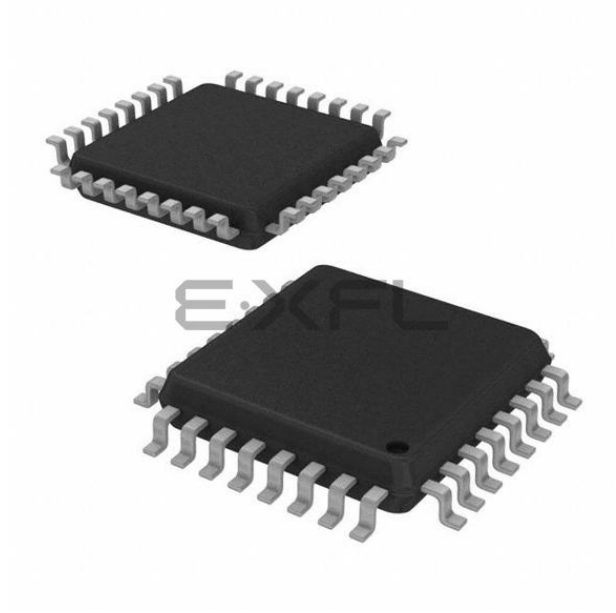


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

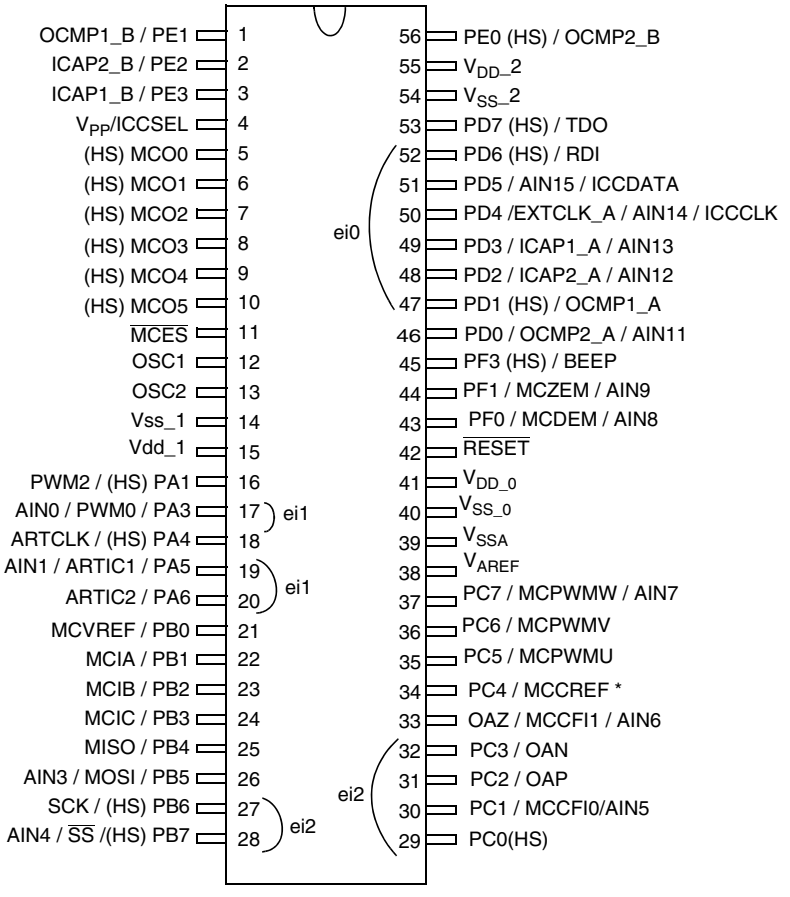
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	Not For New Designs
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	LINbusSCI
Peripherals	LVD, Motor Control PWM, POR, PWM, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fmc1k2t3

PIN DESCRIPTION (Cont'd)

Figure 5. 56-Pin SDIP Package Pinouts



(HS) 20mA high sink capability
eix associated external interrupt vector

* Once the MTC peripheral is ON, the pin PC4 is configured to an alternate function. PC4 is no longer usable as a digital I/O

3 REGISTER & MEMORY MAP

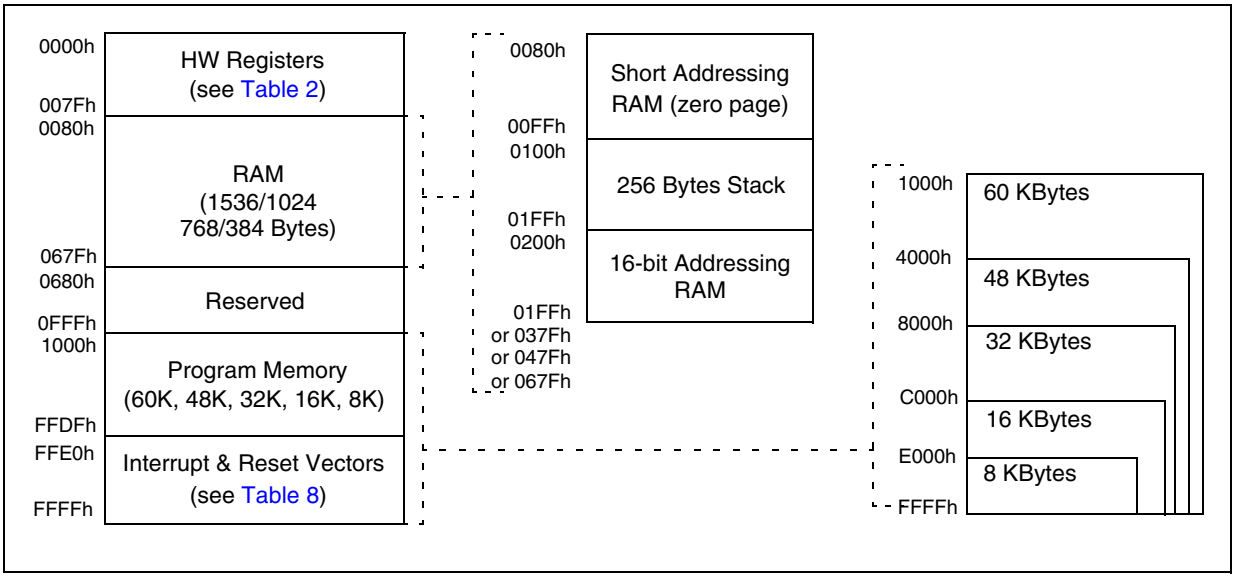
As shown in Figure 8, the MCU is capable of addressing 64K bytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, up to 2Kbytes of RAM and up to 60Kbytes of user program memory. The RAM space includes up to 256 bytes for the stack from 0100h to 01FFh.

The highest address bytes contain the user reset and interrupt vectors.

IMPORTANT: Memory locations marked as “Reserved” must never be accessed. Accessing a reserved area can have unpredictable effects on the device.

Figure 8. Memory Map



As shown in Figure 9, the MCU is capable of addressing 64K bytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, up to 1536 bytes of RAM and up to 60 Kbytes of user program memory. The RAM space includes up to 256 bytes for the stack from 0100h to 01FFh.

The highest address bytes contain the user reset and interrupt vectors.

6.4 MAIN CLOCK CONTROLLER WITH REAL-TIME CLOCK AND BEEPER (MCC/RTC)

The Main Clock Controller consists of three different functions:

- a programmable CPU clock prescaler
- a clock-out signal to supply external devices
- a real-time clock timer with interrupt capability

Each function can be used independently and simultaneously.

6.4.1 Programmable CPU Clock Prescaler

The programmable CPU clock prescaler supplies the clock for the ST7 CPU and its internal peripherals. It manages Slow power saving mode (See Section 8.2 SLOW MODE for more details).

The prescaler selects the f_{CPU} main clock frequency and is controlled by three bits in the MCCSR register: CP[1:0] and SMS.

6.4.2 Clock-out Capability

The clock-out capability is an alternate function of an I/O port pin that outputs a f_{OSC2} clock to drive external devices. It is controlled by the MCO bit in the MCCSR register.

CAUTION: When selected, the clock out pin suspends the clock during Active-halt mode.

6.4.3 Real-time Clock Timer (RTC)

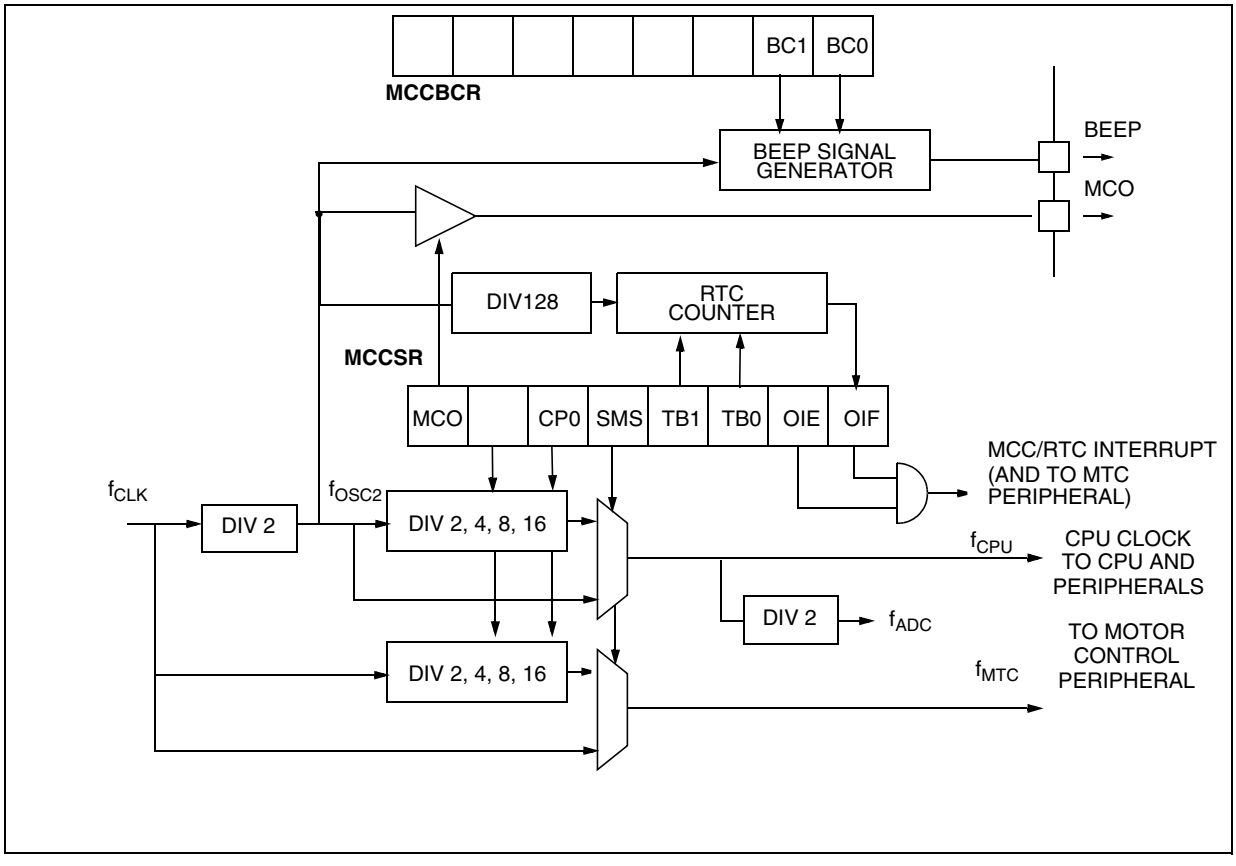
The counter of the real-time clock timer allows an interrupt to be generated based on an accurate real-time clock. Four different time bases depending directly on f_{OSC2} are available. The whole functionality is controlled by four bits of the MCCSR register: TB[1:0], OIE and OIF.

When the RTC interrupt is enabled (OIE bit set), the ST7 enters Active-halt mode when the HALT instruction is executed. See Section 8.4 ACTIVE-HALT AND HALT MODES for more details.

6.4.4 Beeper

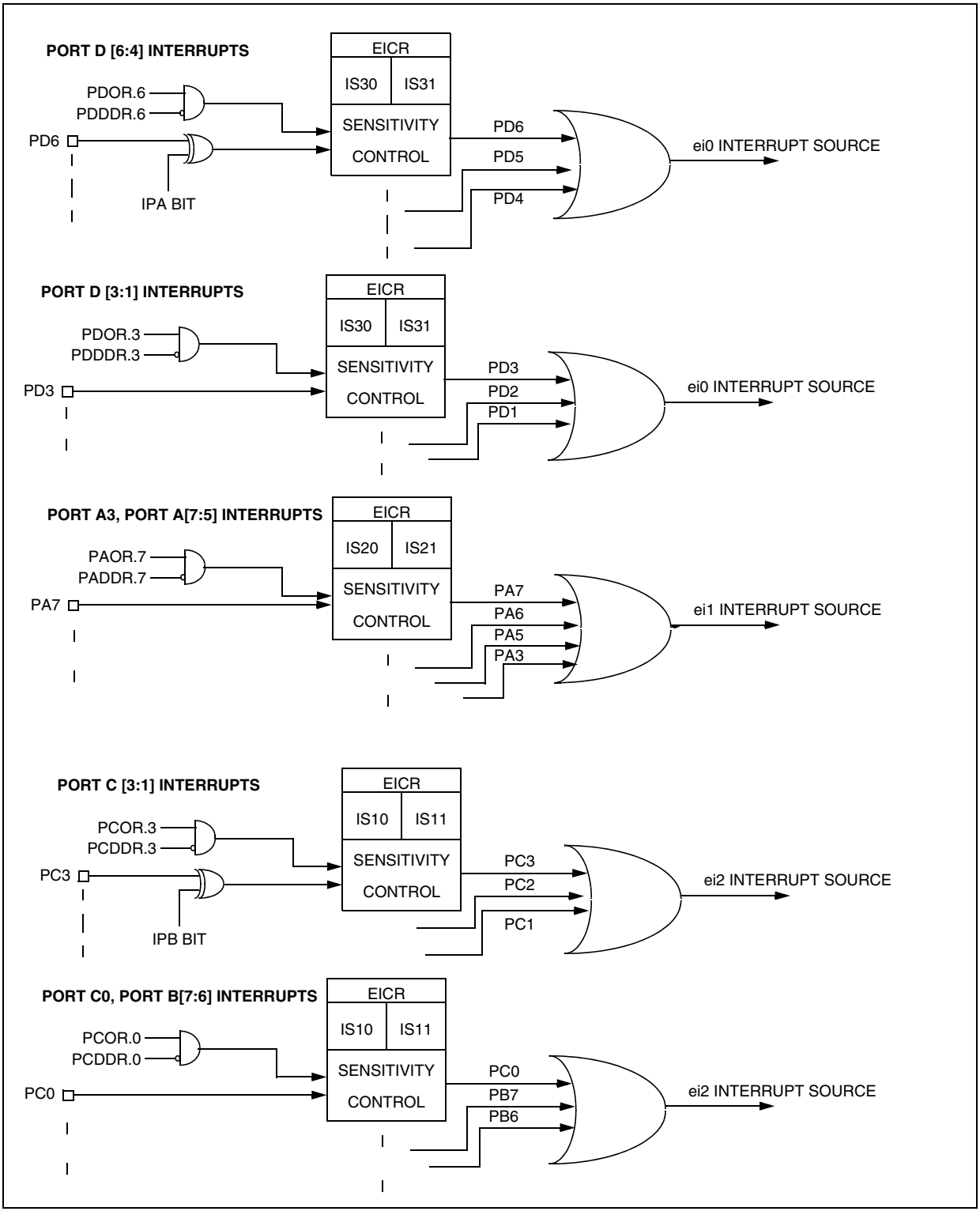
The beep function is controlled by the MCCBCR register. It can output three selectable frequencies on the BEEP pin (I/O port alternate function).

Figure 19. Main Clock Controller (MCC/RTC) Block Diagram



INTERRUPTS (Cont'd)

Figure 24. External Interrupt Control bits



(Cont'd)

Read Only

Reset Value: 0000 0000 (00h)

The three least significant bits are not used.

7							0
ICF1	OCF1	TOF	ICF2	OCF2	TIMD	0	0

Bit 7 =
0: No input capture (reset value).
1: An input capture has occurred on the ICAP1 pin or the counter has reached the OC2R value in PWM mode. To clear this bit, first read the SR register, then read or write the low byte of the IC1R (IC1LR) register.

Bit 6 =
0: No match (reset value).
1: The content of the free running counter has matched the content of the OC1R register. To clear this bit, first read the SR register, then read or write the low byte of the OC1R (OC1LR) register.

Bit 5 =
0: No timer overflow (reset value).
1: The free running counter rolled over from FFFFh to 0000h. To clear this bit, first read the SR register, then read or write the low byte of the CR (CLR) register.

Reading or writing the ACLR register does not clear TOF.

Bit 4 =
0: No input capture (reset value).
1: An input capture has occurred on the ICAP2 pin. To clear this bit, first read the SR register, then read or write the low byte of the IC2R (IC2LR) register.

Bit 3 =
0: No match (reset value).
1: The content of the free running counter has matched the content of the OC2R register. To clear this bit, first read the SR register, then read or write the low byte of the OC2R (OC2LR) register.

Bit 2 =
This bit is set and cleared by software. When set, it freezes the timer prescaler and counter and disabled the output functions (OCMP1 and OCMP2 pins) to reduce power consumption. Access to the timer registers is still available, allowing the timer configuration to be changed while it is disabled.
0: Timer enabled
1: Timer prescaler, counter and outputs disabled

Bits 1:0 = Reserved, must be kept cleared.

SERIAL PERIPHERAL INTERFACE (cont'd)

SPI CONTROL/STATUS REGISTER (SPICSR)

Read/Write (some bits Read Only)
Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	OVR	MODF	-	SOD	SSM	SSI

Bit 7 = **SPIF**

This bit is set by hardware when a transfer has been completed. An interrupt is generated if SPIE = 1 in the SPICR register. It is cleared by a software sequence (an access to the SPICSR register followed by a write or a read to the SPIDR register).
0: Data transfer is in progress or the flag has been cleared.
1: Data transfer between the device and an external device has been completed.

Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

Bit 6 = **WCOL**
This bit is set by hardware when a write to the SPIDR register is done during a transmit sequence. It is cleared by a software sequence (see Figure 60).
0: No write collision occurred
1: A write collision has been detected

Bit 5 = **OVR** S
This bit is set by hardware when the byte currently being received in the shift register is ready to be transferred into the SPIDR register while SPIF = 1 (See Section 10.4.5.2). An interrupt is generated if SPIE = 1 in the SPICR register. The OVR bit is cleared by software reading the SPICSR register.
0: No overrun error
1: Overrun error detected

Bit 4 = **MODF**
This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see Section 10.4.5.1 Master Mode Fault (MODF)). An SPI interrupt can be generated if SPIE = 1 in the SPICR register. This bit is cleared by a software sequence (An access to the SPICSR register while MODF = 1 followed by a write to the SPICR register).
0: No master mode fault detected
1: A fault in master mode has been detected

Bit 3 = Reserved, must be kept cleared.

Bit 2 = **SOD**
This bit is set and cleared by software. When set, it disables the alternate function of the SPI output (MOSI in master mode / MISO in slave mode)
0: SPI output enabled (if SPE = 1)
1: SPI output disabled

Bit 1 = **SSM**
This bit is set and cleared by software. When set, it disables the alternate function of the SPI $\overline{SS}$ pin and uses the SSI bit value instead. See Section 10.4.3.2 Slave Select Management.
0: Hardware management ($\overline{SS}$ managed by external pin)
1: Software management (internal $\overline{SS}$ signal controlled by SSI bit. External $\overline{SS}$ pin free for general-purpose I/O)

Bit 0 = **SSI**
This bit is set and cleared by software. It acts as a 'chip select' by controlling the level of the $\overline{SS}$ slave select signal when the SSM bit is set.
0: Slave selected
1: Slave deselected

SPI DATA I/O REGISTER (SPIDR)

Read/Write
Reset Value: Undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

The SPIDR register is used to transmit and receive data on the serial bus. In a master device, a write to this register will initiate transmission/reception of another byte.

Notes: During the last clock cycle the SPIF bit is set, a copy of the received data byte in the shift register is moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read.

While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

Warning: A write to the SPIDR register places data directly into the shift register for transmission.

A read to the SPIDR register returns the value located in the buffer and not the content of the shift register (see Figure 55).

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

CONTROL REGISTER 2 (SCICR2)

Read/Write
Reset Value: 0000 0000 (00h)

7							0
TIE	TCIE	RIE	ILIE	TE	RE	RWU[1)]	SBK[1)]

[1)]This bit has a different function in LIN mode, please refer to the LIN mode register description.

Bit 7 = **TIE** *Transmitter interrupt enable*
This bit is set and cleared by software.
0: Interrupt is inhibited
1: In SCI interrupt is generated whenever TDRE = 1 in the SCISR register

Bit 6 = **TCIE** *Transmission complete interrupt enable*
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SCI interrupt is generated whenever TC = 1 in the SCISR register

Bit 5 = **RIE** *Receiver interrupt enable*
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SCI interrupt is generated whenever OR = 1 or RDRF = 1 in the SCISR register

Bit 4 = **ILIE** *Idle line interrupt enable*
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SCI interrupt is generated whenever IDLE = 1 in the SCISR register.

Bit 3 = **TE** *Transmitter enable*
This bit enables the transmitter. It is set and cleared by software.
0: Transmitter is disabled
1: Transmitter is enabled

Notes:

– During transmission, a "0" pulse on the TE bit ("0" followed by "1") sends a preamble (idle line) after the current word.

– When TE is set there is a 1 bit-time delay before the transmission starts.

Bit 2 = **RE** *Receiver enable*
This bit enables the receiver. It is set and cleared by software.
0: Receiver is disabled in the SCISR register
1: Receiver is enabled and begins searching for a start bit

Bit 1 = **RWU** *Receiver wake-up*
This bit determines if the SCI is in mute mode or not. It is set and cleared by software and can be cleared by hardware when a wake-up sequence is recognized.
0: Receiver in active mode
1: Receiver in mute mode

Notes:

– Before selecting Mute mode (by setting the RWU bit) the SCI must first receive a data byte, otherwise it cannot function in Mute mode with wake-up by Idle line detection.

– In Address Mark Detection Wake-Up configuration (WAKE bit = 1) the RWU bit cannot be modified by software while the RDRF bit is set.

Bit 0 = **SBK** *Send break*
This bit set is used to send break characters. It is set and cleared by software.
0: No break character is transmitted
1: Break characters are transmitted

Note: If the SBK bit is set to "1" and then to "0", the transmitter will send a BREAK word at the end of the current word.

DATA REGISTER (SCIDR)

Read/Write
Reset Value: Undefined

Contains the Received or Transmitted data character, depending on whether it is read from or written to.

7							0
DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0

The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR).
The TDR register provides the parallel interface between the internal bus and the output shift register (see Figure 62).
The RDR register provides the parallel interface between the input shift register and the internal bus (see Figure 62).

MOTOR CONTROLLER (Cont'd)

All detections of Z_n events are done during a short measurement window while the high side switch is turned off. For this reason the PWM signal is applied on the high side switches.

When the high side switch is off, the high side winding is tied to 0V by the free-wheeling diode, the low side winding voltage is also held at 0V by the low side ON switch and the complete BEMF voltage is present on the third winding: detection is then possible.

Table 24. Step Configuration Summary

Configuration		Step					
		Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	Σ_6
Phase state register	Current direction	A to B	A to C	B to C	B to A	C to A	C to B
	High side	T0	T0	T2	T2	T4	T4
	Low side	T3	T5	T5	T1	T1	T3
	OO[5:0] bits in MPHST register	001001	100001	100100	000110	010010	011000
BEMF input	Measurement done on:	MCIC	MCIB	MCIA	MCIC	MCIB	MCIA
	IS[1:0] bits in MPHST register	10	01	00	10	01	00
BEMF edge	Back EMF shape	Falling	Rising	Falling	Rising	Falling	Rising
	CPB bit in MCRB register (ZVD bit = 0)	0	1	0	1	0	1
Hardware or Hardware-simulated demagnetization	Voltage on measured point at the start of demagnetization	0V	HV	0V	HV	0V	HV
	HDM-SDM bits in MCRB register	10	11	10	11	10	11
Demagnetization switch	PWM side selection to accelerate demagnetization	Low Side	High Side	Low Side	High Side	Low Side	High Side
	Driver selection to accelerate demagnetization	T3	T0	T5	T2	T1	T4

For a detailed description of the MTC registers, see Section 10.6.13.

10.6.4 Application Example: AC Induction Motor Drive

Although the command sequence is rather different between a PM BLDC and an AC three-phase induction motor, the Motor Controller can be configured to generate three-phase sinusoidal voltages.

A timer with three independent PWM channels is available for this purpose. Based on each of the PWM reference signal, two complemented PWM signals with deadtime are generated on the output pins (6 in total), to drive directly an inverter with triple half bridge topology.

The variable voltage levels to be applied on the motor terminals come from continuously varying duty cycle, from one PWM period to the other (refer to Figure 75 on page 144). The PWM counter generates a dedicated Update event (U event) which:

– updates automatically the compare registers setting the duty cycle to avoid time critical issues and ensure glitchless PWM operation.

– generates a dedicated U interrupt in which the values for the next coming update event are loaded in compare preload registers.

The shape of the output voltage (voltage, frequency, sinewave, trapezoid, ...) is completely managed by the applicative software, in charge of computing the compare values to be loaded for a given PWM duty-cycle (refer to Figure 76).

MOTOR CONTROLLER (Cont'd)

10.6.6.8 Position Sensor Mode

In position sensor mode (SR=1 in MCRA register), the rotor position information is given to the peripheral by means of logical data on the three inputs MCIA, MCIB and MCIC (Hall sensors).

For each step one of these three inputs is selected (IS[1:0] bits in register MPHST) in order to detect the Z event. Be careful that the phase comparator is OFF until CKE and /or DAC bits are set in MCRA register.

In sensor mode, Demagnetization and the related features (such as the special PWM configuration, D_S or D_H management, programmable filter) are not available (see Table 32)

Table 32. Demagnetisation access

SR bit MCRA register	Demagnetisation feature availabilty
1	NO
0	YES

In sensor mode configuration the rotor detection doesn't need a particular phase configuration to perform the measurement and a Z event can be read from any detection window. The sampling is done at a selectable frequency (f_{SCF}), see Table 82. This means that Z event position sensoring is more precise than it is in sensorless mode.

There is no minimum off time required for current control PWM in sensor mode so the minimum off time is set automatically to 0µs as soon as the SR bit is set in the MCRA register and a true 100% duty cycle can be set in the PWM compare U register for the PWM generation in voltage mode.

In Sensor mode, the ZEF[3:0] bits in the MZFR register are active and can be used to define the number of consecutive Z samples needed to generate the active event.

Procedure for reading sensor inputs in Direct Access mode: In Direct Access mode, the sensors can be read either when the clock are enabled or disabled (depending on CKE it in MCRA register). To read the sensor data the following steps have to be performed:

1. Select Direct Access Mode (DAC bit in MCRA register)
2. Select the appropriate MCIx input pin by means of the IS[1:0] bits in the MPHST register
3. Read the comparator output (HST bit in the MREF register)

MOTOR CONTROLLER (Cont'd)

10.6.6.10 Commutation Noise Filter

For D event detection and for Z event detection (when SPLG bit is set while DS[3:0] bits are reset), sampling is done at f_{SCF} during the PWM ON or OFF time ("Sampling block" on page 159). To avoid any erroneous detection due to PWM commutation noise, an hardware filter of 1µs (for $f_{PER-IPH}$ = 4Mhz) when PWM is put ON and when PWM is put OFF has been implemented. This means that, with sampling at 1MHz (1µs), due to this filter, 1 sample are ignored directly after the commutation.

This filter is active all the time for the D event and it is active for the Z event when the SPLG bit is set and DS[3:0] bits are cleared (meaning that the Z event is sampled at high frequency during the PWM ON or OFF time).

Table 34. Sensor/sensorless mode and D & Z event selection

SR bit	SPLG bit	DS[3:0] bits	Mode	OS[2:0] bits use	Event detection sampling clock	Sampling behaviour for Z event detection	Window and Event Filters	Behaviour of the output PWM
0	0	000	Sensors not used	Enabled	D: f_{SCF} Z: SA&OT config. PWM frequency	At the end of the off time of the PWM signal	D Window Filter DWF[3:0] after C event D Event Filter DEF[3:0] after DWF Z Window Filter ZWF[3:0] after D event Z Event Filter ZEF[3:0] after ZWF See Table 30 on page 152	"Before D" behaviour, "between D and Z" behaviour and "after Z" behaviour
0	1	000	Sensors not used	Enabled	D: f_{SCF} Z: f_{SCF}	During off time or ON time of the PWM signal		"Before D" behaviour, "between D and Z" behaviour and "after Z" behaviour
0	0	Not equal to 000	Sensors not used	Enabled	D: f_{SCF} Z: SA&OT config. PWM frequency	During ON time of the PWM signal		"Before D" behaviour, "between D and Z" behaviour and "after Z" behaviour
0	1	Not equal to 000	Sensors not used	Enabled	D: f_{SCF} Z: f_{SCF}	During ON time of the PWM signal		"Before D" behaviour, "between D and Z" behaviour and "after Z" behaviour
1	x	xxx	Position Sensors used	OS1 disabled	Z: f_{SCF}	During OFF time or ON time of the PWM signal	No Z Window Filter Only Z Event Filter is active in Sensor mode	"Before Z" behaviour and "after Z" behaviour

Note: For f_{SCF} selection, see Table 82

MOTOR CONTROLLER (Cont'd)

Note

If only one encoder output is available, it may be input either on MCIA or MCIB and an encoder clock signal will still be generated (in this case the frequency will be 50% less than with two inputs.

The state of EDIR bit will depend on signals present on MCIA and MCIB pins, the result will be given by the sampling of MCIA with MCIB falling edges.

10.6.6.14 Summary

Input Detection block set-up for the different available modes is summarized in the Table 35.

Table 35. Input Detection Block set-up

Input Detection Block Mode	Sensor Type	Edge sensitivity	SR bit	TES[1:0] bits (Tacho Edge Selection)	IS[1:0] bits (Input Selection)
Position Sensor	Hall, Optical,...	Both rising and falling edges	1	00	00 01 10
Sensorless	N/A	N/A	0	00	00 01 10
Speed Sensor	Incremental Encoder	Both rising and falling edges (imposed)	x	Any configuration different from 00: 01 10 11	11
	Tachogenerator, Hall, Optical...	Rising edge		01	00 01 10
		Falling edge		10	00 01 10
		Both rising and falling edges		11	00 01 10

MOTOR CONTROLLER (Cont'd)

10.6.9.3 Dead Time Generator

When using typical triple half bridge topology for power converters, precautions must be taken to avoid short circuits in half bridges. This is ensured by driving high and low side switches with complementary signals and by managing the time between the switching-off and the switching-on instants of the adjacent switches.

This time is usually known as deadtime and has to be adjusted depending on the devices connected to the PWM outputs and their characteristics (intrinsic delays of level-shifters, delays due to power switches,...).

When driving motors in six-step mode, the deadtime generator function also allows synchronous rectification to be performed on the switch adjacent to the one where PWM is applied to reduce conduction losses.

For each of the three PWM channels, there is one 6-bit Dead Time generator available.

It generates two output signals: A and B.

The A output signal is the same as the input phase signal except for the rising edge, which is delayed relative to the input signal rising edge.

The B output signal is the opposite of the input phase signal except the rising edge which is delayed relative to the input signal falling edge.

Figure 111 shows the relationship between the output signals of the deadtime register and its inputs.

If the delay is greater than the width of the active phase (A or B) then the corresponding pulse is not generated (see Figure 112 and Figure 113).

Figure 111. Dead Time waveforms

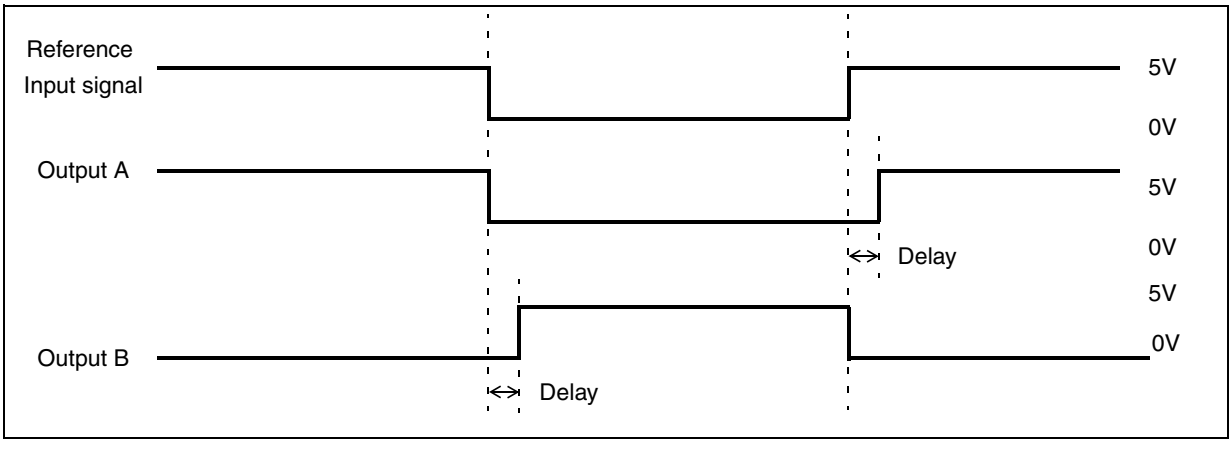
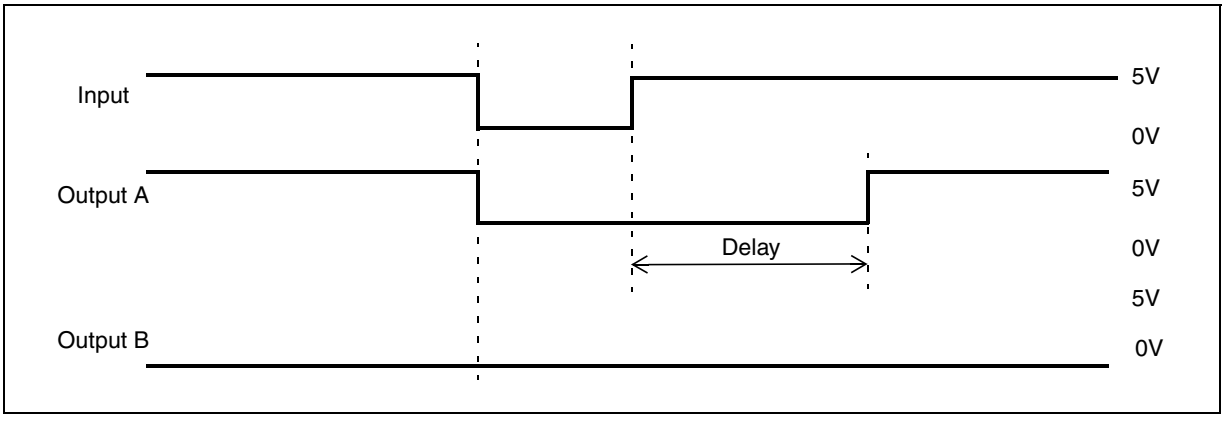


Figure 112. Dead time waveform with delay greater than the negative PWM pulse



MOTOR CONTROLLER (Cont'd)

10.6.10 PWM Generator Block

The PWM generator block produces three independent PWM signals based on a single carrier frequency with individually adjustable duty cycles.

Depending on the motor driving method, one or three of these signals may be redirected to the other functional blocks of the motor control peripheral, using the PCN bit in the MDTG register.

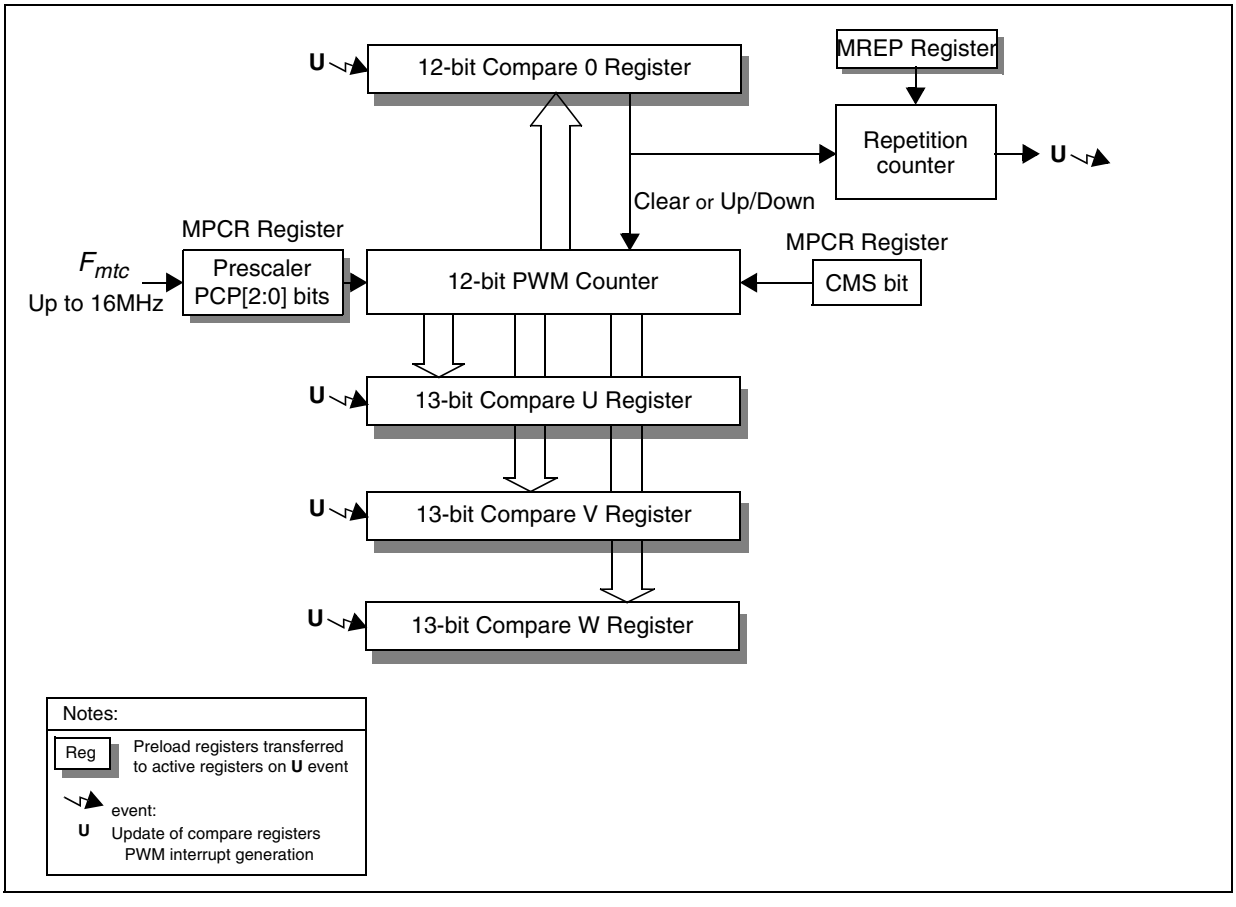
When driving PM BLDC motors in six-step mode (voltage mode only, either sensored or sensorless) a single PWM signal (Phase U) is used to supply the Input Stage, PWM and Channel Manager blocks according to the selected modes.

For other kind of motors requiring independent PWM control for each of the three phases, all PWM signals (Phases U, V and W) are directed to the channel manager, in which deadtime or a high frequency carrier may be added. This is the case of AC induction motors or PMAC motors for instance, supplied with 120° shifted sinewaves in voltage mode.

10.6.10.1 Main Features

- 12-bit PWM free-running Up/Down Counter with up to 16MHz input clock (F_{mtc}).
- Edge-aligned and center-aligned PWM operating modes
- Possibility to re-load compare registers twice per PWM period in center-aligned mode
- Full-scale PWM generation
- PWM update interrupt generation
- 8-bit repetition counter
- 8-bit PWM mode
- Timer re-synchronisation feature

Figure 117. PWM generator block diagram



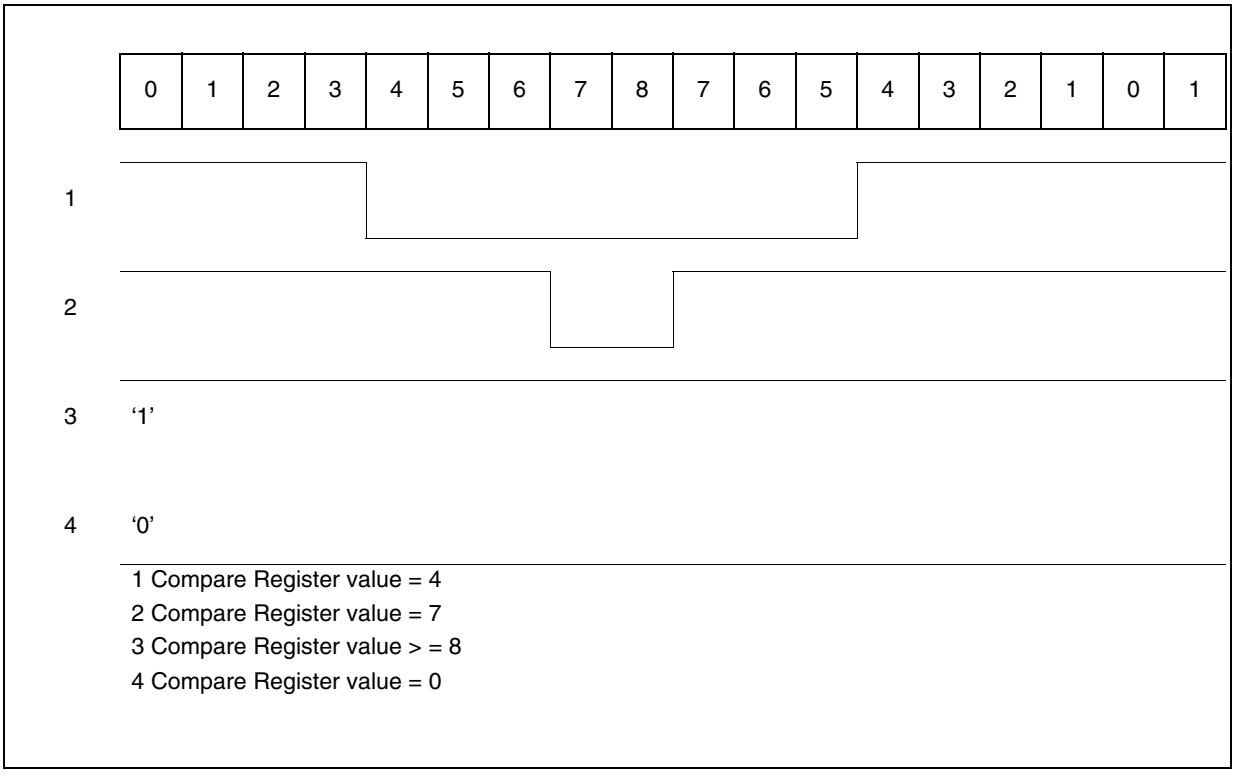
MOTOR CONTROLLER (Cont'd)

If the 13-bit Compare register value is greater than the extended Compare 0 Register (the 13th bit is set to '0'), the corresponding PWM output signal is held at '1'.

If the 13-bit Compare register value is 0, the corresponding PWM output signal is held at '0'.

Figure 119 shows some center-aligned PWM waveforms in an example where the Compare 0 register value = 8.

Figure 119. Center-aligned PWM Waveforms (Compare 0 Register = 8)



MOTOR CONTROLLER (Cont'd)

MOTOR CURRENT FEEDBACK REGISTER (MCFR)
Read/Write
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
RPGS	RST	CFF2	CFF1	CFF0	CFW2	CFW1	CFW0

Bit 7= **RPGS:** *Register Page Selection:*
0: Access to registers mapped in page 0
1: Access to registers mapped in page 1

Bit 6= **RST**: *Reset MTC registers.*
Software can set this bit to reset all MTC registers without resetting the ST7.
0: No MTC register reset
1: Reset all MTC registers

Bits 5:3 = **CFF[2:0]**: *Current Feedback Filter bits*
These bits select the number of consecutive valid samples (when the current is above the limit) needed to generate the active event. Sampling is done at $f_{PERIPH}/4$.

Table 67. Current Feedback Filter Setting

CFF2	CFF1	CFF0	Current Feedback Samples
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

Bits 2:0 = **CFW[2:0]**: *Current Window Filter bits:*

These bits select the length of the blanking window activated each time PWM is turned ON. The filter blanks the output of the current comparator.

Table 68. Current Feedback Window Setting

CFW2	CFW1	CFW0	Blanking Window
0	0	0	Blanking window off
0	0	1	0.5µs
0	1	0	1µs
0	1	1	1.5µs
1	0	0	2µs
1	0	1	2.5µs
1	1	0	3µs
1	1	1	3.5µs

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

Register Name	7	6	5	4	3	2	1	0
MCPWL Reset Value	CPWL7 0	CPWL6 0	CPWL5 0	CPWL4 0	CPWL3 0	0	0	0
MCPVH Reset Value	CPVH7 0	CPVH6 0	CPVH5 0	CPVH4 0	CPVH3 0	CPVH2 0	CPVH1 0	CPVH0 0
MCPVL Reset Value	CPVL7 0	CPVL6 0	CPVL5 0	CPVL4 0	CPVL3 0	0	0	0
MCPUH Reset Value	CPUH7 0	CPUH6 0	CPUH5 0	CPUH4 0	CPUH3 0	CPUH2 0	CPUH1 0	CPUH0 0
MCPUL Reset Value	CPUL7 0	CPUL6 0	CPUL5 0	CPUL4 0	CPUL3 0	0	0	0
MCP0H Reset Value	0	0	0	0	CP0H3 1	CP0H2 1	CP0H1 1	CP0H0 1
MCP0L Reset Value	CP0L7 1	CP0L6 1	CP0L5 1	CP0L4 1	CP0L3 1	CP0L2 1	CP0L1 1	CP0L0 1

Table 84. MTC Page 1 Register Map and Reset Values

Register Name	7	6	5	4	3	2	1	0
MDTG Reset Value	PCN 1	DTE 1	DTG5 1	DTG4 1	DTG3 1	DTG2 1	DTG1 1	DTG0 1
MPOL Reset Value	ZVD 0	REO 0	OP5 1	OP4 1	OP3 1	OP2 1	OP1 1	OP0 1
MPWME Reset Value	DG 0	PWMW 0	PWMV 0	PWMU 0	OT3 0	OT2 0	OT1 0	OT0 0
MCONF Reset Value	DS3 0	DS2 0	DS1 0	DS0 0	SOI 0	SOM 0	XT16 1	XT8 0
MPAR Reset Value	TES1 0	TES0 0	OE5 0	OE4 0	OE3 0	OE2 0	OE1 0	OE0 0
MZFR Reset Value	ZEF3 0	ZEF2 0	ZEF1 0	ZEF0 0	ZWF3 1	ZWF2 1	ZWF1 1	ZWF0 1
MSCR Reset Value	ZSV 0	0	0	0	SCF1 0	SCF0 0	ECM 0	DISS 0

INSTRUCTION SET OVERVIEW (Cont'd)

11.2 INSTRUCTION GROUPS

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may be subdivided into 13 main groups as illustrated in the following table:

Load and Transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/Decrement	INC	DEC						
Compare and Tests	CP	TNZ	BCP					
Logical operations	AND	OR	XOR	CPL	NEG			
Bit Operation	BSET	BRES						
Conditional Bit Test and Branch	BTJT	BTJF						
Arithmetic operations	ADC	ADD	SUB	SBC	MUL			
Shift and Rotates	SLL	SRL	SRA	RLC	RRC	SWAP	SLA	
Unconditional Jump or Call	JRA	JRT	JRF	JP	CALL	CALLR	NOP	RET
Conditional Branch	JRxx							
Interruption management	TRAP	WFI	HALT	IRET				
Condition Code Flag modification	SIM	RIM	SCF	RCF				

Using a pre-byte

The instructions are described with one to four opcodes.

In order to extend the number of available opcodes for an 8-bit CPU (256 opcodes), three different prebyte opcodes are defined. These prebytes modify the meaning of the instruction they precede.

The whole instruction becomes:

PC-2 End of previous instruction

PC-1 Prebyte

PC opcode

PC+1 Additional word (0 to 2) according to the number of bytes required to compute the effective address

These prebytes enable instruction in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or the instruction using direct addressing mode. The prebytes are:

PDY 90 Replace an X based instruction using immediate, direct, indexed, or inherent addressing mode by a Y one.

PIX 92 Replace an instruction using direct, direct bit, or direct relative addressing mode to an instruction using the corresponding indirect addressing mode.
It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.

PIY 91 Replace an instruction using X indirect indexed addressing mode by a Y one.

11.2.1 Illegal Opcode Reset

In order to provide enhanced robustness to the device against unexpected behaviour, a system of illegal opcode detection is implemented. If a code to be executed does not correspond to any opcode or prebyte value, a reset is generated. This, combined with the Watchdog, allows the detection and recovery from an unexpected fault or interference.

Note: A valid prebyte associated with a valid opcode forming an unauthorized combination does not generate a reset.

SUPPLY CURRENT CHARACTERISTICS (Cont'd)

12.4.2 Wait and Slow-wait Modes

Symbol	Parameter	Conditions		Typ	Max [1]	Unit
I_{DD}	Supply current in Wait mode [2] (see Figure 133)	$4.5V \le V_{DD} \le 5.5V$	f_{OSC}=16MHz, f_{CPU}=8MHz	8	12	mA
	Supply current in Slow-wait mode [2] (see Figure 134)		f_{OSC}=16MHz, f_{CPU}=500kHz	3.5	5	

Figure 133. Typical I_{DD} in Wait vs. f_{CPU}

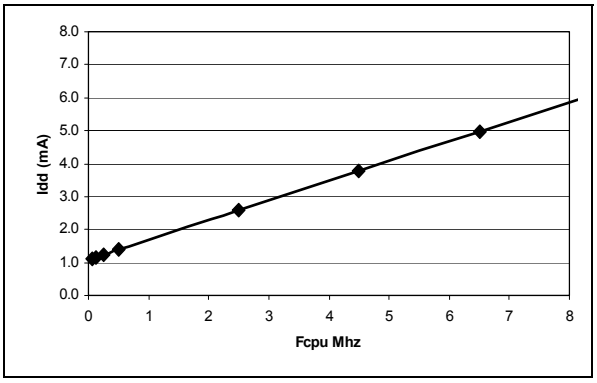
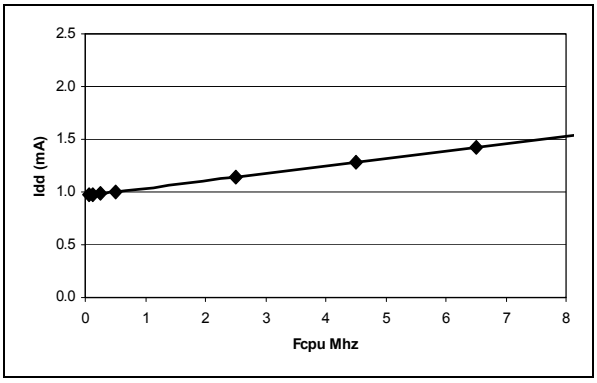


Figure 134. Typical I_{DD} in Slow-wait vs. f_{CPU}



Notes:

1. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
2. Measurements are done in the following conditions:

- Program executed from RAM, CPU running with RAM access. The increase in consumption when executing from Flash is 50%.
- All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load)
- All peripherals in reset state.
- LVD disabled.
- Clock input (OSC1) driven by external square wave.
- In Slow and Slow-wait mode, f_{CPU} is based on f_{OSC} divided by 32.

To obtain the total current consumption of the device, add the clock source (Section 12.5.3) and the peripheral power consumption.

PACKAGE CHARACTERISTICS (Cont'd)

13.2 THERMAL CHARACTERISTICS

Symbol	Ratings	Value	Unit
R_{thJA}	Package thermal resistance (junction to ambient) LQFP80 14x14 LQFP64 14x14 LQFP44 10x10 LQFP32 7x7 SDIP32 400mil SDIP56 600mil	 55 55 68 80 63 45	°C/W
T_{Jmax}	Maximum junction temperature[1)]	150	°C
P_{Dmax}	Power dissipation [2)]	500	mW

Notes:

1. The maximum chip-junction temperature is based on technology characteristics.

2. The maximum power dissipation is obtained from the formula $P_D = (T_J - T_A) / R_{thJA}$.

The power dissipation of an application can be defined by the user with the formula: $P_D=P_{INT}+P_{PORT}$ where P_{INT} is the chip internal power ($I_{DD} \times V_{DD}$) and P_{PORT} is the port power dissipation depending on the ports used in the application.

IMPORTANT NOTES (Cont'd)

15.2 CLEARING ACTIVE INTERRUPTS OUTSIDE INTERRUPT ROUTINE

When an active interrupt request occurs at the same time as the related flag or interrupt mask is being cleared, the CC register may be corrupted.

Concurrent interrupt context

The symptom does not occur when the interrupts are handled normally, i.e. when:

- The interrupt request is cleared (flag reset or interrupt mask) within its own interrupt routine

- The interrupt request is cleared (flag reset or interrupt mask) within any interrupt routine

- The interrupt request is cleared (flag reset or interrupt mask) in any part of the code while this interrupt is disabled

If these conditions are not met, the symptom can be avoided by implementing the following sequence:

Perform SIM and RIM operation before and after resetting an active interrupt request

Ex:

```
SIM
reset flag or interrupt mask
RIM
```

Nested interrupt context

The symptom does not occur when the interrupts are handled normally, i.e. when:

- The interrupt request is cleared (flag reset or interrupt mask) within its own interrupt routine

- The interrupt request is cleared (flag reset or interrupt mask) within any interrupt routine with higher or identical priority level

- The interrupt request is cleared (flag reset or interrupt mask) in any part of the code while this interrupt is disabled

If these conditions are not met, the symptom can be avoided by implementing the following sequence:

```
PUSH CC
SIM
reset flag or interrupt mask
POP CC
```

15.3 TIMD SET SIMULTANEOUSLY WITH OC INTERRUPT

If the 16-bit timer is disabled at the same time as the output compare event occurs then the output compare flag gets locked and cannot be cleared before the timer is enabled again.

15.3.1 Impact on the application

If the output compare interrupt is enabled, then the output compare flag cannot be cleared in the timer interrupt routine. Consequently the interrupt service routine is called repeatedly and the application gets stuck which causes the watchdog reset if enabled by the application.

15.3.2 Workaround

Disable the timer interrupt before disabling the timer. While enabling, first enable the timer, then enable the timer interrupts.

Perform the following to disable the timer

– TACR1 or TBCR1 = 0x00h; // Disable the compare interrupt.

– TACSR | or TBCSR |= 0x40; // Disable the timer.

– Perform the following to enable the timer again

– TACSR & or TBCSR & = ~0x40; // Enable the timer.

– TACR1 or TBCR1 = 0x40; // Enable the compare interrupt.

15.4 LINSCI LIMITATIONS

15.4.1 LINSCI wrong break duration

SCI Mode

A single break character is sent by setting and resetting the SBK bit in the SCICR2 register. In some cases, the break character may have a longer duration than expected:

- 20 bits instead of 10 bits if M=0

- 22 bits instead of 11 bits if M=1.

In the same way, as long as the SBK bit is set, break characters are sent to the TDO pin. This may lead to generate one break more than expected.

Occurrence

The occurrence of the problem is random and proportional to the baudrate. With a transmit frequency of 19200 baud (fCPU=8MHz and SCIBRR=0xC9), the wrong break duration occurrence is around 1%.