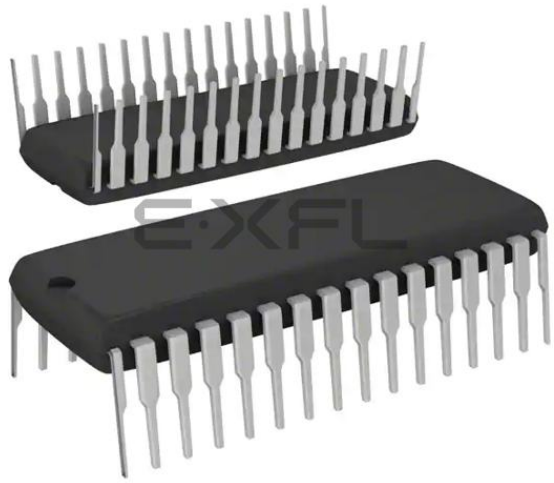


Welcome to E-XFL.COM

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
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	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	32-DIP (0.600", 15.24mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fmc1k2b6

3 REGISTER & MEMORY MAP

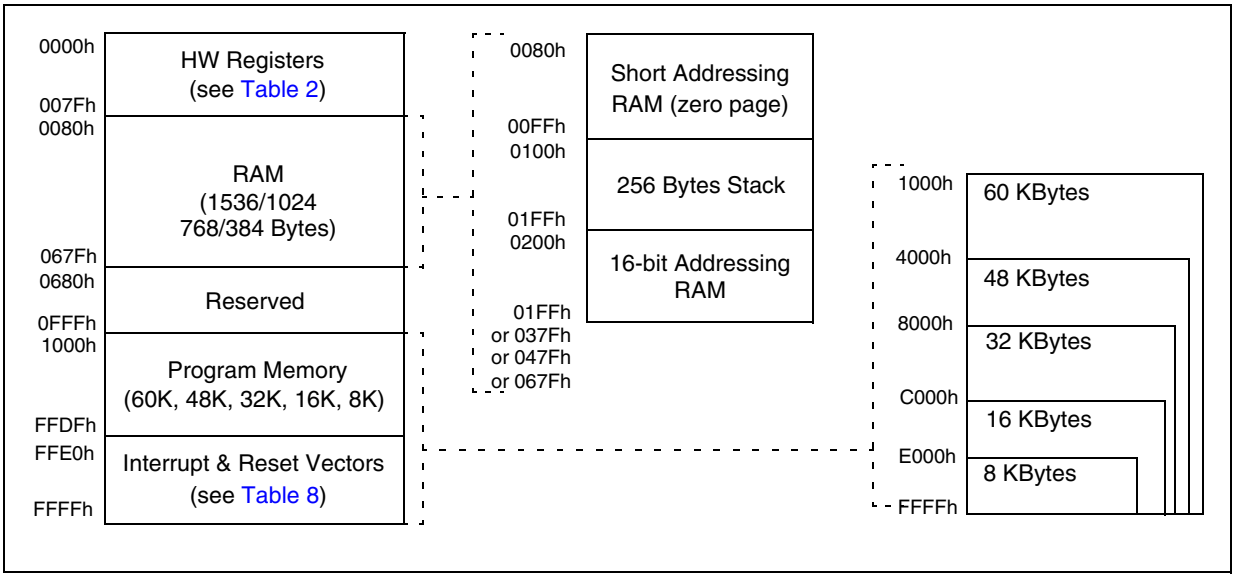
As shown in Figure 8, the MCU is capable of addressing 64K bytes of memories and I/O registers.

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.

Table 2. Hardware Register Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
0024h	ITC	ITSPR0	Interrupt Software Priority Register 0	FFh	R/W
0025h		ITSPR1	Interrupt Software Priority Register 1	FFh	R/W
0026h		ITSPR2	Interrupt Software Priority Register 2	FFh	R/W
0027h		ITSPR3	Interrupt Software Priority Register 3	FFh	R/W
0028h		EICR	External Interrupt Control Register	00h	R/W
0029h	FLASH	FSCR	Flash Control/Status Register	00h	R/W
002Ah	WATCHDOG	WDGCR	Window Watchdog Control Register	7Fh	R/W
002Bh		WDGWR	Window Watchdog Window Register	7Fh	R/W
002Ch	MCC	MCCSR	Main Clock Control / Status Register	00h	R/W
002Dh		MCCBCR	Main Clock Controller: Beep Control Register	00h	R/W
002Eh	ADC	ADCCSR	Control/Status Register	00h	R/W
002Fh		ADCDRMSB	Data Register MSB	00h	Read Only
0030h		ADCDRLSB	Data Register LSB	00h	Read Only
0031h	TIMER A	TACR2	Timer A Control Register 2	00h	R/W
0032h		TACR1	Timer A Control Register 1	00h	R/W
0033h		TACSR	Timer A Control/Status Register	xxh	R/W
0034h		TAIC1HR	Timer A Input Capture 1 High Register	xxh	Read Only
0035h		TAIC1LR	Timer A Input Capture 1 Low Register	xxh	Read Only
0036h		TAOC1HR	Timer A Output Compare 1 High Register	80h	R/W
0037h		TAOC1LR	Timer A Output Compare 1 Low Register	00h	R/W
0038h		TACHR	Timer A Counter High Register	FFh	Read Only
0039h		TACLR	Timer A Counter Low Register	FCh	Read Only
003Ah		TAACHR	Timer A Alternate Counter High Register	FFh	Read Only
003Bh		TAACLR	Timer A Alternate Counter Low Register	FCh	Read Only
003Ch		TAIC2HR	Timer A Input Capture 2 High Register	xxh	Read Only
003Dh		TAIC2LR	Timer A Input Capture 2 Low Register	xxh	Read Only
003Eh		TAOC2HR	Timer A Output Compare 2 High Register	80h	R/W
003Fh		TAOC2LR	Timer A Output Compare 2 Low Register	00h	R/W
0040h	SIM	SICSR	System Integrity Control/Status Register	000x000x b	R/W
0041h	TIMER B	TBCR2	Timer B Control Register 2	00h	R/W
0042h		TBCR1	Timer B Control Register 1	00h	R/W
0043h		TBCSR	Timer B Control/Status Register	xxh	R/W
0044h		TBIC1HR	Timer B Input Capture 1 High Register	xxh	Read Only
0045h		TBIC1LR	Timer B Input Capture 1 Low Register	xxh	Read Only
0046h		TBOC1HR	Timer B Output Compare 1 High Register	80h	R/W
0047h		TBOC1LR	Timer B Output Compare 1 Low Register	00h	R/W
0048h		TBCHR	Timer B Counter High Register	FFh	Read Only
0049h		TBCLR	Timer B Counter Low Register	FCh	Read Only
004Ah		TBACHR	Timer B Alternate Counter High Register	FFh	Read Only
004Bh		TBACLR	Timer B Alternate Counter Low Register	FCh	Read Only
004Ch		TBIC2HR	Timer B Input Capture 2 High Register	xxh	Read Only
004Dh		TBIC2LR	Timer B Input Capture 2 Low Register	xxh	Read Only
004Eh		TBOC2HR	Timer B Output Compare 2 High Register	80h	R/W
004Fh		TBOC2LR	Timer B Output Compare 2 Low Register	00h	R/W

5 CENTRAL PROCESSING UNIT

5.1 INTRODUCTION

This CPU has a full 8-bit architecture and contains six internal registers allowing efficient 8-bit data manipulation.

5.2 MAIN FEATURES

- Enable executing 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes (with indirect addressing mode)
- Two 8-bit index registers
- 16-bit stack pointer
- Low power Halt and Wait modes
- Priority maskable hardware interrupts
- Non-maskable software/hardware interrupts

5.3 CPU REGISTERS

The six CPU registers shown in Figure 11 are not present in the memory mapping and are accessed by specific instructions.

Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

Index Registers (X and Y)

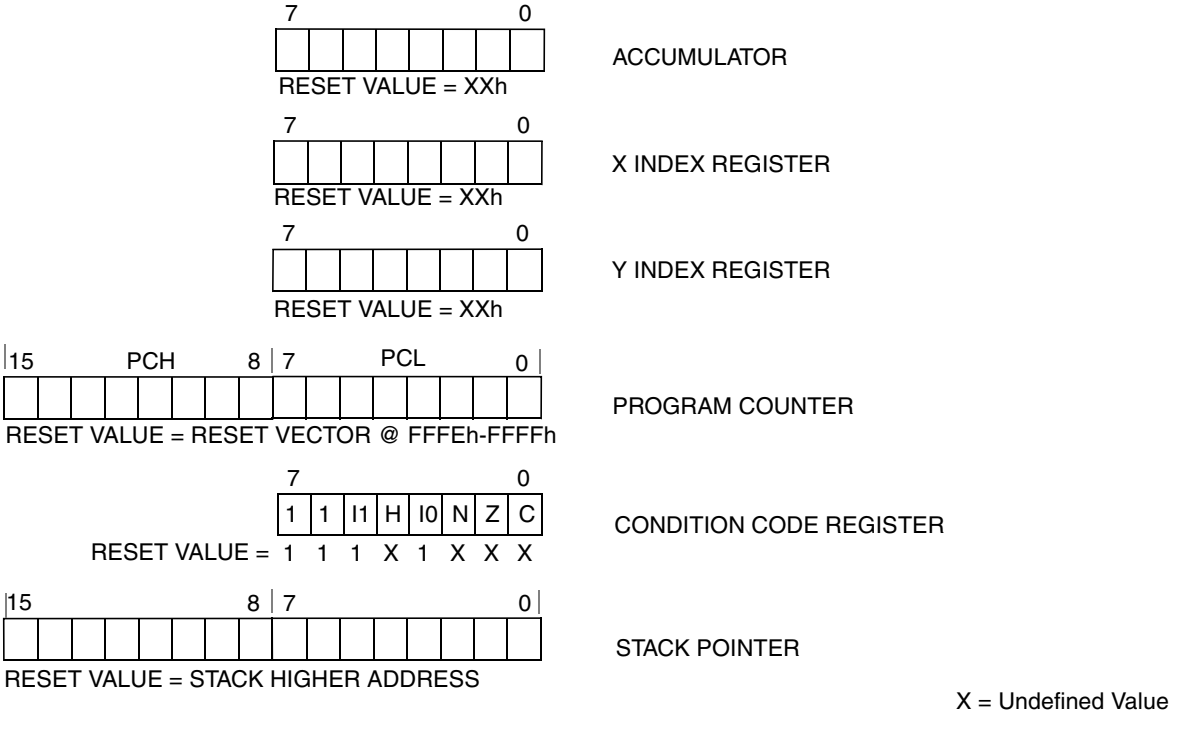
These 8-bit registers are used to create effective addresses or as temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures.

Program Counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

Figure 11. CPU Registers



8 POWER SAVING MODES

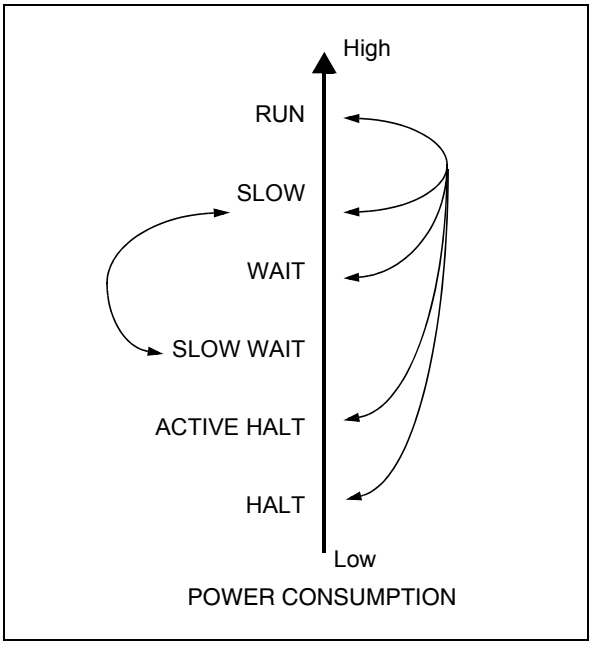
8.1 INTRODUCTION

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,

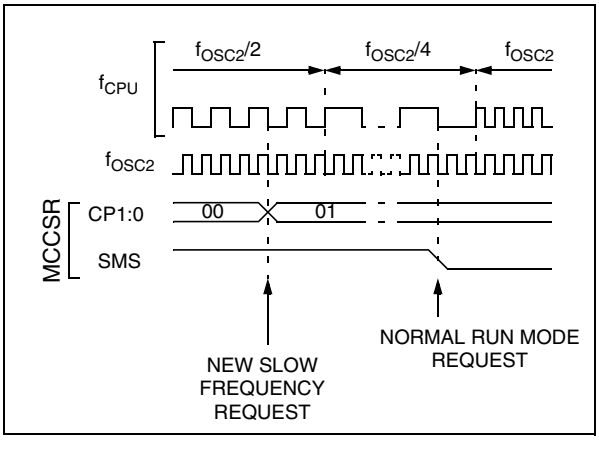
– To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

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



I/O PORTS (Cont'd)

CAUTION: The alternate function must not be activated as long as the pin is configured as input with interrupt, in order to avoid generating spurious interrupts.

Analog alternate function

When the pin is used as an ADC input, the I/O must be configured as floating input. The analog multiplexer (controlled by the ADC registers) switches the analog voltage present on the selected pin to the common analog rail which is connected to the ADC input.

It is recommended not to change the voltage level or loading on any port pin while conversion is in progress. Furthermore it is recommended not to have clocking pins located close to a selected analog pin.

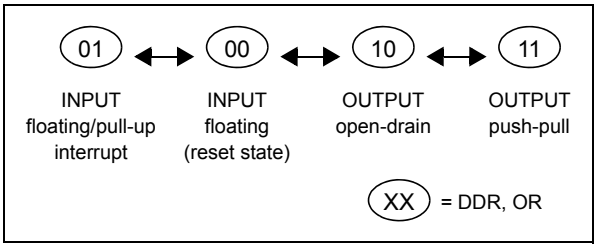
WARNING: The analog input voltage level must be within the limits stated in the absolute maximum ratings.

9.3 I/O PORT IMPLEMENTATION

The hardware implementation on each I/O port depends on the settings in the DDR and OR registers and specific feature of the I/O port such as ADC Input or true open drain.

Switching these I/O ports from one state to another should be done in a sequence that prevents unwanted side effects. Recommended safe transitions are illustrated in Figure 33 on page 57. Other transitions are potentially risky and should be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 33. Interrupt I/O Port State Transitions



9.4 LOW POWER MODES

Mode	Description
Wait	No effect on I/O ports. External interrupts cause the device to exit from Wait mode.
Halt	No effect on I/O ports. External interrupts cause the device to exit from Halt mode.

9.5 INTERRUPTS

The external interrupt event generates an interrupt if the corresponding configuration is selected with DDR and OR registers 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
External interrupt on selected external event	-	DDRx ORx	Yes	

10.3 16-BIT TIMER

10.3.1 Introduction

The timer consists of a 16-bit free-running counter driven by a programmable prescaler.

It may be used for a variety of purposes, including pulse length measurement of up to two input signals (*input capture*) or generation of up to two output waveforms (*output compare* and *PWM*).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the CPU clock prescaler.

Some devices of the ST7 family have two on-chip 16-bit timers. They are completely independent, and do not share any resources. They are synchronized after a Device reset as long as the timer clock frequencies are not modified.

This description covers one or two 16-bit timers. In the devices with two timers, register names are prefixed with TA (Timer A) or TB (Timer B).

10.3.2 Main Features

- Programmable prescaler: f_{CPU} divided by 2, 4 or 8.
- Overflow status flag and maskable interrupt
- External clock input (must be at least 4 times slower than the CPU clock speed) with the choice of active edge
- Output compare functions with
  - 2 dedicated 16-bit registers
  - 2 dedicated programmable signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Input capture functions with
  - 2 dedicated 16-bit registers
  - 2 dedicated active edge selection signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Pulse width modulation mode (PWM)
- One pulse mode
- Reduced Power Mode
- 5 alternate functions on I/O ports (ICAP1, ICAP2, OCMP1, OCMP2, EXTCLK)*

The Block Diagram is shown in Figure 44.

***Note:** Some timer pins may not available (not bonded) in some devices. Refer to the device pin out description.

When reading an input signal on a non-bonded pin, the value will always be '1'.

10.3.3 Functional Description

10.3.3.1 Counter

The main block of the Programmable Timer is a 16-bit free running upcounter and its associated 16-bit registers. The 16-bit registers are made up of two 8-bit registers called high & low.

Counter Register (CR):

- Counter High Register (CHR) is the most significant byte (MS Byte).
- Counter Low Register (CLR) is the least significant byte (LS Byte).

Alternate Counter Register (ACR)

- Alternate Counter High Register (ACHR) is the most significant byte (MS Byte).
- Alternate Counter Low Register (ACLR) is the least significant byte (LS Byte).

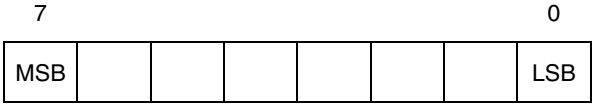
These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (Timer overflow flag), located in the Status register, (SR), (see note at the end of paragraph titled 16-bit read sequence).

Writing in the CLR register or ACLR register resets the free running counter to the FFFCh value.
Both counters have a reset value of FFFCh (this is the only value which is reloaded in the 16-bit timer). The reset value of both counters is also FFFCh in One Pulse mode and PWM mode.

The timer clock depends on the clock control bits of the CR2 register, as illustrated in Table 16 Clock Control Bits. The value in the counter register repeats every 131 072, 262 144 or 524 288 CPU clock cycles depending on the CC[1:0] bits.
The timer frequency can be $f_{CPU}/2$, $f_{CPU}/4$, $f_{CPU}/8$ or an external frequency.

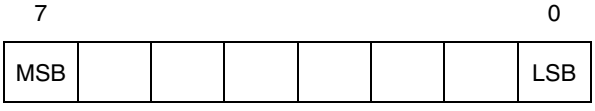
Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the high part of the counter value (transferred by the input capture 1 event).



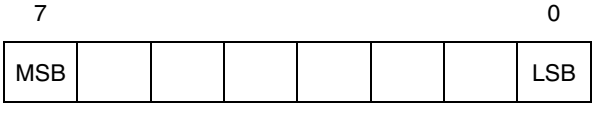
Read/Write
Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.



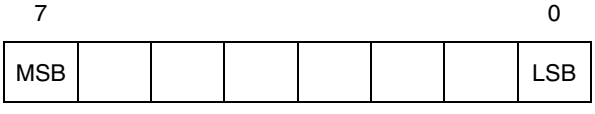
Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the low part of the counter value (transferred by the input capture 1 event).



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

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.



(cont'd)

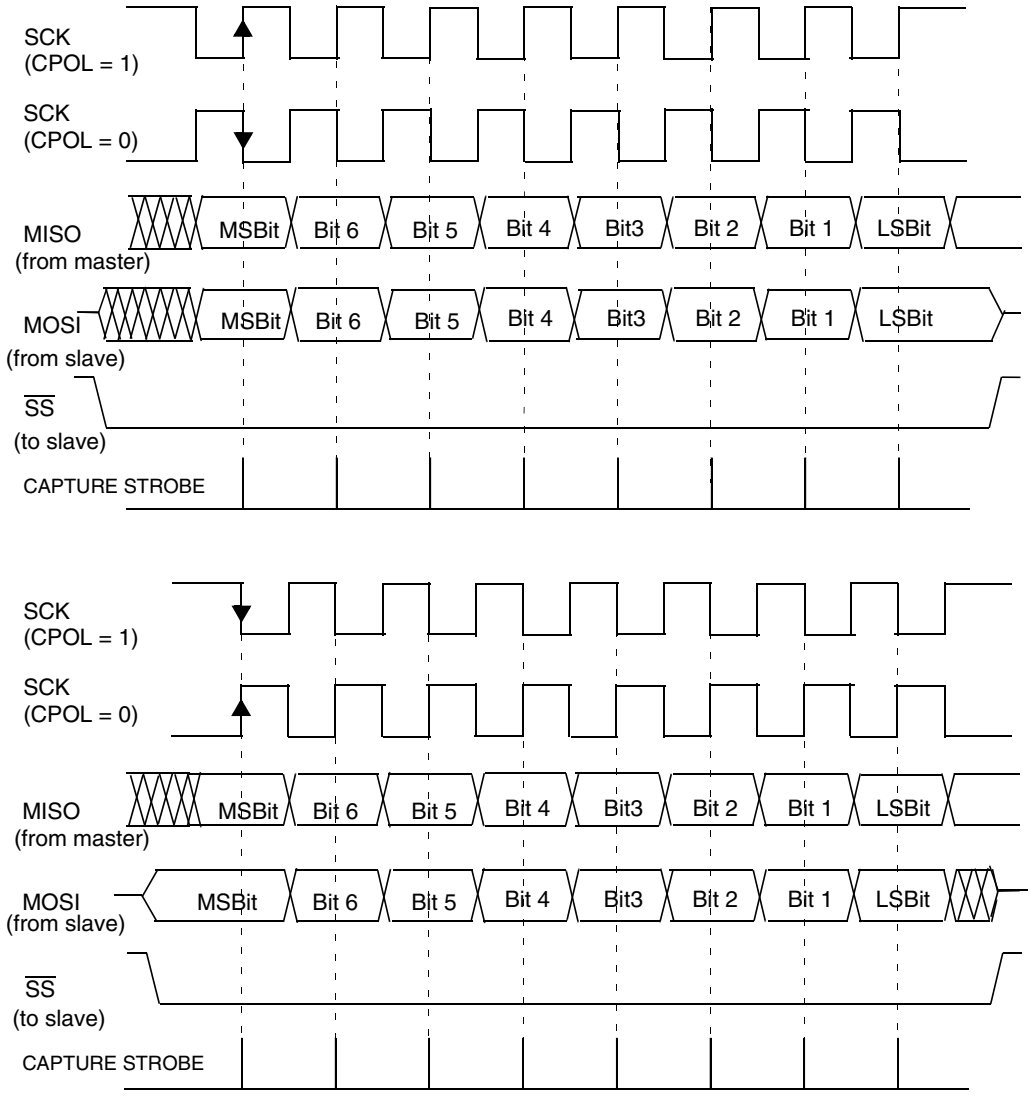
Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits (See Figure 59).

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.

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

10.5.9.5 LIN Baud Rate

Baud rate programming is done by writing a value in the LPR prescaler or performing an automatic resynchronization as described below.

Automatic Resynchronization

To automatically adjust the baud rate based on measurement of the LIN Synch Field:

– Write the nominal LIN Prescaler value (usually depending on the nominal baud rate) in the LPFR / LPR registers.

– Set the LASE bit to enable the Auto Synchronization Unit.

When Auto Synchronization is enabled, after each LIN Synch Break, the time duration between five falling edges on RDI is sampled on f_{CPU} and the result of this measurement is stored in an internal 15-bit register called SM (not user accessible) (See Figure 68). Then the LDIV value (and its associated LPFR and LPR registers) are automatically updated at the end of the fifth falling edge. During LIN Synch field measurement, the SCI state machine is stopped and no data is transferred to the data register.

10.5.9.6 LIN Slave Baud Rate Generation

In LIN mode, transmission and reception are driven by the LIN baud rate generator

Note: LIN Master mode uses the Extended or Conventional prescaler register to generate the baud rate.

If LINE bit = 1 and LSLV bit = 1 then the Conventional and Extended Baud Rate Generators are disabled: the baud rate for the receiver and transmitter are both set to the same value, depending on the LIN Slave baud rate generator:

$$Tx = Rx = \frac{f_{CPU}}{(16*LDIV)}$$

with:

LDIV is an unsigned fixed point number. The mantissa is coded on 8 bits in the LPR register and the fraction is coded on 4 bits in the LPFR register.

If LASE bit = 1 then LDIV is automatically updated at the end of each LIN Synch Field.

Three registers are used internally to manage the auto-update of the LIN divider (LDIV):

- LDIV_NOM (nominal value written by software at LPR/LPFR addresses)

- LDIV_MEAS (results of the Field Synch measurement)

- LDIV (used to generate the local baud rate)

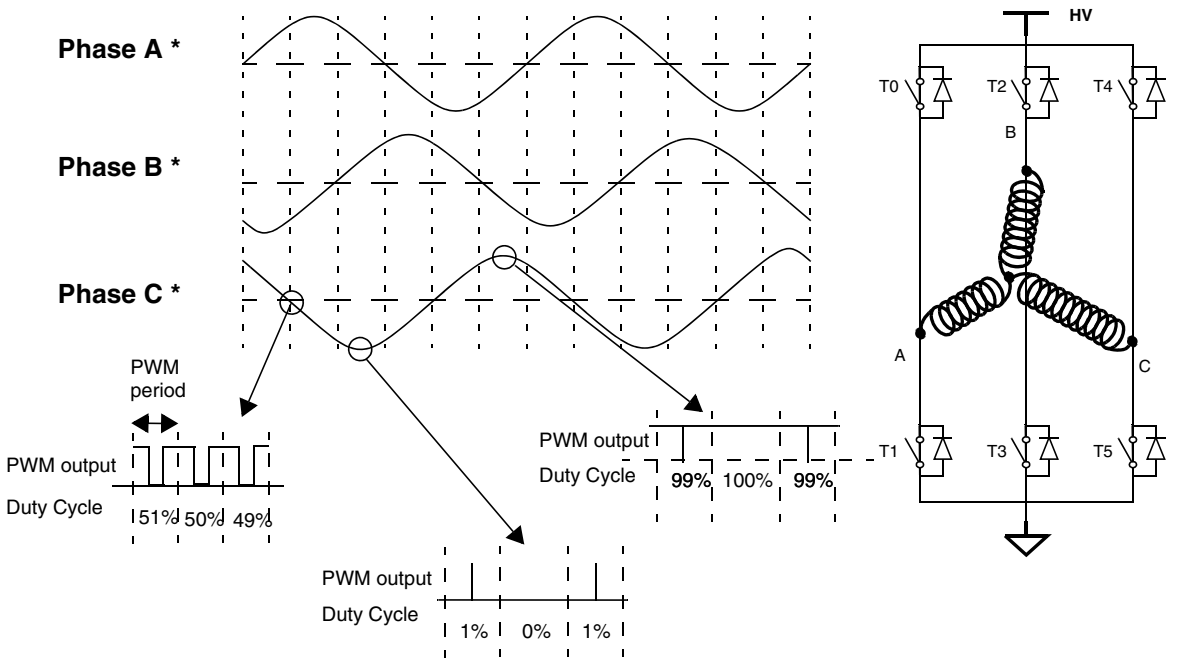
The control and interactions of these registers is explained in Figure 69 and Figure 70. It depends on the LDUM bit setting (LIN Divider Update Method)

Note:

As explained in Figure 69 and Figure 70, LDIV can be updated by two concurrent actions: a transfer from LDIV_MEAS at the end of the LIN Sync Field and a transfer from LDIV_NOM due to a software write of LPR. If both operations occur at the same time, the transfer from LDIV_NOM has priority.

MOTOR CONTROLLER (Cont'd)

Figure 76. Typical command signals of a three-phase induction motor

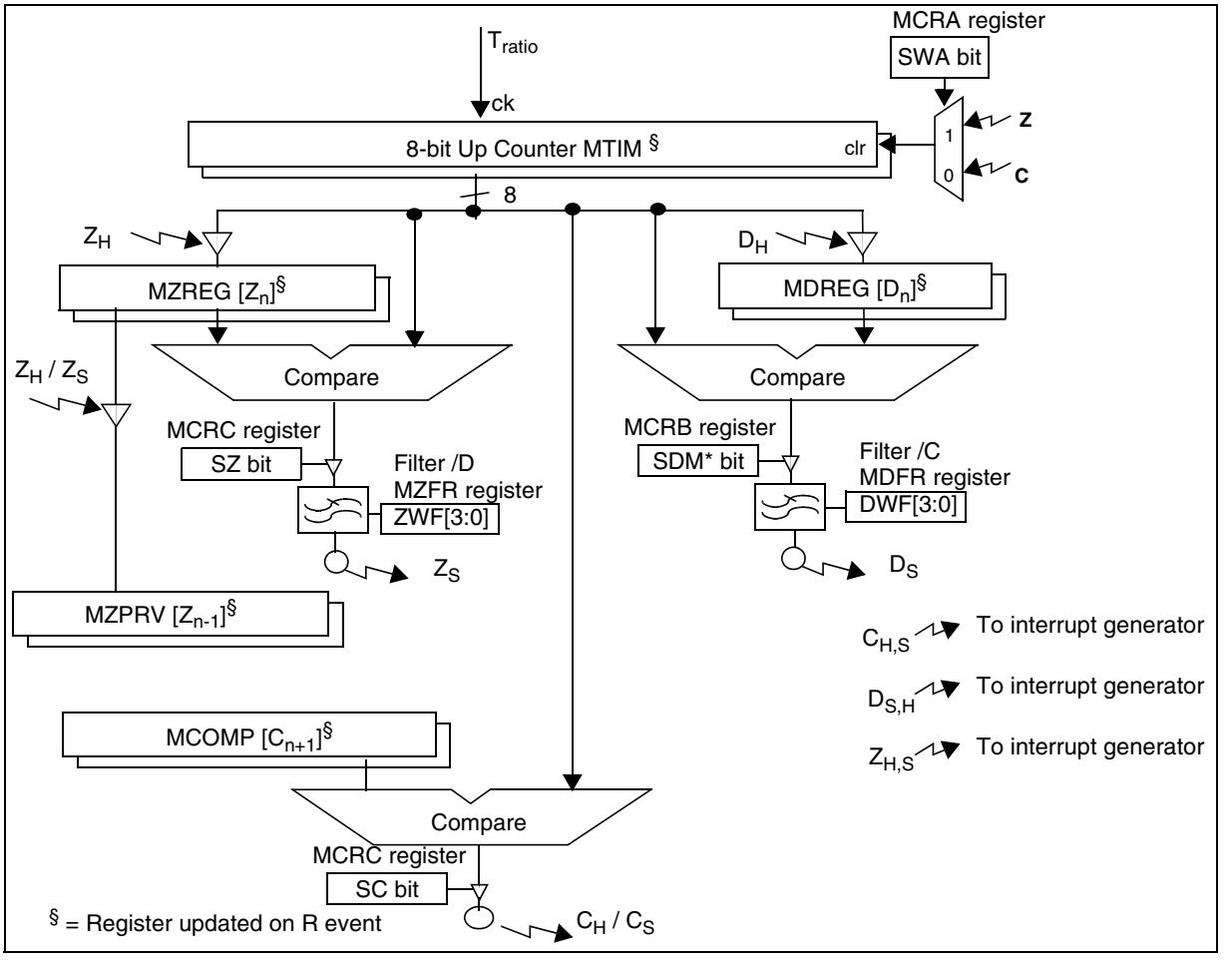


* These simplified chronograms represent the phase voltages after low-pass filtering of the PWM outputs reference signals

MOTOR CONTROLLER (Cont'd)

10.6.7 Delay Manager

Figure 91. Overview of MTIM Timer in Switched and Autoswitched Mode



This part of the MTC contains all the time-related functions, its architecture is based on an 8-bit shift left/shift right timer shown in Figure 91. The MTIM timer includes:

– An auto-updated prescaler

– A capture/compare register for simulated demagnetization simulation (MDREG)

– Two cascaded capture and one compare registers (MZREG and MZPRV) for storing the times between two consecutive BEMF zero crossings (Z_H events) and for zero-crossing event simulation (Z_S)

– An 8x8 bit multiplier for auto computing the next commutation time

– One compare register for phase commutation generation (MCOMP)

The MTIM timer module can work in two main modes when driving synchronous motors in six-steps mode.

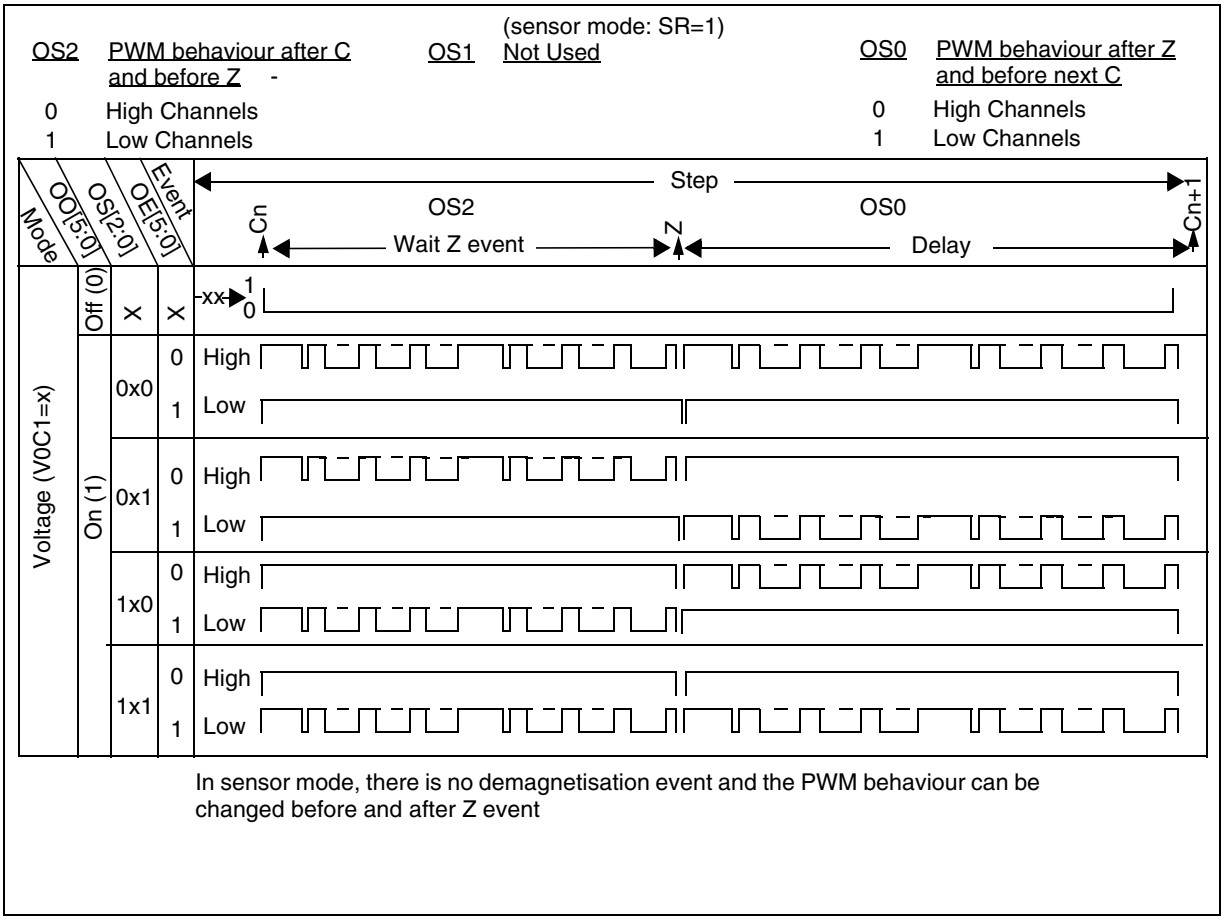
In switched mode the user must process the step duration and commutation time by software.

In autoswitched mode the commutation action is performed automatically depending on the rotor position information and register contents. This is called the hardware commutation event C_H. When enabled by the SC bit in the MCRC register, commutation can also be simulated by writing a value directly in the MCOMP register that is compared with the MTIM value. This is called simulated commutation C_S (See "Built-in Checks and Controls for simulated events" on page 175.).

Both in switched mode and autoswitched mode , if the SC bit in the MCRC register is set (software commutation enabled), no comparison between

MOTOR CONTROLLER (Cont'd)

Figure 110. PWM application in Voltage or Current Sensor Mode (see Table 62)



In sensor mode, there is no demagnetisation event and the PWM behaviour can be changed before and after Z event

INSTRUCTION SET OVERVIEW (Cont'd)

Mnemo	Description	Function/Example	Dst	Src	I1	H	I0	N	Z	C
ADC	Add with Carry	A = A + M + C	A	M		H		N	Z	C
ADD	Addition	A = A + M	A	M		H		N	Z	C
AND	Logical And	A = A . M	A	M				N	Z	
BCP	Bit compare A, Memory	tst (A . M)	A	M				N	Z	
BRES	Bit Reset	bres Byte, #3	M							
BSET	Bit Set	bset Byte, #3	M							
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	M							C
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	M							C
CALL	Call subroutine									
CALLR	Call subroutine relative									
CLR	Clear		reg, M					0	1	
CP	Arithmetic Compare	tst(Reg - M)	reg	M				N	Z	C
CPL	One Complement	A = FFH-A	reg, M					N	Z	1
DEC	Decrement	dec Y	reg, M					N	Z	
HALT	Halt				1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC			I1	H	I0	N	Z	C
INC	Increment	inc X	reg, M					N	Z	
JP	Absolute Jump	jp [TBL.w]								
JRA	Jump relative always									
JRT	Jump relative									
JRF	Never jump	jrf *								
JRIH	Jump if ext. INT pin = 1	(ext. INT pin high)								
JRIL	Jump if ext. INT pin = 0	(ext. INT pin low)								
JRH	Jump if H = 1	H = 1 ?								
JRNH	Jump if H = 0	H = 0 ?								
JRM	Jump if I1:0 = 11	I1:0 = 11 ?								
JRNM	Jump if I1:0 <> 11	I1:0 <> 11 ?								
JRMI	Jump if N = 1 (minus)	N = 1 ?								
JRPL	Jump if N = 0 (plus)	N = 0 ?								
JREQ	Jump if Z = 1 (equal)	Z = 1 ?								
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?								
JRC	Jump if C = 1	C = 1 ?								
JRNC	Jump if C = 0	C = 0 ?								
JRULT	Jump if C = 1	Unsigned <								
JRUGE	Jump if C = 0	Jmp if unsigned >=								
JRUGT	Jump if (C + Z = 0)	Unsigned >								

12.6 MEMORY CHARACTERISTICS

12.6.1 RAM and Hardware Registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode [1]	Halt mode (or RESET)	1.6			V

12.6.2 FLASH Memory

DUAL VOLTAGE HDFLASH MEMORY						
Symbol	**Parameter**	**Conditions**	**Min [2]**	**Typ**	**Max [2]**	**Unit**
f_{CPU}	Operating frequency	Read mode	0		8	MHz
		Write / Erase mode	1		8	
V_{PP}	Programming voltage [3]	4.5V ≤V_{DD} ≤5.5V	11.4		12.6	V
I_{PP}	V_{PP} current[4] [5]	Read (V_{PP}=12V)			200	µA
		Write / Erase			30	mA
t_{VPP}	Internal V_{PP} stabilization time			10		µs
t_{RET}	Data retention	T_A=85°C	40			years
		T_A=105°C	15			
		T_A=125°C	7			
N_{RW}	Write erase cycles	T_A=25°C	100			cycles
T_{PROG} T_{ERASE}	Programming or erasing temperature range		-40	25	85	°C

Notes:

1. Minimum V_{DD} supply voltage without losing data stored in RAM (in Halt mode or under RESET) or in hardware registers (only in Halt mode). Not tested in production.
2. Data based on characterization results, not tested in production.
3. V_{PP} must be applied only during the programming or erasing operation and not permanently for reliability reasons.
4. Data based on simulation results, not tested in production
5. In Write/Erase mode the I_{DD} supply current consumption is the same as in Run mode (section 12.4.1 on page 252)

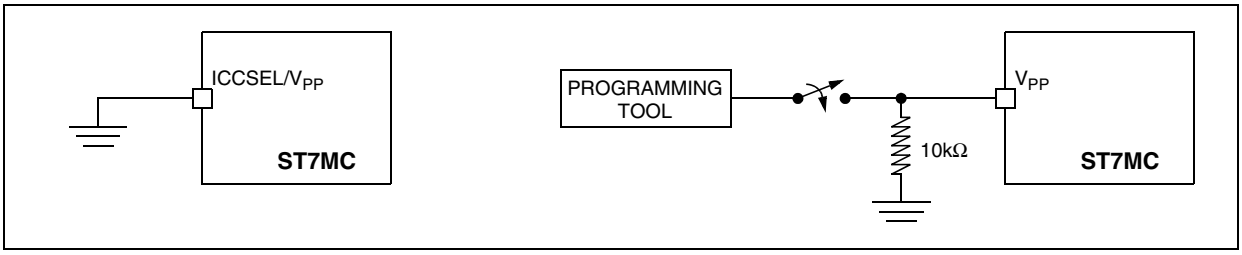
CONTROL PIN CHARACTERISTICS (Cont'd)

12.9.2 ICCSEL/V_{PP} Pin

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

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IL}	Input low level voltage [1]		V_{SS}	0.2	V
V_{IH}	Input high level voltage [1] [2]	ICC mode entry	V_{DD}-0.1	12.6	
I_L	Input leakage current	$V_{IN}=V_{SS}$		±1	µA

Figure 146. Two typical Applications with V_{PP} Pin [3]

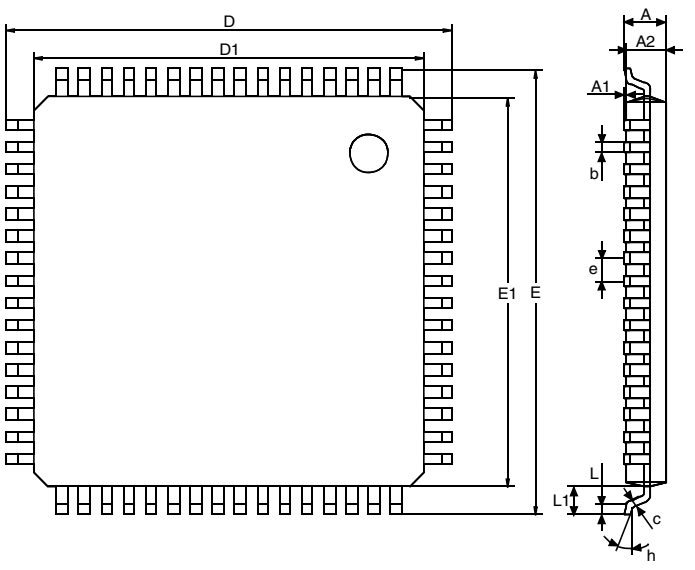


Notes:

1. Data based on design simulation and/or technology characteristics, not tested in production.
2. V_{PP} is also used to program the flash, refer to the Flash characteristics.
3. When the ICC mode is not required by the application ICCSEL/V_{PP} pin must be tied to V_{SS}.

PACKAGE CHARACTERISTICS (Cont'd)

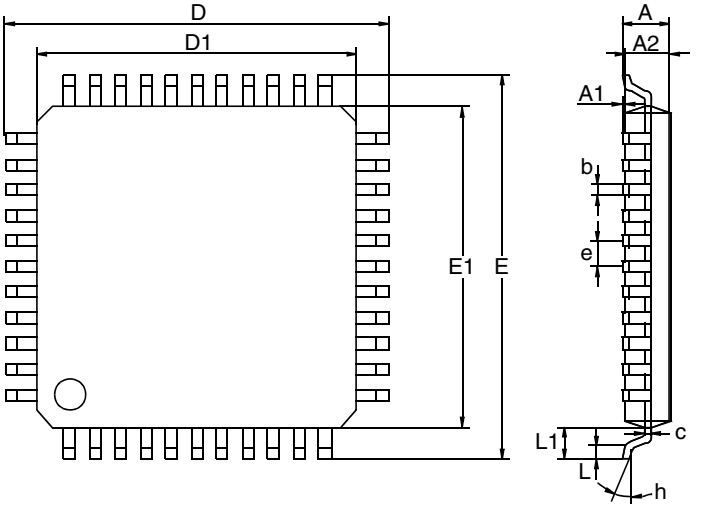
Figure 163. 64-Pin Low Profile Quad Flat Package (14x14)



Dim.	mm			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.0630
A1	0.05		0.15	0.0020		0.0059
A2	1.35	1.40	1.45	0.0531	0.0551	0.0571
b	0.30	0.37	0.45	0.0118	0.0146	0.0177
c	0.09		0.20	0.0035		0.0079
D		16.00			0.6299	
D1		14.00			0.5512	
E		16.00			0.6299	
E1		14.00			0.5512	
e		0.80			0.0315	
θ	0°	3.5°	7°	0°	3.5°	7°
L	0.45	0.60	0.75	0.0177	0.0236	0.0295
L1		1.00			0.0394	
	Number of Pins					
N	64					

Note 1. Values in inches are converted from mm and rounded to 4 decimal digits.

Figure 164. 44-Pin Low Profile Quad Flat Package



Dim.	mm			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.0630
A1	0.05		0.15	0.0020		0.0059
A2	1.35	1.40	1.45	0.0531	0.0551	0.0571
b	0.30	0.37	0.45	0.0118	0.0146	0.0177
C	0.09		0.20	0.0035		0.0079
D		12.00			0.4724	
D1		10.00			0.3937	
E		12.00			0.4724	
E1		10.00			0.3937	
e		0.80			0.0315	
θ	0°	3.5°	7°	0°	3.5°	7°
L	0.45	0.60	0.75	0.0177	0.0236	0.0295
L1		1.00			0.0394	
	Number of Pins					
N	44					

Note 1. Values in inches are converted from mm and rounded to 4 decimal digits.

14 ST7MC DEVICE CONFIGURATION AND ORDERING INFORMATION

Each device is available for production in ROM versions and in user programmable versions (FLASH) as well as in factory coded versions (FASTROM). ST7MC are ROM devices. ST7PMC devices are Factory Advanced Service Technique ROM (FASTROM) versions: they are programmed Flash devices.

ST7FMC Flash devices are shipped to customers with a default content (FFh), while ROM/FASTROM factory coded parts contain the code supplied by the customer. This implies that Flash devices have to be configured by the customer using the Option Bytes while the ROM devices are factory-configured.

14.1 FLASH OPTION BYTES

	STATIC OPTION BYTE 0								STATIC OPTION BYTE 1							
	7							0	7							0
	WDG		CKSEL	VD		$\overline{RSTC}$	DIV2	FMP_R	PKG						MCO	
	HALT	SW		1	0				2	1	0					
Default value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The option bytes allow the hardware configuration of the microcontroller to be selected. They have no address in the memory map and can be accessed only in programming mode (for example using a standard ST7 programming tool). The default content of the FLASH is fixed to FFh. This means that all the options have “1” as their default value.

OPTION BYTE 0

OPT7= **WDG HALT** *Watchdog and Halt mode*
This option bit determines if a RESET is generated when entering Halt mode while the Watchdog is active.
0: No Reset generation when entering Halt mode
1: Reset generation when entering Halt mode

OPT6= **WDG SW** *Hardware or software watchdog*
This option bit selects the watchdog type.
0: Hardware (watchdog always enabled)
1: Software (watchdog to be enabled by software)

OPT5 = **CKSEL** *Clock Source Selection.*
0: PLL clock selected[1)]
1: Oscillator clock selected

Note 1: Even if PLL clock is selected, a clock signal must always be present (refer to Figure 13. on page 28)

OPT4:3= **VD[1:0]** *Voltage detection*
These option bits enable the voltage detection block (LVD, and AVD).

Selected Low Voltage Detector	VD1	VD0
LVD and AVD On	0	0
LVD On and AVD Off	0	1
LVD and AVD Off	1	0
	1	1

OPT2 = **RSTC** *RESET clock cycle selection*
This option bit selects the number of CPU cycles applied during the RESET phase and when exiting Halt mode. For resonator oscillators, it is advised to select 4096 due to the long crystal stabilization time.
0: Reset phase with 4096 CPU cycles
1: Reset phase with 256 CPU cycles
Note: When the PLL clock is selected (CKSEL=0), the reset clock cycle selection is forced to 4096 CPU cycles.

OPT1= **DIV2** *Divider by 2*
1: DIV2 divider disabled with OSCIN = 8MHz
0: DIV2 divider enabled (in order to have 8 MHz required for the PLL with OSCIN =16 Mhz))

OPT0= **FMP_R** *Flash memory read-out protection*
Readout protection, when selected provides a protection against program memory content extraction and against write access to Flash memory. This protection is based on a read/write protection of the memory in test modes and ICP mode. Erasing the option bytes when the FMP_R option is selected causes the whole user memory to be erased first and the device can be reprogrammed. Refer to the ST7 Flash Programming Reference Manual and section 4.3.1 on page 22 for more details.
0: Read-out protection enabled
1: Read-out protection disabled

ST7MC DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

ST7MC MICROCONTROLLER OPTION LIST
(Last update: September 2008)

Customer: . . . . . .
Address: . . . . . .
. . . . . .
Contact: . . . . . .
Phone No: . . . . . .
Reference/ROM or FASTROM Code* : . . . . . .
*The ROM or FASTROM code name is assigned by STMicroelectronics.
ROM or FASTROM code must be sent in .S19 format. .Hex extension cannot be processed.

Device type/memory size/package (check only one option):

ROM	8K	16K	32K	48K	60K
LQFP32:	[ ]				
LQFP44:		[ ]			

FASTROM	8K	16K	32K	48K	60K
LQFP32:	[ ]	[ ]			
LQFP44:		[ ]	[ ]		
LQFP64:			[ ]	[ ]	
LQFP80:					[ ]

Conditioning (check only one option):

[ ] Tape & Reel [ ] Tray

Special Marking [ ] No [ ] Yes "_ _ _ _ _ _ _ _ _ _ " (10 char. max)

Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Temperature range [ ] - 40°C to + 85°C [ ] - 40°C to + 125°C

MCO (Motor Control Output state under reset)	[ ] Hiz	[ ] Low	[ ] High
DIV2	[ ] Disabled		[ ] Enabled
CKSEL	[ ] Oscillator clock		[ ] PLL clock
Watchdog Selection	[ ] Software Activation		[ ] Hardware Activation
Halt when Watchdog on	[ ] Reset		[ ] No reset
Readout Protection	[ ] Disabled		[ ] Enabled
LVD Reset	[ ] Disabled		[ ] Enabled
AVD Interrupt (if LVD enabled)	[ ] Disabled		[ ] Enabled
Reset Delay	[ ] 256 Cycles		[ ] 4096 Cycles

Supply Operating Range in the application: . . . . . .

Notes . . . . . .

Date . . . . . .

Signature . . . . . .

Please download the latest version of this option list from:
http://www.st.com/mcu > downloads > ST7 microcontrollers > Option list

ST7MC DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

14.3 DEVELOPMENT TOOLS

Development tools for the ST7 microcontrollers include a complete range of hardware systems and software tools from STMicroelectronics and third-party tool suppliers. The range of tools includes solutions to help you evaluate microcontroller peripherals, develop and debug your application, and program your microcontrollers.

14.3.1 Starter kits

ST offers complete, affordable **starter kits** and full-featured that allow you to evaluate microcontroller features and quickly start developing ST7 applications. Starter kits are complete hardware/software tool packages that include features and samples to to help you quickly start developing your application.

14.3.2 Development and debugging tools

Application development for ST7 is supported by fully optimizing **C Compilers** and the **ST7 Assembler-Linker** toolchain, which are all seamlessly integrated in the ST7 integrated development environments in order to facilitate the debugging and fine-tuning of your application. The Cosmic C Compiler is available in a free version that outputs up to 16K of code.

The range of hardware tools includes full-featured **ST7-EMU2B series emulators** and the low-cost **RLink** in-circuit debugger/programmer. These tools are supported by the **ST7 Toolset** from STMicroelectronics, which includes the STVD7 integrated development environment (IDE) with high-level language debugger, editor, project manager and integrated programming interface.

14.3.3 Programming tools

During the development cycle, the **ST7-EMU3 series emulators** and the **RLink** provide in-circuit programming capability for programming the Flash microcontroller on your application board.

ST also provides a low-cost dedicated in-circuit programmer, the **ST7-STICK**, as well as **ST7 Socket Boards** which provide all the sockets required for programming any of the devices in a specific ST7 sub-family on a platform that can be used with any tool with in-circuit programming capability for ST7.

For production programming of ST7 devices, ST's third-party tool partners also provide a complete range of gang and automated programming solutions, which are ready to integrate into your production environment.

14.3.4 Order codes for ST7MC development tools

Table 91. Development tool order codes for the ST7MC family

MCU	Starter kit	Emulator	Programming tool
ST7MC1 ST7MC2	ST7MC-KIT/BLDC	ST7MDT50-EMU3	ST7-STICK[1)2)] STX-RLINK[3)]

1. Add suffix /EU, /UK or /US for the power supply for your region

2. Parallel port connection to PC

3. RLink with ST7 tool set

For additional ordering codes for spare parts and accessories, refer to the online product selector at www.st.com/mcu.

IMPORTANT NOTES (Cont'd)

Figure 170. Revision Marking on Box Label and Device Marking