poles}{2}} \qquad (6)$$

where poles is the number of motor poles.

Current Feedback and Current Protections

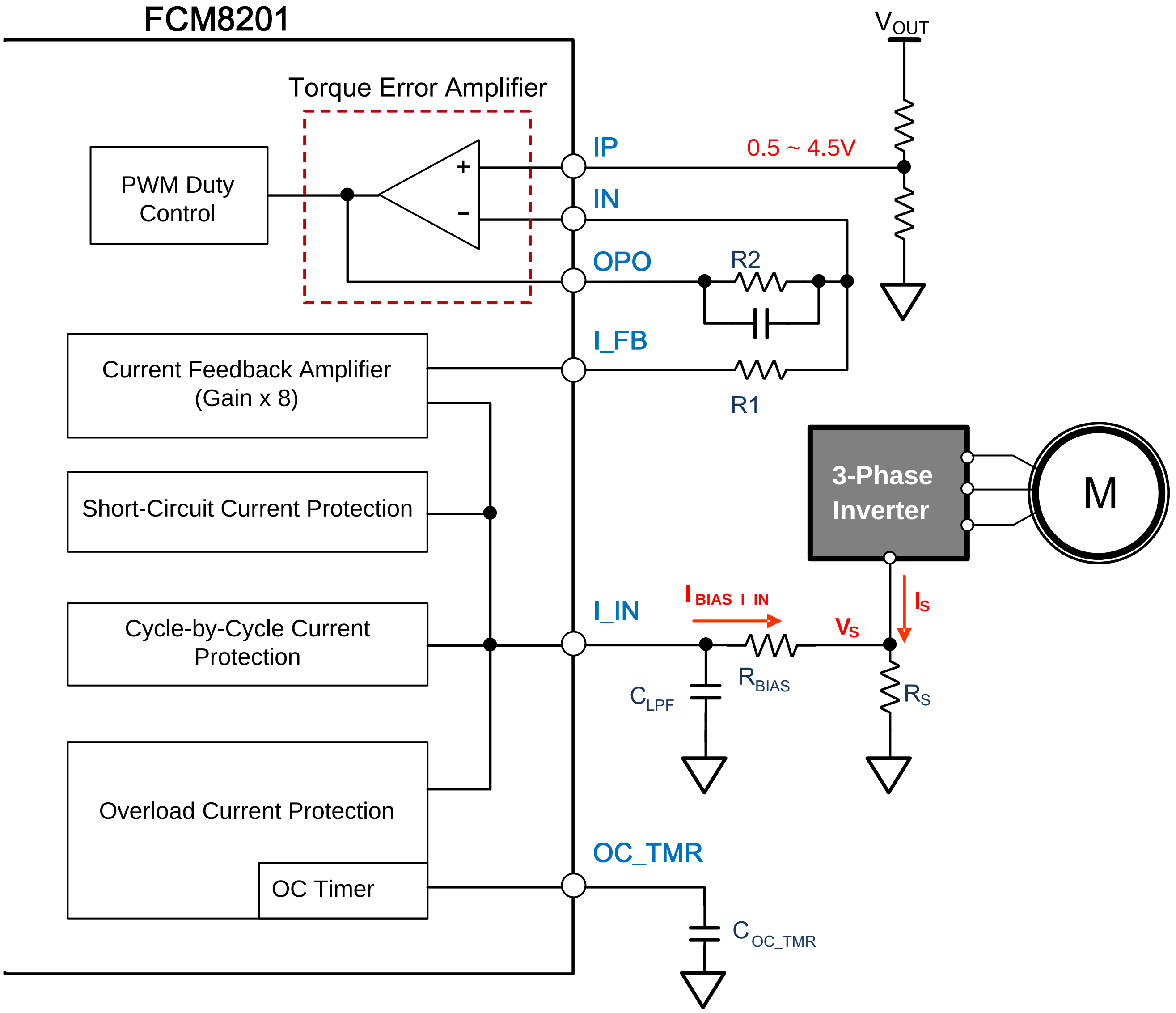


Figure 22. Block Diagram of Current Feedback and Protections

Figure 22 shows the internal current feedback block diagram and external application circuit. The current feedback block includes the current feedback amplifier, torque error amplifier, and over-current protections.

After the current feedback signal (I_IN) passes through a low-pass filter to screen out high-frequency noise, the current feedback amplifier amplifies the current eightfold before the current is output via the I_FB pin. This is the main function of the current feedback amplifier. An external application circuit is provided for applications such as torque-loop control or automatic adjustment of current leading phase.

I_IN pin outputs 50µA bias current ($I_{BIAS\_I\_IN}$) and then, by means of the additional 10KΩ bias resistor (R_{BIAS}), the benchmark level of load current is raised to 0.5V.

The application circuit lies between the I_IN pin and R_{BIAS}; somewhere near I_IN pin there should be a filter capacitor C_{LPF} for grounding. High-frequency noise above PWM frequency is filtered to prevent noise from triggering abnormal operation of current protection. When PWM frequency falls between 15 and 30KHz and R_{BIAS} is 10KΩ, the capacitance should fall between about 220pF and 470pF. Otherwise, excessive capacitance leads to inaccuracy of protection function while insufficient capacitance causes failure to filter all noise, likely to trigger abnormal operation of current protection.

The relationship between the current feedback input signal and the output (I_IN vs. I_FB) can be expressed by:

$$V_{I\_FB} = (V_S \times 8) + (I_{BIAS\_I\_IN} \times R_{BIAS}) \tag{7}$$

Application of Torque Feedback Control

Although FCM8201 comes with open-loop speed control design; if the combination of I_FB pin and torque error amplifier is used, it is also possible to realize application of torque feedback control.

In the application circuit with torque feedback control, the IP pin can serve as the input signal for torque control. When the current feedback signal I_IN has been amplified by I_FB in the torque error amplifier, it is compared with torque control feedback signal IP. Once I_FB is greater than the setting for IP, duty width of PWM starts to decrease to reduce the torque output of motor. If I_FB is lower than the setting for IP, the duty of PWM is determined by the speed loop's DUTY pin or DUTY_REG.

If the application circuit does not need torque feedback control, Figure 23 should be followed for connecting torque error amplifier. The circuit connection method can take the IP pin as the limit setting for maximum PWM duty; the maximum PWM duty is determined by the external divided resistance of IP pin.

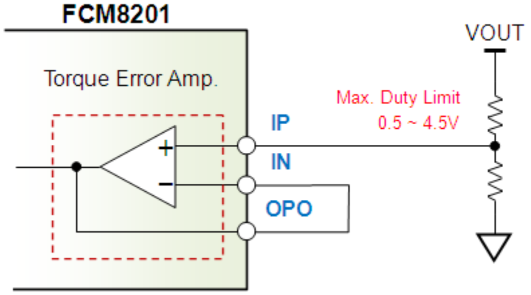


Figure 23.Application without Torque -Loop Control

Application of Over-Current Protections

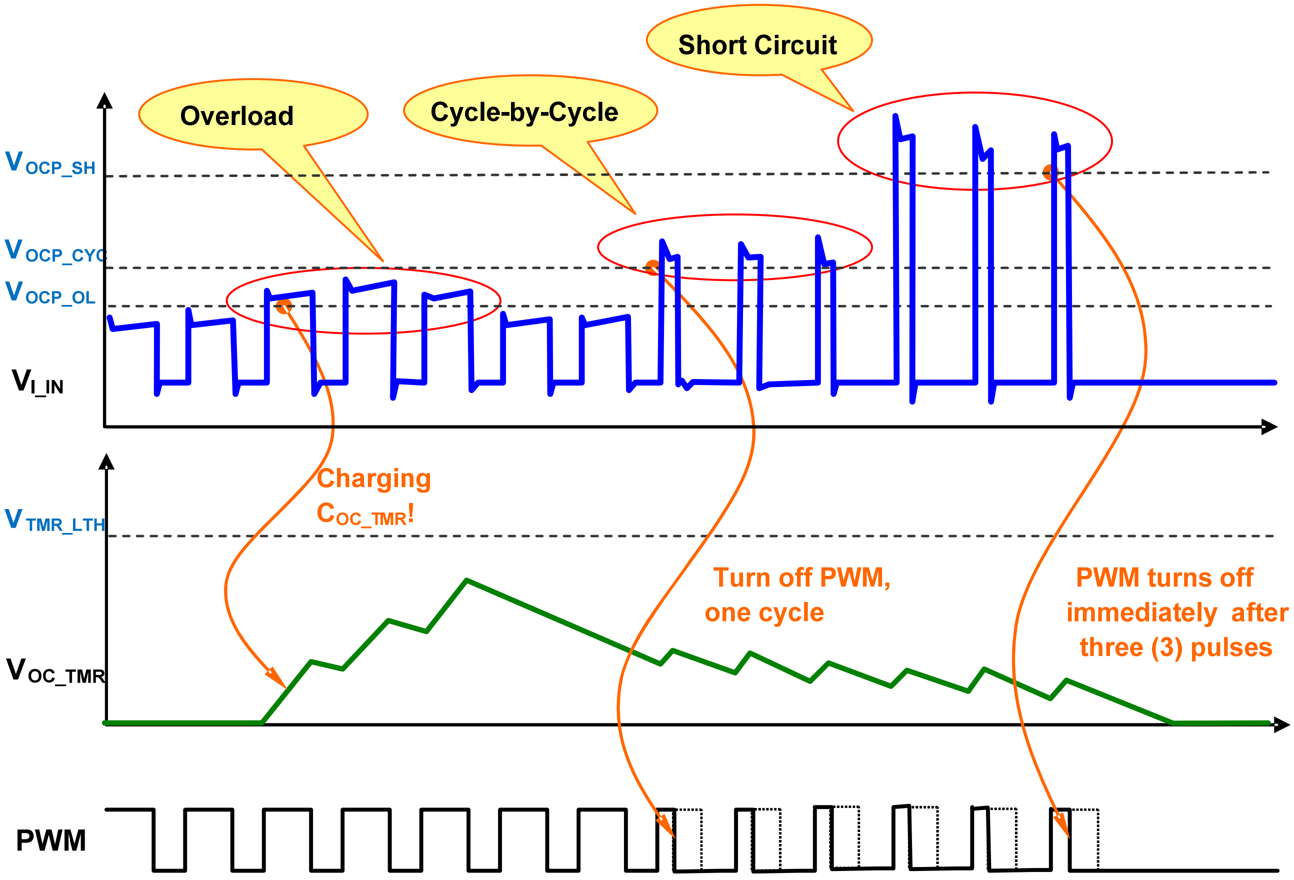


Figure 24.Operating of Current Protections

As shown in Figure 24, FCM8201 provides three-stage over-current protection. The first stage is overload current protection, where the threshold voltage (V_{OCP_OL}) of protection operation is 1.4V. The second stage is cycle-by-cycle current protection for each cycle of PWM, where the threshold voltage (V_{OCP_CYC}) of protection operation is 1.5V. The final stage is short-circuit current protection, where the threshold voltage (V_{OCP_SH}) of protection operation is 2.5V.

Overload Current Protection

The time setting of the over-current timer (OC timer) for overload protection can be determined by a capacitor on the OC_TMR pin or the OC_TMR[2:0] register.

The default OC timer is the OC_TMR pin and an externally connected capacitor. Once the voltage of I_IN goes above V_{OCP_OL}, the OC_TMR pin provides a current at 40μA

(I_{TMR_CHG}) to charge the capacitor. When the voltage of I_IN has dropped back to somewhere below V_{OCP_OL}, the OC_TMR pin provides a current at 10µA (I_{TMR_DIS}) to discharge the capacitor. If the voltage of I_IN keeps exceeding V_{OCP_OL}, the OC_TMR keeps charging the capacitor. Once the voltage on the capacitor has been charged to 2.5V (typical), FCM8201 immediately turns off all motor drive output signals (U, V, W, X, Y, and Z). Moreover, it latches the close status until the FREE/nST pin transitions to a high level (free) and then goes back to the low-level status (start). Only then does FCM8201 disable the latch and restore output.

This protection function allows sufficient torque for the motor. Within the safe time set by the OC timer, the motor can run with a load higher than the rated load without damaging power transistor of the driver circuit. In the application field of electric vehicle and electric machine tool, such protection function is necessary.

If the application circuit does not need this protection, ground OC_TMR pin to disable this function.

Detailed descriptions are provided in the SPI Operation Mode section below to address use of the OC_TMR[2:0] register as time setting for OC timer.

Cycle-by-Cycle Current Protection

As long as the voltage of the I_IN pin exceeds the threshold voltage V_{OCP_CYC} of the cycle-by-cycle current protection, FCM8201 immediately reduces the normal PWM duty to the minimum duty output.

The purpose of this protection function is to protect the power transistor of the driver circuit from damage resulting from excessive peak current generated on PWM output as a result of instantaneous load change during motor running.

Short-Circuit Current Protection

When the voltage of the I_IN pin has three to four PWM pulses exceeding V_{OCP_SH}, the short-current protection is triggered. FCM8201 immediately closes all motor drive output signals and latches the close status until the FREE/nST pin transits to a high level (free) and then goes back to low-level status (start). Only then does FCM8201 disable the latch and restore output.

Voltage Protection for System and Motor Driver

FCM8201 provides multiple voltage protection functions, including: over-voltage (OV) and under-voltage (UV) protection for system working voltage (+12V), UV protection for 5V voltage regulator and OV protection for motor driver. The first three protection functions avoid motor drive failure resulting from abnormal system voltage. The last protects the driver circuit and power transistor from being burnt down by the excessive back-EMF voltage generated during motor running.

OV and UV Protection for System

If the system working voltage exceeds 18V, FCM8201 enables OV protection status and closes all motor drive output signals (U, V, W, X, Y and Z); the motor is set to free running. If the working voltage goes below 8V, UV protection is enabled and closes all motor drive outputs and reset itself.

When the output of the 5V voltage regulator goes below 4V, FCM8201 closes all motor drive output signals and the motor is set to free running.

OV Protection of Motor Driver

Braking and instantaneous speed / load change during operation of BLDC motors may generate back-EMF voltage and feed it back to input voltage of motor drive, bringing about a sharp rise in input voltage. If the sharp voltage rise is not appropriately checked, it may burn down the power transistors and the whole driver circuit.

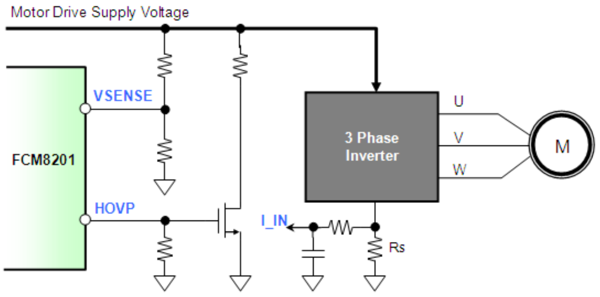


Figure 25. Application Circuit of Motor Drive OVP

Basic application circuit of motor drive OV protection is shown in Figure 25. With the divider resistor, FCM8201 can use the VSENSE pin to detect the voltage of motor drive input voltage. Once the voltage of the VSENSE pin exceeds 4.5V, the OV protection function of motor drive triggers. At this moment, FCM8201 immediately turns off all drive output signals and keeps the back-EMF from rising. Meanwhile the HOVP pin also outputs high level to enable conduction of power transistor of discharge circuit. In this way, the back-EMF fed back to input voltage is quickly exhausted and the level of input voltage is forced to fall to a safe working level. Along with the disappearance of back-EMF, the level of VSENSE pin goes below 4.0V and all motor drive output signals are restored to a normal level.

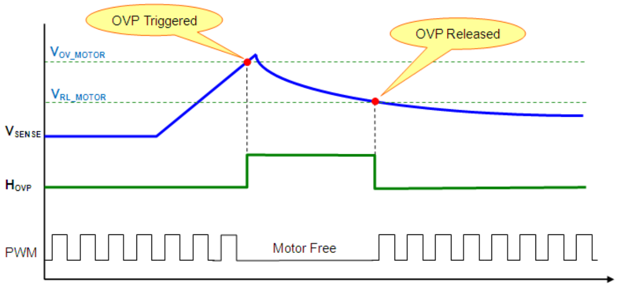


Figure 26. Operating Waveform of Motor Drive OVP

SPI Operation Mode

In SPI Mode, FCM8201 provides more motor control functions than in Stand-Alone Mode. The microcontroller may use SPI interface to control FCM8201 to realize more advanced motor control applications.

Advanced functions only available in SPI Mode include:

- Square-wave drive supports PWM-ON Commutation Mode, which can reduce noise of square-wave drive.
- Dead time provides four time settings: 2.0 /2.5 / 3.5 / 4.0µs, offering more safety in avoiding power transistors from being burnt down.
- Three working range settings for Hall frequency of angle predictor (0.4~40Hz / 0.8~80Hz / 3.2~320Hz), offering a wider speed range available for sine-wave drive operation.
- OC timer can save PCB space and capacitor material cost by setting time duration in internal register.
- Synchronous rectifying function can be turned off or controlled by using OC_TMR pin.
- The function of Low-Side Minimum PWM Output is available so that during PWM-ON square-wave driving, the bootstrap capacitor of the high/low-side driver IC is kept from charge failure.
- Current-leading phase correction can perform more accurate current phase calibration and improve output efficiency of sine-wave drive by reading I_FB_REG and writing ANG_REG.
- FCM8201 can be forced into Sine-Wave Drive Mode.
- The logic level of three signals - HA / HB / HC can be separately configured as inverse, providing more flexibility in choosing Hall sensors.
- The function of Hall Signals Regulation can be enabled to prevent Hall signals from becoming abnormal due to noise interference.
- Cyclic Redundancy Check (CRC) function of SPI interface can be enabled to avoid abnormal operation of motor control due to communication errors.
- Voltage level of VSENSE pin can be read by the microcontroller using SPI. In this way, UV protection or more flexible OV protection can be achieved.
- Voltage level of RT pin can be read by microcontroller using SPI. In this way, greater flexibility in the over-temperature protection can be achieved.
- Cause of all failures can be identified by using STATUS_REG.

Entering and Exiting SPI Mode

FCM8201 uses Stand-Alone Mode by default. To use SPI Mode, each time the machine is started, the master microcontroller should use XP and XN signals to complete the signal toggle procedure shown in Figure 27. In this way, FCM8201 activates SPI Mode.

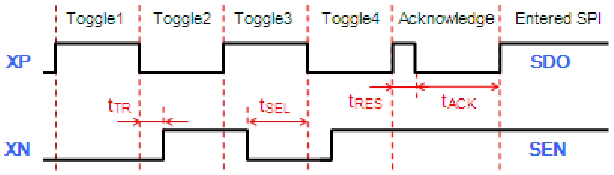


Figure 27. Timing Diagram of Entering SPI Mode

After four consecutive repetitions of HIGH/LOW toggle, MCU should keep XP/XN at high-level output of open-drain (collector). If FCM8201 confirms activation of SPI Mode, within 5µs (t_{RES}) the XP signal is lowered to a low-level status and stays for about 1ms (t_{ACK}) so that MCU is provided with sufficient time to acknowledge that FCM8201 has activated SPI Mode. Afterwards, XP pin changes to SDO function and XN pin to SEN function.

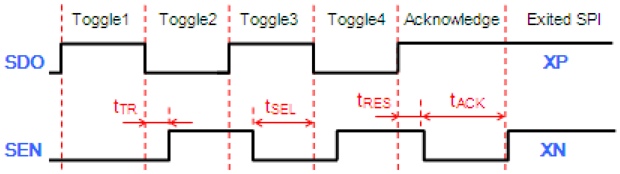


Figure 28. Timing Diagram of Exiting SPI Mode

To return from SPI Mode to Stand-Alone Mode, MCU has to send four consecutive HIGH/LOW toggle signals from SDO/SEN pins and keep SDO/SEN at high-level output of open-drain (collector). If FCM8201 confirms exit of SPI Mode, within 5µs (t_{RES}) SEN signal is lowered to a low-level status and stays for about 1ms (t_{ACK}) so that MCU is provided with sufficient time to acknowledge that FCM8201 has exited SPI Mode. Then SDO changes back to XP function and SEN to XN function.

Table 3. Timing Specification of SPI Mode Enter and Exit

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t_{TR}	Signals Toggle Transient Time	f_{SYS}=1.28MHz	4		12	µs
t_{SEL}	Signals Toggle Stable Time	f_{SYS}=1.28MHz	12		100	µs
t_{RES}	Response Time	f_{SYS}=1.28MHz		4		µs
t_{ACK}	Acknowledge Time	f_{SYS}=1.28MHz		1		ms

Example of Program for Entering and Quitting SPI Mode

```
sbit    PIN_XP_SDO = P2^6;              // to declare MCU I/O Port P2.6 to be XP/SDO pin
sbit    PIN_XN_SEN = P2^4;              // to declare MCU I/O Port P2.4 to be XN/SEN pin

void SPI_Enter_Routine()                // to execute subprogram for entering SPI mode
{
        unsigned char ReTry_Count = 3;
        while (--ReTry_Count)
        {
               Send_Signals_Toggle();
               Delay_1µs (10);                 // waiting for 10µs
               if (!PIN_XP_SDO) return;
               Delay_100µs (15);               // waiting for 1.5ms
        }
}

void SPI_Exit_Routine()                 // to execute subprogram for quitting SPI mode
{
        unsigned char ReTry_Count = 3;
        while (--ReTry_Count)
        {
               Send_Signals_Toggle();
               Delay_1µs (10);
               if (!PIN_XN_SEN) return;
               Delay_100µs (15);
        }
}

void Send_Signals_Toggle()              //  subprogram  for  sending  XP/XN  high/low  toggle  signals
four times
{
        PIN_XP_SDO = 1                  // Toggle 1
        PIN_XN_SEN = 0;
        Delay_1µs (50);
        PIN_XP_SDO = 0;                 // Toggle 2
        PIN_XN_SEN = 1;
        Delay_1µs (50);
        PIN_XP_SDO = 1;                 // Toggle 3
        PIN_XN_SEN = 0;
        Delay_ µs (50);
        PIN_XP_SDO = 0;                 // Toggle 4
        PIN_XN_SEN = 1;
        Delay_1µs (50);
        PIN_XP_SDO = 1;                 // XP/XN keeping Open-Drain (Collector) high output
        PIN_XN_SEN = 1;
}
```

Reading and Writing Operation of SPI Mode

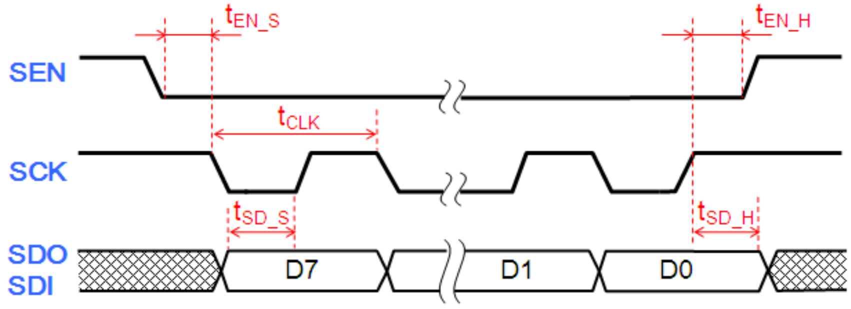


Figure 29. Bus Timing Diagram of SPI

Table 4. Timing Specification of SPI

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
t_{EN_S}	Setup Time of SEN	f_{SYS}=1.28MHz		2		µs
t_{EN_H}	Hold Time of SEN	f_{SYS}=1.28MHz		2		µs
t_{SD_S}	Setup Time of SDO/SDI	f_{SYS}=1.28MHz	0			µs
t_{SD_H}	Hold Time of SDO/SDI	f_{SYS}=1.28MHz		2		µs
t_{CLK}	Cycle Time of SCK	f_{SYS}=1.28MHz	12.5			µs
t_{CLK_TO}	Time Out of SCK	f_{SYS}=1.28MHz		100		ms

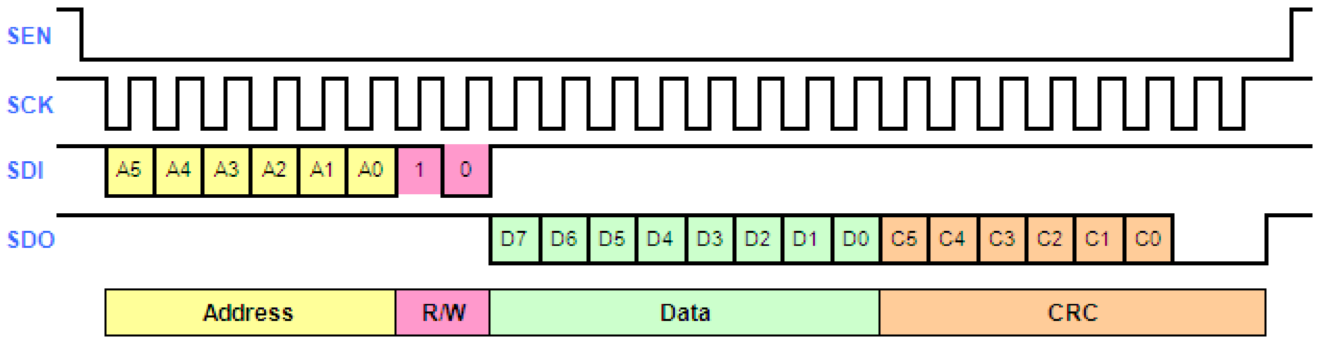


Figure 30. Read Operation Format

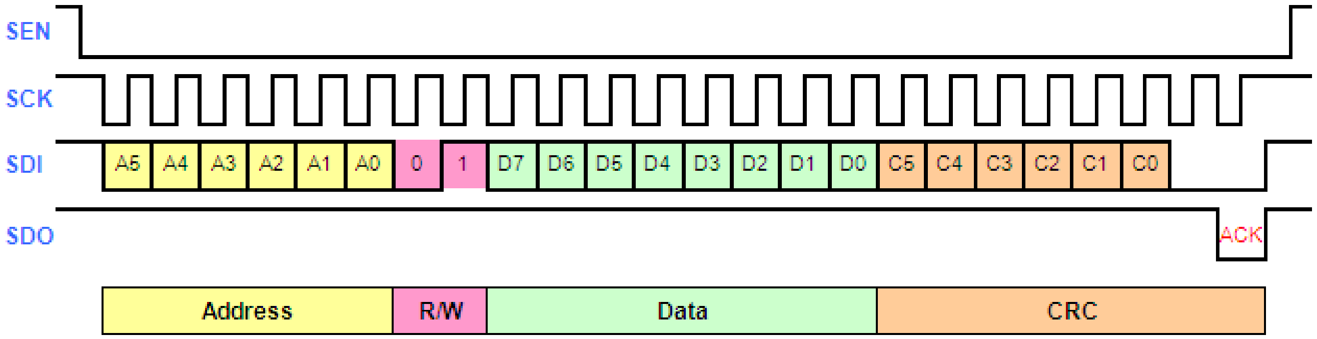


Figure 31. Write Operation Format

SPI interface offers the error detection function of 6-bit cyclic redundancy check (CRC). By default, this function is disabled. To enable it to prevent communication error from causing motor control failure, set CRC_ON bit of WDT_REG to 1.

Whether CRC function is enabled or not, each reading / writing operation digit has a length of 24 bits. The very first 6 bits stand for the address of the register to be read / written; the following 2 bits, the control command to be executed for reading or writing (Read = 1,0; Write = 0,1); the next 8 bits, data to be read / written from / to the register; the preceding 6 bits of the last 8 bits, CRC calculation value; and the lattermost two bits, invalid bits.

Calculation scope of CRC falls between A5 to D0, 16 bits in all. During reading operation, the master MCU has to calculate the CRC of these 16 bits by itself; then a check is performed for consistency with the CRC sent by FCM8201 via SDO. Inconsistency of the two CRC contents indicates error in communications for the present reading operation. MCU has to re-execute the reading operation. During a writing operation, FCM8201 automatically checks consistency between CRC self-calculated and the CRC sent by MCU. If consistent, upon the time at the final bit, SDO is changed to low-level status (i.e. ACK bit) to indicate no error in communications. If MCU does not detect any ACK bit on the SCK rising edge of the final bit, that indicates there is an error in the writing operation just performed and the MCU must repeat execution.

CRC formula:

$$g(x) = x^6 + x + 1 \tag{8}$$

Example of Program for CRC Calculation

ADDR: means the address of the register to be read / written.

RW: means the control command to be executed for reading or writing.

DATA: means the data to be read / written from / to the register.

```
int      Cal_CRC(int ADDR, RW, DATA)
{
         unsigned long Data_String, CRC_Accum, CRC_Result;
         int i;

         Data_String = (((ADDR & 0x3F) << 2 | (RW & 0x3)) << 8) | (DATA & 0xFF);
         for ( i = 0; i <= 15; i++ )
         {
                if ( (Data_String & 0x8000) == 0x8000 )
                {
                       CRC_Accum = Data_String & 0xFE00;
                       CRC_Accum = CRC_Accum ^ 0x8600;
                       Data_String = (Data_String & 0x01FF) | CRC_Accum;
                }
                if ( i == 15 )
                       break;
                Data_String = Data_String << 1;
         }
         CRC_Result = ((Data_String & 0xFE00) >> 9) & 0x3F;
         return CRC_Result;
}
```

SPI Readable / Writeable Register

Table 5. List of SPI Readable / Writeable Registers

Address	Name	Description
0x00	CNTL_REG	System Control Register
0x01	PWM_REG	PWM Control Register
0x02	ANG_REG	Angle Shift Control Register
0x03	SVM_REG	Sine-Wave Control Register
0x04	HALL_REG	Hall Signals Control Register
0x05	**Reserved**	
0x06	WDT_REG	Watch Dog Timer Control Register
0x07	**Reserved**	
0x08	DUTY_REG	Duty Control Register
0x09	IP_REG	Error Amplifier IP Pin Control Register

Table 6. CNTL_REG, System Control Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	**TMR_CLR**	**OC_TMR2**	**OC_TMR1**	**OC_TMR0**	**IP_EA**	**DT_EA**	**CW/CCW**	**FREE/nST**
Default	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**

TMR_CLR: OC Timer clearing bit.
1: OC Timer clearing.
0: OC Timer normal operation.

OC_TMR[2:0]: Bit for setting OC Timer counter time. When I_IN pin voltage is greater than 1.4V, the counter counts upwards; less than 1.4V, downwards. The clock frequency for upward-counting is four times that of downward counting.
0,0,0: The counter does not work; the time of OC timer is determined by the capacitor of the external OC_TMR pin.
0,0,1: Upward counting time = 2^18 ÷ f_{SYS}. When f_{SYS}=1.28MHz, the upward counting takes 0.2s while the downward one, 0.8s.
0,1,0: Upward counting time = 2^19 ÷ f_{SYS}. When f_{SYS}=1.28MHz, the upward counting takes 0.41s while the downward one, 1.64s.
0,1,1: Upward counting time = 2^20 ÷ f_{SYS}. When f_{SYS}=1.28MHz, the upward counting takes 0.82s while the downward one, 3.28s.
1,0,0: Upward counting time = 2^21 ÷ f_{SYS}. When f_{SYS}=1.28MHz, the upward counting takes 1.64s while the downward one, 6.55s.
1,0,1: Upward counting time = 2^22 ÷ f_{SYS}. When f_{SYS}=1.28MHz, the upward counting takes 3.28s while the downward one, 13.11s.
1,1,0: Upward counting time = 2^23 ÷ f_{SYS}. When f_{SYS}=1.28MHz, the upward counting takes 6.55s while the downward one, 26.21s.
1,1,1: Upward counting time = 2^24 ÷ f_{SYS}. When f_{SYS}=1.28MHz, the upward counting takes 13.11s while the downward one, 52.43s.

IP_EA: Bit for enabling IP_REG.
0: Duty width of PWM is controlled by external IP pin.
1: Duty width of PWM is controlled by internal IP_REG.

DT_EA: Bit for enabling DUTY_REG.
0: Duty width of PWM is controlled by external DUTY pin.
1: Duty width of PWM is controlled by internal DUTY_REG.

CW/CCW: Direction control bit.
0: CCW.
1: CW.

FREE/nST: Bit for controlling free operation or starting.
0: motor starts (nST).
1: motor free running (FREE).

Table 7. PWM_REG, PWM Control Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	PMOD	na	DT1	DT0	SEQ_TBL	SYN_OFF	EXT_SYN	LPWM
Default	0	0	0	0	0	0	0	0

PMOD: Bit for selecting PWM drive mode.
0: After angle predictor makes a correct prediction, FCM8201 switches to sine-wave PWM drive.
1: After angle predictor makes a correct prediction, FCM8201 switches to square-wave PWM drive.

DT[1:0]: Bit for setting dead time.
0,0: 2.5µs
0,1: 2.0µs
1,0: 4.0µs
1,1: 3.5µs

SEQ_TBL: Bit for setting Commutation Mode of square-wave drive PWM.
0: PWM-PWM Mode commutation.
1: PWN-ON Mode commutation.

SYN_OFF: Bit for controlling synchronous rectifying function of square-wave drive.
0: Enable synchronous rectifying function of square-wave drive.
1: Disable synchronous rectifying function of square-wave drive.

EXT_SYN: On/off for synchronous rectifying function, bit for setting control by external pin OC_TMR.
0: On/off for synchronous rectifying function, to be determined by SYN_OFF bit.
1: When the value set for OC_TMR [2:0] of CNTL_REG is not 0,0,0, on/off of synchronous rectifying function is determined by HIGH/LOW status of external pin OC_TMR.

LPWM: Bit for enabling low-side minimum PWM output function.
0: Disable low-side minimum PWM output function.
1: Enable low-side minimum PWM output function.

Note:

1. When PWM-ON is selected for square-drive PWM Commutation Mode, LPWM is set as 1. When PWM output is disabled, FCM8201, at the low-side of the three phases, permanently outputs minimum PWM duty. This prevents high-side MOSFET conduction failure resulting from shortage in opportunity for charging bootstrap capacitor of high/low-side driver IC.

Table 8. ANG_REG, Angle Shift Control Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	ANG_SEL	ARNG1	ARNG0	ANG4	ANG3	ANG2	ANG1	ANG0
Default	0	0	0	0	0	0	0	0

ANG_SEL: For selecting input for angle shift control.
0: Angle shift is controlled by external AS pin.
1: Angle shift is controlled by internal ANG[4:0] bits.

ARNG[1:0]: For setting Hall frequency range available for operation of angle predictor.
0,0: Hall frequency available for operation ranges from 0.8Hz to 80Hz (48 ~ 4800rpm at two-poled motor).
0,1: Hall frequency available for operation ranges from 0.4Hz to 40Hz (24 ~ 2400rpm at two-poled motor).
1,x: Hall frequency available for operation ranges from 3.2Hz to 320Hz (192 ~ 19,200 rpm at two-poled motor).

ANG[4:0]: For setting shift angle.
Setting values 0 ~ 31 are equivalent to shift angle 0° ~ 58°.

Table 9. SVM_REG, Sine-Wave Control Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	SIN_MAU	SIN_EA	na	na	na	na	na	na
Default	0	0	0	0	0	0	0	0

SIN_MAU: Bit for enabling forced activation of sine-wave drive.
0: Sine-wave drive is automatically enabled when conditions are met.
1: Sine-wave drive is forced to be enabled whether or not conditions are met.

Note:

2. When sine-wave drive is forced to be enabled, PWM output failure may occur when motor speed exceeds prediction range of angle predictor or the revolving speed changes too quickly!

SIN_EA: For selecting PWM Output Mode when sine-wave drive is forced to be enabled.
0: square-wave drive PWM output.
1: sine-wave drive PWM output.

Note:

3. Forced sine-wave can be eventually enabled only when both bits of SIN_MAU and SIN_EA are set to 1.

Table 10. HALL_REG, Hall Signals Control Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	na	na	na	na	HREG	HC_INV	HB_INV	HA_INV
Default	0	0	0	0	0	0	0	0

HREG: For setting on/off for Hall signals regulation.
0: Disable Hall signals regulation.
1: Enable Hall signals regulation.

Note:

4. Whether Hall signal regulation is enabled or not, FCM8201 is provided with a debounce function for Hall signal to filter out 3 ~ 6µs bounce noise. Enabling Hall signal regulation can further filter out abnormal signals that are more harmful than bounce noise and prevent Hall signals from discontinuity that causes pauses in motor running.

HC_INV: Inverse setting of HC input signal.
0: No inverse for HC input.
1: Inverse for HC input.

HB_INV: Reverse setting of HB input signal.
0: No inverse for HB input.
1: Inverse for HB input.

HA_INV: Reverse setting of HA input signal.
0: No inverse for HA input.
1: Inverse for HA input.

Table 11. WDT_REG, Watch Dog Timer Control Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	OSL_DIS	OTL_EA	na	CRC_ON	WDT_EN	CLR	W_TMR1	W_TMR0
Default	0	0	0	0	0	0	0	0

OSL_DIS: For setting latch disabling for pin open/short-circuit protection. Only the R_CLK pin is provided with short protection function; R_CLK and RT pins are provided with open-circuit protection.
0: Open/short-circuit protection does not provide latch function; so long as open/short status ends, FCM8201 immediately restores PWM output.
1: After triggering of open/short-circuit protection, all PWM outputs are immediately turned off and latched until the next time FREE/nST pin re-transitions to LOW level or until FREE/nST bit of CNTL_REG is set to 1 and then set to 0.

***OTL_EA*:** For enabling of over-temperature protection latch.
0: Over-temperature protection does not provide latch function; so long as over-temperature status ends, FCM8201 immediately restores PWM output.
1: After triggering of over-temperature protection, all PWM outputs are immediately turned off and latched until the next time FREE/nST pin re-transitions to LOW level or until FREE/nST bit of CNTL_REG is set to 1 and then set to 0.

***CRC_ON*:** For enabling of SPI CRC function.
0: SPI CRC is disabled.
1: SPI CRC is enabled.

***WDT_EN*:** For enabling of watch dog timer (WDT).
0: WDT is disabled.
1: WDT is enabled.

***W_TMR[1:0]*:** For setting time for watch dog timer.
0,0: When f_{SYS} = 1.28MHz, timing duration is 0.25s.
0,1: When f_{SYS} = 1.28MHz, timing duration is 0.5s.
1,0: When f_{SYS} = 1.28MHz, timing duration is 1s.
1,1: When f_{SYS} = 1.28MHz, timing duration is 2s.

Table 12. DUTY_REG, Duty Control Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	DUTY7	DUTY6	DUTY5	DUTY4	DUTY3	DUTY2	DUTY1	DUTY0
Default	0	0	0	0	0	0	0	0

***DUTY[7:0]*:** For setting duty width of PWM. When the DT_EA bit of CNTL_REG is set to 1, DUTY[7:0] takes the place of the DUTY pin to control duty width of PWM.
A set value between 0 and 255 is equivalent to DUTY pin voltage 0.5V ~ 4.5V. (0 = Zero Duty, 255 = Full Duty).

Table 13. IP_REG, Error Amplifier IP Pin Control Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0
Default	0	0	0	0	0	0	0	0

***IP[7:0]*:** For setting internal voltage level for IP pin for error amplifier. When IP_EA bit of CNTL_REG is set to 1, the written value of IP[7:0] substitutes an external IP to become the IP pin voltage level of error amplifier. A set value between 0 and 255 is equivalent to IP pin voltage 0.5V ~ 4.5V.

SPI Read-Only Register

Table 14. List of SPI Read Only Registers

Address	Name	Description
0x20	AS_REG	AS Pin Voltage Level Read Register
0x21	VSENSE_REG	VSENSE Pin Voltage Level Read Register
0x22	I_FB_REG	I_FB Pin Voltage Level Read Register
0x23	RT_REG	RT Pin Voltage Level Read Register
0x24	Reserved	
0x25		
0x26	HPERH_REG	Hall Period Counter HIGH Byte Register
0x27	HPERL_REG	Hall Period Counter LOW Byte Register
0x28	STATUS_REG	System Status Register

Table 15. AS_REG, AS Pin Voltage Level Read Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	na	na	na	AS4	AS3	AS2	AS1	AS0

AS[4:0]: ADC value read at the AS pin voltage level.

$$AS(V)=\left(AS[4:0]\times 0.125V\right)+0.5V \tag{9}$$

Table 16. VSENSE_REG, VSENSE Pin Voltage Level Read Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	na	na	na	VS4	VS3	VS2	VS1	VS0

VS[4:0]: ADC value read at the VSENSE pin voltage level.

$$VSENSE(V)=\left(VS[4:0]\times 0.125V\right)+0.5V \tag{10}$$

Table 17. I_FB_REG, I_FB Pin Voltage Level Read Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	na	na	na	IFB4	IFB3	IFB2	IFB1	IFB0

IFB[4:0]: ADC value read at the I_FB pin voltage level.

$$I\_FB(V)=\left(IFB[4:0]\times 0.125V\right)+0.5V \tag{11}$$

Table 18. RT_REG, RT Pin Voltage Level Read Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	na	na	na	RT4	RT3	RT2	RT1	RT0

RT[4:0]: ADC value read at the RT pin voltage level.

$$RT(V)=\left(RT[4:0]\times 0.125V\right)+0.5V \tag{12}$$

Table 19. HPERH_REG, Hall Period Counter High Byte Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	HP15	HP14	HP13	HP12	HP11	HP10	HP9	HP8

Table 20. HPERL_REG, Hall Period Counter Low Byte Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	HP7	HP6	HP5	HP4	HP3	HP2	HP1	HP0

HP[15:0]: Time length counting value for each configuration change of Hall signal HA/B/C.

$$t = \frac{HP[15:0]}{CLK_{HPER}} \qquad (13)$$

Notes:

5. If ARNG[1:0]=0,0, then, $CLK_{HPER} = f_{SYS} \div 4$.
6. If ARNG[1:0]=0,1, then, $CLK_{HPER} = f_{SYS} \div 8$.
7. If ARNG[1:0]=1,x, then, $CLK_{HPER} = f_{SYS}$.

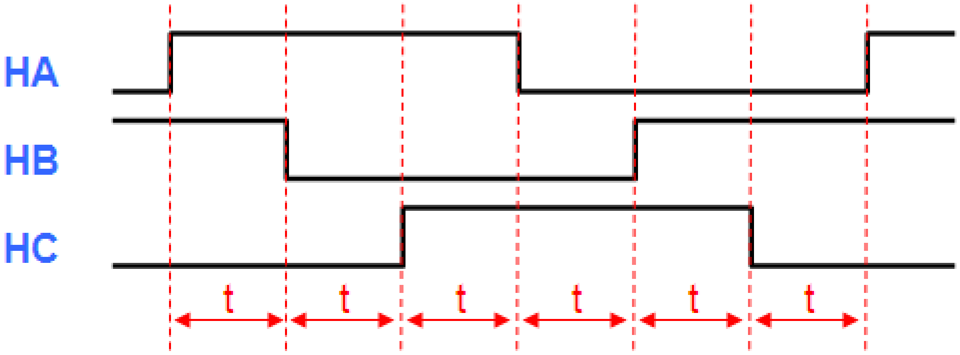


Figure 32. Time Counting of Hall Period Counter Registers

Note:

8. When reading these two registers, always read HPERH_REG before reading HPERL_REG. Once HPERH_REG has been read, FCM8201 immediately suspends content update of both registers. FCM8201 does not re-count the new Hall signal change duration until HPERL_REG is also read.

Table 21. STATUS_REG, System Status Register

Bit	b7	b6	b5	b4	b3	b2	b1	b0
Name	OT	OC	OS	OV	H_ERR	DIR	WDT	SHORT

OT: Read value 1 indicates over-temperature protection has been triggered (RT pin voltage less than 1.0V).

OC: Read value 1 indicates overload over-current protection has been triggered.

OS: Read value 1 indicates R_CLK and RT pins open/short-circuit protection has been triggered.

OV: Read value 1 indicates over-voltage protection has been triggered (VSENSE pin voltage higher than 4.5V).

H_ERR: Read value 1 indicates error in Hall input signals (HA/B/C has such signal configuration as 0/0/0 or 1/1/1).

DIR: Read value 1 indicates the direction and order of Hall signals are not consistent with direction of internal PWM sequencer.

WDT: Read value 1 indicates timing of watch dog timer is timeout.

SHORT: Read value 1 indicates short-circuit over-current protection has been triggered (I_IN pin voltage higher than 2.5V).

Related Datasheets

FCM8201 — 3-Phase Sinusoidal Brushless DC Motor Controller

DISCLAIMER
FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION, OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY
FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION.
As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.