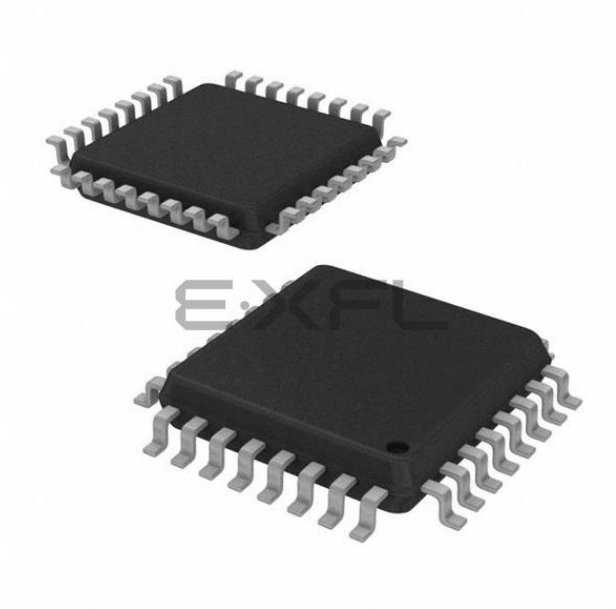


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

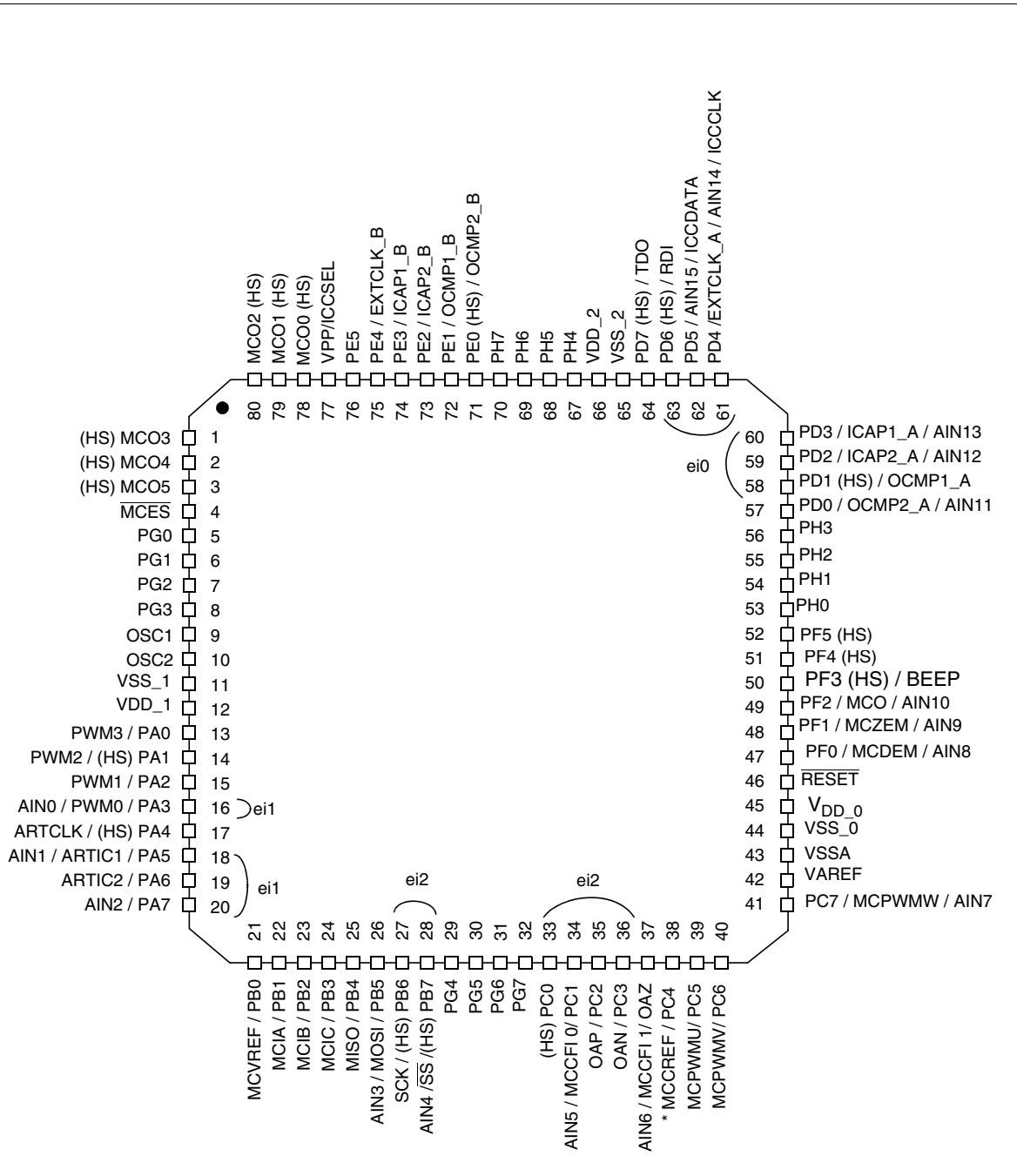
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 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	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fmc1k4t6

2 PIN DESCRIPTION

Figure 2. 80-Pin LQFP 14x14 Package Pinout



(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

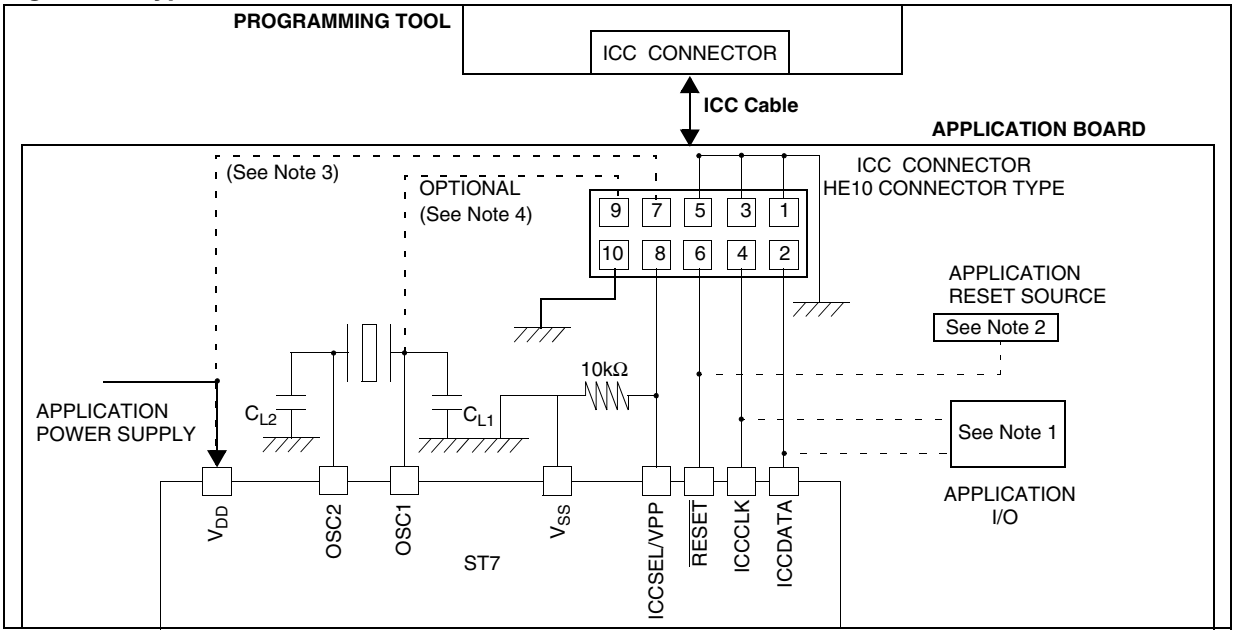
FLASH PROGRAM MEMORY (Cont'd)

4.4 ICC INTERFACE

ICC (In-Circuit Communication) needs a minimum of four and up to six pins to be connected to the programming tool (see Figure 10). These pins are:

- $\overline{\text{RESET}}$: device reset
- V_{SS}: device power supply ground
- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input/output serial data pin
- ICCSEL/V_{PP}: programming voltage
- OSC1(or OSCIN): main clock input for external source (optional)
- V_{DD}: application board power supply (see Figure 10, Note 3)

Figure 10. Typical ICC Interface



Notes:

1. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the Programming Tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.

2. During the ICC session, the programming tool must control the $\overline{\text{RESET}}$ pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5mA at high level (push pull output or pull-up resistor<1K). A schottky diode can be used to isolate the application RESET circuit in this case. When using a classical RC network with R > 1K or a reset management IC with open drain output and pull-up resistor > 1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.

3. The use of Pin 7 of the ICC connector depends on the Programming Tool architecture. This pin must be connected when using most ST Programming Tools (it is used to monitor the application power supply). Please refer to the Programming Tool manual.

4. Pin 9 has to be connected to the OSC1 or OSCIN pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte. ST7 devices with multi-oscillator capability need to have OSC2 grounded in this case.

INTERRUPTS (Cont'd)

Table 9. Nested Interrupts Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0024h		ei1		ei0		MCC + SI		MCES	
	ISPR0	I1_3	I0_3	I1_2	I0_2	I1_1	I0_1		
	Reset Value	1	1	1	1	1	1	1	1
0025h		MTC C/D		MTC R/Z		MTC U/CL		ei2	
	ISPR1	I1_7	I0_7	I1_6	I0_6	I1_5	I0_5	I1_4	I0_4
	Reset Value	1	1	1	1	1	1	1	1
0026h		SCI		TIMER B		TIMER A		SPI	
	ISPR2	I1_11	I0_11	I1_10	I0_10	I1_9	I0_9	I1_8	I0_8
	Reset Value	1	1	1	1	1	1	1	1
0027h						PWMART		AVD	
	ISPR3	I1_15	I0_15	I1_14	I0_14	I1_13	I0_13	I1_12	I0_12
	Reset Value	1	1	1	1	1	1	1	1
0028h	**EICR**	IS11	IS10	IPB	IS21	IS20	IPA	0	0
	Reset Value	0	0	0	0	0	0		

I/O PORTS (Cont'd)

9.5.1 I/O Port Implementation

The I/O port register configurations are summarised as follows.

Standard Ports

PA4, PA2:0, PB5:0, PC7:4, PD7:6, PE5:0, PF5:0, PG7:0, PH7:0

MODE	DDR	OR
floating input	0	0
pull-up input	0	1
open drain output	1	0
push-pull output	1	1

Interrupt Ports

PA6, PA3, PB6, PC3, PC1, PD5, PD4, PD2 (with pull-up)

MODE	DDR	OR
floating input	0	0
pull-up interrupt input	0	1
open drain output	1	0
push-pull output	1	1

PA7, PA5, PB7, PC2, PC0, PD6, PD3, PD1 (without pull-up)

MODE	DDR	OR
floating input	0	0
floating interrupt input	0	1
open drain output	1	0
push-pull output	1	1

Table 12. Port Configuration

Port	Pin name	Input		Output	
		OR = 0	OR = 1	OR = 0	OR = 1
Port A	PA7, PA5	floating	floating interrupt	open drain	push-pull
	PA6, PA3	floating	pull-up interrupt	open drain	push-pull
	PA2:0	floating	pull-up	open drain	push-pull
Port B	PB7	floating	floating interrupt	open drain	push-pull
	PB6	floating	pull-up interrupt	open drain	push-pull
	PB5:0	floating	pull-up	open drain	push-pull
Port C	PC7:4	floating	pull-up	open drain	push-pull
	PC3, PC1	floating	pull-up interrupt	open drain	push-pull
	PC2, PC0	floating	floating interrupt	open drain	push-pull
Port D	PD7, PD0	floating	pull-up	open drain	push-pull
	PD6, PD3, PD1	floating	floating interrupt	open drain	push-pull
	PD5, PD4, PD2	floating	pull-up interrupt	open drain	push-pull
Port E	PE5:0	floating	pull-up	open drain	push-pull
Port F	PF5:0	floating	pull-up	open drain	push-pull
Port G	PG7:0	floating	pull-up	open drain	push-pull
Port H	PH7:0	floating	pull-up	open drain	push-pull

(Cont'd)

Each Timer is associated with three control and status registers, and with six pairs of data registers (16-bit values) relating to the two input captures, the two output compares, the counter and the alternate counter.

Read/Write

Reset Value: 0000 0000 (00h)

7							0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1

Bit 7 = *Input Capture Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is generated whenever the ICF1 or ICF2 bit of the SR register is set.

Bit 6 = *Output Compare Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is generated whenever the OCF1 or OCF2 bit of the SR register is set.

Bit 5 = *Timer Overflow Interrupt Enable.*
0: Interrupt is inhibited.
1: A timer interrupt is enabled whenever the TOF bit of the SR register is set.

Bit 4 = *Forced Output Compare 2.*
This bit is set and cleared by software.
0: No effect on the OCMP2 pin.
1: Forces the OLVL2 bit to be copied to the OCMP2 pin, if the OC2E bit is set and even if there is no successful comparison.

Bit 3 = *Forced Output Compare 1.*
This bit is set and cleared by software.
0: No effect on the OCMP1 pin.
1: Forces OLVL1 to be copied to the OCMP1 pin, if the OC1E bit is set and even if there is no successful comparison.

Bit 2 = *Output Level 2.*
This bit is copied to the OCMP2 pin whenever a successful comparison occurs with the OC2R register and OCxE is set in the CR2 register. This value is copied to the OCMP1 pin in One Pulse Mode and Pulse Width Modulation mode.

Bit 1 = *Input Edge 1.*
This bit determines which type of level transition on the ICAP1 pin will trigger the capture.
0: A falling edge triggers the capture.
1: A rising edge triggers the capture.

Bit 0 = *Output Level 1.*
The OLVL1 bit is copied to the OCMP1 pin whenever a successful comparison occurs with the OC1R register and the OC1E bit is set in the CR2 register.

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

10.5.5 SCI Mode - Functional Description

Conventional Baud Rate Generator Mode

The block diagram of the Serial Control Interface in conventional baud rate generator mode is shown in Figure 62.

It uses four registers:

– 2 control registers (SCICR1 and SCICR2)

– A status register (SCISR)

– A baud rate register (SCIBRR)

Extended Prescaler Mode

Two additional prescalers are available in extended prescaler mode. They are shown in Figure 64.

– An extended prescaler receiver register (SCIERPR)

– An extended prescaler transmitter register (SCIETPR)

10.5.5.1 Serial Data Format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the SCICR1 register (see Figure 63).

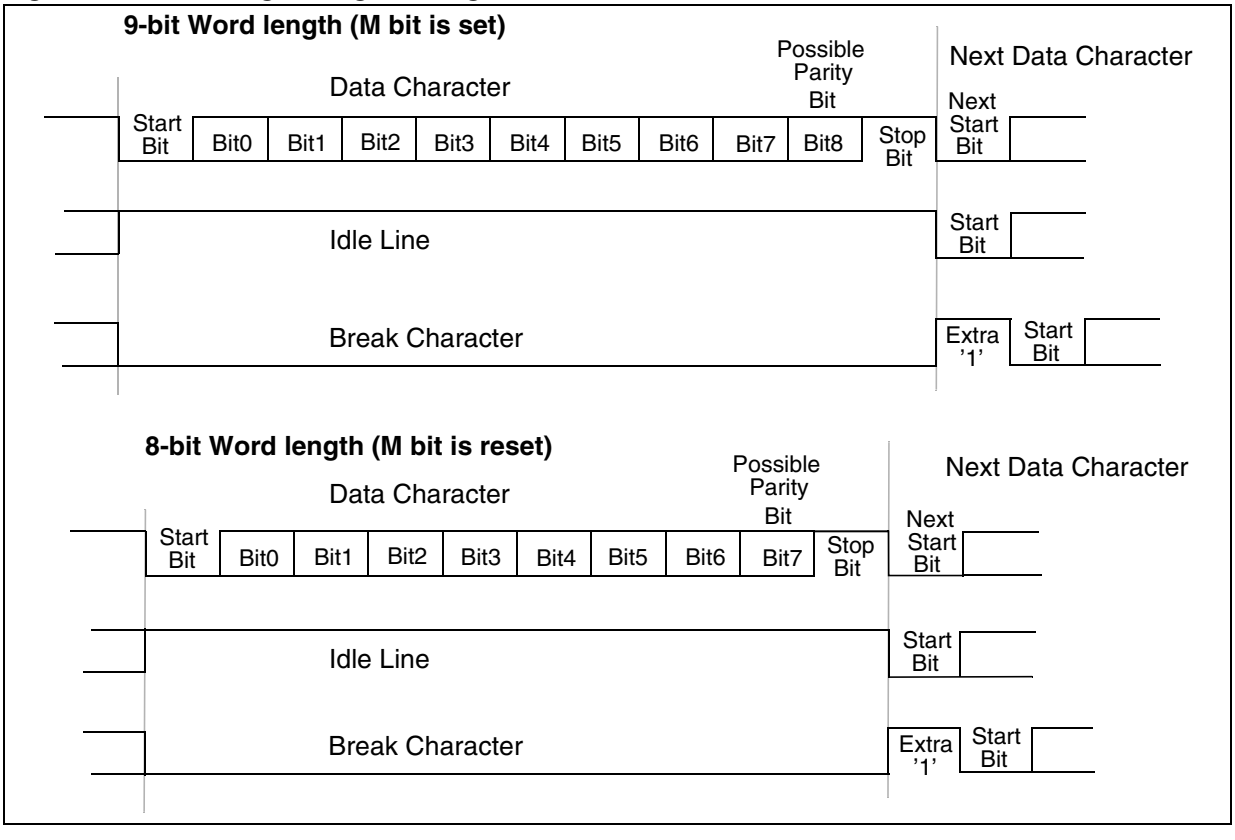
The TDO pin is in low state during the start bit.

The TDO pin is in high state during the stop bit.

An Idle character is interpreted as a continuous logic high level for 10 (or 11) full bit times.

A Break character is a character with a sufficient number of low level bits to break the normal data format followed by an extra "1" bit to acknowledge the start bit.

Figure 63. Word Length Programming



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

10.5.9.4 LIN Error Detection

LIN Header Error Flag

The LIN Header Error Flag indicates that an invalid LIN Header has been detected.

When a LIN Header Error occurs:

– The LHE flag is set

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

If autosynchronization is enabled (LASE bit = 1), this can mean that the LIN Synch Field is corrupted, and that the SCI is in a blocked state (LSF bit is set). The only way to recover is to reset the LSF bit and then to clear the LHE bit.

– The LHE bit is reset by an access to the SCISR register followed by a read of the SCIDR register.

LHE/OVR Error Conditions

When Auto Resynchronization is disabled (LASE bit = 0), the LHE flag detects:

– That the received LIN Synch Field is not equal to 55h.

– That an overrun occurred (as in standard SCI mode)

– Furthermore, if LHDM is set it also detects that a LIN Header Reception Timeout occurred (only if LHDM is set).

When the LIN auto-resynchronization is enabled (LASE bit = 1), the LHE flag detects:

– That the deviation error on the Synch Field is outside the LIN specification which allows up to +/-15.5% of period deviation between the slave and master oscillators.

– A LIN Header Reception Timeout occurred. If $T_{HEADER} > T_{HEADER\_MAX}$ then the LHE flag is set. Refer to Figure 67. (only if LHDM is set to 1)

– An overflow during the Synch Field Measurement, which leads to an overflow of the divider registers. If LHE is set due to this error then the SCI goes into a blocked state (LSF bit is set).

– That an overrun occurred on Fields other than the Synch Field (as in standard SCI mode)

Deviation Error on the Synch Field

The deviation error is checking by comparing the current baud rate (relative to the slave oscillator) with the received LIN Synch Field (relative to the master oscillator). Two checks are performed in parallel:

– The first check is based on a measurement between the first falling edge and the last falling edge of the Synch Field. Let us refer to this period deviation as D:

If the LHE flag is set, it means that:

D > 15.625%

If LHE flag is not set, it means that:

D < 16.40625%

If 15.625% ≤D < 16.40625%, then the flag can be either set or reset depending on the dephasing between the signal on the RDI line and the CPU clock.

– The second check is based on the measurement of each bit time between both edges of the Synch Field: this checks that each of these bit times is large enough compared to the bit time of the current baud rate.

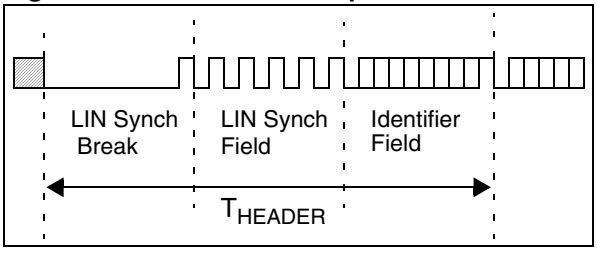
When LHE is set due to this error then the SCI goes into a blocked state (LSF bit is set).

LIN Header Time-out Error

When the LIN Identifier Field Detection Method is used (by configuring LHDM to 1) or when LIN auto-resynchronization is enabled (LASE bit = 1), the LINSCI automatically monitors the T_{HEADER_MAX} condition given by the LIN protocol.

If the entire Header (up to and including the STOP bit of the LIN Identifier Field) is not received within the maximum time limit of 57 bit times then a LIN Header Error is signalled and the LHE bit is set in the SCISR register.

Figure 67. LIN Header Reception Timeout



The time-out counter is enabled at each break detection. It is stopped in the following conditions:
- A LIN Identifier Field has been received
- An LHE error occurred (other than a timeout error).
- A software reset of LSF bit (transition from high to low) occurred during the analysis of the LIN Synch Field or

If LHE bit is set due to this error during the LIN Synchr Field (if LASE bit = 1) then the SCI goes into a blocked state (LSF bit is set).

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

LIN HEADER LENGTH REGISTER (LHLR)
Read Only
Reset Value: 0000 0000 (00h).

7							0
LHL7	LHL6	LHL5	LHL4	LHL3	LHL2	LHL1	LHL0

Note: In LIN Slave mode when LASE = 1 or LHDM = 1, the LHLR register is accessible at the address of the SCIERPR register.

Otherwise this register is always read as 00h.

Bits 7:0 = **LHL[7:0]** *LIN Header Length.*

This is a read-only register, which is updated by hardware if one of the following conditions occurs:
- After each break detection, it is loaded with "FFh".
- If a timeout occurs on T_{HEADER}, it is loaded with 00h.
- After every successful LIN Header reception (at the same time than the setting of LHDF bit), it is loaded with a value (LHL) which gives access to the number of bit times of the LIN header length (T_{HEADER}). The coding of this value is explained below:

LHL Coding:

T_{HEADER_MAX} = 57

LHL(7:2) represents the mantissa of (57 - T_{HEADER})

LHL(1:0) represents the fraction (57 - T_{HEADER})

LHL[7:2]	Mantissa (57 - T_{HEADER})	Mantissa (T_{HEADER})
0h	0	57
1h	1	56
...	...	...
39h	56	1
3Ah	57	0
3Bh	58	Never Occurs
...	...	...
3Eh	62	Never Occurs
3Fh	63	Initial value

LHL[1:0]	Fraction (57 - T_{HEADER})
0h	0
1h	1/4
2h	1/2
3h	3/4

Example of LHL coding:

Example 1: LHL = 33h = 001100 11b

LHL(7:3) = 1100b = 12d

LHL(1:0) = 11b = 3d

This leads to:

Mantissa (57 - T_{HEADER}) = 12d

Fraction (57 - T_{HEADER}) = 3/4 = 0.75

Therefore:

(57 - T_{HEADER}) = 12.75d

and T_{HEADER} = 44.25d

Example 2:

57 - T_{HEADER} = 36.21d

LHL(1:0) = rounded(4*0.21d) = 1d

LHL(7:2) = Mantissa (36.21d) = 36d = 24h

Therefore LHL(7:0) = 10010001 = 91h

Example 3:

57 - T_{HEADER} = 36.90d

LHL(1:0) = rounded(4*0.90d) = 4d

The carry must be propagated to the matissa:

LHL(7:2) = Mantissa (36.90d) + 1 = 37d =

Therefore LHL(7:0) = 10110000 = A0h

MOTOR CONTROLLER (Cont'd)

10.6.6.13 Encoder Mode (IS[1:0] = 11)

Figure 90 shows the signals delivered by a standard digital incremental encoder and associated information:

- Two 90° phased square signals with variable frequency proportional to the speed; they must be connected to MCIA and MCIB input pins,
- Clock derived from incoming signal edges,
- Direction information determined by the relative phase shift of input signals (+ or -90°).

The Incremental Encoder Interface block aims at extracting these signals. As input logic is both rising and falling edge sensitive (independently from TES[1:0] bits setting), resulting clock frequency is four times the one of the input signals, thus increasing resolution for measurements.

It may be noticed that Direction bit (EDIR bit in MCRC register) is read only and that it does'nt affect counting direction of clocked timer (cf Section). As a result, one cannot extract position information from encoder inputs during speed reversal.

Figure 89. Tacho Capture events configured by the TES[1:0] bits

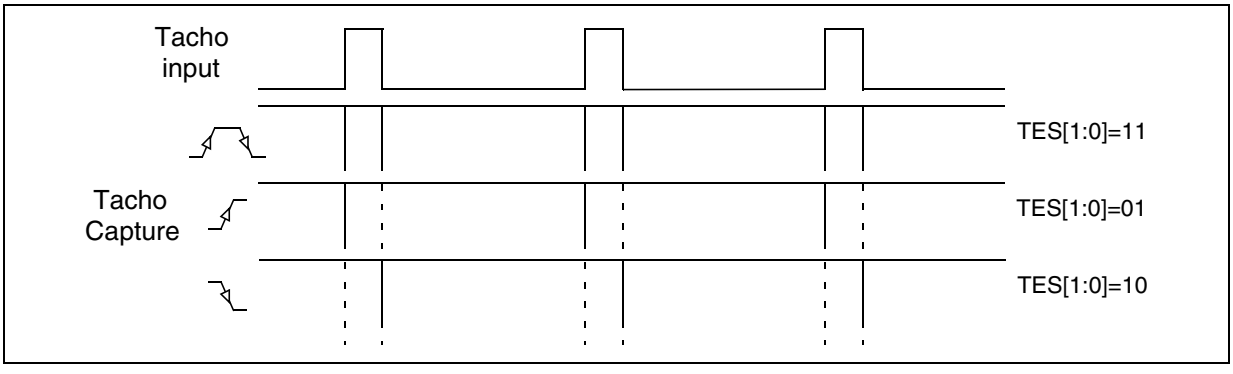
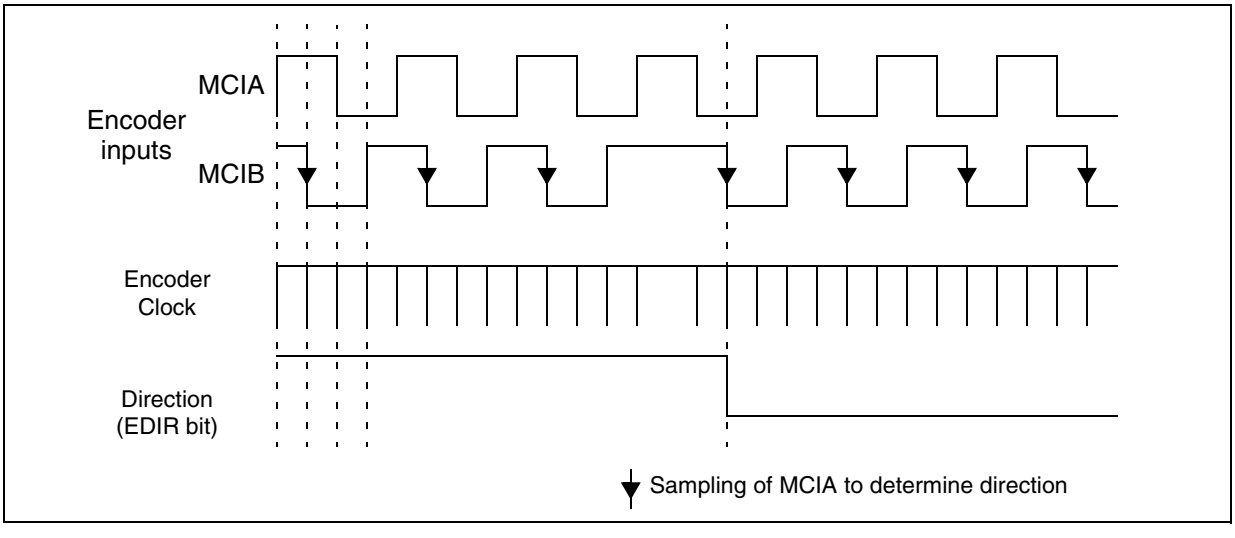


Figure 90. Incremental Encoder output signals and derived information



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)

10.6.7.5 Speed Measurement Mode

Motor speed can be measured using two methods depending on sensor type: period measurement or pulse counting. Typical sensor handling is described here.

Incremental encoders allows accurate speed measurement by providing a large number of pulses per revolution (ppr) with ppr rates up to several thousands; the higher the ppr rate, the higher the resolution. The proposed method consists of counting the number of clock cycles issued by the Incremental Encoder Interface (Encoder Clock) during a fixed time window (refer to Figure 100).

The tachogenerator has a much lower ppr rate than the encoder (typically factor 10). In this context, it is more meaningful to measure the period between Tacho Captures (i.e. relevant transitions of the incoming signals). Accuracy is imposed by the reference clock, i.e. the CPU clock (refer to Figure 99).

Figure 99. Tachogenerator period acquisition using MTIM timer

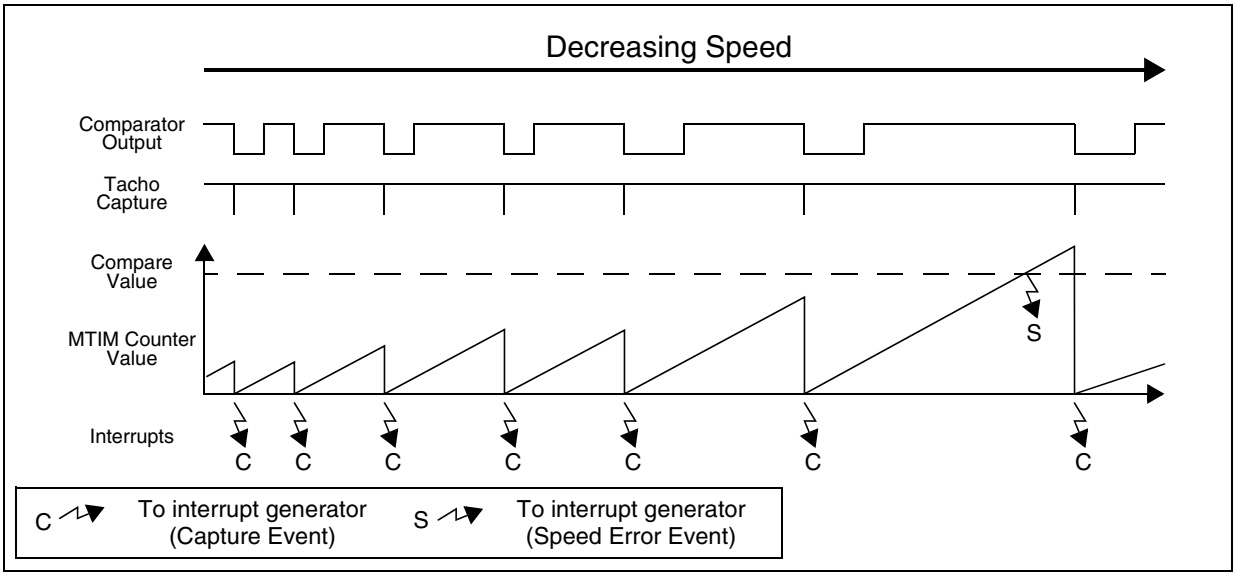
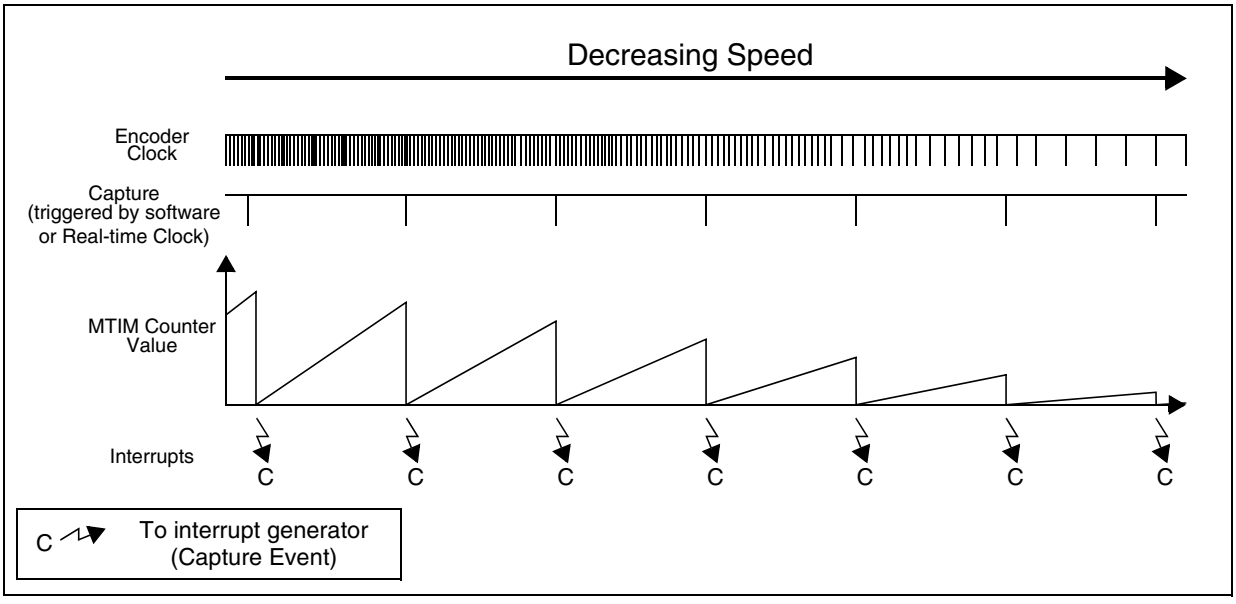


Figure 100. Encoder Clock frequency measure using MTIM timer



MOTOR CONTROLLER (Cont'd)

Three kinds of interrupt can be generated in Speed Sensor Mode, as summarized in Figure 104:

– C interrupt, when a capture event occurs; this interrupt shares resources (Mask bit and Flag) with the Commutation event in Switched/Autoswitched Mode, as these modes are mutually exclusive.

– RPI/RMI interrupts occur when the ST[3:0] bits of the MPSR register are changed, either automatically or by hardware.

– S interrupt occurs when a Speed Error happens (i.e. a successful comparison between [MTIM:MTIML] and [MDREG:FF]). This interrupt has the same channel as the Emergency Stop interrupt (MCES), as it also warns the user about abnormal system operation. The respective Flag bits have to be tested in the interrupt service routine to differentiate Speed Errors from Emergency Stop events.

These interrupts may be masked individually.

Note on Delay Manager Initialization in Speed Measurement Mode: In order to set-up the [MTIM:MTIML] counter properly before any speed measurement, the following procedure must be applied:

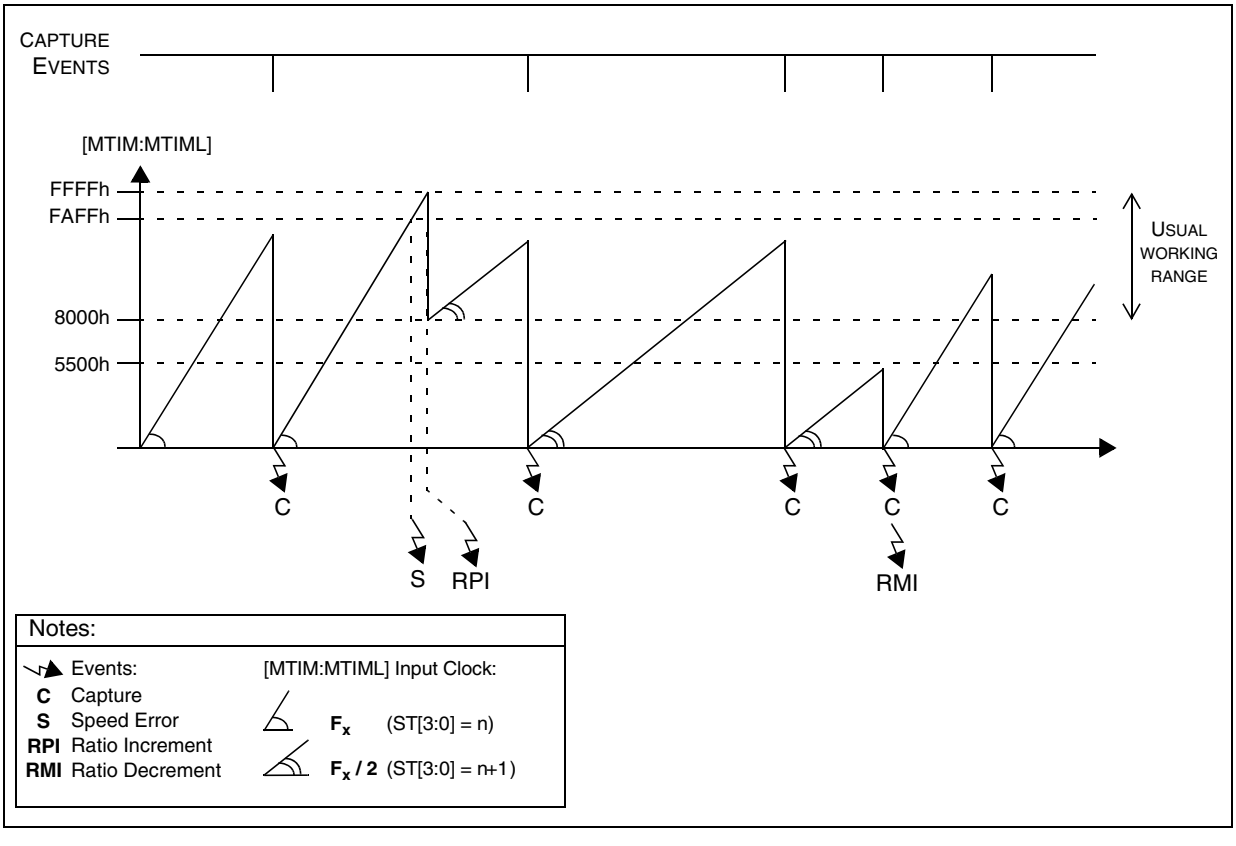
– The peripheral clock must be disabled (resetting the CKE bit in the MCRA register) to allow write access to ST[3:0], MTIM and MTIML (refer to Table 41),

– MTIM, MTIML must be reset and appropriate values must be written in the ST[3:0] prescaler adapt to the frequency of the signal being measured and to allow speed measurement with sufficient resolution.

Note on MTIML: The Least Significant Byte of the counter (MTIML) is not used when working in Position Sensor or Sensorless Modes.

Debug option: a signal reflecting the capture events may be output on a standard I/O port for debugging purposes. Refer to section10.6.7.3 on page 172 for more details.

Figure 104. Prescaler auto-change example



MOTOR CONTROLLER (Cont'd)

INTERRUPT STATUS REGISTER (MISR)
Read/Write
Reset Value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
PUI	RPI	RMI	CLI	EI	ZI	DI	CI

Bit 7 = **PUI:** *PWM Update Interrupt flag.*

This bit is set by hardware when all the PWM Compare register are transferred from the preload to the active registers. The corresponding interrupt allows the user to refresh the preload registers before the next PWM update event defined with MREP register.
0: No PWM Update interrupt pending
1: PWM Update Interrupt pending

Bit 6 = **RPI**: *Ratio Increment interrupt flag.*
Autoswitched mode (SWA bit =1)**:**
0: No R+ interrupt pending
1: R+ Interrupt pending

Switched mode (SWA bit =0)**:**
0: No R+ action
1: The hardware will increment the ST[3:0] bits when the next commutation occurs and shift all timer registers right.

Speed Sensor mode (SWA bit =x, TES[1:0] bits =01, 10, 11)**:**
0: No R+ interrupt pending
1: R+ Interrupt pending

Bit 5 = **RMI**: *Ratio Decrement interrupt flag.*
Autoswitched mode (SWA bit =1)**:**
0: No R- interrupt pending
1: R- Interrupt pending
Switched mode (SWA bit =0)**:**
0: No R- action
1: The hardware will decrement the ST[3:0] bits when the next commutation occurs and shift all timer registers left.

Speed Sensor mode (SWA bit =x, TES[1:0] bits =01, 10, 11)**:**
0: No R- interrupt pending
1: R- Interrupt pending

Bit 4 = **CLI**: *Current Limitation interrupt flag.*
0: No Current Limitation interrupt pending
1: Current Limitation interrupt pending

Bit 3 = **EI**: *Emergency stop Interrupt flag.*
0: No Emergency stop interrupt pending
1: Emergency stop interrupt pending

Bit 2 = **ZI**: *BEMF Zero-crossing interrupt flag.*
0: No BEMF Zero-crossing Interrupt pending
1: BEMF Zero-crossing Interrupt pending

Bit 1 = **DI**: *End of Demagnetization interrupt flag.*
0: No End of Demagnetization interrupt pending
1: End of Demagnetization interrupt pending

Bit 0 = **CI**: *Commutation / Capture interrupt flag*
0: No Commutation / Capture Interrupt pending
1: Commutation / Capture Interrupt pending

Note 1: Loading value FFh in the MISR register will reset the PWM generator counter and transfer the compare preload registers in the active registers by generating a U event (PUI bit set to 1). Refer to "Timer Re-synchronisation" on page 206.

Note 2: When several MTC interrupts are enabled at the same time the BRES instruction must not be used to avoid unwanted clearing of status flags: if a second interrupt occurs while BRES is executed (which performs a read-modify-write sequence) to clear the flag of a first interrupt, the flag of the second interrupt may also be cleared and the corresponding interrupt routine will not be serviced. It is thus recommended to use a load instruction to clear the flag, with a value equal to the logical complement of the bit. For instance, to clear the PUI flag:

ld MISR, # 0x7F.

Note 3: **In Autoswitched mode** (SWA=1 in the MRCA register): As all bits in the MISR register are status flags, they are set by internal hardware signals and must be cleared by software. Any attempt to write them to 1 will have no effect (they will be read as 0) without interrupt generation.

In Switched mode (SWA=0 in the MRCA register):

To avoid any losing any interrupts when modifying the RMI and RPI bits the following instruction sequence is recommended:

ld MISR, # 0x9F ; reset both RMI & RPI bits

ld MISR, # 0xBF ; set RMI bit

ld MISR, # 0xDF ; set RPI bit

MOTOR CONTROLLER (Cont'd)

Effect on PWM generator: the PWM generator 12-bit counter is reset as soon as CKE = 0; this ensures that the PWM signals start properly in all cases. When these bits are set, all registers with preload on Update event are transferred to active registers.

Bit 5 = **SR**: *Sensor ON/OFF.*
0: Sensorless mode
1: Position Sensor mode

Table 57. Sensor Mode Selection

SR bit	Mode	OS[2:0] bits	Behaviour of the output PWM
0	Sensors not used	OS[2:0] bits enabled	"Between C_n&D" behaviour, "between D&Z" behaviour and "between Z&C_{n+1}" behaviour
1	Sensors used	OS1 disabled	"Between C_n&Z" behaviour and "between Z&C_{n+1}" behaviour

See also Table 61 and Table 62

Bit 4 = **DAC**: *Direct Access to phase state register.*
0: No Direct Access (reset value). In this mode the preload value of the MPHST and MCRB registers is taken into account at the C event.
1: Direct Access enabled. In this mode, write a value in the MPHST register to access the outputs directly.

Note: In Direct Access Mode (DAC bit is set in MCRA register), a C event is generated as soon as there is a write access to the OO[5:0] bits in MPHST register. In this case, the PWM low/high selection is done by the OS0 bit in the MCRB register.

Table 58. DAC Bit Meaning

MOE bit	DAC bit	Effect on Output
0	x	Reset state depending on the option bit
1	0	Standard running mode.
1	1	MPHST register value (depending on MPOL, MPAR register values and PWM setting) see Table 74

Bit 3 = **V0C1**: *Voltage/Current Mode*
0: Voltage Mode
1: Current Mode

Bit 2 = **SWA**: *Switched/Autoswitched Mode*
0: Switched Mode
1: Autoswitched Mode

Note 1 : after reset, in autoswitched mode (SWA =1) , the motor control peripheral is waiting for a C commutation event.

Note 2: After reset, a C event is immediately generated when CKE and SWA are simultaneaously set due to a nil value of MCOMP.

Bit 1 = **PZ**: *Protection from parasitic Zero-crossing event detection*
0: Protection disabled
1: Protection enabled

Note: If the PZ bit is set, the Z event filter (ZEF[3:0] in the MZFR register is ignored.

Bit 0 = **DCB**: *Data Capture bit*
0: Use MZPRV (Z_N-1) for multiplication
1: Use MZREG (Z_N) for multiplication

Table 59. Multiplier Result

DCB bit	Commutation Delay
0	MCOMP = MWGHT x MZPRV / 256
1	MCOMP = MWGHT x MZREG / 256

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

MOTOR CONTROL CHARACTERISTICS (Cont'd)

Figure 150. Example 1: Waveforms for Zero-crossing Detection with Sampling at the end of PWM off-time

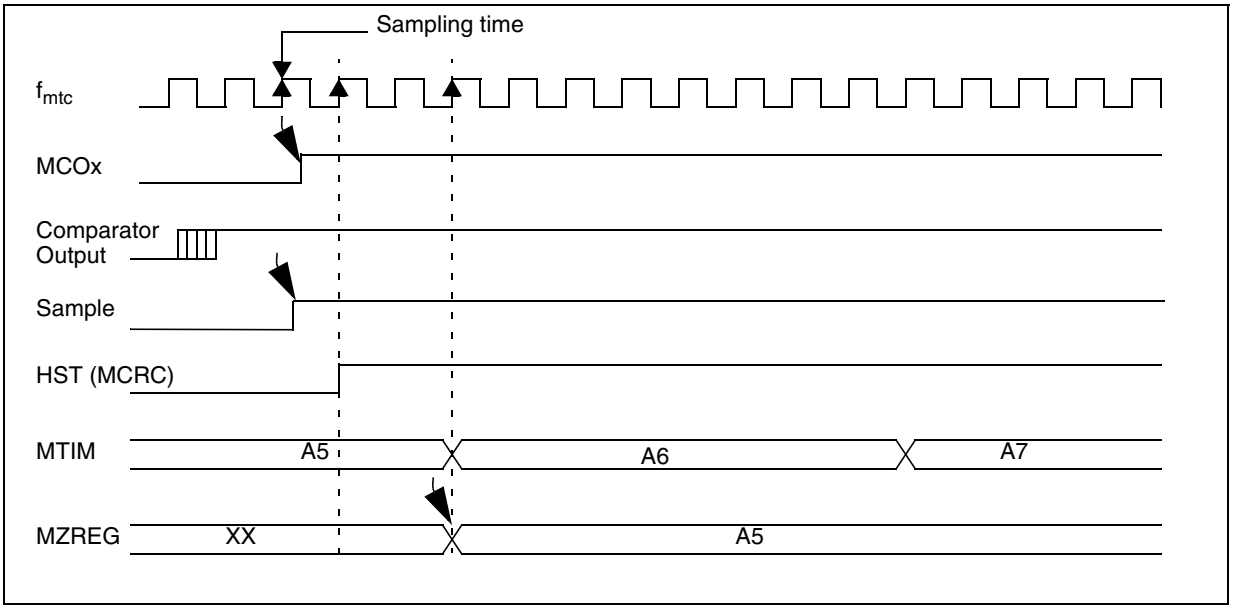
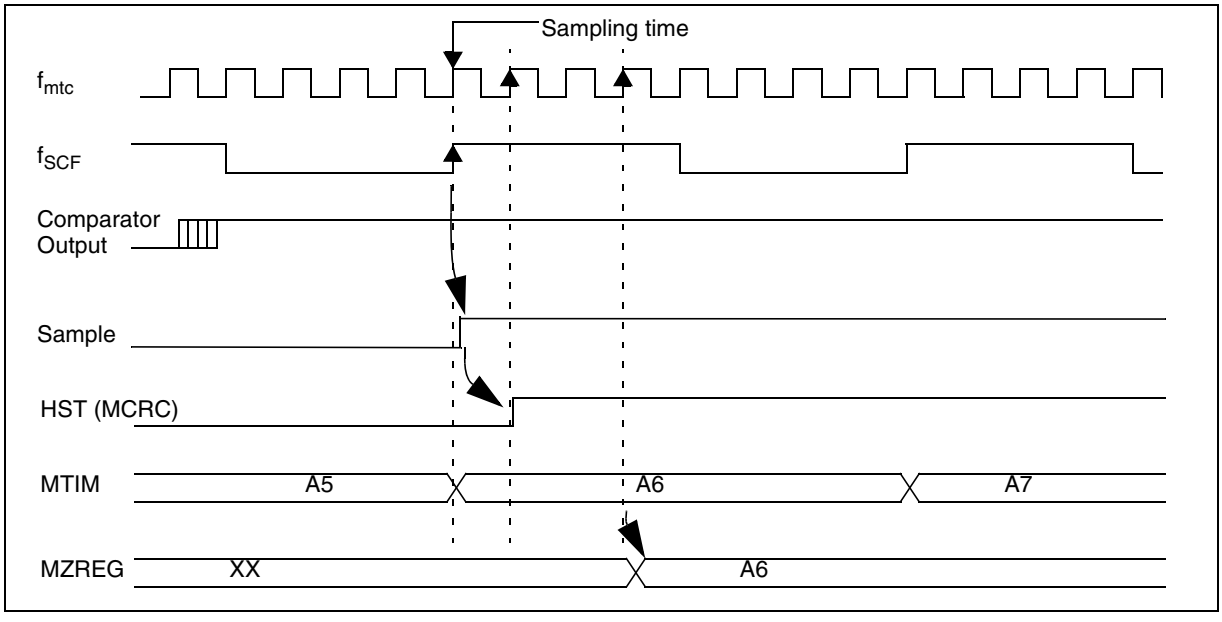


Figure 151. Example 2: Waveforms for Zero-crossing Detection with Sampling at f_{SCF}



MOTOR CONTROL CHARACTERISTICS (Cont'd)

12.12.3 Input Stage (Current Feedback Comparator + Sampling)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IN}	Comparator input voltage range		V_{SSA} - 0.1		V_{DD} + 0.1	V
V_{offset}	Comparator offset error			5	40 [1]	mV
I_{offset}	Input offset current				1	µA
t_{propag}	Comparator propagation delay [1]			35	100	ns
$t_{startup}$	Start-up filter duration [2]	Time waited before sampling when comparator is turned ON, i.e. CKE=1 or DAC=1 (with f_{PERIPH} = 4MHz)	3			µs
$t_{sampling}$	Digital sampling delay [3]	Time needed to turn OFF the MCOs when comparator output rises (CFF=0)	4 / f_{MTC} (see Figure 155)			
		Time between a comparator toggle (current loop event) and bit CL becoming set (CFF=0)	2 / f_{MTC} (see Figure 155)			
		Time needed to turn OFF the MCOs when comparator output rises (CFF=x)	(1+x) * (4 / f_{PERIPH}) + (3 / f_{mtc}) (see Figure 156)			
		Time between a comparator toggle (current loop event) and bit CL becoming set (CFF=x)	(1+x) * (4 / f_{PERIPH}) + (1 / f_{mtc}) (see Figure 156)			

Notes:

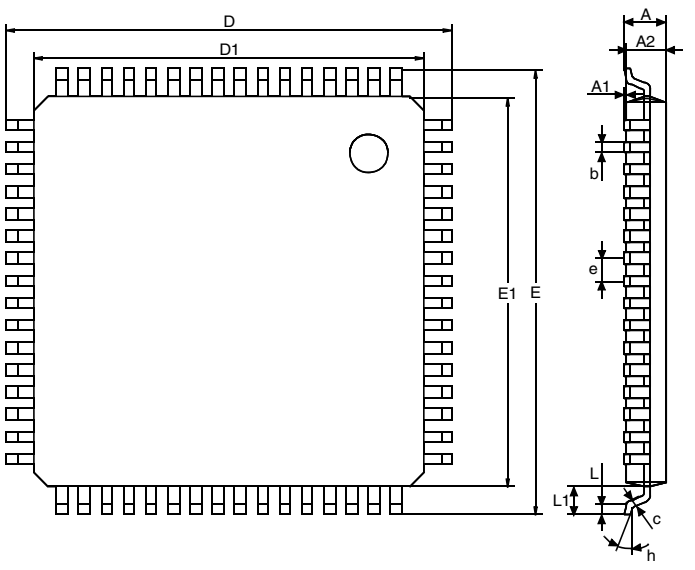
1. The comparator accuracy depends on the environment. In particular, the following cases may reduce the accuracy of the comparator and must be avoided:
– Negative injection current on the I/Os close to the comparator inputs
– Switching on I/Os close to the comparator inputs
– Negative injection current on not used comparator input (MCCFI0 or MCCFI1)
– Switching with a high dV/dt on not used comparator input (MCCFI0 or MCCFI1)
These phenomena are even more critical when a big external serial resistor is added on the inputs.

2. This filter is implemented to wait for comparator stabilization and avoid any wrong information during start-up.

3.This delay represents the number of clock cycles needed to generate an event as soon as the comparator ouput changes.
Example: When CFF=0 (detection is based on a single detection), MCO outputs are turned OFF at the 4th clock cycle after comparator commutation, i.e. there is a variation of ($1/f_{mtc}$) or (4 / f_{PERIPH}) depending on the case.

PACKAGE CHARACTERISTICS (Cont'd)

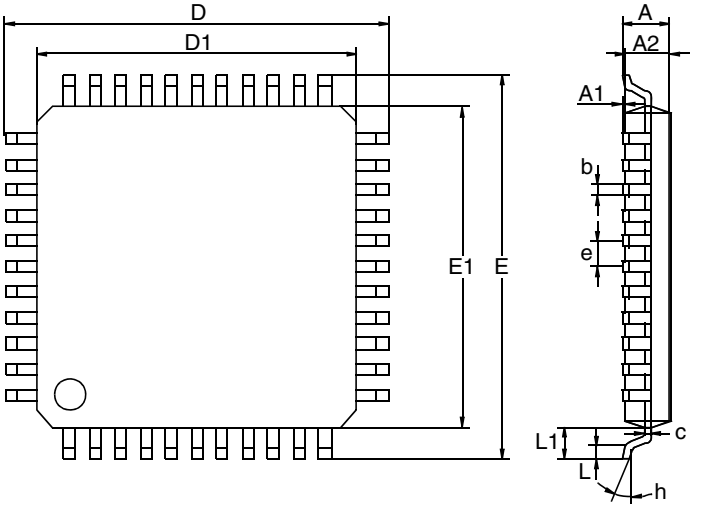
Figure 163. 64-Pin Low Profile Quad Flat Package (14x14)



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.

ST7MC DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

OPTION BYTE 1

OPT7:5= **PKG[2:0]** *package selection*
These option bits are used to select the device package.

Selected Package	PKG2	PKG1	PKG0
LQFP32 / SDIP32	0	0	0
LQFP44	0	0	1
SDIP 56	0	1	0
LQFP64	0	1	1
LQFP80	1	x	x

OPT4:2= Reserved

OPT1:0 = **MCO** *Motor Control Output Options*
MCO port under reset.

Motor Control Output	bit 1	bit 0
HiZ	0	0
Low	0	1
High	1	0
HiZ	1	1