Welcome to E-XFL.COM

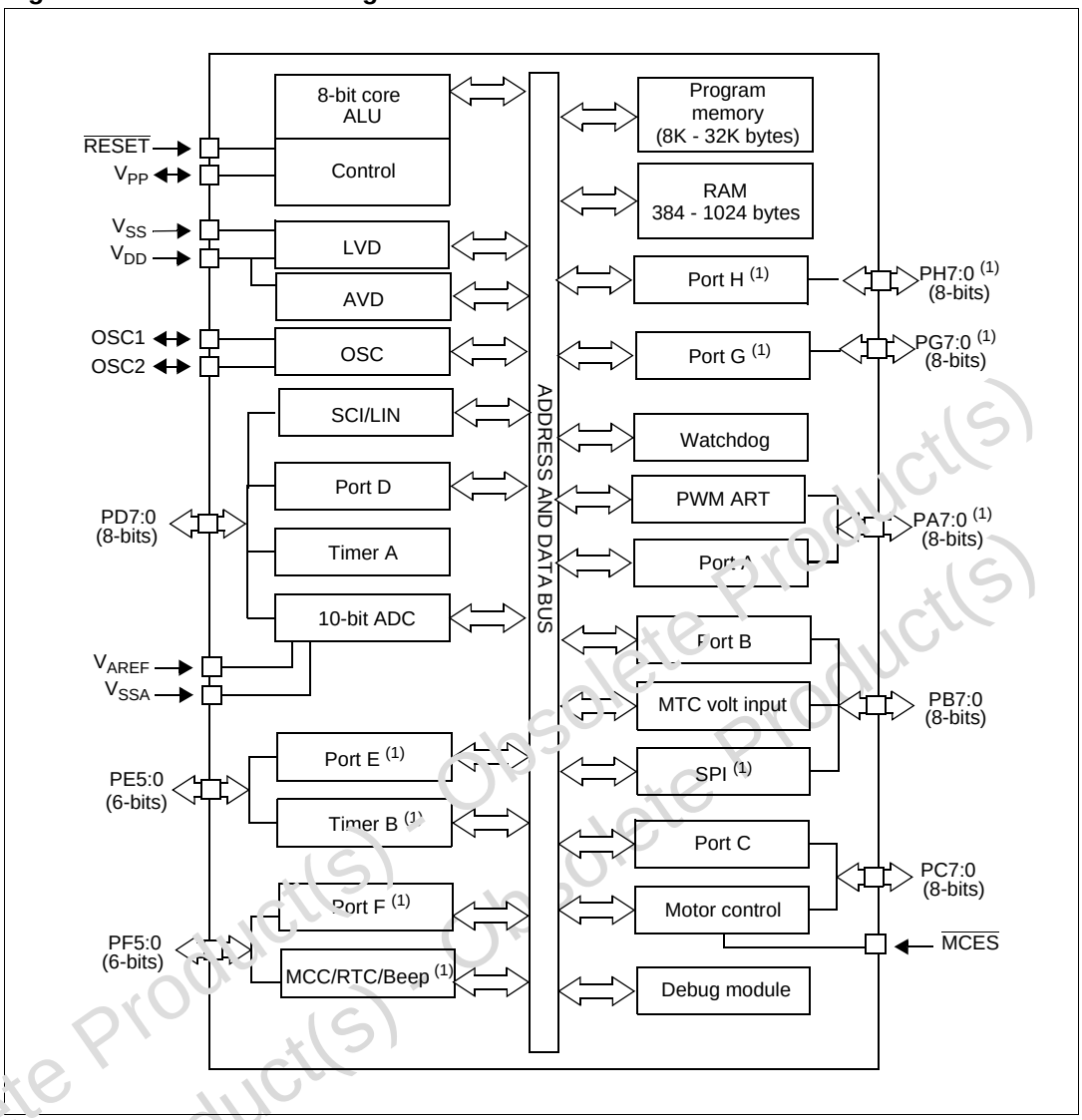
What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	LINbusSCI, SPI
Peripherals	LVD, Motor Control PWM, POR, PWM, WDT
Number of I/O	26
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fmc2s6tc

Figure 1. Device block diagram



1. On some devices only; see *Table 2: Device pin description on page 23*

6.5 System integrity management (SI)

The system integrity management block contains the low voltage detector (LVD), auxiliary voltage detector (AVD) and clock security system (CSS) functions. It is managed by the SICSR register.

Note: A reset can also be triggered following the detection of an illegal opcode or prebyte code. Refer to Section 11.2.2 on page 309 for further details.

6.5.1 Low voltage detector (LVD)

The low voltage detector function (LVD) generates a static reset when the V_{DD} supply voltage is below a V_{IT-} reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The V_{IT-} reference value for a voltage drop is lower than the V_{IT+} reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

The LVD reset circuitry generates a reset when V_{DD} is below:

- V_{IT+} when V_{DD} is rising
- V_{IT-} when V_{DD} is falling

The LVD function is illustrated in *Figure 13*.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is above V_{IT-}, the MCU can only be in two modes:

- under full software control
- in static safe reset

In these conditions, secure operation is always ensured for the application without the need for external reset hardware.

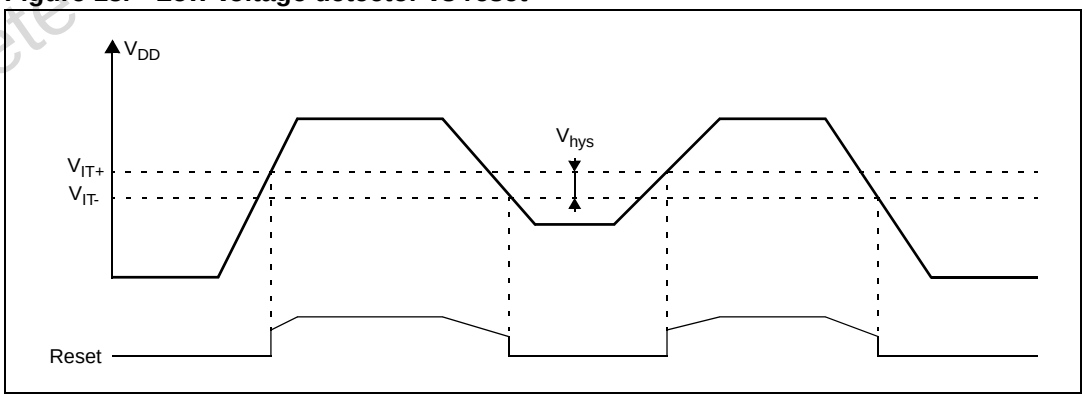
During a low voltage detector reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

Note: 1 The LVD allows the device to be used without any external reset circuitry.

2 The LVD is an optional function which can be selected by option byte.

3 It is recommended to make sure that the V_{DD} supply voltage rises monotonously when the device is exiting from reset, to ensure the application functions properly.

Figure 13. Low voltage detector vs reset



6.5.2 Auxiliary voltage detector (AVD)

The voltage detector function (AVD) is based on an analog comparison between a $V_{IT-(AVD)}$ and $V_{IT+(AVD)}$ reference value and the V_{DD} main supply. The V_{IT-} reference value for falling voltage is lower than the V_{IT+} reference value for rising voltage in order to avoid parasitic detection (hysteresis).

The output of the AVD comparator is directly readable by the application software through a real time status bit (AVDF) in the SICSR register. This bit is read only.

Caution: The AVD function is active only if the LVD is enabled through the option byte (see *Section 14.1 on page 356*).

Monitoring the V_{DD} main supply

If the AVD interrupt is enabled, an interrupt is generated when the voltage crosses the $V_{IT+(AVD)}$ or $V_{IT-(AVD)}$ threshold (AVDF bit toggles).

In the case of a drop in voltage, the AVD interrupt acts as an early warning, allowing software to shut down safely before the LVD resets the microcontroller. See *Figure 14*.

The interrupt on the rising edge is used to inform the application that the V_{DD} warning state is over.

If the voltage rise time t_{rv} is less than 256 or 4096 CPU cycles (depending on the reset delay selected by option byte), no AVD interrupt is generated when $V_{IT+(AVD)}$ is reached.

If t_{rv} is greater than 256 or 4096 cycles then:

- If the AVD interrupt is enabled before the $V_{IT+(AVD)}$ threshold is reached, then two AVD interrupts are received: the first when the AVDIE bit is set, and the second when the threshold is reached.
- If the AVD interrupt is enabled after the $V_{IT+(AVD)}$ threshold is reached then only one AVD interrupt occurs.

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.

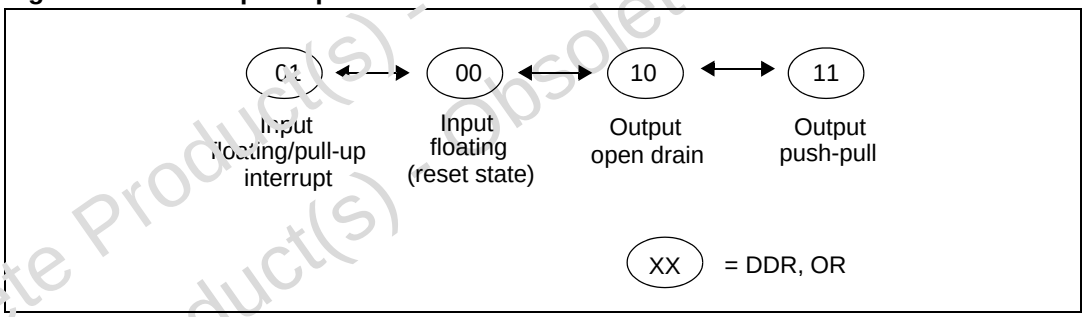
Warning: The analog input voltage level must be within the limits stated in the absolute maximum ratings.

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 29*. Other transitions are potentially risky and should be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 29. Interrupt I/O port state transitions



9.4 Low power modes

Table 27. Effect of low power modes on I/O ports

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.

Table 56. SPICR register description (continued)

Bit	Name	Function
4	MSTR	Master mode This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$ = 0 (see *Master mode fault (MODF) on page 133*). 0: Slave mode 1: Master mode. The function of the SCK pin changes from an input to an output and the functions of the MISO and MOSI pins are reversed.
3	CPOL	Clock polarity This bit is set and cleared by software. This bit determines the idle state of the serial clock. The CPOL bit affects both the master and slave modes. 0: SCK pin has a low level idle state 1: SCK pin has a high level idle state *Note: If CPOL is changed at the communication byte boundaries, the SPI must be disabled by resetting the SPE bit.*
2	CPHA	Clock phase This bit is set and cleared by software. 0: The first clock transition is the first data capture edge 1: The second clock transition is the first capture edge *Note: The slave must have the same CPOL and CPHA settings as the master.*
1:0	SPR[1:0]	Serial clock frequency These bits are set and cleared by software. Used with the SPR2 bit, they select the baud rate of the SPI serial clock SCK output by the SPI in master mode. *Note: These 2 bits have no effect in slave mode.*

SPI control/status register (SPICSR)

SPICSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SPIF	WCOL	OVR	MODF	Reserved	SOD	SSM	SSI
RO	RO	RO	RO	-	R/W	R/W	R/W

Table 57. SPICSR register description

Bit	Name	Function
7	SPIF	Serial peripheral data transfer flag 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.*
6	WCOL	Write collision status 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 59*). 0: No write collision occurred 1: A write collision has been detected

Figure 61. SCI block diagram (in conventional baud rate generator mode)

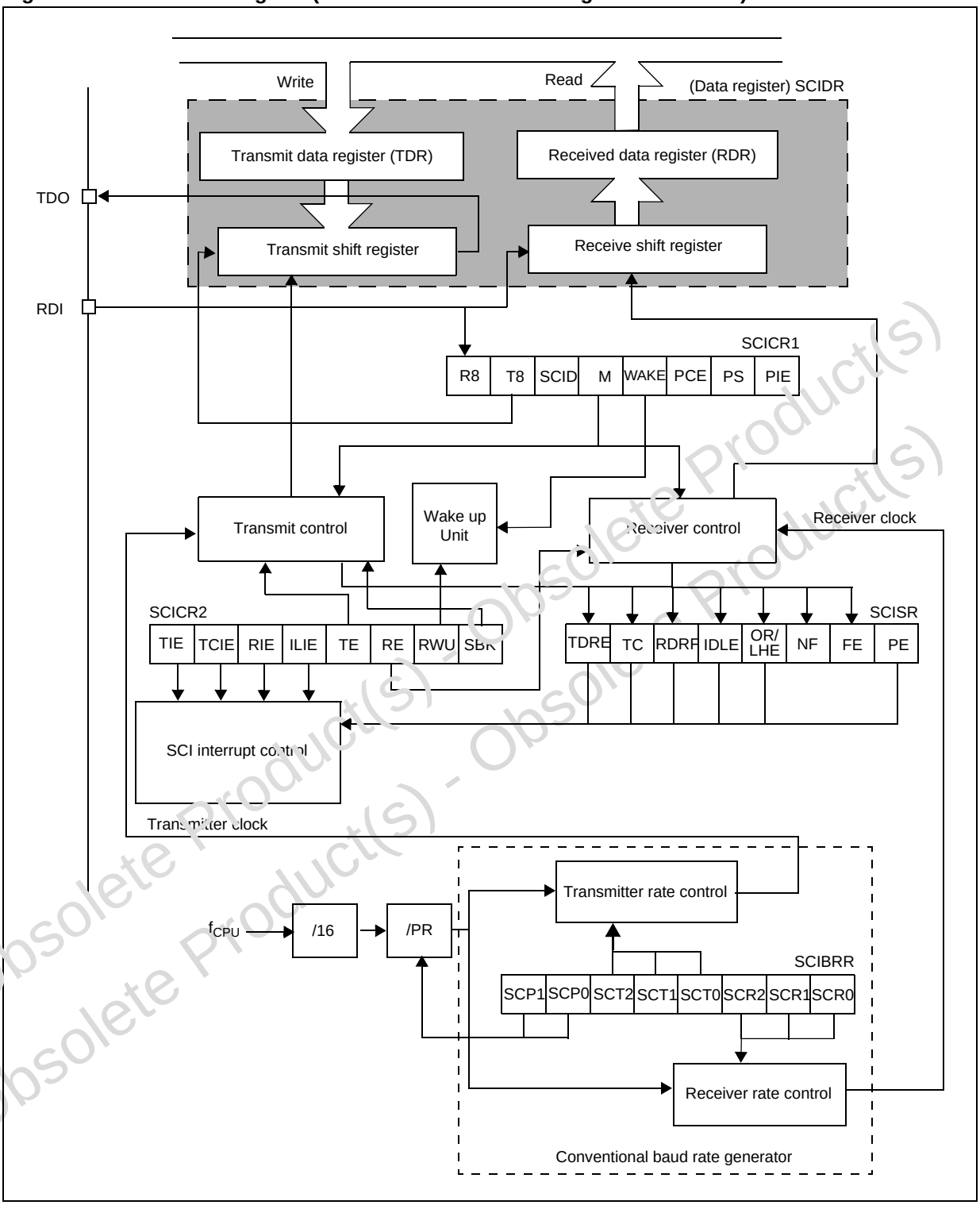
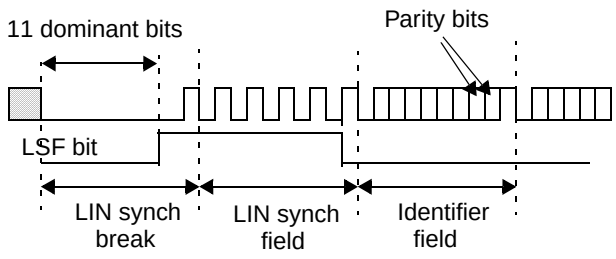


Table 71. SCICR3 register description (continued)

Bit	Name	Function
3	LHDM	LIN header detection method This bit is set and cleared by software. It is only usable in LIN slave mode. It enables the header detection method. In addition if the RWU bit in the SCICR2 register is set, the LHDM bit selects the wake-up method (replacing the WAKE bit). 0: LIN synch break detection method 1: LIN identifier field detection method
2	LHIE	LIN header interrupt enable This bit is set and cleared by software. It is only usable in LIN slave mode. 0: LIN header interrupt is inhibited 1: An SCI interrupt is generated whenever LHDF = 1
1	LHDF	LIN header detection flag This bit is set by hardware when a LIN header is detected and cleared by a software sequence (an access to the SCISR register followed by a read of the SCICR3 register). It is only usable in LIN slave mode. 0: No LIN header detected 1: LIN header detected *Note: The header detection method depends on the LHDM bit:* *- If LHDM = 0, a header is detected as a LIN synch break* *- If LHDM = 1, a header is detected as a LIN Identifier, meaning that a LIN synch break field + a LIN synch field + a LIN identifier field have been consecutively received.*
0	LSF	LIN synch field state This bit indicates that the LIN synch field is being analyzed. It is only used in LIN slave mode. In auto synchronization mode (LASE bit = 1), when the SCI is in the LIN synch field state it waits or counts the falling edges on the RDI line. It is set by hardware as soon as a LIN synch break is detected and cleared by hardware when the LIN synch field analysis is finished (see *Figure 73*). This bit can also be cleared by software to exit LIN Synch state and return to idle mode. 0: The current character is not the LIN synch field 1: LIN synch field state (LIN synch field undergoing analysis)

Figure 73. LSF bit set and clear



The shape of the output voltage (voltage, frequency, sinewave, trapezoid, ...) is completely managed by the application software, in charge of computing the compare values to be loaded for a given PWM duty-cycle (refer to *Figure 77*).

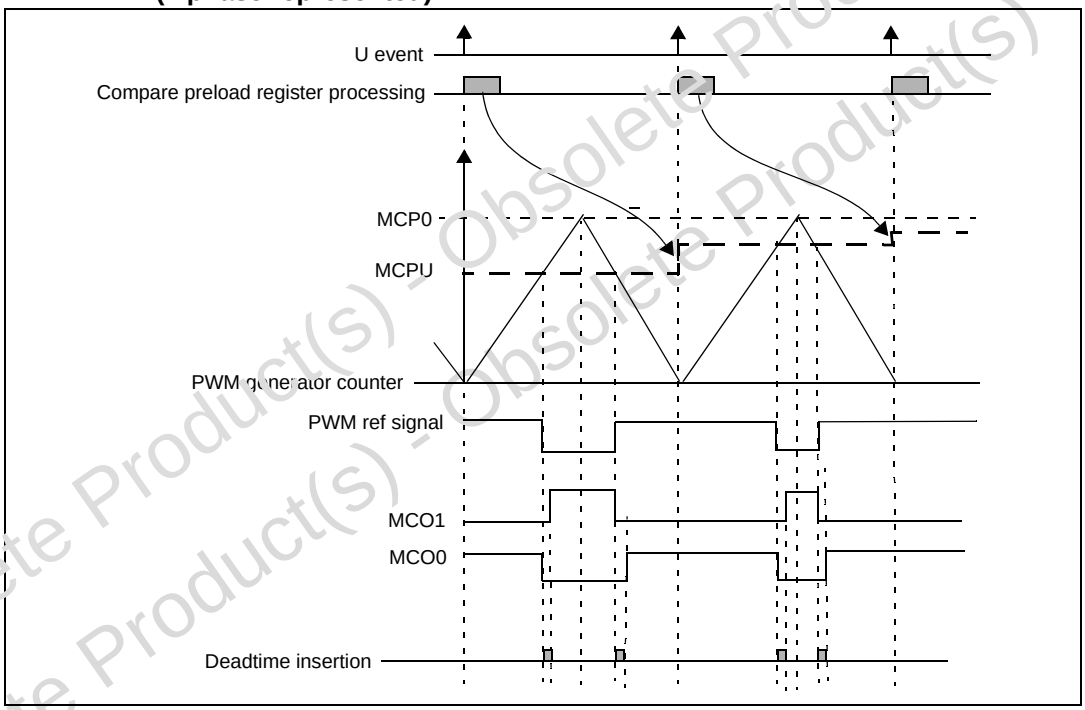
Finally, the PWM modulated voltage generated by the power stage is smoothed by the motor inductance to get sinusoidal currents in the stator windings.

The induction motor being asynchronous, there is no need to synchronize the rotor position to the sinewave generation phase in most of the applications.

Part of the MTC dedicated to delay computation and event sampling can thus be reconfigured to perform speed acquisition of the most common speed sensor, without the need of an additional standard timer.

This speed measurement timer with clear-on-capture and clock prescaler auto-setting allows to keep the CPU load to a minimum level while taking benefit of the embedded input comparator and edge detector.

Figure 76. Complementary PWM generation for three-phase induction motor (1 phase represented)



Specific applications can require sampling for the Z event detection only during the ON time of the PWM signal. This can happen when the PWM signal is applied only on the low side switches for Z event detection. In this case, during the OFF time of the PWM signal, the phase voltage is tied to the application voltage V and no back-EMF signal can be seen. During the ON time of the PWM signal, the phase voltage can be compared to the neutral point voltage and the Z event can be detected. Therefore, it is possible to add a programmable delay before sampling (which is normally done when the PWM signal is switched ON) to perform the sampling during the ON time of the PWM signal. This delay is set with the DS [3:0] bits in the MCONF register.

Table 91. Delay length before sampling(1)

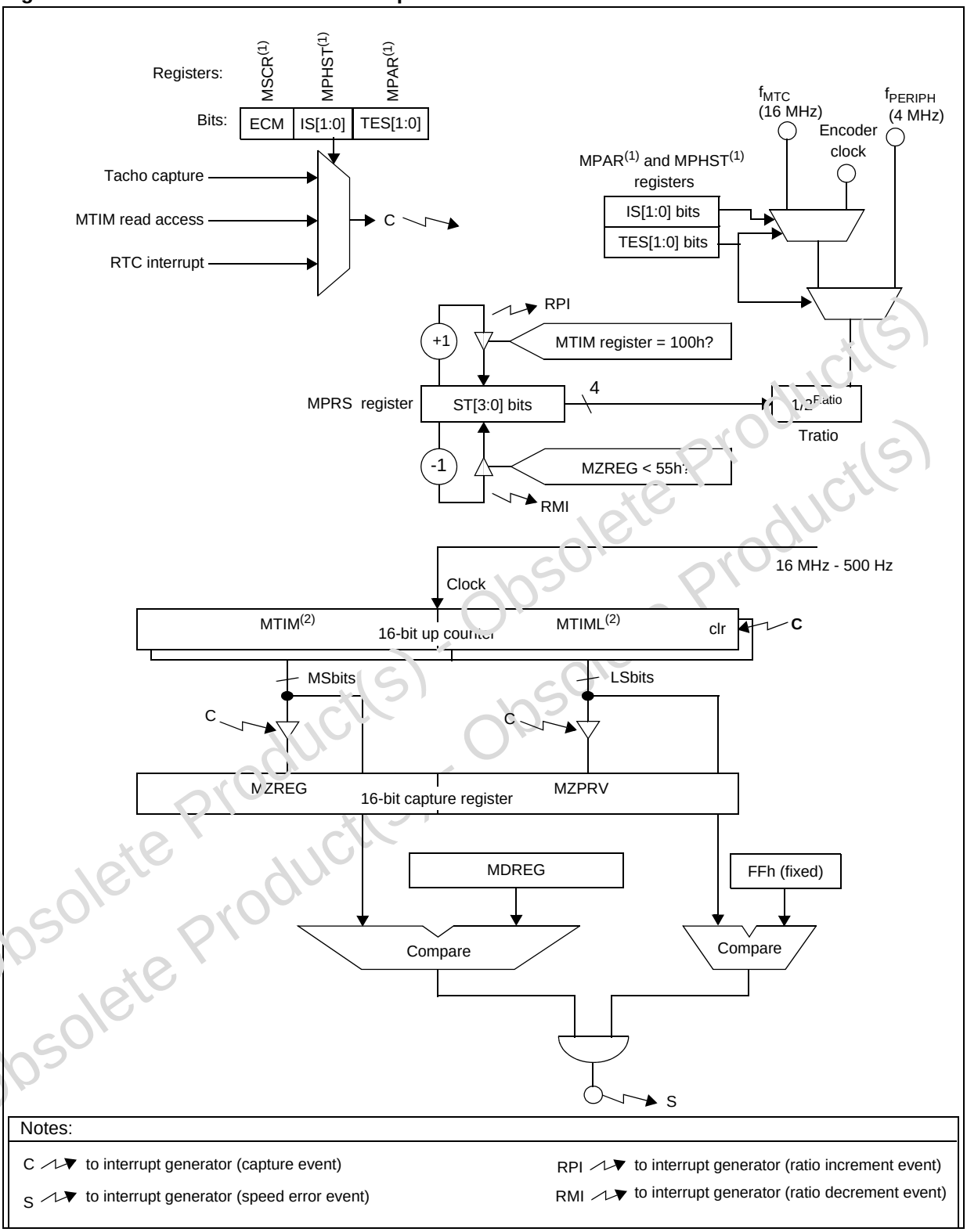
DS3	DS2	DS1	DS0	Delay added to sample at T_{ON}
0	0	0	0	No delay added. Sample during T_{OFF}
0	0	0	1	2.5 µs
0	0	1	0	5 µs
0	0	1	1	7.5 µs
0	1	0	0	10 µs
0	1	0	1	12.5 µs
0	1	1	0	15 µs
0	1	1	1	17.5 µs
1	0	0	0	20 µs
1	0	0	1	22.5 µs
1	0	1	0	25 µs
1	0	1	1	27.5 µs
1	1	0	0	30 µs
1	1	0	1	32.5 µs
1	1	1	0	35 µs
1	1	1	1	37.5 µs

1. Times are indicated for 4 MHz f_{PERIPH}.

As soon as a delay is set in the DS[3:0] bits, the minimum OFF time for the PWM signal is no longer required and it is automatically set to 0µs in current mode in the internal sampling clock and a true 100% duty cycle can be set in the 12-bit PWM generator compare U register if needed.

Depending on the frequency and the duty cycle of the PWM signal, the delay inserted before sampling could cause it sample the signal OFF time instead of the ON time. In this case an interrupt can be generated and the sample is not taken into account. When a sample occurs outside the PWM signal ON time, the SOI bit in the MCONF register is set and an interrupt request is generated if the SOM bit is set in the MCONF register. This interrupt is enabled only if a delay value has been set in the DS[3:0] bits. In this case, the sampling is done at the PWM frequency but only during the ON time of the PWM signal. *Figure 85* and *Figure 86* show in detail the generation of the sampling order when the delay is added.

Figure 103. Overview of MTIM timer in speed measurement mode



Notes:

C ↝ to interrupt generator (capture event)

S ↝ to interrupt generator (speed error event)

RPI ↝ to interrupt generator (ratio increment event)

RMI ↝ to interrupt generator (ratio decrement event)

1. Register set-up described in *Speed sensor mode on page 208*.
2. Register updated on R event.

Warning: **As the MCRB register contains preload bits ~~with~~, it has to be written as a complete byte. A bit set or bit reset instruction on a non-preload bit resets ~~has the effect of resetting~~ all the preload bits.**

Control register C (MCRC)

MCRB Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SEI/OI	EDIR/HZ	SZ	SC	SPLG	VR[2:0]		
R/W	RO	R/W	R/W	R/W	R/W		

Table 136. MCRC register description

Bit	Name	Function
7	SEI/OI	Speed error interrupt flag/MTIM overflow flag **Position sensor or sensorless mode** (TES[1:0] bits = 00): OI: MTIM overflow flag This flag signals an overflow of the MTIM timer. It has to be cleared by software. 0: No MTIM timer overflow 1: MTIM timer overflow *Note: No interrupt is associated with this flag.* **Speed sensor mode** (TES[1:0] bits = 01, 10, 11): SEI: Speed error interrupt flag 0: No tacho error interrupt pending 1: Tacho error interrupt pending
6	EDIR/HZ	Encoder Direction bit/ Hardware zero-crossing event bit **Position sensor or sensorless mode** (TES[1:0] bits = 00)**:** HZ**:** Hardware zero-crossing event bit This read/write bit selects if the Z event is hardware or not. 0: No hardware zero-crossing event 1: Hardware zero-crossing event **Speed sensor mode** (TES[1:0] bits = 01, 10, 11)**:** EDIR: Encoder direction bit This bit is read-only. As the rotation direction depends on encoder outputs and motor phase connections, this bit cannot indicate absolute direction. It therefore gives the relative phase-shift (that is, advance/delay) between the two signals in quadrature output by the encoder (see *Figure 91*). 0: MCIA input delayed compared to MCIB input 1: MCIA input in advance compared to MCIB input
5	SZ	Simulated zero-crossing event bit 0: No simulated zero-crossing event 1: Simulated zero-crossing event

Warning: **Access to preload registers: Special care has to be taken with preload registers, especially when using the ST7 BSET and BRES instructions on MTC registers.**
For instance, while writing to the MPHST register, the value in the preload register is written. However, while reading at the same address, the current value in the register and not the value of the preload register is obtained.
Excepted for three-phase PWM generator's registers, all preload registers are loaded in the active registers at the same time. In normal mode this is done automatically when a C event occurs, however in direct access mode (DAC bit = 1) the preload registers are loaded as soon as a value is written in the MPHST register.

Caution: **Access to write-once bits:** Special care has to be taken with write-once bits in MPOL and MDTG registers; these bits have to be first accessed during the set-up. Any access to the other bits (not write-once) through a BRES or a BSET instruction locks the content of write-once bits (no possibility for the core to distinguish individual bit access: Read/write internal signal acts on a whole register only). This protection is then only unlocked after a processor hardware reset.

Deadtime generator register (MDTG)

MDTG — Reset value: 1111 1111 (FFh)

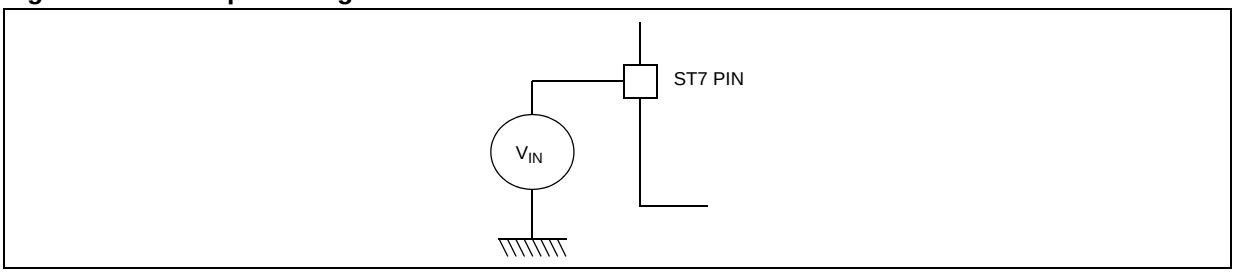
7	6	5	4	3	2	1	0
PCN	DTE	DTG[5:0]					
R/W	(1)	Write once only					

1. Write once-only bit if PCN bit is set, read/write if PCN bit is reset.

Table 185. Instruction set overview (continued)

Mnemo	Description	Function/example	Dst	Src	I1	H	I0	N	Z	C
JRUGT	Jump if (C + Z = 0)	Unsigned >								
JRULE	Jump if (C + Z = 1)	Unsigned <=								
LD	Load	dst <= src	reg, M	M, reg				N	Z	
MUL	Multiply	X,A = X * A	A, X, Y	X, Y, A		0				0
NEG	Negate (2's compl)	neg $10	reg, M					N	Z	C
NOP	No operation									
OR	OR operation	A = A + M	A	M				N	Z	
Pop	Pop from the stack	pop reg	reg	M						
		pop CC	CC	M	I1	H	I0	N	Z	C
Push	Push onto the stack	Push Y	M	reg, CC						
RCF	Reset carry flag	C = 0								0
RET	Subroutine return									
RIM	Enable interrupts	I1:0 = 10 (level 0)			1		0			
RLC	Rotate left true C	C <= A <= C	reg, M					N	Z	C
RRC	Rotate right true C	C => A => C	reg, M					N	Z	C
RSP	Reset stack pointer	S = Max allowed								
SBC	Substract with carry	A = A - M - C	A	M				N	Z	C
SCF	Set carry flag	C = 1								1
SIM	Disable interrupts	I1:0 = 11 (level 3)			1		1			
SLA	Shift left arithmetic	C <= A <= 0	reg, M					N	Z	C
SLL	Shift left logic	C <= A <= 0	reg, M					N	Z	C
SRL	Shift right logic	0 => A => C	reg, M					0	Z	C
SRA	Shift right arithmetic	A7 => A => C	reg, M					N	Z	C
SUB	Substraction	A = A - M	A	M				N	Z	C
SWAP	SWAP nibbles	A7-A4 <=> A3-A0	reg, M					N	Z	
TNZ	Test for neg and zero	tnz lbl1						N	Z	
TRAP	S/W trap	S/W interrupt			1		1			
WFI	Wait for interrupt				1		0			
XOR	Exclusive OR	A = A XOR M	A	M				N	Z	

Figure 130. Pin input voltage



12.2 Absolute maximum ratings

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

12.2.1 Voltage characteristics

Table 186. Voltage characteristics

Symbol	Ratings	Maximum value	Unit
V_{DD} - V_{SS}	Supply voltage	6.5	V
V_{PP} - V_{SS}	Programming voltage	13	
V_{IN}	Input voltage on any pin(1)(2)	V_{SS}-0.3 to V_{DD}+0.3	
$\|\Delta V_{DDx}\|$ and $\|\Delta V_{SSx}\|$	Variations between different digital power pins	50	mV
$\|V_{SSA}$ - $V_{SSx}\|$	Variations between digital and analog ground pins	50	
$V_{ESD(HBM)}$	Electro-static discharge voltage (Human body model)	See *Section 12.7.3: Absolute maximum ratings (electrical sensitivity) on page 327*	
$V_{ESD(MM)}$	Electro-static discharge voltage (Machine model)		

1. Directly connecting the $\overline{RESET}$ and I/O pins to V_{DD} or V_{SS} could damage the device if an unintentional internal reset is generated or an unexpected change of the I/O configuration occurs (for example, due to a corrupted program counter). To guarantee safe operation, this connection has to be done through a pull-up or pull-down resistor (typical: 4.7kΩ for $\overline{RESET}$, 10kΩ for I/Os). For the same reason, unused I/O pins must not be directly tied to V_{DD} or V_{SS}.
2. $I_{INJ(PIN)}$ must never be exceeded. This is implicitly insured if V_{IN} maximum is respected. If V_{IN} maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by $V_{IN} > V_{DD}$ while a negative injection is induced by $V_{IN} < V_{SS}$.

12.12 Motor control characteristics

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

12.12.1 Internal reference voltage

Table 216. Internal reference voltage

Symbol	Parameter	Conditions	Min	Typ(1)	Max	Unit
V_{REF}	Voltage threshold (VR [2:0] = 000)	VR [2:0] = 000		V_{DD}*0.04		V
		Example: V_{DD} - V_{SSA} = 5V		0.2		
	Voltage threshold (VR [2:0] = 001)	VR [2:0] = 001		V_{DD}*0.12		
		Example: V_{DD} - V_{SSA} = 5V		0.6		
	Voltage threshold (VR [2:0] = 010)	VR [2:0] = 010		V_{DD}*0.2		
		Example: V_{DD} - V_{SSA} = 5V		1.0		
	Voltage threshold (VR [2:0] = 011)	VR [2:0] = 011		V_{DD}*0.3		
		Example: V_{DD} - V_{SSA} = 5V		1.5		
	Voltage threshold (VR [2:0] = 100)	VR [2:0] = 100		V_{DD}*0.4		
		Example: V_{DD} - V_{SSA} = 5V		2.0		
	Voltage threshold (VR [2:0] = 101)	VR [2:0] = 101		V_{DD}*0.5		
		Example: V_{DD} - V_{SSA} = 5V		2.5		
	Voltage threshold (VR [2:0] = 110)	VR [2:0] = 110		V_{DD}*0.7		
		Example: V_{DD} - V_{SSA} = 5V		3.5		
$\Delta V_{REF}/V_{REF}$	Tolerance on V_{REF}			2.5	10	%

1. Unless otherwise specified, typical data are based on T_A = 25°C and V_{DD} - V_{SS} = 5V. They are given only as design guidelines and are not tested.

13.2.2 LQFP32 package

Figure 164. 32-pin low profile quad flat packag outline

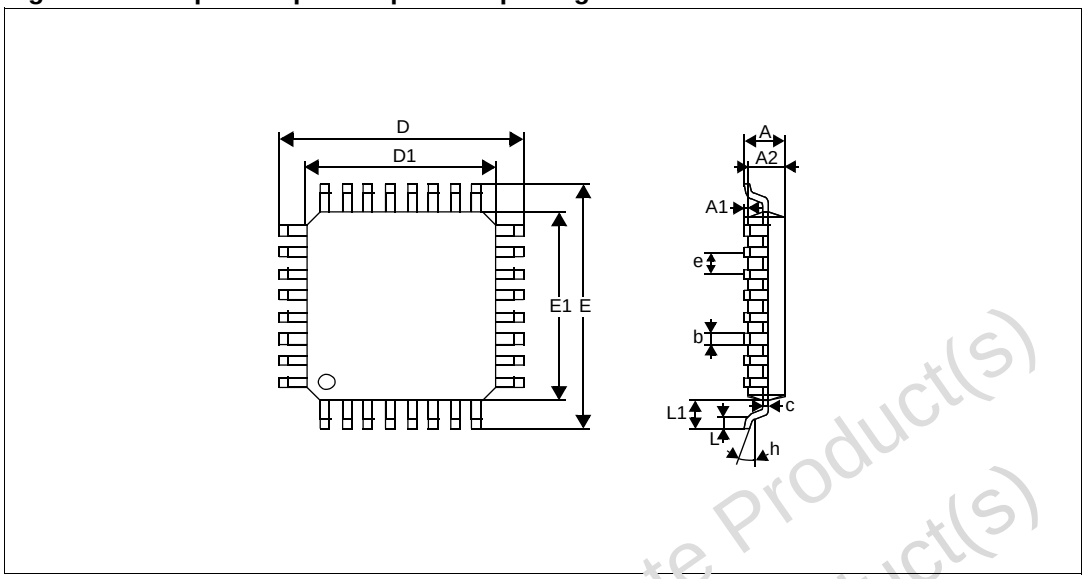


Table 223. 32-pin low profile quad flat package mechanical data

Dimension	mm			inches[1]		
	Minimum	Typical	Maximum	Minimum	Typical	Maximum
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.30	0.37	0.45	0.012	0.015	0.018
c	0.09		0.20	0.004		0.008
D		9.00			0.354	
D1		7.00			0.276	
E		9.00			0.354	
E1		7.00			0.276	
e		0.80			0.031	
θ	0°	3.5°	7°	0°	3.5°	7°
L	0.45	0.60	0.75	0.018	0.024	0.030
L1		1.00			0.039	
	Number of pins					
N	32					

1. Values in inches are converted from mm and rounded to 3 decimal digits.

Occurrence

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

Analysis

The LIN protocol specifies a minimum of 13 bits for the break duration, but there is no maximum value. Nevertheless, the maximum length of the header is specified as (14 + 10 + 10 + 1) x 1.4 = 49 bits. This is composed of:

- The synch break field (14 bits).
- The synch field (10 bits).
- the identifier field (10 bits).

Every LIN frame starts with a break character. Adding an idle character increases the length of each header by 10 bits. When the problem occurs, the header length is increased by 11 bits and becomes ((14 + 11) + 10 + 10 + 1) = 45 bits.

To conclude, the problem is not always critical for LIN communication if the software keeps the time between the sync field and the ID smaller than 4 bits, that is, 208µs at 19200 baud.

Workaround

The workaround is the same as for SCI mode but considering the low probability of occurrence (1%), it may be preferable to keep the break generation sequence as it is.

15.4.2 Header time-out does not prevent wake-up from mute mode

Normally, when LINSCI is configured in LIN slave mode, if a header time-out occurs during a LIN header reception (that is, header length > 57 bits), the LIN Header Error bit (LHE) is set, an interrupt occurs to inform the application but the LINSCI should stay in mute mode, waiting for the next header reception.

Problem description

The LINSCI sampling period is Tbit/16. If a LIN Header time-out occurs between the 9th and the 15th sample of the Identifier Field Stop Bit (refer to *Figure 165*), the LINSCI wakes up from mute mode. Nevertheless, LHE is set and LIN header detection flag (LHDF) is kept cleared.

In addition, if LHE is reset by software before this 15th sample (by accessing the SCISR register and reading the SCIDR register in the LINSCI interrupt routine), the LINSCI generates another LINSCI interrupt (due to the RDRF flag setting).

semaphore. If it is '1' this means that the last interrupt was missed and the interrupt routine is invoked with the call instruction.

To implement the workaround, the following software sequence is to be followed for writing into the PxOR/PxDDR registers. The example is for Port PF1 with falling edge interrupt sensitivity. The software sequence is given for both cases (global interrupt disabled/enabled).

Case 1: Writing to PxOR or PxDDR with global interrupts enabled:

```
LD A,#01
LD sema,A; set the semaphore to '1'
LD A,PFDR
AND A,#02
LD  X,A; store the level before writing to PxOR/PxDDR
LD A,#$90
LD PFDDR,A; Write to PFDDR
LD A,#$ff
LD  PFOR,A ; Write to PFOR
LD A,PFDR
AND A,#02
LD Y,A; store the level after writing to PxOR/PxDDR
LD A,X; check for falling edge
cp A,#02
jrne OUT
TNZ Y
jrne OUT
LD A,sema; check the semaphore status if edge is detected
CP A,#01
jrne OUT
call call_routine; call the interrupt routine
OUT:LD A,#00
LD sema,A
.call_routine; entry to call_routine
Push A
Push X
Push CC
.ext1_rt; entry to interrupt routine
LD A,#00
LD sema,A
IRET
```

Case 2: Writing to PxOR or PxDDR with global interrupts disabled:

```
SIM; set the interrupt mask
LD A,PFDR
AND A,#$02
LD  X,A; store the level before writing to PxOR/PxDDR
LD A,#$90
LD PFDDR,A; Write into PFDDR
LD A,#$ff
LD PFOR,A          ; Write to PFOR
LD A,PFDR
AND A,#$02
LD Y,A; store the level after writing to PxOR/PxDDR
```