Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	LINbusSCI, SPI
Peripherals	LVD, Motor Control PWM, POR, PWM, WDT
Number of I/O	44
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fmc2r7t6

MAIN CLOCK CONTROLLER WITH REAL-TIME CLOCK (Cont'd)

6.4.5 Low Power Modes

Mode	Description
Wait	No effect on MCC/RTC peripheral. MCC/RTC interrupt cause the device to exit from Wait mode.
Active-halt	No effect on MCC/RTC counter (OIE bit is set), the registers are frozen. MCC/RTC interrupt cause the device to exit from Active-halt mode.
Halt	MCC/RTC counter and registers are frozen. MCC/RTC operation resumes when the MCU is woken up by an interrupt with "exit from Halt" capability.

6.4.6 Interrupts

The MCC/RTC interrupt event generates an interrupt if the OIE bit of the MCCSR register is set and the interrupt mask in the CC register is not active (RIM instruction).

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Time base overflow event	OIF	OIE	Yes	No [1]

Note:

The MCC/RTC interrupt wakes up the MCU from Active-halt mode, not from Halt mode.

6.4.7 Register Description

MCC CONTROL/STATUS REGISTER (MCCSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
MCO	CP1	CP0	SMS	TB1	TB0	OIE	OIF

Bit 7 = **MCO** *Main clock out selection*
This bit enables the MCO alternate function on the PF0 I/O port. It is set and cleared by software.
0: MCO alternate function disabled (I/O pin free for general-purpose I/O)
1: MCO alternate function enabled (f_{OSC2} on I/O port)

Note: To reduce power consumption, the MCO function is not active in Active-halt mode.

Bit 6:5 = **CP[1:0]** *CPU clock prescaler*
These bits select the CPU clock prescaler which is applied in the different slow modes. Their action is conditioned by the setting of the SMS bit. These two bits are set and cleared by software

f_{CPU} in Slow mode	CP1	CP0
f_{OSC2} / 2	0	0
f_{OSC2} / 4	0	1
f_{OSC2} / 8	1	0
f_{OSC2} / 16	1	1

Bit 4 = **SMS** *Slow mode select*
This bit is set and cleared by software.
0: Normal mode. $f_{CPU} = f_{OSC2}$
1: Slow mode. f_{CPU} is given by CP1, CP0
See Section 8.2 SLOW MODE and Section 6.4 MAIN CLOCK CONTROLLER WITH REAL-TIME CLOCK AND BEEPER (MCC/RTC) for more details.

Bit 3:2 = **TB[1:0]** *Time base control*

These bits select the programmable divider time base. They are set and cleared by software.

Counter Prescaler	Time Base		TB1	TB0
	f_{OSC2} =4MHz	f_{OSC2}=8MHz		
16000	4ms	2ms	0	0
32000	8ms	4ms	0	1
80000	20ms	10ms	1	0
200000	50ms	25ms	1	1

A modification of the time base is taken into account at the end of the current period (previously set) to avoid an unwanted time shift. This allows to use this time base as a real-time clock.

Bit 1 = **OIE** *Oscillator interrupt enable*
This bit set and cleared by software.
0: Oscillator interrupt disabled
1: Oscillator interrupt enabled
This interrupt can be used to exit from Active-halt mode.
When this bit is set, calling the ST7 software HALT instruction enters the Active-halt power saving mode.

INTERRUPTS (Cont'd)

7.3 INTERRUPTS AND LOW POWER MODES

All interrupts allow the processor to exit the Wait low power mode. On the contrary, only external and other specified interrupts allow the processor to exit from the Halt modes (see column "Exit from Halt" in "Interrupt Mapping" table). When several pending interrupts are present while exiting Halt mode, the first one serviced can only be an interrupt with exit from Halt mode capability and it is selected through the same decision process shown in Figure 21.

Note: If an interrupt, that is not able to Exit from Halt mode, is pending with the highest priority when exiting Halt mode, this interrupt is serviced after the first one serviced.

7.4 CONCURRENT & NESTED MANAGEMENT

The following Figure 22 and Figure 23 show two different interrupt management modes. The first is called concurrent mode and does not allow an interrupt to be interrupted, unlike the nested mode in Figure 23. The interrupt hardware priority is given in this order from the lowest to the highest: MAIN, IT4, IT3, IT2, IT1, IT0, MCES. The software priority is given for each interrupt.

Warning: A stack overflow may occur without notifying the software of the failure.

Figure 22. Concurrent Interrupt Management

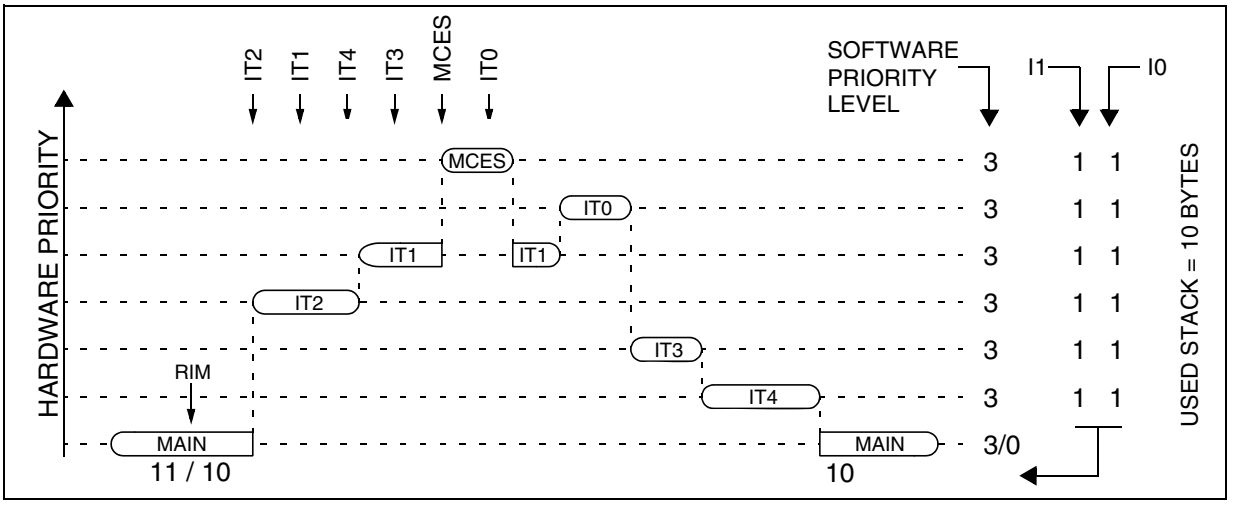
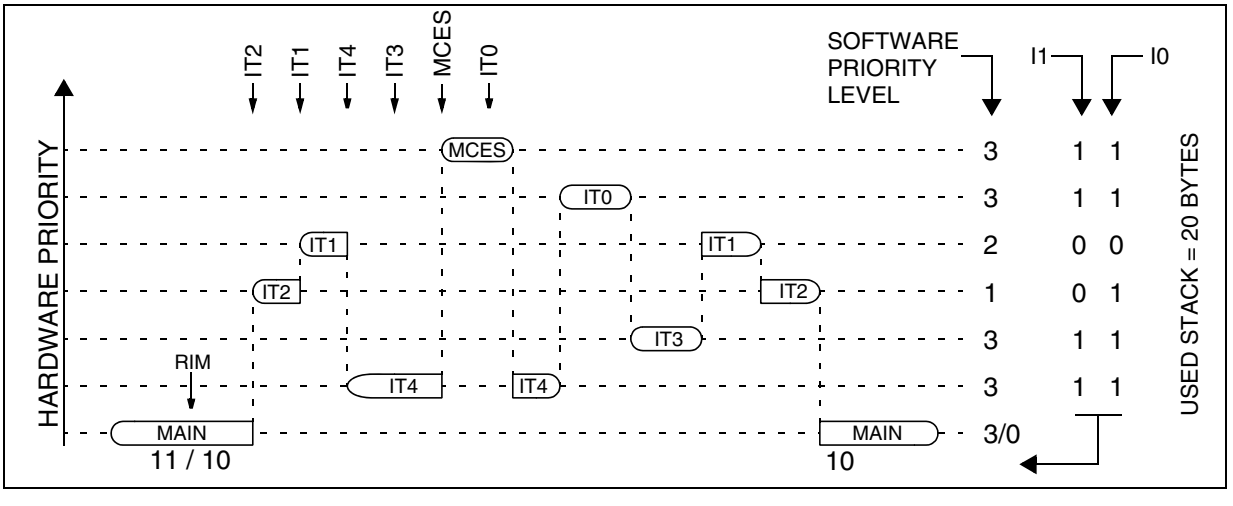


Figure 23. Nested Interrupt Management



8 POWER SAVING MODES

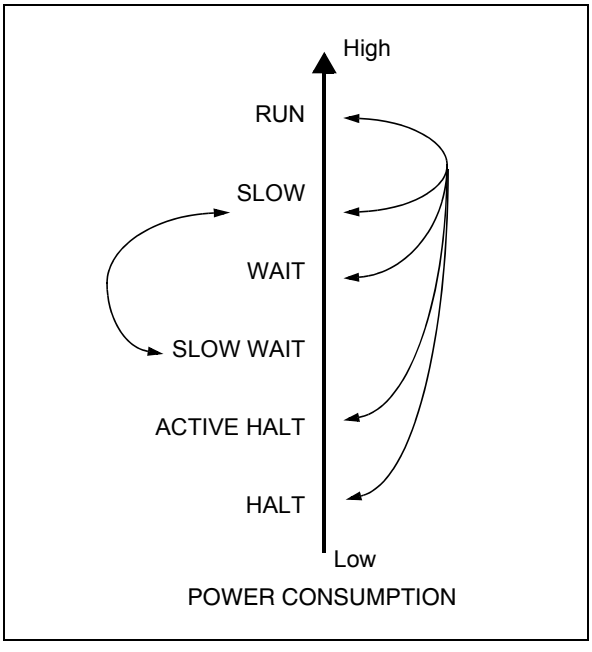
8.1 INTRODUCTION

To give a large measure of flexibility to the application in terms of power consumption, four main power saving modes are implemented in the ST7 (see Figure 25): Slow, Wait (Slow-wait), Active-halt and Halt.

After a RESET the normal operating mode is selected by default (Run mode). This mode drives the device (CPU and embedded peripherals) by means of a master clock which is based on the main oscillator frequency divided or multiplied by 2 (f_{OSC2}).

From Run mode, the different power saving modes may be selected by setting the relevant register bits or by calling the specific ST7 software instruction whose action depends on the oscillator status.

Figure 25. Power Saving Mode Transitions



8.2 SLOW MODE

This mode has two targets:

– To reduce power consumption by decreasing the internal clock in the device,

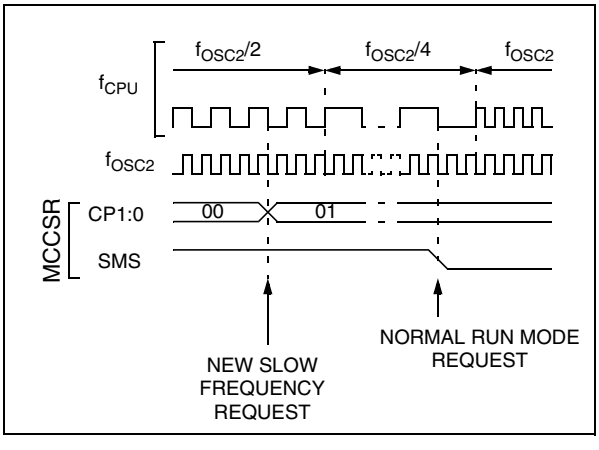
– To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

Slow mode is controlled by three bits in the MCCSR register: the SMS bit which enables or disables Slow mode and two CPx bits which select the internal slow frequency (f_{CPU}).

In this mode, the master clock frequency (f_{OSC2}) can be divided by 2, 4, 8 or 16. The CPU and peripherals are clocked at this lower frequency (f_{CPU}).

Note: Slow-wait mode is activated when entering the Wait mode while the device is already in Slow mode.

Figure 26. Slow Mode Clock Transitions



WINDOW WATCHDOG (Cont'd)

The application program must write in the WDGCR register at regular intervals during normal operation to prevent an MCU reset. This operation must occur only when the counter value is lower than the window register value. The value to be stored in the WDGCR register must be between FFh and C0h (see Figure 35):

– Enabling the watchdog:
When Software Watchdog is selected (by option byte), the watchdog is disabled after a reset. It is enabled by setting the WDGA bit in the WDGCR register, then it cannot be disabled again except by a reset.

When Hardware Watchdog is selected (by option byte), the watchdog is always active and the WDGA bit is not used.

– Controlling the downcounter:
This downcounter is free-running: It counts down even if the watchdog is disabled. When the watchdog is enabled, the T6 bit must be set to prevent generating an immediate reset.
The T[5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset (see Figure 35. Approximate Timeout Duration). The timing varies between a minimum and a maximum value due to the unknown status of the prescaler when writing to the WDGCR register (see Figure 36).

The window register (WDGWR) contains the high limit of the window: To prevent a reset, the downcounter must be reloaded when its value is lower than the window register value and greater than 3Fh. Figure 37 describes the window watchdog process.

Note: The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

– Watchdog Reset on Halt option
If the watchdog is activated and the watchdog reset on halt option is selected, then the HALT instruction will generate a Reset.

10.1.4 Using Halt Mode with the WDG

If Halt mode with Watchdog is enabled by option byte (no watchdog reset on HALT instruction), it is recommended before executing the HALT instruction to refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller.

WINDOW WATCHDOG (Cont'd)

10.1.9 Interrupts

None.

10.1.10 Register Description

CONTROL REGISTER (WDGCR)

Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0

Bit 7 = **WDGA** *Activation bit.*

This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.

0: Watchdog disabled

1: Watchdog enabled

Note: This bit is not used if the hardware watchdog option is enabled by option byte.

Bits 6:0 = **T[6:0]** *7-bit counter (MSB to LSB).*
These bits contain the value of the watchdog counter. It is decremented every 16384 f_{OSC2} cycles (approx.). A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

WINDOW REGISTER (WDGWR)

Read/Write

Reset Value: 0111 1111 (7Fh)

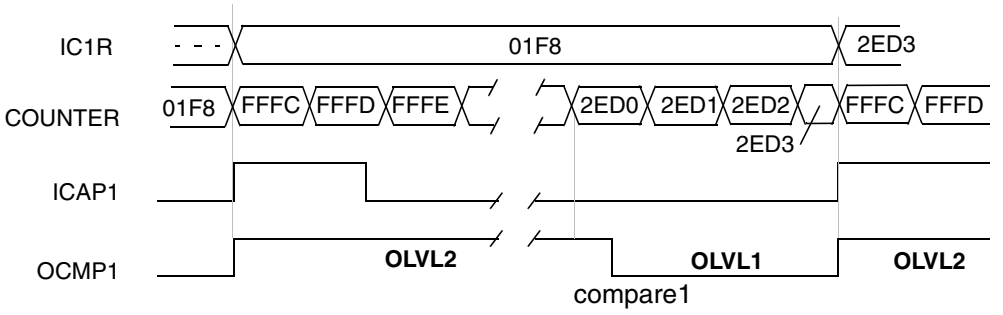
7							0
-	W6	W5	W4	W3	W2	W1	W0

Bit 7 = Reserved

Bits 6:0 = **W[6:0]** *7-bit window value*
These bits contain the window value to be compared to the downcounter.

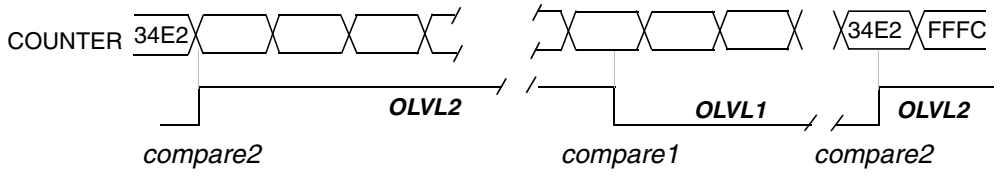
16-BIT TIMER (Cont'd)

Figure 53. One Pulse Mode Timing Example



Note: IEDG1=1, OC1R=2ED0h, OLVL1=0, OLVL2=1

Figure 54. Pulse Width Modulation Mode Timing Example



Note:

10.3.4 Low Power Modes

Wait	No effect on 16-bit Timer. Timer interrupts cause the Device to exit from Wait mode.
Halt	16-bit Timer registers are frozen. In Halt mode, the counter stops counting until Halt mode is exited. Counting resumes from the previous count when the Device is woken up by an interrupt with “exit from Halt mode” capability or from the counter reset value when the Device is woken up by a RESET. If an input capture event occurs on the ICAP*i* pin, the input capture detection circuitry is armed. Consequently, when the Device is woken up by an interrupt with “exit from Halt mode” capability, the ICF*i* bit is set, and the counter value present when exiting from Halt mode is captured into the IC*i*R register.

10.3.5 Interrupts

Input Capture 1 event/Counter reset in PWM mode	ICF1	ICIE	Yes	No
Input Capture 2 event	ICF2		Yes	No
Output Compare 1 event (not available in PWM mode)	OCF1	OCIE	Yes	No
Output Compare 2 event (not available in PWM mode)	OCF2		Yes	No
Timer Overflow event	TOF	TOIE	Yes	No

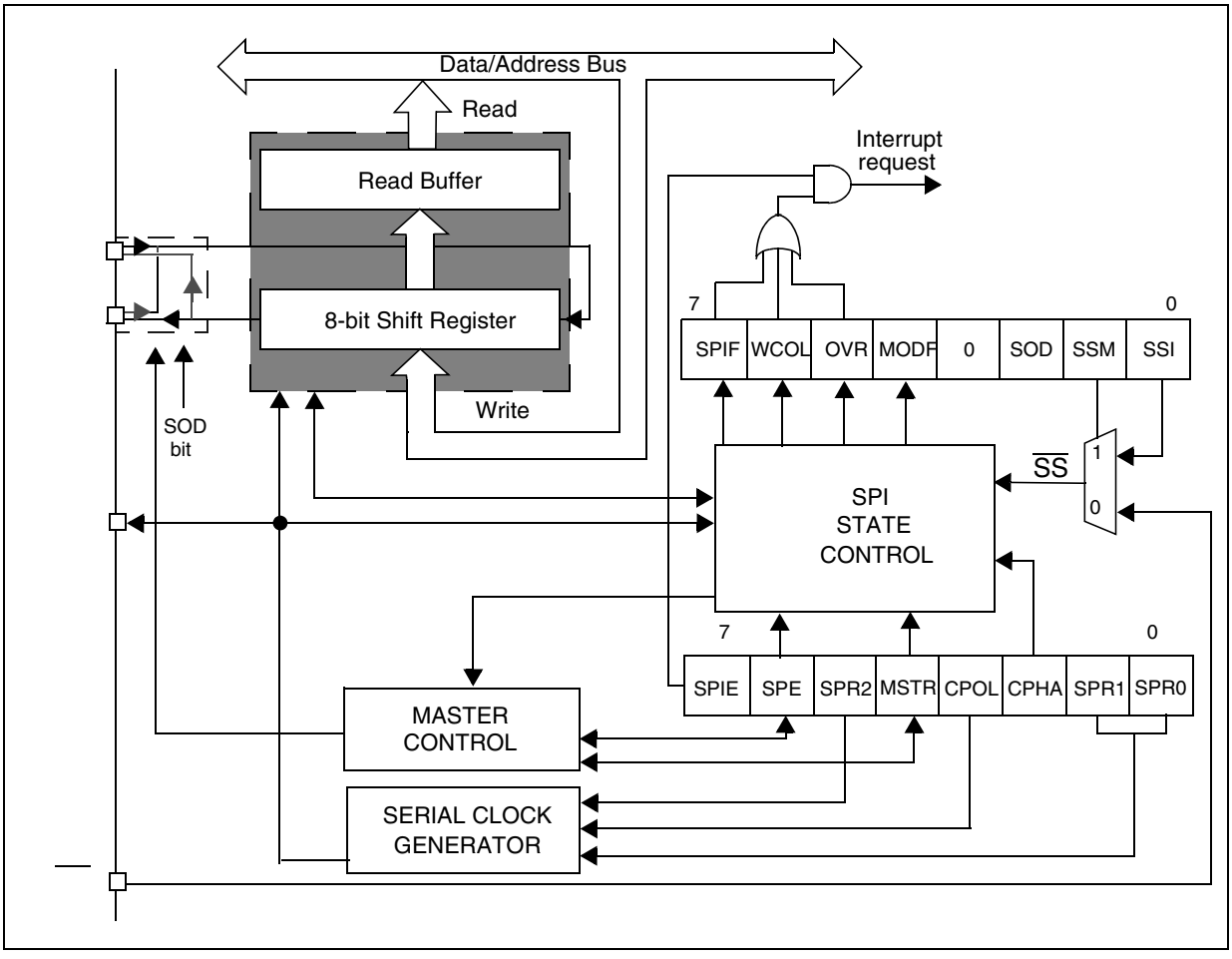
Note:

10.3.6 Summary of Timer modes

Input Capture (1 and/or 2)	Yes	Yes	Yes	Yes
Output Compare (1 and/or 2)	Yes	Yes	Yes	Yes
One Pulse Mode	No	Not Recommended[1)]	No	Partially [2)]
PWM Mode	No	Not Recommended[3)]	No	No

[3)] See note 4 in Section 10.3.3.6 Pulse Width Modulation Mode

(cont'd)


Data/Address Bus
Read
Read Buffer
8-bit Shift Register
Write
SOD
bit
Interrupt
request
7
0
SPIF
WCOL
OVR
MODF
0
SOD
SSM
SSI
SS
1
0
SPI
STATE
CONTROL
7
0
SPIE
SPE
SPR2
MSTR
CPOL
CPHA
SPR1
SPR0
MASTER
CONTROL
SERIAL CLOCK
GENERATOR

(cont'd)

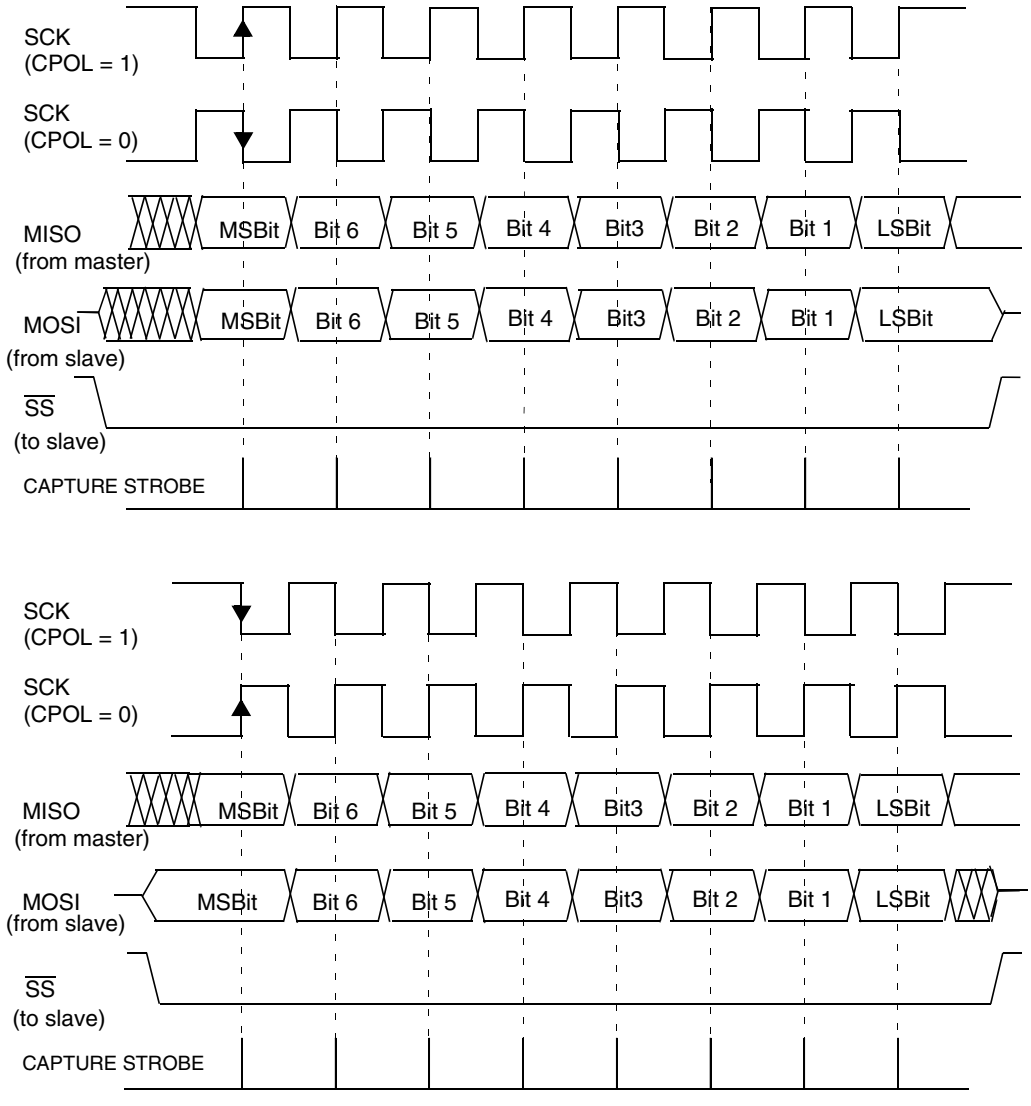
Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits (See Figure 59).

The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL = 1 or pulling down SCK if CPOL = 0).

The combination of the CPOL clock polarity and CPHA (clock phase) bits selects the data capture clock edge.

Figure 59 shows an SPI transfer with the four combinations of the CPHA and CPOL bits. The diagram may be interpreted as a master or slave timing diagram where the SCK pin, the MISO pin and the MOSI pin are directly connected between the master and the slave device.

: If CPOL is changed at the communication byte boundaries, the SPI must be disabled by resetting the SPE bit.



Note: This figure should not be used as a replacement for parametric information. Refer to the Electrical Characteristics chapter.

SERIAL PERIPHERAL INTERFACE (cont'd)

10.4.5.4 Single Master and Multimaster Configurations

There are two types of SPI systems:

– Single Master System

– Multimaster System

Single Master System

A typical single master system may be configured using a device as the master and four devices as slaves (see Figure 61).

The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices.

The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices.

Note: To prevent a bus conflict on the MISO line, the master allows only one active slave device during a transmission.

For more security, the slave device may respond to the master with the received data byte. Then the master will receive the previous byte back from the slave device if all MISO and MOSI pins are connected and the slave has not written to its SPIDR register.

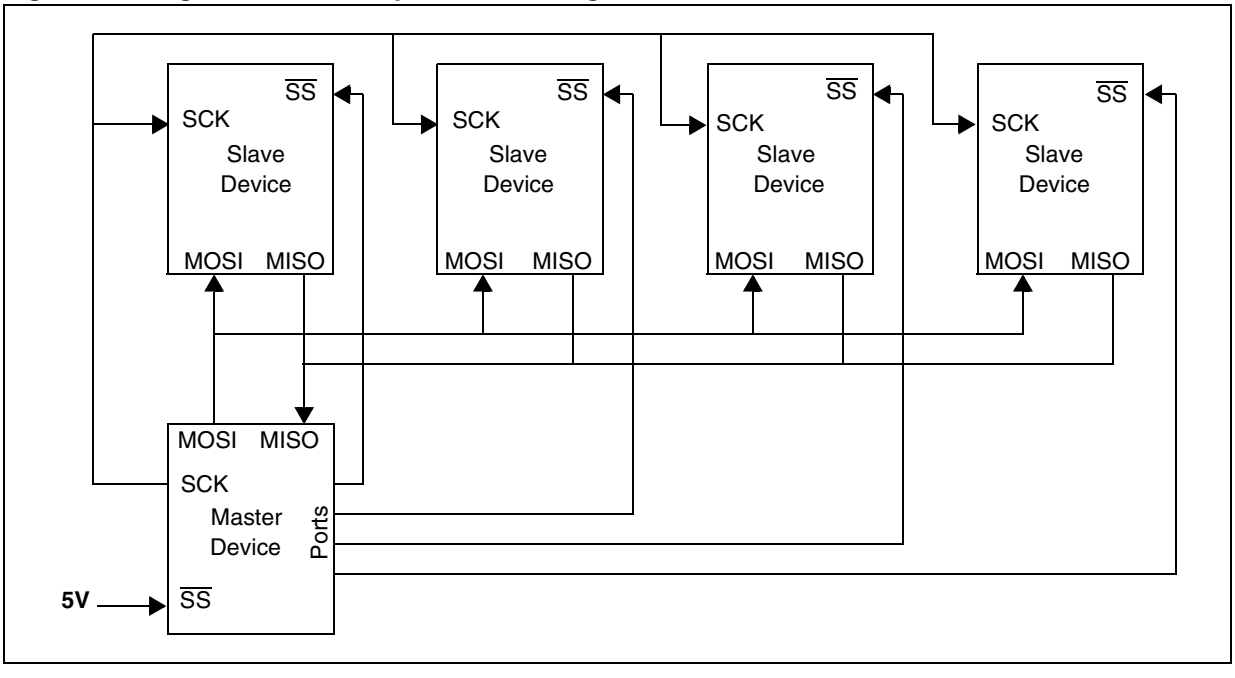
Other transmission security methods can use ports for handshake lines or data bytes with command fields.

Multimaster System

A multimaster system may also be configured by the user. Transfer of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system.

The multimaster system is principally handled by the MSTR bit in the SPICR register and the MODF bit in the SPICSR register.

Figure 61. Single Master / Multiple Slave Configuration



LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Mode) (cont'd)

10.5.9.3 LIN Reception

In LIN mode the reception of a byte is the same as in SCI mode but the LINSCI has features for handling the LIN Header automatically (identifier detection) or semiautomatically (Synch Break detection) depending on the LIN Header detection mode. The detection mode is selected by the LHDM bit in the SCICR3.

Additionally, an automatic resynchronization feature can be activated to compensate for any clock deviation, for more details please refer to Section 10.5.9.5 LIN Baud Rate.

LIN Header Handling by a Slave

Depending on the LIN Header detection method the LINSCI will signal the detection of a LIN Header after the LIN Synch Break or after the Identifier has been successfully received.

Note:

It is recommended to combine the Header detection function with Mute mode. Putting the LINSCI in Mute mode allows the detection of Headers only and prevents the reception of any other characters.

This mode can be used to wait for the next Header without being interrupted by the data bytes of the current message in case this message is not relevant for the application.

Synch Break Detection (LHDM = 0):

When a LIN Synch Break is received:

– The RDRF bit in the SCISR register is set. It indicates that the content of the shift register is transferred to the SCIDR register, a value of 0x00 is expected for a Break.

– The LHDF flag in the SCICR3 register indicates that a LIN Synch Break Field has been detected.

– An interrupt is generated if the LHIE bit in the SCICR3 register is set and the I[1:0] bits are cleared in the CCR register.

– Then the LIN Synch Field is received and measured.

  – If automatic resynchronization is enabled (LASE bit = 1), the LIN Synch Field is not transferred to the shift register: There is no need to clear the RDRF bit.

  – If automatic resynchronization is disabled (LASE bit = 0), the LIN Synch Field is received as a normal character and transferred to the SCIDR register and RDRF is set.

Note:

In LIN slave mode, the FE bit detects all frame error which does not correspond to a break.

Identifier Detection (LHDM = 1):

This case is the same as the previous one except that the LHDF and the RDRF flags are set only after the entire header has been received (this is true whether automatic resynchronization is enabled or not). This indicates that the LIN Identifier is available in the SCIDR register.

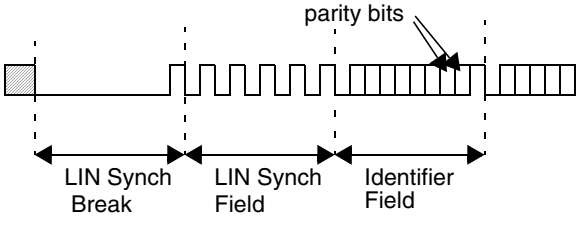
Notes:

During LIN Synch Field measurement, the SCI state machine is switched off: No characters are transferred to the data register.

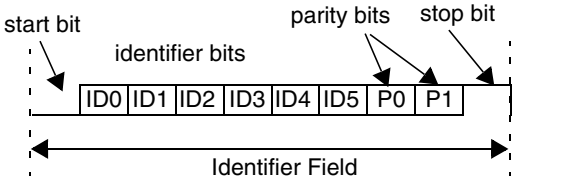
LIN Slave parity

In LIN Slave mode (LINE and LSLV bits are set) LIN parity checking can be enabled by setting the PCE bit.

In this case, the parity bits of the LIN Identifier Field are checked. The identifier character is recognized as the third received character after a break character (included):



The bits involved are the two MSB positions (7th and 8th bits if M = 0; 8th and 9th bits if M = 0) of the identifier character. The check is performed as specified by the LIN specification:



$$P0 = ID0 \oplus ID1 \oplus ID2 \oplus ID4$$

$$P1 = \overline{ID1 \oplus ID3 \oplus ID4 \oplus ID5}$$

M = 0

MOTOR CONTROLLER (Cont'd)

10.6.6.5 Demagnetization (D) Event

At the end of the demagnetization phase, current no longer goes through the free-wheeling diodes. The voltage on the non-excited winding terminal goes from one of the power rail voltages to the common star connection voltage plus the BEMF voltage. In some cases (if the BEMF voltage is positive and the free-wheeling diodes are at ground for example) this end of demagnetization can be seen as a voltage edge on the selected MCIx input and it is called a hardware demagnetization event D_H. See Table 30.

The D event filter can be used to select the number of consecutive D events needed to generate the D_H event.

If enabled by the HDM bit in the MCRB register, the current value of the MTIM timer is captured in register MDREG when this event occurs in order to be able to simulate the demagnetization phase for the next steps.

When enabled by the SDM bit in the MCRB register, demagnetization can also be simulated by comparing the MTIM timer with the MDREG register. This kind of demagnetization is called simulated demagnetization D_S.

If the HDM and SDM bits are both set, the first event that occurs, triggers a demagnetization event. For this to work correctly, a D_S event must not precede a D_H event because the latter could be detected as a Z event.

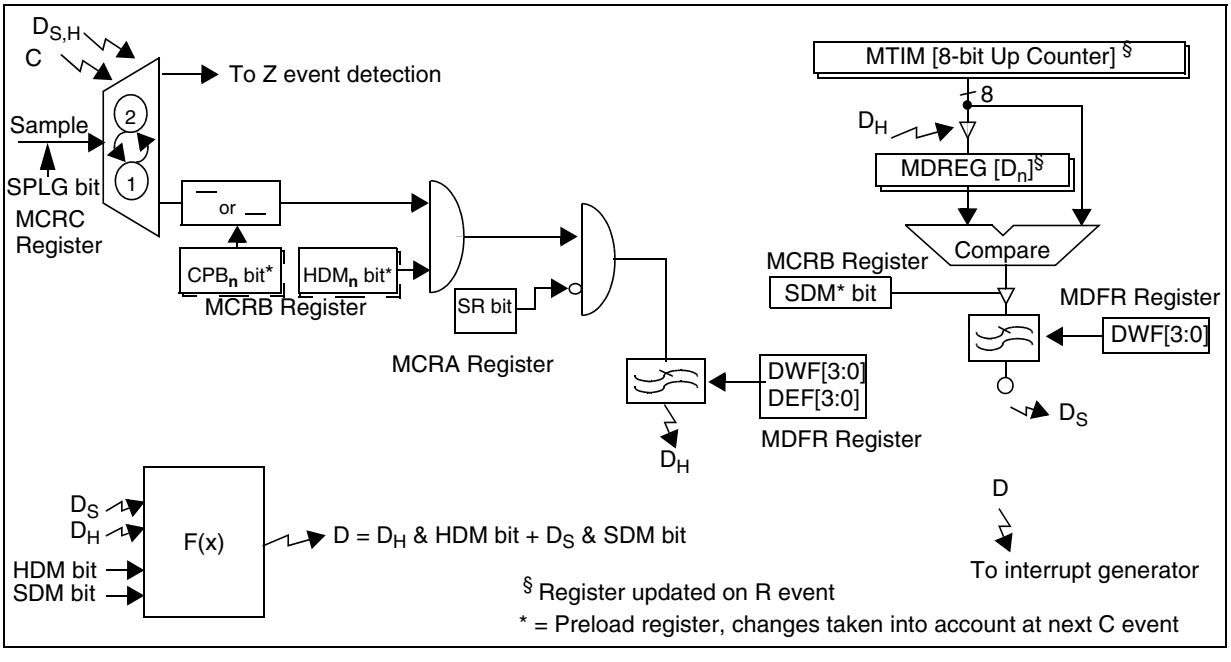
Simulated demagnetization can also be always used if the HDM bit is reset and the SDM bit is set. This mode works as a programmable masking time between the C_H and Z events. To drive the motor securely, the masking time must be always greater than the real demagnetization time in order to avoid a spurious Z event.

When an event occurs, (either D_H or D_S) the DI bit in the MISR register is set and an interrupt request is generated if the DIM bit of register MIMR is set.

Caution 1: Due to the alternate automatic capture and compare of the MTIM timer with MDREG register by D_H and D_S events, the MDREG register should be manipulated with special care.

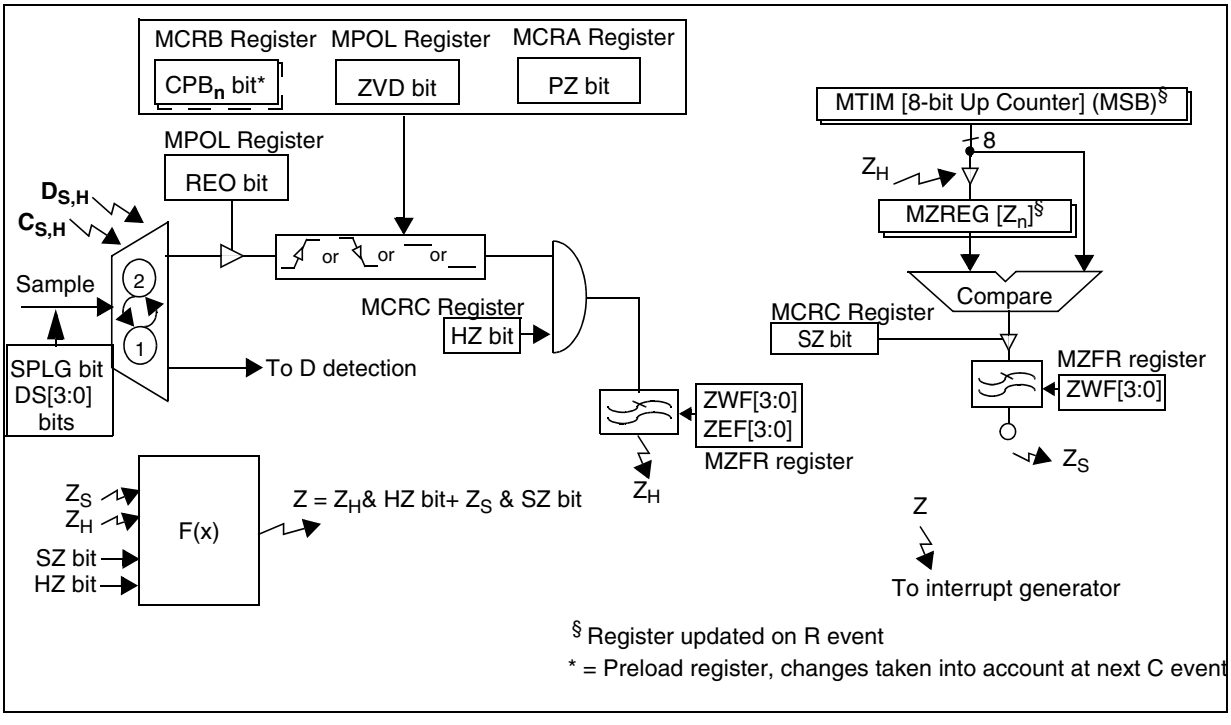
Caution 2: Due to the event generation protection in the MZREG, MCOMP and MDREG registers for Soft Event generation (See "Built-in Checks and Controls for simulated events" on page 175.), the value written in the MDREG register in soft demagnetisation mode (SDM=1) is checked by hardware after the C event. If this value is less than or equal to the MTIM counter value at this moment, the Software demagnetisation event is generated immediately and the MTIM current value overwrites the value in the MDREG register to be able to reuse the right demagnetisation time for another simulated event generation.

Figure 81. D Event Generation Mechanism



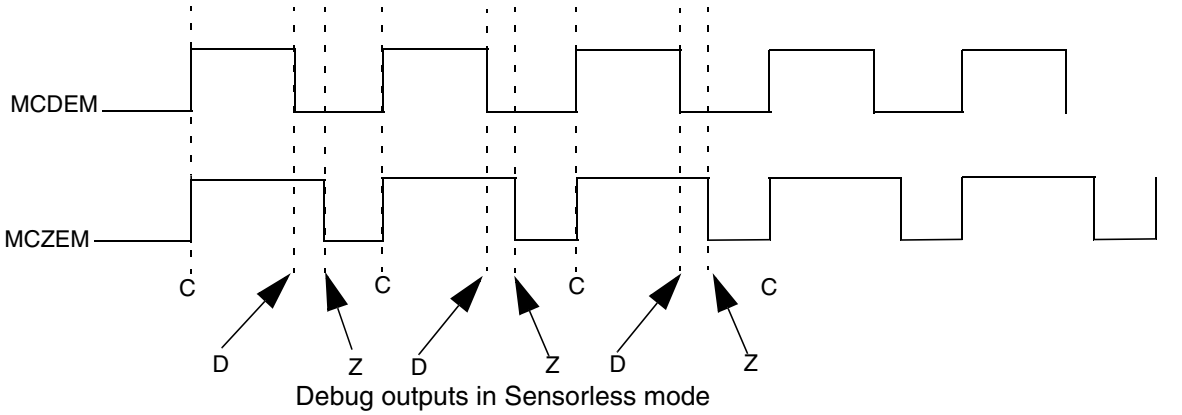
MOTOR CONTROLLER (Cont'd)

Figure 82. Z Event Generation

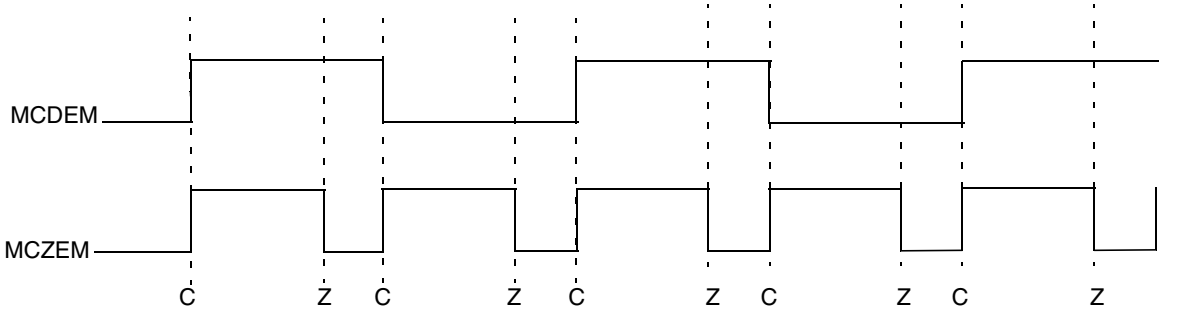


MOTOR CONTROLLER (Cont'd)

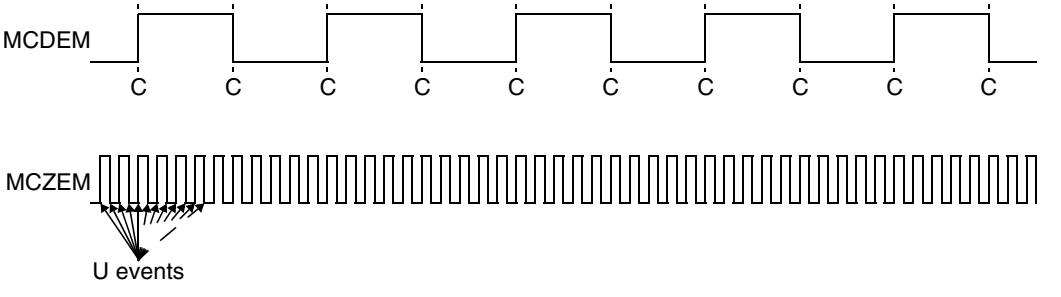
Figure 94. Output on pins MCDEM and MCZEM with debug option (DG bit=1)



Debug outputs in Sensorless mode



Debug outputs in Sensor mode



Debug outputs in Speed Measurement mode (TES[1:0] bits equal to 01, 10 or 11).

MOTOR CONTROLLER (Cont'd)

If the PCN bit is reset, one of the three PWM signals (the one set by the compare U register pair) or the output of the measurement window generator (depending on if the driving mode is voltage or current) is used to provide six-step signals through the PWM manager (to drive a PM BLDC motor for instance).

In that case, DTE behaves like a standard bit (with multiple write capability). When the deadtime generator is enabled (bit DTE=1), some restrictions are applied, summarized in Table 53:

- Channels are now grouped by pairs: Channel[0:1], Channel[2:3], Channel[4:5]; a deadtime generator is allocated to each of these pairs (see cautions below);
- The input signal of the deadtime generator is the active output of the PWM manager for the corresponding channel. For instance, if we consider the Channel[0:1] pair, it may be either Channel0 or Channel1.
- When both channels of a pair are inactive, the corresponding outputs will also stay inactive (this is mandatory to allow BEMF zero-crossing detection).

Table 53 summarizes the functionality of the deadtime generator when the PCN bit is reset. 1(pwm*) means that the corresponding channel is active (1 in the corresponding bit in the MPHST register), and a PWM signal is applied on it (using the MPAR register and the OS[2:0] bits in MCRB register). $\overline{PWM}$ represents the complementary signals (although the duty cycle is slightly different due to deadtime insertion). 0 means that the channel is inactive and 1 means that the channel is active and a logic level 1 is applied on it (no PWM signal).

Table 53. Dead Time generator outputs

PCN = 0; DTE =1; x= 0, 2, 4			
On/Off x (OOx bit)	**On/Off x+1 (OOx+1 bit)**	**MCOx output**	**MCOx+1 output**
0	1 (pwm*)	$\overline{PWM}$	PWM
1 (pwm*)	0	PWM	$\overline{PWM}$
1	1 (pwm*)	0	0
1 (pwm*)	1	0	0
1	0	1	0
0	1	0	1
0	0	0	0

* PWM generation enabled

Warning: Grouping channels by pairs imposes the external connections between the MCO outputs and power devices; the user must therefore pay attention to respect the "recommended schematics" described in Figure 123. on page 228 and Figure 124

Note: As soon as the channels are grouped in pairs, special care has to be taken in configuring the MPAR register for a PM BLDC drive. If both channels of the same pair are both labelled "high" for example and if the PWM is applied on high channels, the active MCO output x (OOx=1 bit in the MPHST register) outputs PWM and the paired MCO output x+1 (OOx+1bit in the MPHST register) outputs $\overline{PWM}$ and vice versa.

Caution: When PCN=0 and a complementary PWM is applied (DTE=1) on one channel of a pair, if both channels are active, this corresponds in output to both channels OFF. This is for security purpose to avoid cross-conduction.

Caution: To clear the DTE bit from reset state of MDTG register (FFh), the PCN bit must be cleared before.

MOTOR CONTROLLER (Cont'd)

Table 62. PWM mode when SR=1

OS2 bit	PWM after C and before Z	OS1 bit	Unused	OS0	PWM after Z and before next C
0	On High Channels	x	x	0	On high channels
				1	On low channels
1	On Low Channels	x	x	0	On high channels
				1	On low channels

Table 63. PWM mode when DAC=1

OS2 bit	Unused	OS1 bit	Unused	OS0	PWM on outputs
x	x	x	x	0	On high channels
				1	On low channels

Warning: As the MCRB register contains preload bits with, it has to be written as a complete byte. A Bit Set or Bit Reset instruction on a non-preload bit will have the effect of resetting all the preload bits.

CONTROL REGISTER C (MCRC)
Read/Write (except EDIR bit)
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SEI / OI	EDIR/ HZ	SZ	SC	SPLG	VR2	VR1	VR0

Bit 7= **SEI/OI**: *Speed Error interrupt flag / MTIM Overflow flag*

Position Sensor or Sensorless mode (TES[1:0] bits =00)**:**

OI: *MTIM Overflow flag*
This flag signals an overflow of the MTIM timer. It has to be cleared by software.
0: No MTIM timer overflow
1: MTIM timer overflow

Note: No interrupt is associated with this flag

Speed Sensor mode (TES[1:0] bits =01, 10, 11)**:**

SEI: *Speed error interrupt flag*
0: No Tacho Error interrupt pending
1: Tacho Error interrupt pending

Bit 6= **EDIR/HZ** : *Encoder Direction bit/ Hardware zero-crossing event bit*

Position Sensor or Sensorless mode (TES[1:0] bits =00)**:**

HZ: *Hardware zero-crossing event bit*
This Read/Write bit selects if the Z event is hardware or not.
0: No hardware zero-crossing event
1: Hardware zero-crossing event

Speed Sensor mode (TES[1:0] bits =01, 10, 11)**:**

EDIR:*Encoder Direction bit*
This bit is Read only. As the rotation direction depends on encoder outputs and motor phase connections, this bit cannot indicate absolute direction. It therefore gives the relative phase-shift (i.e. advance/delay) between the two signals in quadrature output by the encoder (see Figure 90).
0: MCIA input delayed compared to MCIB input.
1: MCIA input in advance compared to MCIB input

Bit 5 = **SZ**: *Simulated zero-crossing event bit*
0: No simulated zero-crossing event
1: Simulated zero-crossing event

Bit 4 = **SC**: *Simulated commutation event bit*
0: Hardware commutation event in auto-switched mode (SWA = 1 in MCRA register)
1: Simulated commutation event in auto-switched mode (SWA = 1 in MCRA register).

Bit 3 = **SPLG**: *Sampling Z event at high frequency in sensorless mode (SR=0)*

This bit enables sampling at high frequency in sensorless mode independently of the PWM signal or only during ON time if the DS[3:0] bits in the MCONF register contain a value. Refer to Table 77, "Sampling Delay," on page 224
0: Normal mode (Z sampling at PWM frequency at the end of the off time)
1: Z event sampled at f_{SCF} (see Table 82)

Note: When the SPLG bit is set, there is no minimum OFF time programmed by the OT [3:0] bits, the OFF time is forced to 0µs. This means that in current mode, the OFF time of the PWM signal will come only from the current loop.

MOTOR CONTROLLER (Cont'd)

MOTOR Z EVENT FILTER REGISTER (MZFR)
Read/Write
Reset Value: 0000 1111 (0Fh)

7	6	5	4	3	2	1	0
ZEF3	ZEF2	ZEF1	ZEF0	ZWF3	ZWF2	ZWF1	ZWF0

Bits 7:4 = **ZEF[3:0]**: *Z Event Filter bits*
These bits select the number of valid consecutive Z events (when the Z event is detected) needed to generate the active event. Sampling is done at the selected f_{SCF} frequency (see Table 82.) or at PWM frequency.

Table 80. Z Event filter Setting

ZEF3	ZEF2	ZEF1	ZEF0	Z event Samples
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

Bits 3:0 = **ZWF[3:0]**: *Z Window Filter bits*
These bits select the length of the blanking window activated at each D event. The filter blanks the Z event detection until the end of the time window.

Table 81. Z Window filter Setting

ZWF3	ZWF2	ZWF1	ZWF0	D to Z window filter in Sensorless Mode (SR=0)	SR=1
0	0	0	0	5 µs	No Window Filter after D event
0	0	0	1	10 µs	
0	0	1	0	15 µs	
0	0	1	1	20 µs	
0	1	0	0	25 µs	
0	1	0	1	30 µs	
0	1	1	0	35 µs	
0	1	1	1	40 µs	
1	0	0	0	60 µs	
1	0	0	1	80 µs	
1	0	1	0	100 µs	
1	0	1	1	120 µs	
1	1	0	0	140 µs	
1	1	0	1	160 µs	
1	1	1	0	180 µs	
1	1	1	1	200 µs	

Note: Times are indicated for 4 MHz f_{PERIPH}

EMC CHARACTERISTICS (Cont'd)

12.7.2 EMI (Electromagnetic interference)

Based on a simple application running on the product (toggling 2 LEDs through the I/O ports), the product is monitored in terms of emission. This emission test is in line with the norm SAE J 1752/ 3 which specifies the board and the loading of each pin.

Symbol	Parameter	Conditions	Device/ Package	Monitored Frequency Band	Max vs. [f_{OSC}/f_{CPU}]		Unit
					8/4MHz	16/8MHz	
S_{EMI}	Peak level	V_{DD}=5V, T_A=+25°C conforming to SAE J 1752/3	Flash/LQFP64	0.1MHz to 30MHz	8	6	dBµV
				30MHz to 130MHz	8	12	
				130MHz to 1GHz	1	9	
				SAE EMI Level	1.5	2.5	-

Notes:

1. Data based on characterization results, not tested in production.
2. Refer to Application Note AN1709 for data on other package types

I/O PORT PIN CHARACTERISTICS (Cont'd)

12.8.2 Output Driving Current

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL} 1)	Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see Figure 141)	V_{DD}=5V	I_{IO}=+5mA		1.2	V
			I_{IO}=+2mA		0.5	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see Figure 142)		I_{IO}=+20mA, $T_A \leq 85°C$ $T_A \geq 85°C$		1.3 1.5	
			I_{IO}=+8mA		0.6	
V_{OH} 2)	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see Figure 143)		I_{IO}=-5mA, $T_A \leq 85°C$ $T_A \geq 85°C$	V_{DD}-1.4 V_{DD}-1.6		
			I_{IO}=-2mA	V_{DD}-0.7		

Figure 141. Typical V_{OL} at V_{DD}=5V (standard)

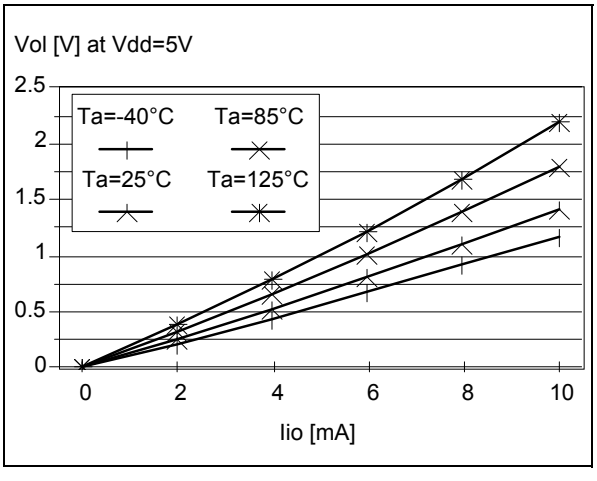


Figure 142. Typical V_{OL} at V_{DD}=5V (high-sink)

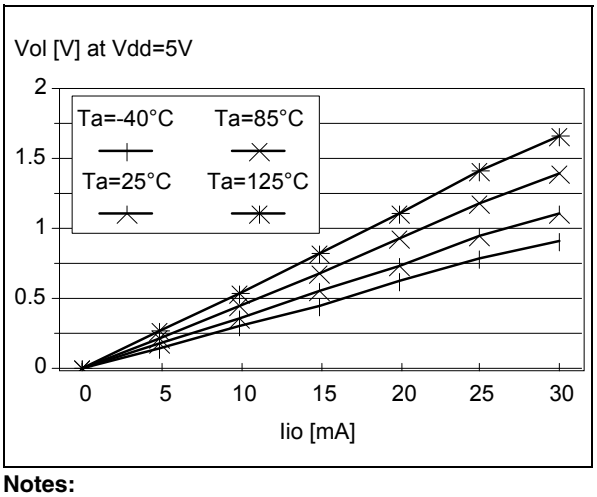
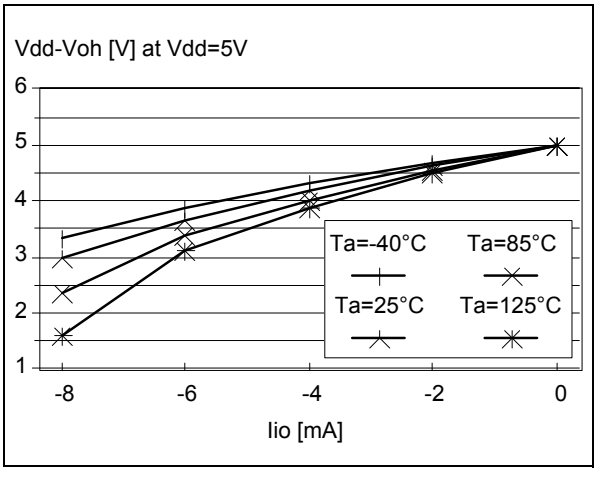


Figure 143. Typical V_{DD}-V_{OH} at V_{DD}=5V



Notes:

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 12.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
2. The I_{IO} current sourced must always respect the absolute maximum rating specified in Section 12.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}.

IMPORTANT NOTES (Cont'd)

Figure 170. Revision Marking on Box Label and Device Marking