



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	I ² C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	32
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f321j9t6

Table of Contents

10.5.3	General Description	88
10.5.4	Clock Phase and Clock Polarity	92
10.5.5	Error Flags	93
10.5.6	Low Power Modes	95
10.5.7	Interrupts	95
10.5.8	Register Description	96
10.6	SERIAL COMMUNICATIONS INTERFACE (SCI)	99
10.6.1	Introduction	99
10.6.2	Main Features	99
10.6.3	General Description	99
10.6.4	Functional Description	101
10.6.5	Low Power Modes	108
10.6.6	Interrupts	108
10.6.7	Register Description	109
10.7	I2C BUS INTERFACE (I2C)	115
10.7.1	Introduction	115
10.7.2	Main Features	115
10.7.3	General Description	115
10.7.4	Functional Description	117
10.7.5	Low Power Modes	121
10.7.6	Interrupts	121
10.7.7	Register Description	122
10.8	10-BIT A/D CONVERTER (ADC)	128
10.8.1	Introduction	128
10.8.2	Main Features	128
10.8.3	Functional Description	129
10.8.4	Low Power Modes	129
10.8.5	Interrupts	129
10.8.6	Register Description	130
11	INSTRUCTION SET	132
11.1	CPU ADDRESSING MODES	132
11.1.1	Inherent	133
11.1.2	Immediate	133
11.1.3	Direct	133
11.1.4	Indexed (No Offset, Short, Long)	133
11.1.5	Indirect (Short, Long)	133
11.1.6	Indirect Indexed (Short, Long)	134
11.1.7	Relative mode (Direct, Indirect)	134
11.2	INSTRUCTION GROUPS	135
12	ELECTRICAL CHARACTERISTICS	138
12.1	PARAMETER CONDITIONS	138
12.1.1	Minimum and Maximum values	138
12.1.2	Typical values	138
12.1.3	Typical curves	138
12.1.4	Loading capacitor	138
12.1.5	Pin input voltage	138
12.2	ABSOLUTE MAXIMUM RATINGS	139

Address	Block	Register Label	Register Name	Reset Status	Remarks
002Ah	WATCHDOG	WDGCR	Watchdog Control Register	7Fh	R/W
002Bh		SICSR	System Integrity Control/Status Register	000x 000x b	R/W
002Ch 002Dh	MCC	MCCSR MCCBCR	Main Clock Control / Status Register Main Clock Controller: Beep Control Register	00h 00h	R/W R/W
002Eh to 0030h	Reserved Area (3 Bytes)				
0031h 0032h 0033h 0034h 0035h 0036h 0037h 0038h 0039h 003Ah 003Bh 003Ch 003Dh 003Eh 003Fh	TIMER A	TACR2 TACR1 TACSR TAIC1HR TAIC1LR TAOC1HR TAOC1LR TACHR TACLR TAACHR TAACLR TAIC2HR TAIC2LR TAOC2HR TAOC2LR	Timer A Control Register 2 Timer A Control Register 1 Timer A Control/Status Register Timer A Input Capture 1 High Register Timer A Input Capture 1 Low Register Timer A Output Compare 1 High Register Timer A Output Compare 1 Low Register Timer A Counter High Register Timer A Counter Low Register Timer A Alternate Counter High Register Timer A Alternate Counter Low Register Timer A Input Capture 2 High Register Timer A Input Capture 2 Low Register Timer A Output Compare 2 High Register Timer A Output Compare 2 Low Register	00h 00h xxxx x0xx b xxh xxh 80h 00h FFh FCh FFh FCh xxh xxh 80h 00h	R/W R/W R/W Read Only Read Only R/W R/W Read Only Read Only Read Only Read Only Read Only Read Only R/W R/W
0040h	Reserved Area (1 Byte)				
0041h 0042h 0043h 0044h 0045h 0046h 0047h 0048h 0049h 004Ah 004Bh 004Ch 004Dh 004Eh 004Fh	TIMER B	TBCR2 TBCR1 TBCSR TBIC1HR TBIC1LR TBOC1HR TBOC1LR TBCHR TBCLR TBACHR TBACLR TBIC2HR TBIC2LR TBOC2HR TBOC2LR	Timer B Control Register 2 Timer B Control Register 1 Timer B Control/Status Register Timer B Input Capture 1 High Register Timer B Input Capture 1 Low Register Timer B Output Compare 1 High Register Timer B Output Compare 1 Low Register Timer B Counter High Register Timer B Counter Low Register Timer B Alternate Counter High Register Timer B Alternate Counter Low Register Timer B Input Capture 2 High Register Timer B Input Capture 2 Low Register Timer B Output Compare 2 High Register Timer B Output Compare 2 Low Register	00h 00h xxxx x0xx b xxh xxh 80h 00h FFh FCh FFh FCh xxh xxh 80h 00h	R/W R/W R/W Read Only Read Only R/W R/W Read Only Read Only Read Only Read Only Read Only Read Only R/W R/W
0050h 0051h 0052h 0053h 0054h 0055h 0056h 0057h	SCI	SCISR SCIDR SCIBRR SCICR1 SCICR2 SCIERP SCIETPR	SCI Status Register SCI Data Register SCI Baud Rate Register SCI Control Register 1 SCI Control Register 2 SCI Extended Receive Prescaler Register Reserved area SCI Extended Transmit Prescaler Register	C0h xxh 00h x000 0000b 00h 00h ---	Read Only R/W R/W R/W R/W R/W R/W

CENTRAL PROCESSING UNIT (Cont'd)**Condition Code Register (CC)**

Read/Write

Reset Value: 111x1xxx

7							0
1	1	I1	H	I0	N	Z	C

The 8-bit Condition Code register contains the interrupt masks and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions.

These bits can be individually tested and/or controlled by specific instructions.

Arithmetic Management BitsBit 4 = **H** *Half carry*.

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instructions. It is reset by hardware during the same instructions.

0: No half carry has occurred.

1: A half carry has occurred.

This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Bit 2 = **N** *Negative*.

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It's a copy of the result 7th bit.

0: The result of the last operation is positive or null.

1: The result of the last operation is negative (that is, the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = **Z** *Zero*.

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.

0: The result of the last operation is different from zero.

1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

Bit 0 = **C** *Carry/borrow*.

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.

0: No overflow or underflow has occurred.

1: An overflow or underflow has occurred.

This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the "bit test and branch", shift and rotate instructions.

Interrupt Management BitsBit 5,3 = **I1, I0** *Interrupt*

The combination of the I1 and I0 bits gives the current interrupt software priority.

Interrupt Software Priority	I1	I0
Level 0 (main)	1	0
Level 1	0	1
Level 2	0	0
Level 3 (= interrupt disable)	1	1

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (IxSPR). They can be also set/cleared by software with the RIM, SIM, IRET, HALT, WFI and PUSH/POP instructions.

See the interrupt management chapter for more details.

SYSTEM INTEGRITY MANAGEMENT (Cont'd)**6.4.4 Register Description****SYSTEM INTEGRITY (SI) CONTROL/STATUS REGISTER (SICSR)**

Read/Write

Reset Value: 000x 000x (00h)

7							0
AVD S	AVD IE	AVD F	LVD RF	0	0	0	WDG RF

Bit 7 = AVDS Voltage Detection selection

This bit is set and cleared by software. Voltage Detection is available only if the LVD is enabled by option byte.

0: Voltage detection on V_{DD} supply

1: Voltage detection on EVD pin

Bit 6 = AVDIE Voltage Detector interrupt enable

This bit is set and cleared by software. It enables an interrupt to be generated when the AVDF flag changes (toggles). The pending interrupt information is automatically cleared when software enters the AVD interrupt routine.

0: AVD interrupt disabled

1: AVD interrupt enabled

Bit 5 = AVDF Voltage Detector flag

This read-only bit is set and cleared by hardware. If the AVDIE bit is set, an interrupt request is generated when the AVDF bit changes value. Refer to Figure 16 and to Section 6.4.2.1 for additional details.

0: V_{DD} or V_{EVD} over $V_{IT+(AVD)}$ threshold

1: V_{DD} or V_{EVD} under $V_{IT-(AVD)}$ threshold

Bit 4 = LVDRF LVD reset flag

This bit indicates that the last Reset was generated by the LVD block. It is set by hardware (LVD re-

set) and cleared by software (writing zero). See WDGRF flag description for more details. When the LVD is disabled by OPTION BYTE, the LVDRF bit value is undefined.

Bits 31 = Reserved, must be kept cleared.

Bit 0 = WDGRF Watchdog reset flag

This bit indicates that the last Reset was generated by the Watchdog peripheral. It is set by hardware (watchdog reset) and cleared by software (writing zero) or an LVD Reset (to ensure a stable cleared state of the WDGRF flag when CPU starts).

Combined with the LVDRF flag information, the flag description is given by the following table.

RESET Sources	LVDRF	WDGRF
External RESET pin	0	0
Watchdog	0	1
LVD	1	X

Application notes

The LVDRF flag is not cleared when another RESET type occurs (external or watchdog), the LVDRF flag remains set to keep trace of the original failure.

In this case, a watchdog reset can be detected by software while an external reset can not.

CAUTION: When the LVD is not activated with the associated option byte, the WDGRF flag can not be used in the application.

INTERRUPTS (Cont'd)

Table 7. Dedicated Interrupt Instruction Set

Instruction	New Description	Function/Example	I1	H	I0	N	Z	C
HALT	Entering Halt mode		1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC	I1	H	I0	N	Z	C
JRM	Jump if I1:0=11 (level 3)	I1:0=11 ?						
JRNM	Jump if I1:0<>11	I1:0<>11 ?						
POP CC	Pop CC from the Stack	Mem => CC	I1	H	I0	N	Z	C
RIM	Enable interrupt (level 0 set)	Load I0 in I1:0 of CC	1		0			
SIM	Disable interrupt (level 3 set)	Load I1 in I1:0 of CC	1		1			
TRAP	Software trap	Software NMI	1		1			
WFI	Wait for interrupt		1		0			

Note: During the execution of an interrupt routine, the HALT, POPCC, RIM, SIM and WFI instructions change the current software priority up to the next IRET instruction or one of the previously mentioned instructions.

POWER SAVING MODES (Cont'd)

8.4 ACTIVE-HALT AND HALT MODES

ACTIVE-HALT and HALT modes are the two lowest power consumption modes of the MCU. They are both entered by executing the 'HALT' instruction. The decision to enter either in ACTIVE-HALT or HALT mode is given by the MCC/RTC interrupt enable flag (OIE bit in MCCR register).

MCCR OIE bit	Power Saving Mode entered when HALT instruction is executed
0	HALT mode
1	ACTIVE-HALT mode

8.4.1 ACTIVE-HALT MODE

ACTIVE-HALT mode is the lowest power consumption mode of the MCU with a real time clock available. It is entered by executing the 'HALT' instruction when the OIE bit of the Main Clock Controller Status register (MCCR) is set (see section 10.2 on page 57 for more details on the MCCR register).

The MCU can exit ACTIVE-HALT mode on reception of an MCC/RTC interrupt or a RESET. When exiting ACTIVE-HALT mode by means of an interrupt, no 256 or 4096 CPU cycle delay occurs. The CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see Figure 27).

When entering ACTIVE-HALT mode, the I[1:0] bits in the CC register are forced to '10b' to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.

In ACTIVE-HALT mode, only the main oscillator and its associated counter (MCC/RTC) are running to keep a wake-up time base. All other peripherals are not clocked except those which get their clock supply from another clock generator (such as external or auxiliary oscillator).

The safeguard against staying locked in ACTIVE-HALT mode is provided by the oscillator interrupt.

Note: As soon as the interrupt capability of one of the oscillators is selected (MCCR.OIE bit set), entering ACTIVE-HALT mode while the Watchdog is active does not generate a RESET.

This means that the device cannot spend more than a defined delay in this power saving mode.

CAUTION: When exiting ACTIVE-HALT mode following an MCC/RTC interrupt, OIE bit of MCCR register must not be cleared before t_{DELAY} after the interrupt occurs ($t_{\text{DELAY}} = 256 \text{ or } 4096 t_{\text{CPU}}$ de-

lay depending on option byte). Otherwise, the ST7 enters HALT mode for the remaining t_{DELAY} period.

Figure 26. ACTIVE-HALT Timing Overview

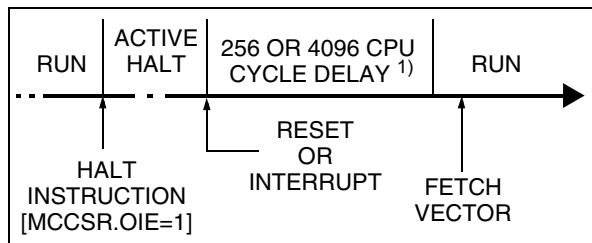
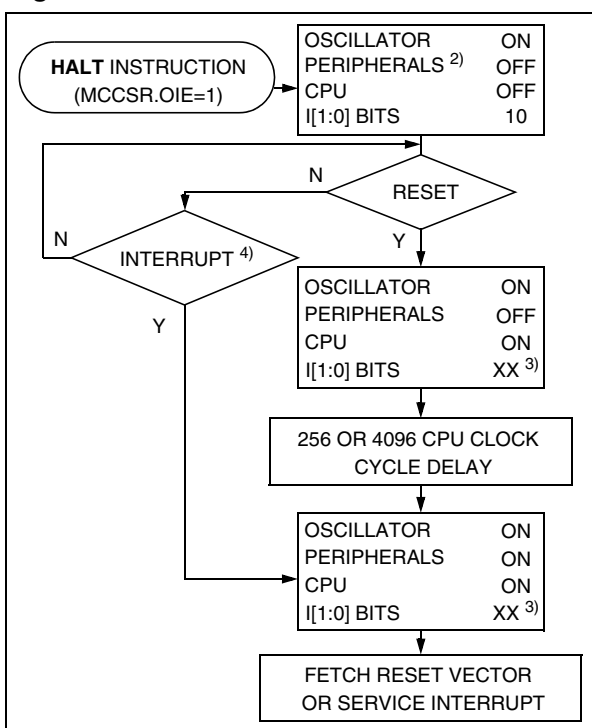


Figure 27. ACTIVE-HALT Mode Flow-chart

**Notes:**

1. This delay occurs only if the MCU exits ACTIVE-HALT mode by means of a RESET.
2. Peripheral clocked with an external clock source can still be active.
3. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and restored when the CC register is popped.
4. Only the MCC/RTC interrupt can exit the MCU from ACTIVE-HALT mode.

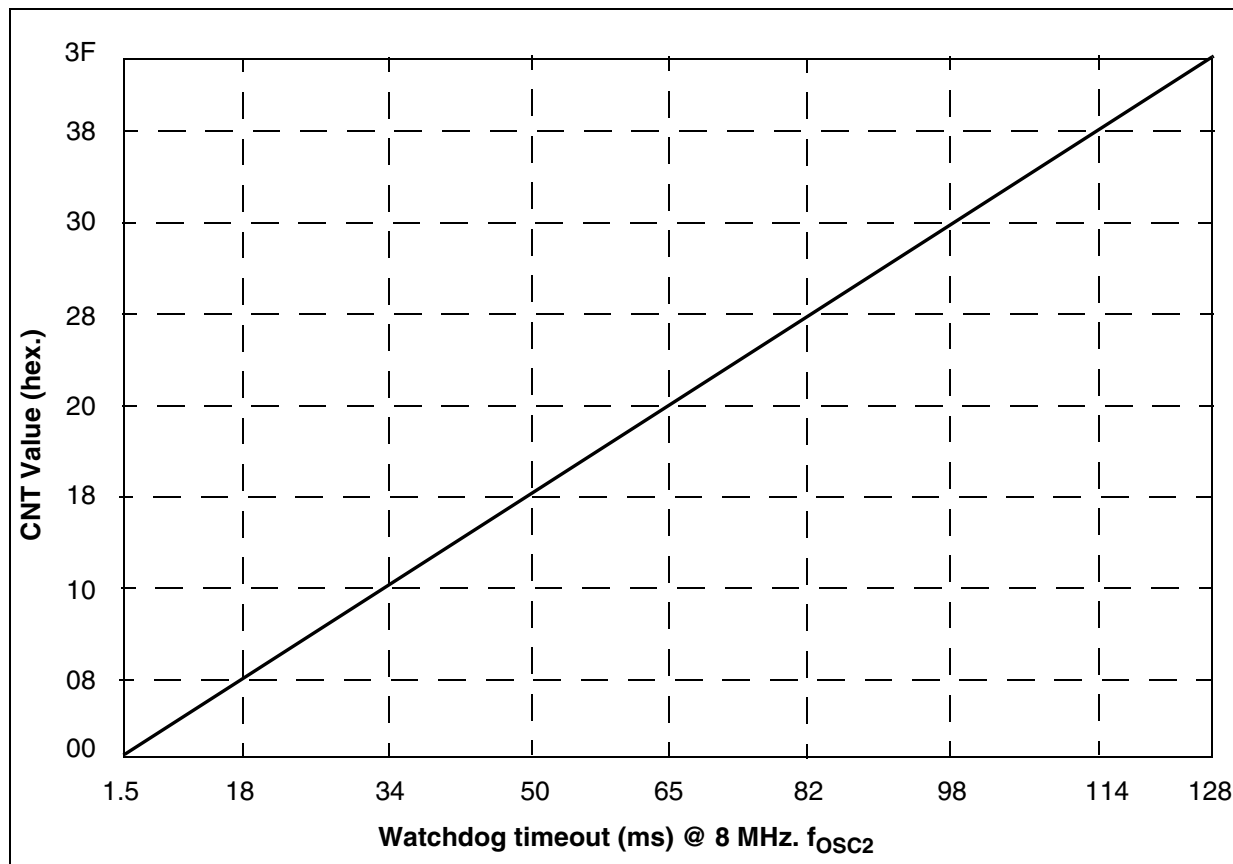
WATCHDOG TIMER (Cont'd)**10.1.4 How to Program the Watchdog Timeout**

Figure 2 shows the linear relationship between the 6-bit value to be loaded in the Watchdog Counter (CNT) and the resulting timeout duration in milliseconds. This can be used for a quick calculation without taking the timing variations into account. If

more precision is needed, use the formulae in Figure 3.

Caution: When writing to the WDGCR register, always write 1 in the T6 bit to avoid generating an immediate reset.

Figure 33. Approximate Timeout Duration



10.3 PWM AUTO-RELOAD TIMER (ART)

10.3.1 Introduction

The Pulse Width Modulated Auto-Reload Timer on-chip peripheral consists of an 8-bit auto reload counter with compare/capture capabilities and of a 7-bit prescaler clock source.

These resources allow five possible operating modes:

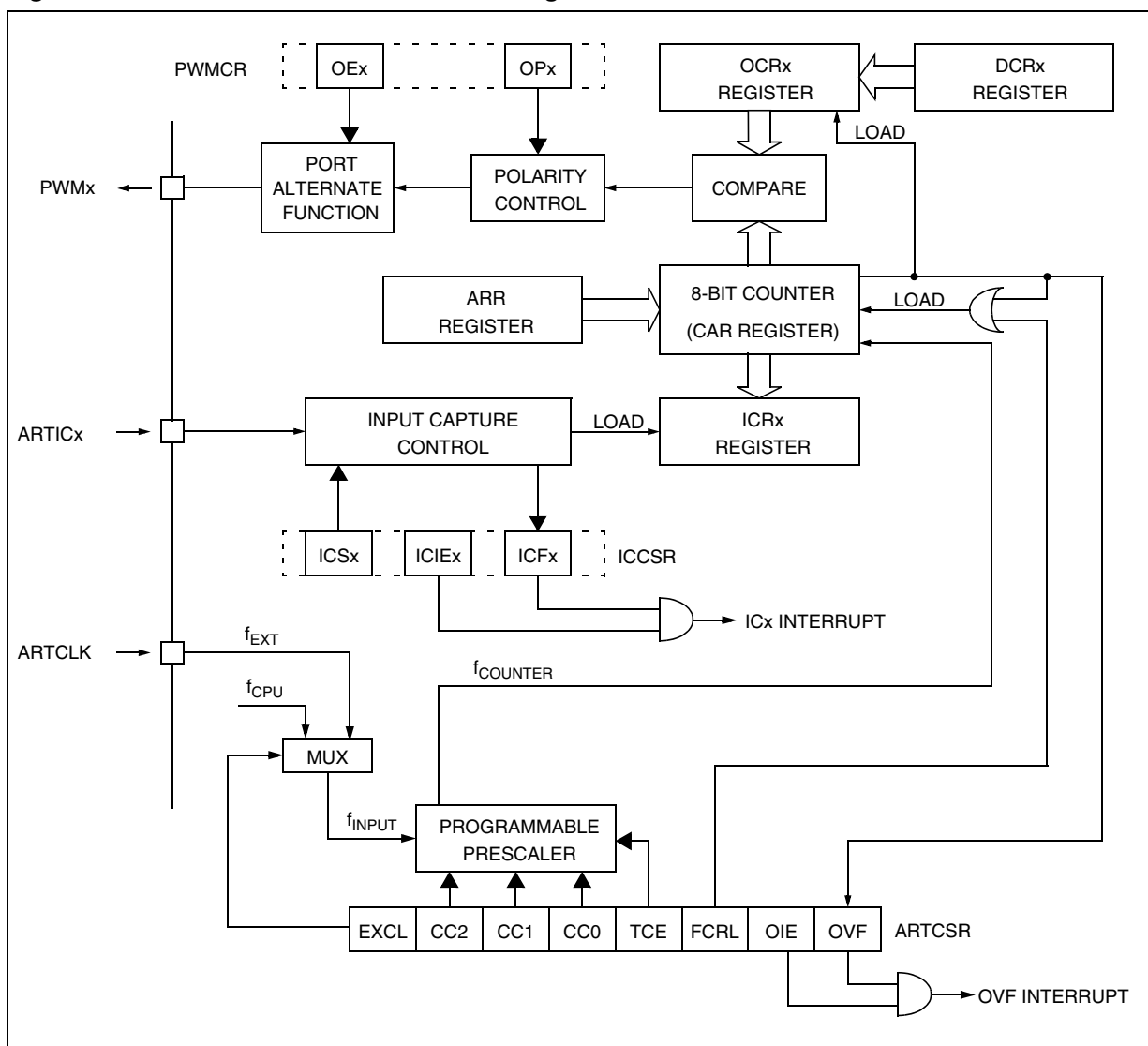
- Generation of up to 4 independent PWM signals
- Output compare and Time base interrupt

- Up to two input capture functions
- External event detector
- Up to two external interrupt sources

The three first modes can be used together with a single counter frequency.

The timer can be used to wake up the MCU from WAIT and HALT modes.

Figure 36. PWM Auto-Reload Timer Block Diagram



16-BIT TIMER (Cont'd)

Figure 46. Input Capture Block Diagram

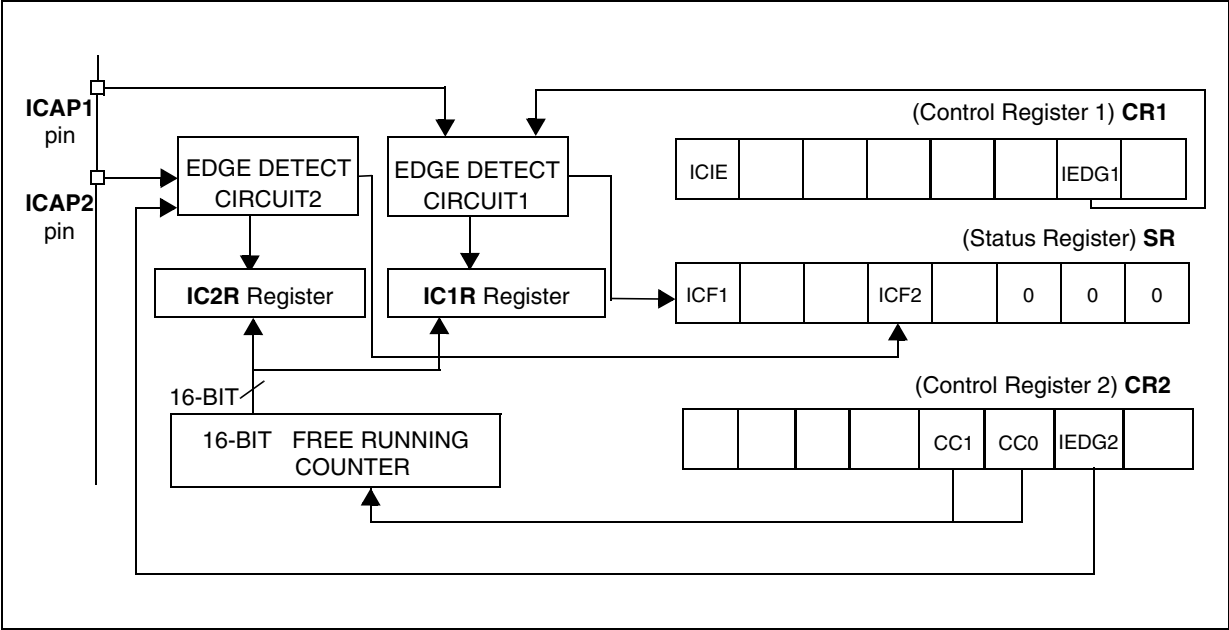
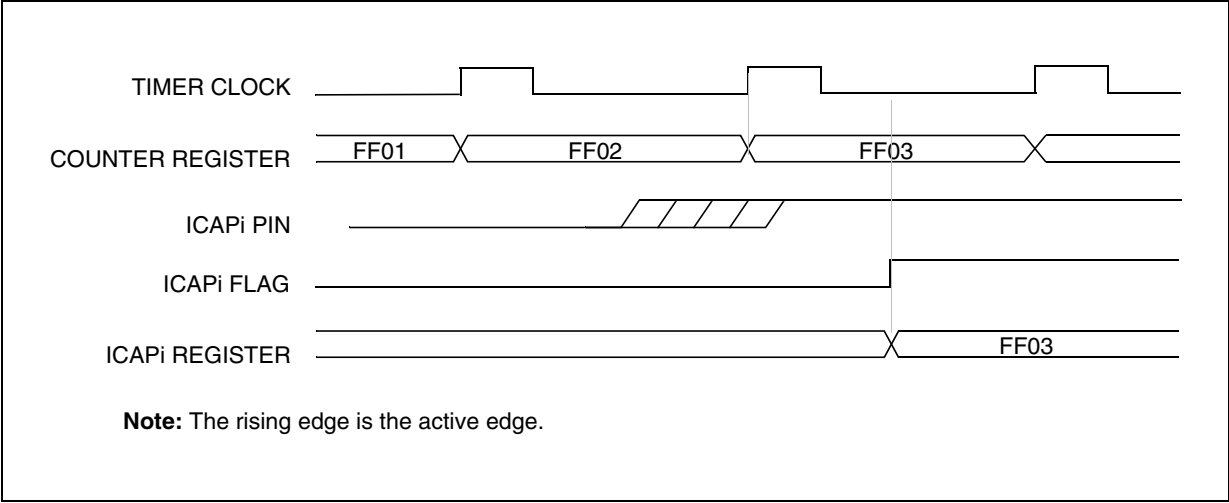


Figure 47. Input Capture Timing Diagram



SERIAL COMMUNICATIONS INTERFACE (Cont'd)**DATA REGISTER (SCIDR)**

Read/Write

Reset Value: Undefined

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

7							0
DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0

The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR).

The TDR register provides the parallel interface between the internal bus and the output shift register (see Figure 1.).

The RDR register provides the parallel interface between the input shift register and the internal bus (see Figure 1.).

BAUD RATE REGISTER (SCIBRR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
SCP1	SCP0	SCT2	SCT1	SCT0	SCR2	SCR1	SCR0

Bits 7:6 = **SCP[1:0]** *First SCI Prescaler*

These 2 prescaling bits allow several standard clock division ranges:

PR Prescaling factor	SCP1	SCP0
1	0	0
3	0	1
4	1	0
13	1	1

Bits 5:3 = **SCT[2:0]** *SCI Transmitter rate divisor*

These 3 bits, in conjunction with the SCP1 & SCP0 bits define the total division applied to the bus clock to yield the transmit rate clock in conventional Baud Rate Generator mode.

TR dividing factor	SCT2	SCT1	SCT0
1	0	0	0
2	0	0	1
4	0	1	0
8	0	1	1
16	1	0	0
32	1	0	1
64	1	1	0
128	1	1	1

Bits 2:0 = **SCR[2:0]** *SCI Receiver rate divisor.*

These 3 bits, in conjunction with the SCP[1:0] bits define the total division applied to the bus clock to yield the receive rate clock in conventional Baud Rate Generator mode.

RR Dividing factor	SCR2	SCR1	SCR0
1	0	0	0
2	0	0	1
4	0	1	0
8	0	1	1
16	1	0	0
32	1	0	1
64	1	1	0
128	1	1	1

10-BIT A/D CONVERTER (ADC) (Cont'd)

10.8.3 Functional Description

The conversion is monotonic, meaning that the result never decreases if the analog input does not and never increases if the analog input does not.

If the input voltage (V_{AIN}) is greater than V_{AREF} (high-level voltage reference) then the conversion result is FFh in the ADCDRH register and 03h in the ADCDRL register (without overflow indication).

If the input voltage (V_{AIN}) is lower than V_{SSA} (low-level voltage reference) then the conversion result in the ADCDRH and ADCDRL registers is 00 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDRH and ADCDRL registers. The accuracy of the conversion is described in the Electrical Characteristics Section.

R_{AIN} is the maximum recommended impedance for an analog input signal. If the impedance is too high, this will result in a loss of accuracy due to leakage and sampling not being completed in the allotted time.

10.8.3.1 A/D Converter Configuration

The analog input ports must be configured as input, no pull-up, no interrupt. Refer to the «I/O ports» chapter. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the ADCCSR register:

- Select the CS[3:0] bits to assign the analog channel to convert.

10.8.3.2 Starting the Conversion

In the ADCCSR register:

- Set the ADON bit to enable the A/D converter and to start the conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete:

- The EOC bit is set by hardware.
- The result is in the ADCDR registers.

A read to the ADCDRH or a write to any bit of the ADCCSR register resets the EOC bit.

To read the 10 bits, perform the following steps:

1. Poll the EOC bit
2. Read the ADCDRL register
3. Read the ADCDRH register. This clears EOC automatically.

Note: The data is not latched, so both the low and the high data register must be read before the next conversion is complete, so it is recommended to disable interrupts while reading the conversion result.

To read only 8 bits, perform the following steps:

1. Poll the EOC bit
2. Read the ADCDRH register. This clears EOC automatically.

10.8.3.3 Changing the conversion channel

The application can change channels during conversion. When software modifies the CH[3:0] bits in the ADCCSR register, the current conversion is stopped, the EOC bit is cleared, and the A/D converter starts converting the newly selected channel.

10.8.4 Low Power Modes

Note: The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed and between single shot conversions.

Mode	Description
WAIT	No effect on A/D Converter
HALT	A/D Converter disabled. After wakeup from Halt mode, the A/D Converter requires a stabilization time t_{STAB} (see Electrical Characteristics) before accurate conversions can be performed.

10.8.5 Interrupts

None.

INSTRUCTION SET OVERVIEW (Cont'd)**11.2 INSTRUCTION GROUPS**

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may

be subdivided into 13 main groups as illustrated in the following table:

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

Using a prebyte

The instructions are described with one to four opcodes.

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

The whole instruction becomes:

PC-2 End of previous instruction
 PC-1 Prebyte
 PC Opcode
 PC+1 Additional word (0 to 2) according to the number of bytes required to compute the effective address

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

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

PIX 92 Replace an instruction using direct, direct bit, or direct relative addressing mode to an instruction using the corresponding indirect addressing mode.

It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.

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

SUPPLY CURRENT CHARACTERISTICS (Cont'd)**12.4.3 On-Chip Peripherals**

Measured on LQFP64 generic board $T_A = 25^\circ\text{C}$ $f_{\text{CPU}}=4\text{MHz}$.

Symbol	Parameter	Conditions	Typ	Unit
$I_{\text{DD(TIM)}}$	16-bit Timer supply current ¹⁾	$V_{\text{DD}}=5.0\text{V}$	50	μA
$I_{\text{DD(ART)}}$	ART PWM supply current ²⁾	$V_{\text{DD}}=5.0\text{V}$	75	μA
$I_{\text{DD(SPI)}}$	SPI supply current ³⁾	$V_{\text{DD}}=5.0\text{V}$	400	μA
$I_{\text{DD(SCI)}}$	SCI supply current ⁴⁾	$V_{\text{DD}}=5.0\text{V}$	400	μA
$I_{\text{DD(I2C)}}$	I2C supply current ⁵⁾	$V_{\text{DD}}=5.0\text{V}$	175	μA
$I_{\text{DD(ADC)}}$	ADC supply current when converting ⁶⁾	$V_{\text{DD}}=5.0\text{V}$	400	μA

Notes:

1. Data based on a differential I_{DD} measurement between reset configuration (timer counter running at $f_{\text{CPU}}/4$) and timer counter stopped (only TIMD bit set). Data valid for one timer.
2. Data based on a differential I_{DD} measurement between reset configuration (timer stopped) and timer counter enabled (only TCE bit set).
3. Data based on a differential I_{DD} measurement between reset configuration (SPI disabled) and a permanent SPI master communication at maximum speed (data sent equal to 55h). This measurement includes the pad toggling consumption.
4. Data based on a differential I_{DD} measurement between SCI low power state (SCID=1) and a permanent SCI data transmit sequence.
5. Data based on a differential I_{DD} measurement between reset configuration (I2C disabled) and a permanent I2C master communication at 100kHz (data sent equal to 55h). This measurement include the pad toggling consumption (27kOhm external pull-up on clock and data lines).
6. Data based on a differential I_{DD} measurement between reset configuration and continuous A/D conversions.

12.8 I/O PORT PIN CHARACTERISTICS

12.8.1 General Characteristics

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

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	Input low level voltage ¹⁾	CMOS ports			$0.3 \times V_{DD}$	V
V_{IH}	Input high level voltage ¹⁾		$0.7 \times V_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis ²⁾			0.7		
$I_{INJ(PIN)}$ ³⁾	Injected Current on PC6 (Flash devices only)	$V_{DD}=5V$	0		+4	mA
	Injected Current on an I/O pin				± 4	
$\Sigma I_{INJ(PIN)}$ ³⁾	Total injected current (sum of all I/O and control pins)				± 25	
I_L	Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$			± 1	μA
I_S	Static current consumption	Floating input mode ⁴⁾		400		
R_{PU}	Weak pull-up equivalent resistor ⁵⁾	$V_{IN}=V_{SS}$ $V_{DD}=5V$	50	120	250	$k\Omega$
C_{IO}	I/O pin capacitance			5		pF
$t_{f(I/O)out}$	Output high to low level fall time ¹⁾	$C_L=50pF$ Between 10% and 90%		25		ns
$t_{r(I/O)out}$	Output low to high level rise time ¹⁾			25		
$t_{w(IT)in}$	External interrupt pulse time ⁶⁾		1			t_{CPU}

Figure 81. Unused I/Os configured as input

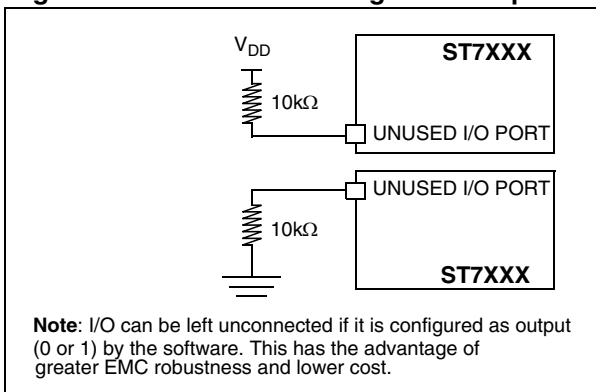
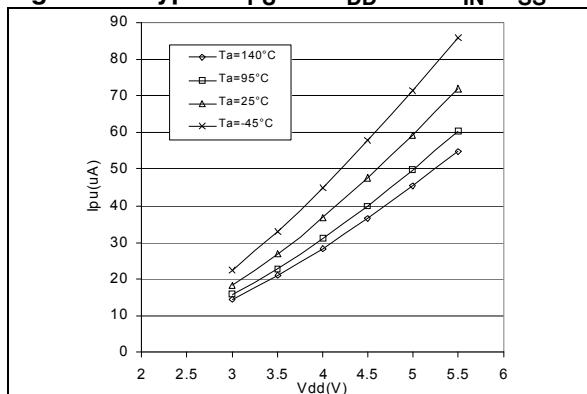


Figure 82. Typical I_{PU} vs. V_{DD} with $V_{IN}=V_{SS}$



Notes:

1. Data based on characterization results, not tested in production.
2. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.
3. When the current limitation is not possible, the V_{IN} maximum must be respected, otherwise refer to $I_{INJ(PIN)}$ specification. A positive injection is induced by $V_{IN} > V_{DD}$ while a negative injection is induced by $V_{IN} < V_{SS}$. Refer to section 12.2.2 on page 139 for more details.
4. Configuration not recommended, all unused pins must be kept at a fixed voltage: using the output mode of the I/O for example and leaving the I/O unconnected on the board or an external pull-up or pull-down resistor (see Figure 81). Static peak current value taken at a fixed V_{IN} value, based on design simulation and technology characteristics, not tested in production. This value depends on V_{DD} and temperature values.
5. The R_{PU} pull-up equivalent resistor is based on a resistive transistor (corresponding I_{PU} current characteristics described in Figure 82).
6. To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

12.9 CONTROL PIN CHARACTERISTICS

12.9.1 Asynchronous $\overline{\text{RESET}}$ Pin

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

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IL}	Input low level voltage ¹⁾				$0.16 \times V_{DD}$	V
V_{IH}	Input high level voltage ¹⁾		$0.85 \times V_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis ²⁾			2.5		V
V_{OL}	Output low level voltage ³⁾	$V_{DD}=5V$ $I_{IO}=+2mA$		0.2	0.5	
I_{IO}	Input current on $\overline{\text{RESET}}$ pin			2		mA
R_{ON}	Weak pull-up equivalent resistor		20	30	120	k Ω
$t_{w(RSTL)out}$	Generated reset pulse duration	Stretch applied on external pulse	0		$42^{6)}$	μs
		Internal reset sources	20	30	$42^{6)}$	μs
$t_{h(RSTL)in}$	External reset pulse hold time ⁴⁾		2.5			μs
$t_{g(RSTL)in}$	Filtered glitch duration ⁵⁾			200		ns

Notes:

1. Data based on characterization results, not tested in production.
2. Hysteresis voltage between Schmitt trigger switching levels.
3. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 12.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS} .
4. To guarantee the reset of the device, a minimum pulse has to be applied to the $\overline{\text{RESET}}$ pin. All short pulses applied on the $\overline{\text{RESET}}$ pin with a duration below $t_{h(RSTL)in}$ can be ignored.
5. The reset network (the resistor and two capacitors) protects the device against parasitic resets, especially in noisy environments.
6. Data guaranteed by design, not tested in production.

DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

Table 30. Suggested List of Socket Types

Device	Socket (supplied with ST7MDT20M-EMU3)	Emulator Adapter (supplied with ST7MDT20M-EMU3)
LQFP64 14 x14	CAB 3303262	CAB 3303351
LQFP64 10 x10	YAMAICHI IC149-064-*75-*5	YAMAICHI ICP-064-6
LQFP44 10 X10	YAMAICHI IC149-044-*52-*5	YAMAICHI ICP-044-5

14.3.4 Socket and Emulator Adapter Information

For information on the type of socket that is supplied with the emulator, refer to the suggested list of sockets in Table 30.

Note: Before designing the board layout, it is recommended to check the overall dimensions of the socket as they may be greater than the dimensions of the device.

For footprint and other mechanical information about these sockets and adapters, refer to the manufacturer's datasheet.

Related Documentation

AN 978: ST7 Visual Develop Software Key Debugging Features

AN 1938: ST7 Visual Develop for ST7 Cosmic C toolset users

AN 1940: ST7 Visual Develop for ST7 Assembler Linker toolset users

Table 31. ST7 Application Notes

IDENTIFICATION	DESCRIPTION
AN1947	ST7MC PMAC SINE WAVE MOTOR CONTROL SOFTWARE LIBRARY
GENERAL PURPOSE	
AN1476	LOW COST POWER SUPPLY FOR HOME APPLIANCES
AN1526	ST7FLITE0 QUICK REFERENCE NOTE
AN1709	EMC DESIGN FOR ST MICROCONTROLLERS
AN1752	ST72324 QUICK REFERENCE NOTE
PRODUCT EVALUATION	
AN 910	PERFORMANCE BENCHMARKING
AN 990	ST7 BENEFITS VS INDUSTRY STANDARD
AN1077	OVERVIEW OF ENHANCED CAN CONTROLLERS FOR ST7 AND ST9 MCUS
AN1086	U435 CAN-DO SOLUTIONS FOR CAR MULTIPLEXING
AN1103	IMPROVED B-EMF DETECTION FOR LOW SPEED, LOW VOLTAGE WITH ST72141
AN1150	BENCHMARK ST72 VS PC16
AN1151	PERFORMANCE COMPARISON BETWEEN ST72254 & PC16F876
AN1278	LIN (LOCAL INTERCONNECT NETWORK) SOLUTIONS
PRODUCT MIGRATION	
AN1131	MIGRATING APPLICATIONS FROM ST72511/311/214/124 TO ST72521/321/324
AN1322	MIGRATING AN APPLICATION FROM ST7263 REV.B TO ST7263B
AN1365	GUIDELINES FOR MIGRATING ST72C254 APPLICATIONS TO ST72F264
AN1604	HOW TO USE ST7MDT1-TRAIN WITH ST72F264
AN2200	GUIDELINES FOR MIGRATING ST7LITE1X APPLICATIONS TO ST7FLITE1XB
PRODUCT OPTIMIZATION	
AN 982	USING ST7 WITH CERAMIC RESONATOR
AN1014	HOW TO MINIMIZE THE ST7 POWER CONSUMPTION
AN1015	SOFTWARE TECHNIQUES FOR IMPROVING MICROCONTROLLER EMC PERFORMANCE
AN1040	MONITORING THE VBUS SIGNAL FOR USB SELF-POWERED DEVICES
AN1070	ST7 CHECKSUM SELF-CHECKING CAPABILITY
AN1181	ELECTROSTATIC DISCHARGE SENSITIVE MEASUREMENT
AN1324	CALIBRATING THE RC OSCILLATOR OF THE ST7FLITE0 MCU USING THE MAINS
AN1502	EMULATED DATA EEPROM WITH ST7 HDFLASH MEMORY
AN1529	EXTENDING THE CURRENT & VOLTAGE CAPABILITY ON THE ST7265 VDDF SUPPLY
AN1530	ACCURATE TIMEBASE FOR LOW-COST ST7 APPLICATIONS WITH INTERNAL RC OSCILLATOR
AN1605	USING AN ACTIVE RC TO WAKEUP THE ST7LITE0 FROM POWER SAVING MODE
AN1636	UNDERSTANDING AND MINIMIZING ADC CONVERSION ERRORS
AN1828	PIR (PASSIVE INFRARED) DETECTOR USING THE ST7FLITE05/09/SUPERLITE
AN1946	SENSORLESS BLDC MOTOR CONTROL AND BEMF SAMPLING METHODS WITH ST7MC
AN1953	PFC FOR ST7MC STARTER KIT
AN1971	ST7LITE0 MICROCONTROLLED BALLAST
PROGRAMMING AND TOOLS	
AN 978	ST7 VISUAL DEVELOP SOFTWARE KEY DEBUGGING FEATURES
AN 983	KEY FEATURES OF THE COSMIC ST7 C-COMPILER PACKAGE
AN 985	EXECUTING CODE IN ST7 RAM
AN 986	USING THE INDIRECT ADDRESSING MODE WITH ST7
AN 987	ST7 SERIAL TEST CONTROLLER PROGRAMMING
AN 988	STARTING WITH ST7 ASSEMBLY TOOL CHAIN
AN1039	ST7 MATH UTILITY ROUTINES

Table 31. ST7 Application Notes

IDENTIFICATION	DESCRIPTION
AN1071	HALF DUPLEX USB-TO-SERIAL BRIDGE USING THE ST72611 USB MICROCONTROLLER
AN1106	TRANSLATING ASSEMBLY CODE FROM HC05 TO ST7
AN1179	PROGRAMMING ST7 FLASH MICROCONTROLLERS IN REMOTE ISP MODE (IN-SITU PROGRAMMING)
AN1446	USING THE ST72521 EMULATOR TO DEBUG AN ST72324 TARGET APPLICATION
AN1477	EMULATED DATA EEPROM WITH XFLASH MEMORY
AN1527	DEVELOPING A USB SMARTCARD READER WITH ST7SCR
AN1575	ON-BOARD PROGRAMMING METHODS FOR XFLASH AND HDFLASH ST7 MCUS
AN1576	IN-APPLICATION PROGRAMMING (IAP) DRIVERS FOR ST7 HDFLASH OR XFLASH MCUS
AN1577	DEVICE FIRMWARE UPGRADE (DFU) IMPLEMENTATION FOR ST7 USB APPLICATIONS
AN1601	SOFTWARE IMPLEMENTATION FOR ST7DALI-EVAL
AN1603	USING THE ST7 USB DEVICE FIRMWARE UPGRADE DEVELOPMENT KIT (DFU-DK)
AN1635	ST7 CUSTOMER ROM CODE RELEASE INFORMATION
AN1754	DATA LOGGING PROGRAM FOR TESTING ST7 APPLICATIONS VIA ICC
AN1796	FIELD UPDATES FOR FLASH BASED ST7 APPLICATIONS USING A PC COMM PORT
AN1900	HARDWARE IMPLEMENTATION FOR ST7DALI-EVAL
AN1904	ST7MC THREE-PHASE AC INDUCTION MOTOR CONTROL SOFTWARE LIBRARY
AN1905	ST7MC THREE-PHASE BLDC MOTOR CONTROL SOFTWARE LIBRARY
SYSTEM OPTIMIZATION	
AN1711	SOFTWARE TECHNIQUES FOR COMPENSATING ST7 ADC ERRORS
AN1827	IMPLEMENTATION OF SIGMA-DELTA ADC WITH ST7FLITE05/09
AN2009	PWM MANAGEMENT FOR 3-PHASE BLDC MOTOR DRIVES USING THE ST7FMC
AN2030	BACK EMF DETECTION DURING PWM ON TIME BY ST7MC

```

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 In-
terrupts 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
LD A,X ; check for falling edge
cp A,$02
jrne OUT
TNZ Y
jrne OUT
LD A,$01
LD sema,A ; set the semaphore to '1' if edge is
detected
RIM ; reset the interrupt mask
LD A,sema ; check the semaphore status
CP A,$01

```

```

jrne OUT
call call_routine; call the interrupt routine
RIM
OUT: RIM
JP while_loop
.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

```

15.1.6 Clearing active interrupts outside interrupt routine

When an active interrupt request occurs at the same time as the related flag is being cleared, an unwanted reset may occur.

Note: clearing the related interrupt mask will not generate an unwanted reset

Concurrent interrupt context

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

when:

- The interrupt flag is cleared within its own interrupt routine
- The interrupt flag is cleared within any interrupt routine
- The interrupt flag is cleared in any part of the code while this interrupt is disabled

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

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

Example:

```

SIM
reset interrupt flag
RIM

```

Nested interrupt context:

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

when:

- The interrupt flag is cleared within its own interrupt routine

16 REVISION HISTORY

Table 32. Revision History

Date	Revision	Description of Changes
18-Oct-2004	1.10	<p>Added “related documentation” section in specific chapters throughout document</p> <p>Flash readout protection sentence added section 4.3.1 on page 18</p> <p>I²C Chapter updated, (Section 10.7)</p> <p>V_t POR max modified in section 12.4 on page 142</p> <p>Added Figure 89 on page 160</p> <p>Modified description of $t_{w(RSTL)out}$ in “Asynchronous RESET Pin” on page 159</p> <p>Added Note on PE2 pin in Table 2, “Device Pin Description,” on page 10</p> <p>Modified V_{AREF} min in “10-BIT ADC CHARACTERISTICS” on page 167</p> <p>Modified I_{INJ} for PC6 in Section 12.8</p> <p>Updated ADC accuracy data and notes in section 12.12.3 on page 170 and “KNOWN LIMITATIONS” on page 186</p> <p>Added “Clearing active interrupts outside interrupt routine” on page 187</p> <p>I2C multimaster bug added in Known Limitations, Section 15.1.10</p> <p>Please read carefully the “KNOWN LIMITATIONS” on page 186</p>
13-Mar-2009	2	<p>Updated root part numbers and device summary table on coverpage.</p> <p>Removed temperature ranges 1, 5, and 7.</p> <p>Renamed all TQFP packages, LQFP.</p> <p>Updated note 6 below Table 2, “Device Pin Description,” on page 10.</p> <p>Updated data retention in “Features” and Section 12.6.2 FLASH Memory.</p> <p>Updated Figure 50 “Output Compare Timing Diagram, f_{TIMER} = f_{CPU}/4” .</p> <p>Updated Figure 94 “SPI Master Timing Diagram 1)” .</p> <p>Added Section 10.8.3.3 “Changing the conversion channel” on page 129.</p> <p>Modified “Absolute Maximum Ratings (Electrical Sensitivity)” on page 155.</p> <p>Update Note 4 in Section 12.8 I/O PORT PIN CHARACTERISTICS “General Characteristics” .</p> <p>Updated notes in Section 13.2 THERMAL CHARACTERISTICS.</p> <p>Removed recommended wave soldering profile and recommended reflow soldering oven profile, and added ECOPACK text in “SOLDERING AND GLUEABILITY INFORMATION” on page 174 (Section 13.3).</p> <p>Removed automotive temperature ranges from Section 12.3.1 General Operating Conditions, and from Figure 106 “ROM Factory Coded Device Types” and option list. Modified Section 14.2 “DEVICE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE” on page 177: replace list or order codes by ordering scheme.</p> <p>Added Section 15.1.5 External interrupt missed, and Section 15.2.1 Internal RC Oscillator with LVD inSection 15 KNOWN LIMITATIONS. Modified Section 15.1.9 “TIMD set simultaneously with OC interrupt” on page 189.</p>