AN205559

F²MC-8FX Family MB95200 Series 8-Bit Microcontroller BLDC Motor Hall Sensor 120° Driver Method

Associated Part Family: MB95200 Series

This document describes how to implement 120° conduction hall sensor BLDC (Brushless DC) motor.

Contents

1 Introduction

This document describes how to implement 120° conduction hall sensor BLDC (Brushless DC) motor control by MB95200 series 8-bit microcontroller.

2 System Principle and Theory

This section describes multi-pulse generator (MPG) and 120° hall sensor driver BLDC motor.

2.1 Description of Multi-pulse Generator

The multi-pulse generator consists of a 16-bit PPG timer, a 16-bit reload timer and a waveform sequencer. By using the waveform sequencer, 16-bit PPG timer output signal can be directed to multi-pulse generator output (OPT5 to OPT0) according to the input signal of multi-pulse generator (SNI2 to SNI0). Meanwhile, the OPT5 to OPT0 output signal can be terminated by DTTI input in case of emergency. The OPT5 to OPT0 output signals are synchronized with the PPG signal in order to eliminate the unwanted glitch.

2.1.1 Block Diagram of Multi-pulse Generator

Figure 1. Block Diagram of Multi-pulse Generator

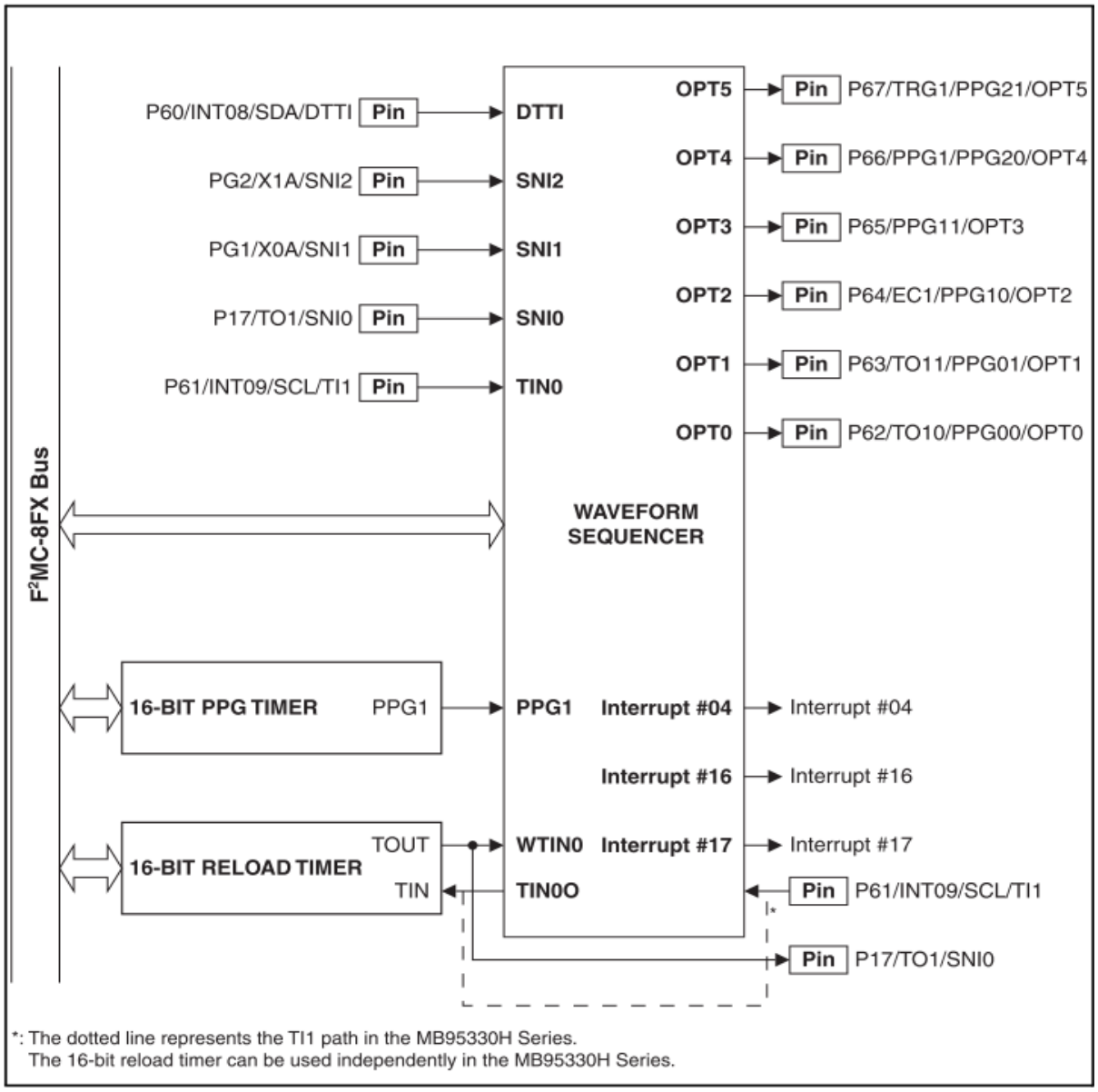

Figure 1 shows the block diagram of multi-pulse generator.

- 16-bit PPG Timer

  The 16-bit PPG timer is used to provide the PPG signal for waveform sequencer. Details of 16-bit PPG timer are described in Hardware Manual of MB95330H Series.

- 16-bit Reload Timer

  The 16-bit reload timer is used to act as interval timer for waveform sequencer. Details of 16-bit reload timer are described in Hardware Manual of MB95330H Series.

- Waveform Sequencer

  The waveform sequencer is the core of multi-pulse generator, which can generate various waveforms.

2.1.2 Registers of Multi-pulse Generator

Figure 2. Registers of Multi-pulse Generator

Output control register (upper)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPCUR	0066_H	DTIE	DTIF	NRSL	OPS2	OPS1	OPS0	WTIF	WTIE	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Output control register (lower)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPCLR	0067_H	PDIF	PDIE	OPE5	OPE4	OPE3	OPE2	OPE1	OPE0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Output data register (upper)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPDUR	$0FDC_H$	BNKF	RDA2	RDA1	RDA0	OP51	OP50	OP41	OP40	$0000XXXX_B$
		R	R	R	R	R	R	R	R	

Output data register (lower)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPDLR	$0FDD_H$	OP31	OP30	OP21	OP20	OP11	OP10	OP01	OP00	$XXXXXXXX_B$
		R	R	R	R	R	R	R	R	

Output data buffer registers (upper)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPDBURB - OPDBUR0	$0FC4_H$ - $0FDA_H$	BNKF	RDA2	RDA1	RDA0	OP51	OP50	OP41	OP40	00000000_B
(Even addresses)		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Output data buffer registers (lower)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
OPDBLRB - OPDBLR0	$0FC5_H$ - $0FDB_H$	OP31	OP30	OP21	OP20	OP11	OP10	OP01	OP00	00000000_B
(Odd addresses)		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable (The read value is the same as the write value.)
R : Read only (The read value is indeterminate.)
x : Indeterminate

(Continued)

(Continued)

Input control register (upper)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
IPCUR	0068_H	WTS1	WTS0	CPIF	CPIE	CPD2	CPD1	CPD0	CMPE	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Input control register (lower)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
IPCLR	0069_H	CPE1	CPE0	SNC2	SNC1	SNC0	SEE2	SEE1	SEE0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare clear register (upper)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CPCUR	$0FDE_H$	CL15	CL14	CL13	CL12	CL11	CL10	CL09	CL08	$XXXXXXXX_B$
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Compare clear register (lower)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
CPCLR	$0FDF_H$	CL07	CL06	CL05	CL04	CL03	CL02	CL01	CL00	$XXXXXXXX_B$
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Timer buffer register (upper)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
TMBUR	$0FE2_H$	T15	T14	T13	T12	T11	T10	T09	T08	$XXXXXXXX_B$
		R	R	R	R	R	R	R	R	

Timer buffer register (lower)

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
TMBLR	$0FE3_H$	T07	T06	T05	T04	T03	T02	T01	T00	$XXXXXXXX_B$
		R	R	R	R	R	R	R	R	

Timer control status register

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
TCSR	$006B_H$	TCLR	MODE	ICLR	ICRE	TMEN	CLK2	CLK1	CLK0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Noise cancellation control register

Address		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
NCCR	$006A_H$	S21	S20	S11	S10	S01	S00	D1	D0	00000000_B
		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

R/W : Readable/writable (The read value is the same as the write value.)
R : Read only (The read value is indeterminate.)
x : Indeterminate

Figure 2 shows registers of multi-pulse generator. For more detailed information, please refer to Chapter 24 in Hardware Manual of MB95330H Series.

2.2 Description of Hall Sensor 120° Driver

2.2.1 Block Diagram of Hall Sensor 120° Driver

Below is the brief operating principle for MCU to drive motor with Hall sensor.

- The multi-pulse generator outputs six switch signals to drive IGBT inverter.
- The three-channel Hall sensor signals are detected by MCU input capture to achieve motor position.
- The one-channel over-current signal is output to MCU by IGBT inverter to protect the whole system.

Figure 3. Block Diagram of Hall Sensor 120° Driver

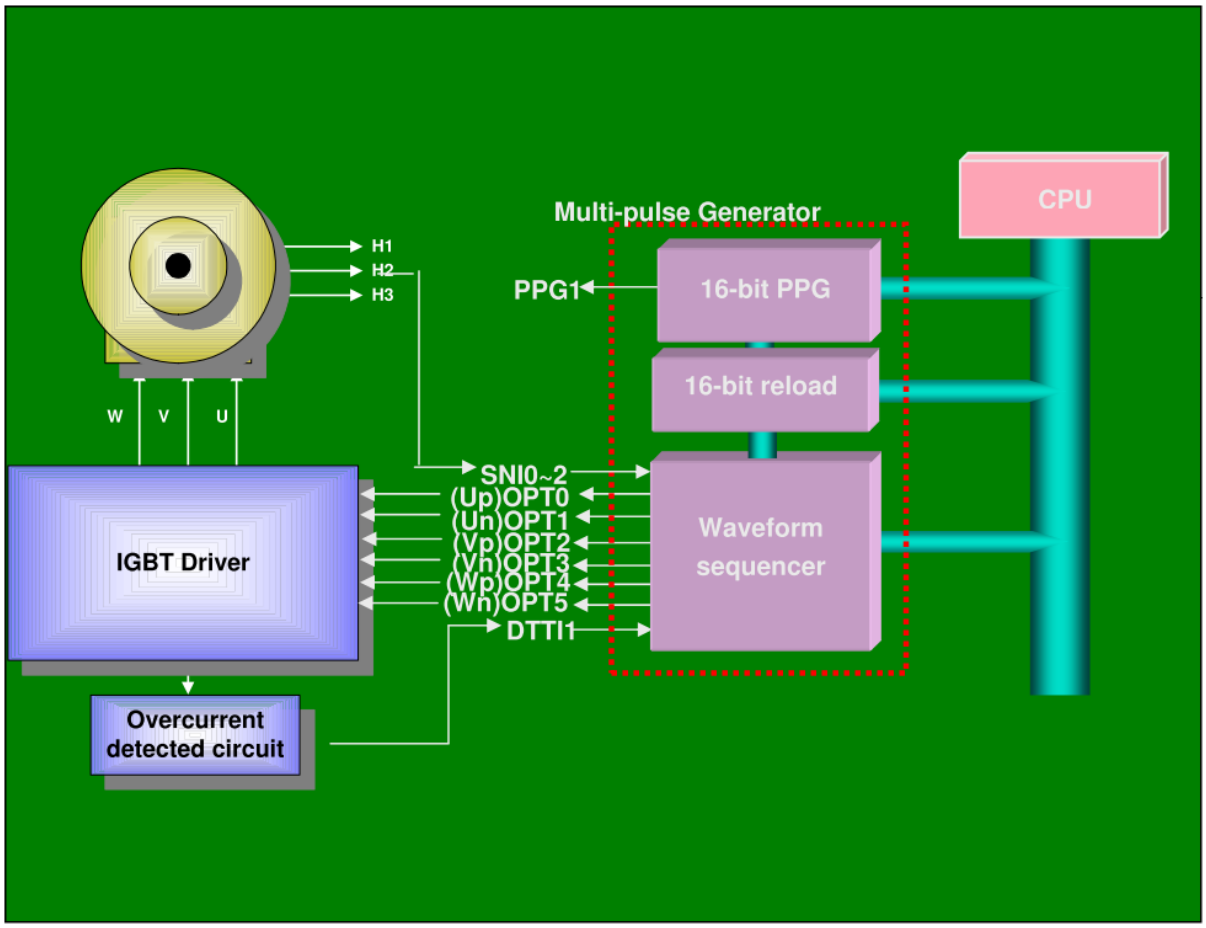

Figure 3 shows the system block diagram of Hall sensor 120° driver BLDC motor.

One electrical cycle is divided into 6 states. The relationship between three-channel Hall sensor signals (H1, H2, H3) and six-channel inverter switch signals (Up, Un, Vp, Vn, Wp, Wn) is shown as below:

Figure 4. Relationship between Hall Sensor Signals and Inverter Switch Signals

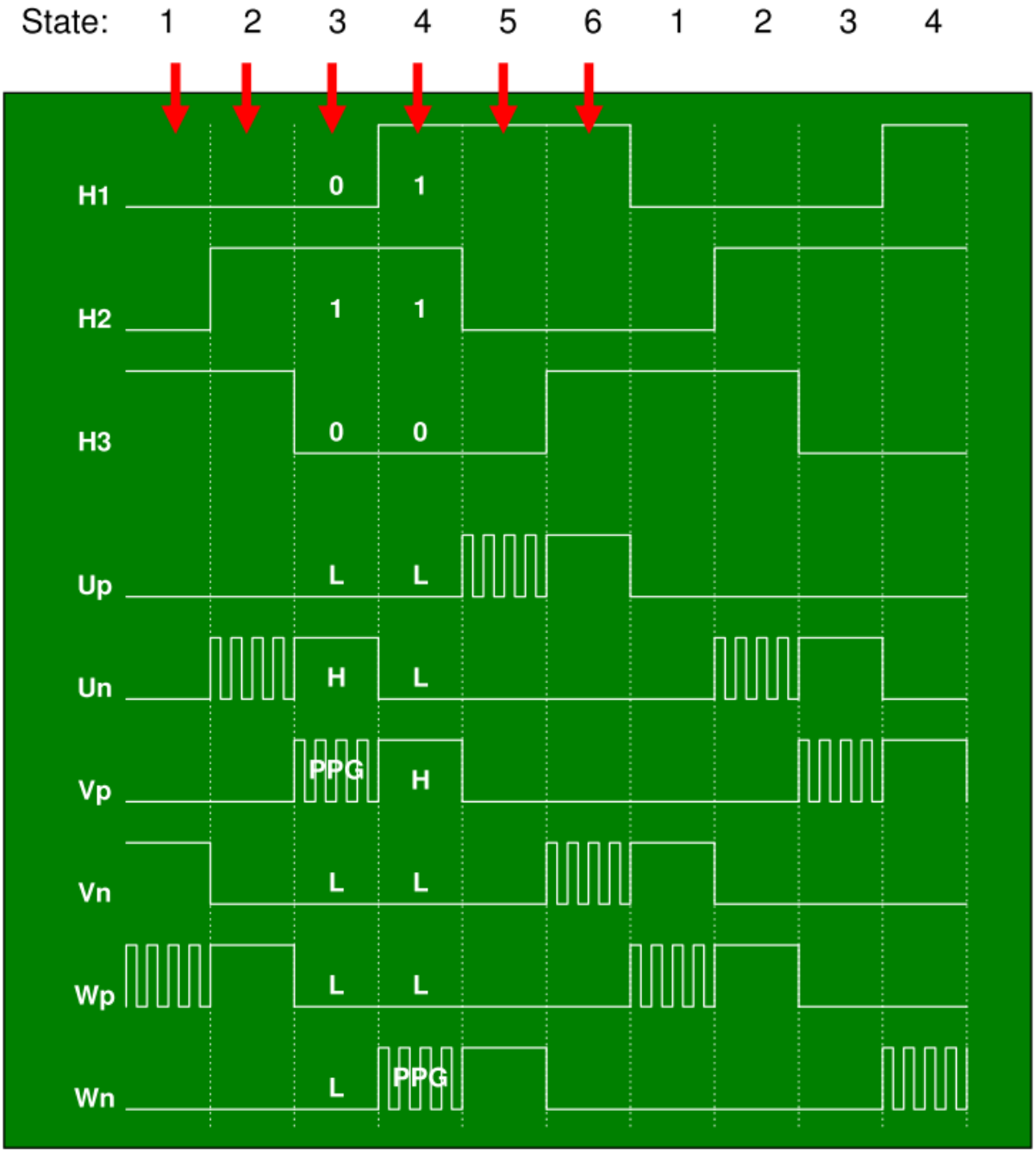

Figure 4 shows the relationship between three-channel Hall sensor signals and six-channel inverter switch signals.

2.2.2 MPG Macro Setting for 120° Motor Driver

1. Trigger signal setting for MPG macro

   The OPS2 to OPS0 are used to select the OPDR register write timing control operation mode. For example:

   - OPCR_OPS = 0;
     // Data transfer from OPDBR0 to OPDR after OPDBR0 is written by software.
   - OPCR_OPS = 1;
     // Data transfer from OPDBR to OPDR is triggered by the 16-bit reload timer underflow.
   - OPCR_OPS = 2;
     // Data transfer from OPDBR to OPDR is triggered by the position detection input.

   In this system, data transferred from OPDBR to OPDR should be triggered by the position detection input for Hall sensor 120° motor driver. For detailed information, please refer to Chapter 24 in Hardware Manual of MB95330H Series.

2. OPTx output waveform selection

   The OPTx pin outputs “L” level, “H” level or “PPG” waveform by setting OPDBR. For example:

   - OPDBLR_OP01 = 0;
   - OPDBLR_OP00 = 0;
     // Data Setting for OPT0 pin to output “L” level.
   - OPDBLR_OP01 = 0;
   - OPDBLR_OP00 = 1;
     // Setting for OPT0 pin to output the output of the PPG timer.
   - OPDBLR_OP01 = 1;
   - OPDBLR_OP00 = 0;
     // Setting for OPT0 pin to output the inverted output of the PPG timer.
   - OPDBLR_OP01 = 1;
   - OPDBLR_OP00 = 1;
     // Setting for OPT0 pin to output “H” level.

   For detailed information, please refer to Chapter 24 in Hardware Manual of MB95330H Series.

3. PPG synchronization setting for output waveform

   In order to avoid short pulse (or glitch) during sequencer state changes, the write timing (WTO) needs to be delayed and synchronized with the next coming edge of PPG output waveform. For example:

   - IPCUR_WTS1 = 1;
   - IPCUR_WTS0 = 1;

     // both edges synchronization. ↑ & ↓

   For detailed information, please refer to Chapter 24 in Hardware Manual of MB95330H Series.

4. Noise cancellation control setting

   When the noise cancellation function is selected, the time for fixing an output pin at the inactive level is delayed for about 4, 8, 16 or 32 machine clocks by the noise cancellation circuit. For example:

   - NCCR = 0xFF;

     // Cancel 32-cycle noise.

   For detailed information, please refer to Chapter 24 in Hardware Manual of MB95330H Series.

3 Usage of BLDC Motor

This section describes usage memory for BLDC motor.

3.1 Start Motor

```
/*---------------------------------------------------------------------*/
/* Name     :     void MPG_Initial(void)
/* Input    :     NO
/* Output   :     NO
/* Description:   MPG control and motor start
/*---------------------------------------------------------------------*/
void MPG_Control_Start(void)
{
      IPCUR = 0xC0;     // disable comparison operation,enalbe PPG synch
      IPCLR = 0xFF;     // enalbe SIN0~SIN2 edge detection
      NCCR = 0xFF;      // select 32 cycle noise
      OPCUR = 0xA4;     // 16 bit reload time trigger and enable DTTI
      OPCLR = 0x3F;
}
```

To start BLDC motor, use 16-bit reload timer underflow trigger for multi-pulse generator (MPG).

3.2 Stop Motor

To stop BLDC motor, MPG is switched to software trigger and the OPT5 to OPT0 output low level when motor is rotating.

```
/*---------------------------------------------------------------------*/
/* Name     :     void Motor_Stop(void)
/* Input    :     NO
/* Output   :     NO
/* Description:   motor stop work main function
/*---------------------------------------------------------------------*/
void Motor_Stop(void)
{
      OPCUR = 0xA0;     // software trigger driver motor and enable DTTI
      OPDBRH0 = 0x00;   // stop motor work
      OPDBRL0 = 0x00;
}
```

3.3 Speed UP and Speed Down

The motor speeds up by increasing PPG duty value while keeping PPG cycle value. The motor speeds down by decreasing PPG duty value while keeping PPG cycle value.

```
/*---------------------------------------------------------------------*/
/* Name     :     void Speed_Setting(void)
/* Input    :     NO
/* Output   :     NO
/* Description:   speed duty check and setting
/*---------------------------------------------------------------------*/
void Speed_Setting(void)
{
      if(Work_Status_Flag != Motor_Work_Status) return;
      if(Current_Duty == Duty_Min||Current_Duty == Duty_Max) return;
      if(Setting_Freq > Current_Freq){          // speed up
            if(Current_Duty < (Duty_Max-1)){
                  Current_Duty = Current_Duty+1;
            }
            else{
                  Current_Duty = Duty_Max;
            }
      }
      else if(Setting_Freq < Current_Freq){     // speed down
            if(Current_Duty > (Duty_Min+1)){
                  Current_Duty = Current_Duty-1;
            }
            else{
                  Current_Duty = Duty_Min;
            }
      }
      else{                                     // setting ok
            Current_Duty = Current_Duty;
      }
      PDUTH = Current_Duty>>8;                  // set ppg duty 200(16%)
      PDUTL = Current_Duty;
}
```

Notes: Current_Duty: current PPG duty value;

Setting_Freq: set motor rotate frequency;

Current_Freq: current motor rotating frequency;

3.4 Clockwise and Count Clockwise

The motor rotates clockwise or counts clockwise by setting different OPDBUR0 to OPDBUR5 switch sequence.

```
/*---------------------------------------------------------------------*/
/* Name     :     void MPG_Clockwise_Initial(void)
/* Input    :     NO
/* Output   :     NO
/* Description:   Motor rotate clockwise or count clockwise setting
/*---------------------------------------------------------------------*/
void MPG_Clockwise_Initial(void)
{
      if(Motor_Reverse_Flag == 0){        // clockwise
            OPDBRH0 = 0x10;               // 1 step
            OPDBRL0 = 0x34;
            OPDBRH1 = 0x21;               // 2 step
            OPDBRL1 = 0x0C;
            OPDBRH2 = 0x33;               // 3 step
            OPDBRL2 = 0x40;
            OPDBRH3 = 0x40;               // 4 step
            OPDBRL3 = 0xC1;
            OPDBRH4 = 0x54;               // 5 step
            OPDBRL4 = 0x03;
            OPDBRH5 = 0x0C;               // 6 step
            OPDBRL5 = 0x10;
      }
      else{                               //count clockwise
            OPDBRH0 = 0x50;               // 1 step
            OPDBRL0 = 0x34;
            OPDBRH1 = 0x01;               // 2 step
            OPDBRL1 = 0x0C;
            OPDBRH2 = 0x13;               // 3 step
            OPDBRL2 = 0x40;
            OPDBRH3 = 0x20;               // 4 step
            OPDBRL3 = 0xC1;
            OPDBRH4 = 0x34;               // 5 step
            OPDBRL4 = 0x03;
            OPDBRH5 = 0x4C;               // 6 step
            OPDBRL5 = 0x10;
      }
}
```

4 Notes on Using BLDC Motor Hall Sensor 120° Driver

The section describes notes on using BLDC motor Hall sensor 120° driver.

4.1 Notes on Relation of Feedback Signal with OPTx Waveform

The relation of feedback signal with OPTx output waveform varies depending on BLDC motor, so user should first check the relation of them, and then set registers in code.

4.2 Notes on Starting Motor

To use Hall sensor 120° drive BLDC motor, first, set 16-bit reload timer trigger to start BLDC motor and then switch to position detection trigger to drive motor when the BLDC motor is rotated normally.

4.3 Notes on Operation of DTTI Input Control

In system, DTTI over-current protection should be enabled. The OPT5 to OPT0 are fixed at inactive level when the low input level is placed at the DTTI pin.

Even while the output is fixed at the inactive level by the input of the DTTI pin, the timer keeps running, the position detection function does not stop and the data transfer from the output data buffer register (OPDBR) to the output data register (OPDR) is continued for waveform generation, but no waveform is output to the OPT5 to OPT0 pins.

5 Performance Investigation

The section describes performance investigation for motor driver by MB95200 series.

The system operation includes motor speed setting, motor rotating speed measurement, PPG duty adjusting (speed adjusting), key process and feedback single detection, and the system performance is shown as below:

- Running period time: 51.8 us (518 cycles).
- ROM size: 2157bit.
- RAM size: 112bit.

Notes:

The system MCLK (machine clock) = 10 MHZ.

6 Additional Information

For more information on Cypress MB95200 products, please visit following website:

http://www.cypress.com/MB95200

Document History

Document Title: AN205559 – F²MC-8FX Family MB95200 Series 8-Bit Microcontroller BLDC Motor Hall Sensor 120° Driver Method

Document Number: 002-05559

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	CBZH	04/23/2009	Initial release
			05/06/2009	Modify according Document Feedback
*A	5265921	CBZH	05/10/2016	Migrated Spansion Application Note MCU-AN-500041-E-11 to Cypress format.
*B	5842086	AESATMP9	08/02/2017	Updated logo and copyright.

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

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

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