



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

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	48
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	64-LQFP
Supplier Device Package	•
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f321ar9t6

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Table of Contents

14.1 FLAS	H OPTION BYTES	175
14.2 DEVI0	CE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE	177
14.3 DEVE	LOPMENT TOOLS	181
14.3.1	Starter kits	181
14.3.2	Development and debugging tools	181
14.3.3	Programming tools	181
14.3.4	Socket and Emulator Adapter Information	182
14.4 ST7 A	PPLICATION NOTES	183
15 KNOWN L	IMITATIONS	186
15.1 ALL F	LASH AND ROM DEVICES	186
15.1.1	External RC option	186
15.1.2	Safe Connection of OSC1/OSC2 Pins	186
15.1.3	Reset pin protection with LVD Enabled	186
15.1.4	Unexpected Reset Fetch	186
15.1.5	External interrupt missed	186
15.1.6	Clearing active interrupts outside interrupt routine	187
15.1.7	SCI Wrong Break duration	189
15.1.8	16-bit Timer PWM Mode	189
15.1.9	TIMD set simultaneously with OC interrupt	189
15.1.1		189
15.2 ALL F		189
15.2.1	Internal RC Oscillator with LVD	189
15.3 LIMIT	ATIONS SPECIFIC TO REV Q AND REV S FLASH DEVICES	189
15.3.1	ADC Accuracy	189
15.4 LIMIT		190
15.4.1	LVD Operation	190
15.4.2	LVD Startup behaviour	191
15.4.3	AVD not supported	191
15.4.4		191
15.4.5		191
15.4.6	Puil-up not present on PE2	191
15.4.7	$\square eau-out protection with LVD \dots$	101
		102
		152

2 PIN DESCRIPTION

Figure 2. 64-Pin LQFP 14x14 and 10x10 Package Pinout



Figure 3. 44-Pin LQFP Package Pinout

57/



INTERRUPTS (Cont'd)

Servicing Pending Interrupts

As several interrupts can be pending at the same time, the interrupt to be taken into account is determined by the following two-step process:

- the highest software priority interrupt is serviced,
- if several interrupts have the same software priority then the interrupt with the highest hardware priority is serviced first.

Figure 19 describes this decision process.

Figure 19. Priority Decision Process



When an interrupt request is not serviced immediately, it is latched and then processed when its software priority combined with the hardware priority becomes the highest one.

Note 1: The hardware priority is exclusive while the software one is not. This allows the previous process to succeed with only one interrupt.

Note 2: TLI, RESET and TRAP can be considered as having the highest software priority in the decision process.

Different Interrupt Vector Sources

Two interrupt source types are managed by the ST7 interrupt controller: the non-maskable type (RESET, TRAP) and the maskable type (external or from internal peripherals).

Non-Maskable Sources

These sources are processed regardless of the state of the 11 and 10 bits of the CC register (see Figure 18). After stacking the PC, X, A and CC registers (except for RESET), the corresponding vector is loaded in the PC register and the 11 and 10 bits of the CC are set to disable interrupts (level 3). These sources allow the processor to exit HALT mode.

TRAP (Non Maskable Software Interrupt)

This software interrupt is serviced when the TRAP instruction is executed. It will be serviced according to the flowchart in Figure 18.

Caution: TRAP can be interrupted by a TLI.

RESET

The RESET source has the highest priority in the ST7. This means that the first current routine has the highest software priority (level 3) and the highest hardware priority.

See the RESET chapter for more details.

Maskable Sources

Maskable interrupt vector sources can be serviced if the corresponding interrupt is enabled and if its own interrupt software priority (in ISPRx registers) is higher than the one currently being serviced (I1 and I0 in CC register). If any of these two conditions is false, the interrupt is latched and thus remains pending.

TLI (Top Level Hardware Interrupt)

This hardware interrupt occurs when a specific edge is detected on the dedicated TLI pin. It will be serviced according to the flowchart in Figure 18 as a trap.

Caution: A TRAP instruction must not be used in a TLI service routine.

External Interrupts

External interrupts allow the processor to exit from HALT low power mode. External interrupt sensitivity is software selectable through the External Interrupt Control register (EICR).

External interrupt triggered on edge will be latched and the interrupt request automatically cleared upon entering the interrupt service routine.

If several input pins of a group connected to the same interrupt line are selected simultaneously, these will be logically ORed.

Peripheral Interrupts

Usually the peripheral interrupts cause the MCU to exit from HALT mode except those mentioned in the "Interrupt Mapping" table. A peripheral interrupt occurs when a specific flag is set in the peripheral status registers and if the corresponding enable bit is set in the peripheral control register. The general sequence for clearing an interrupt is based on an access to the status register followed by a read or write to an associated register.

Note: The clearing sequence resets the internal latch. A pending interrupt (i.e. waiting for being serviced) will therefore be lost if the clear sequence is executed.

57

INTERRUPTS (Cont'd)

Table 8. Interrupt Mapping

N°	Source Block	Description	Register Label	Priority Order	Exit from HALT ³⁾	Address Vector
	RESET	Reset	NI/A		yes	FFFEh-FFFFh
	TRAP	Software interrupt			no	FFFCh-FFFDh
0	TLI	External top level interrupt	EICR		yes	FFFAh-FFFBh
1	MCC/RTC	Main clock controller time base interrupt	MCCSR	Higher	yes	FFF8h-FFF9h
2	ei0	External interrupt port A30		Priority	yes	FFF6h-FFF7h
3	ei1	External interrupt port F20	N/A		yes	FFF4h-FFF5h
4	ei2	External interrupt port B30			yes	FFF2h-FFF3h
5	ei3	External interrupt port B74			yes	FFF0h-FFF1h
6		Not used				FFEEh-FFEFh
7	SPI	SPI peripheral interrupts	SPICSR		yes ¹	FFECh-FFEDh
8	TIMER A	TIMER A peripheral interrupts	TASR	▼	no	FFEAh-FFEBh
9	TIMER B	TIMER B peripheral interrupts	TBSR		no	FFE8h-FFE9h
10	SCI	SCI Peripheral interrupts	SCISR	Lower	no	FFE6h-FFE7h
11	AVD	Auxiliary Voltage detector interrupt	SICSR	Priority	no	FFE4h-FFE5h
12	I2C	I2C Peripheral interrupts	(see periph)		no	FFE2h-FFE3h
13	PWM ART	PWM ART interrupt	ARTCSR		yes ²	FFE0h-FFE1h

Notes:

- 1. Exit from HALT possible when SPI is in slave mode.
- 2. Exit from HALT possible when PWM ART is in external clock mode.
- 3. Only a RESET or MCC/RTC interrupt can be used to wake-up from Active Halt mode.

7.6 EXTERNAL INTERRUPTS

7.6.1 I/O Port Interrupt Sensitivity

The external interrupt sensitivity is controlled by the IPA, IPB and ISxx bits of the EICR register (Figure 22). This control allows to have up to 4 fully independent external interrupt source sensitivities.

Each external interrupt source can be generated on four (or five) different events on the pin:

- Falling edge
- Rising edge

57/

Falling and rising edge

- Falling edge and low level
- Rising edge and high level (only for ei0 and ei2)

To guarantee correct functionality, the sensitivity bits in the EICR register can be modified only when the I1 and I0 bits of the CC register are both set to 1 (level 3). This means that interrupts must be disabled before changing sensitivity.

The pending interrupts are cleared by writing a different value in the ISx[1:0], IPA or IPB bits of the EICR.

9 I/O PORTS

9.1 INTRODUCTION

The I/O ports offer different functional modes: – transfer of data through digital inputs and outputs

- and for specific pins:
- external interrupt generation
- alternate signal input/output for the on-chip peripherals.

An I/O port contains up to 8 pins. Each pin can be programmed independently as digital input (with or without interrupt generation) or digital output.

9.2 FUNCTIONAL DESCRIPTION

Each port has two main registers:

- Data Register (DR)
- Data Direction Register (DDR)
- and one optional register:
- Option Register (OR)

Each I/O pin may be programmed using the corresponding register bits in the DDR and OR registers: Bit X corresponding to pin X of the port. The same correspondence is used for the DR register.

The following description takes into account the OR register, (for specific ports which do not provide this register refer to the I/O Port Implementation section). The generic I/O block diagram is shown in Figure 1

9.2.1 Input Modes

The input configuration is selected by clearing the corresponding DDR register bit.

In this case, reading the DR register returns the digital value applied to the external I/O pin.

Different input modes can be selected by software through the OR register.

Notes:

1. Writing the DR register modifies the latch value but does not affect the pin status.

2. When switching from input to output mode, the DR register has to be written first to drive the correct level on the pin as soon as the port is configured as an output.

3. Do not use read/modify/write instructions (BSET or BRES) to modify the DR register as this might corrupt the DR content for I/Os configured as input.

External interrupt function

When an I/O is configured as Input with Interrupt, an event on this I/O can generate an external interrupt request to the CPU. Each pin can independently generate an interrupt request. The interrupt sensitivity is independently programmable using the sensitivity bits in the EICR register.

Each external interrupt vector is linked to a dedicated group of I/O port pins (see pinout description and interrupt section). If several input pins are selected simultaneously as interrupt sources, these are first detected according to the sensitivity bits in the EICR register and then logically ORed.

The external interrupts are hardware interrupts, which means that the request latch (not accessible directly by the application) is automatically cleared when the corresponding interrupt vector is fetched. To clear an unwanted pending interrupt by software, the sensitivity bits in the EICR register must be modified.

9.2.2 Output Modes

The output configuration is selected by setting the corresponding DDR register bit. In this case, writing the DR register applies this digital value to the I/O pin through the latch. Then reading the DR register returns the previously stored value.

Two different output modes can be selected by software through the OR register: Output push-pull and open-drain.

DR register value and output pin status:

DR	Push-pull	Open-drain
0	V _{SS}	Vss
1	V _{DD}	Floating

9.2.3 Alternate Functions

When an on-chip peripheral is configured to use a pin, the alternate function is automatically selected. This alternate function takes priority over the standard I/O programming.

When the signal is coming from an on-chip peripheral, the I/O pin is automatically configured in output mode (push-pull or open drain according to the peripheral).

When the signal is going to an on-chip peripheral, the I/O pin must be configured in input mode. In this case, the pin state is also digitally readable by addressing the DR register.

Note: Input pull-up configuration can cause unexpected value at the input of the alternate peripheral input. When an on-chip peripheral use a pin as input and output, this pin has to be configured in input floating mode.



10 ON-CHIP PERIPHERALS

10.1 WATCHDOG TIMER (WDG)

10.1.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the T6 bit becomes cleared.

10.1.2 Main Features

- Programmable free-running downcounter
- Programmable reset
- Reset (if watchdog activated) when the T6 bit reaches zero
- HALT Optional reset on instruction (configurable by option byte)
- Hardware Watchdog selectable by option byte

10.1.3 Functional Description

The counter value stored in the Watchdog Control register (WDGCR bits T[6:0]), is decremented every 16384 f_{OSC2} cycles (approx.), and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T[6:0]) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling the reset pin low for typically 30µs.

The application program must write in the WDGCR register at regular intervals during normal operation to prevent an MCU reset. This downcounter is free-running: it counts down even if the watchdog is disabled. The value to be stored in the WDGCR register must be between FFh and C0h:

- The WDGA bit is set (watchdog enabled)
- The T6 bit is set to prevent generating an immediate reset
- The T[5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset (see Figure 2. Approximate Timeout Duration). The timing varies between a minimum and a maximum value due to the unknown status of the prescaler when writing to the WDGCR register (see Figure 3).

Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

If the watchdog is activated, the HALT instruction will generate a Reset.



Figure 32. Watchdog Block Diagram

WATCHDOG TIMER (Cont'd)

10.1.5 Low Power Modes

Mode	Description		
SLOW	No effect on	Watchdog.	
WAIT	No effect on	Watchdog.	
	OIE bit in	WDGHALT bit	
	MCCSR	in Option	
	register	Byte	
HALT	0	0	No Watchdog reset is generated. The MCU enters Halt mode. The Watch- dog counter is decremented once and then stops counting and is no longer able to generate a watchdog reset until the MCU receives an external inter- rupt or a reset.
			If an external interrupt is received, the Watchdog restarts counting after 256 or 4096 CPU clocks. If a reset is generated, the Watchdog is disabled (reset state) unless Hardware Watchdog is selected by option byte. For application recommendations see Section 0.1.7 below.
	0	1	A reset is generated.
	1	x	No reset is generated. The MCU enters Active Halt mode. The Watchdog counter is not decremented. It stop counting. When the MCU receives an oscillator interrupt or external interrupt, the Watchdog restarts counting immediately. When the MCU receives a reset the Watchdog restarts counting after 256 or 4096 CPU clocks.

10.1.6 Hardware Watchdog Option

If Hardware Watchdog is selected by option byte, the watchdog is always active and the WDGA bit in the WDGCR is not used. Refer to the Option Byte description.

10.1.7 Using Halt Mode with the WDG (WDGHALT option)

The following recommendation applies if Halt mode is used when the watchdog is enabled.

 Before executing the HALT instruction, refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller.

10.1.8 Interrupts

None.

5/

10.1.9 Register Description CONTROL REGISTER (WDGCR)

Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
WDGA	Т6	T5	T4	Т3	T2	T1	то

Bit 7 = **WDGA** Activation bit.

This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.

0: Watchdog disabled

1: Watchdog enabled

Note: This bit is not used if the hardware watchdog option is enabled by option byte.

Bit 6:0 = **T[6:0]** 7-bit counter (MSB to LSB).

These bits contain the value of the watchdog counter. It is decremented every 16384 f_{OSC2} cycles (approx.). A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK (Cont'd)

Bit 0 = **OIF** Oscillator interrupt flag

This bit is set by hardware and cleared by software reading the MCCSR register. It indicates when set that the main oscillator has reached the selected elapsed time (TB1:0).

0: Timeout not reached

1: Timeout reached

57/

CAUTION: The BRES and BSET instructions must not be used on the MCCSR register to avoid unintentionally clearing the OIF bit.

MCC BEEP CONTROL REGISTER (MCCBCR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	BC1	BC0

Bit 7:2 = Reserved, must be kept cleared.

Bit 1:0 = **BC[1:0]** *Beep control*

These 2 bits select the PF1 pin beep capability.

BC1	BC0	Beep mode with f _{OSC2} =8MHz					
0	0	Off					
0	1	~2-KHz	Output				
1	0	~1-KHz	Beep signal				
1	1	~500-Hz	~50% duty cycle				

The beep output signal is available in ACTIVE-HALT mode but has to be disabled to reduce the consumption.

Table 15. Main Clock Controller Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
002Rh	SICSR	AVDS	AVDIE	AVDF	LVDRF				WDGRF
002011	Reset Value	0	0	0	х	0	0	0	х
002Ch	MCCSR	MCO	CP1	CP0	SMS	TB1	TB0	OIE	OIF
002Ch	Reset Value	0	0	0	0	0	0	0	0
0000h	MCCBCR							BC1	BC0
002DN	Reset Value	0	0	0	0	0	0	0	0

ON-CHIP PERIPHERALS (Cont'd)

10.3.3 Register Description

CONTROL / STATUS REGISTER (ARTCSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							
EXCL	CC2	CC1	CC0	TCE	FCRL	OIE	OVF

Bit 7 = **EXCL** External Clock

This bit is set and cleared by software. It selects the input clock for the 7-bit prescaler.

0: CPU clock.

1: External clock.

Bit 6:4 = **CC[2:0]** Counter Clock Control These bits are set and cleared by software. They determine the prescaler division ratio from f_{INPUT} .

fcounter	With f _{INPUT} =8 MHz	CC2	CC1	CC0
f _{INPUT}	8 MHz	0	0	0
f _{INPUT} / 2	4 MHz	0	0	1
f _{INPUT} / 4	2 MHz	0	1	0
f _{INPUT} / 8	1 MHz	0	1	1
f _{INPUT} / 16	500 kHz	1	0	0
f _{INPUT} / 32	250 kHz	1	0	1
f _{INPUT} / 64	125 kHz	1	1	0
f _{INPUT} / 128	62.5 kHz	1	1	1

Bit 3 = TCE Timer Counter Enable

This bit is set and cleared by software. It puts the timer in the lowest power consumption mode.

0: Counter stopped (prescaler and counter frozen).1: Counter running.

Bit 2 = FCRL Force Counter Re-Load

This bit is write-only and any attempt to read it will yield a logical zero. When set, it causes the contents of ARTARR register to be loaded into the counter, and the content of the prescaler register to be cleared in order to initialize the timer before starting to count.

Bit 1 = **OIE** Overflow Interrupt Enable

This bit is set and cleared by software. It allows to enable/disable the interrupt which is generated when the OVF bit is set.

0: Overflow Interrupt disable.

1: Overflow Interrupt enable.

Bit 0 = **OVF** Overflow Flag

57/

This bit is set by hardware and cleared by software reading the ARTCSR register. It indicates the transition of the counter from FFh to the ARTARR value. 0: New transition not yet reached

1: Transition reached

COUNTER ACCESS REGISTER (ARTCAR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0

Bit 7:0 = CA[7:0] Counter Access Data

These bits can be set and cleared either by hardware or by software. The ARTCAR register is used to read or write the auto-reload counter "on the fly" (while it is counting).

AUTO-RELOAD REGISTER (ARTARR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

Bit 7:0 = AR[7:0] Counter Auto-Reload Data

These bits are set and cleared by software. They are used to hold the auto-reload value which is automatically loaded in the counter when an overflow occurs. At the same time, the PWM output levels are changed according to the corresponding OPx bit in the PWMCR register.

This register has two PWM management functions:

- Adjusting the PWM frequency
- Setting the PWM duty cycle resolution

PWM Frequency vs Resolution:

ARTARR	Resolution	f _{PWM}			
value	nesolution	Min	Max		
0	8-bit	~0.244 kHz	31.25 kHz		
[0127]	> 7-bit	~0.244 kHz	62.5 kHz		
[128191]	> 6-bit	~0.488 kHz	125 kHz		
[192223]	> 5-bit	~0.977 kHz	250 kHz		
[224239]	> 4-bit	~1.953 kHz	500 kHz		

16-BIT TIMER (Cont'd)

rigure 40. Counter rinning Diagram, in	
CPU CLOCK	
INTERNAL RESET	
TIMER CLOCK	
– COUNTER REGISTER _	\ FFFD\ FFFE\ FFFF\ 0000 \ 0001 \ 0002 \ 0003 \
TIMER OVERFLOW FLAG (TOF)	

Figure 44. Counter Timing Diagram, Internal Clock Divided by 4

Figure 42 Counter Timing Diagram Internal Clock Divided by 2



Figure 45. Counter Timing Diagram, Internal Clock Divided By 8

CPU CLOCK	
INTERNAL RESET	1
TIMER CLOCK	
COUNTER REGISTER	FFFC FFFD 0000
TIMER OVERFLOW FLAG (TOF)	

Note: The MCU is in reset state when the internal reset signal is high, when it is low the MCU is running.

16-BIT TIMER (Cont'd)

10.4.3.5 One Pulse Mode

One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

The One Pulse mode uses the Input Capture1 function and the Output Compare1 function.

Procedure:

To use One Pulse mode:

- 1. Load the OC1R register with the value corresponding to the length of the pulse (see the formula in the opposite column).
- 2. Select the following in the CR1 register:
 - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after the pulse.
 - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin during the pulse.
 - Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin must be configured as floating input).
- 3. Select the following in the CR2 register:
 - Set the OC1E bit, the OCMP1 pin is then dedicated to the Output Compare 1 function.
 - Set the OPM bit.
 - Select the timer clock CC[1:0] (see Table 1).



Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and OLVL2 bit is loaded on the OCMP1 pin, the ICF1 bit is set and the value FFFDh is loaded in the IC1R register.

Because the ICF1 bit is set when an active edge occurs, an interrupt can be generated if the ICIE bit is set.

Clearing the Input Capture interrupt request (that is, clearing the ICF*i* bit) is done in two steps:

1. Reading the SR register while the ICF*i* bit is set.

2. An access (read or write) to the ICiLR register.

The OC1R register value required for a specific timing application can be calculated using the following formula:

Where:

t = Pulse period (in seconds)

 f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits, see Table 1)

If the timer clock is an external clock the formula is:

$$OC_{i}R = t * f_{EXT} - 5$$

Where:

t = Pulse period (in seconds)

f_{EXT} = External timer clock frequency (in hertz)

When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin, (See Figure 10).

Notes:

- 1. The OCF1 bit cannot be set by hardware in One Pulse mode but the OCF2 bit can generate an Output Compare interrupt.
- 2. When the Pulse Width Modulation (PWM) and One Pulse mode (OPM) bits are both set, the PWM mode is the only active one.
- 3. If OLVL1 = OLVL2 a continuous signal will be seen on the OCMP1 pin.
- 4. The ICAP1 pin can not be used to perform input capture. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each time a valid edge occurs on the ICAP1 pin and ICF1 can also generates interrupt if ICIE is set.
- 5. When One Pulse mode is used OC1R is dedicated to this mode. Nevertheless OC2R and OCF2 can be used to indicate a period of time has been elapsed but cannot generate an output waveform because the level OLVL2 is dedicated to the One Pulse mode.

47

16-BIT TIMER (Cont'd)

Table 18. 16-B	it Timer	Register	Мар	and	Reset	Values
----------------	----------	----------	-----	-----	-------	--------

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
Timer A: 32	CR1	ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
Timer B: 42	Reset Value	0	0	0	0	0	0	0	0
Timer A: 31	CR2	OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG
Timer B: 41	Reset Value	0	0	0	0	0	0	0	0
Timer A: 33	CSR	ICF1	OCF1	TOF	ICF2	OCF2	TIMD	-	-
Timer B: 43	Reset Value	х	х	х	х	х	0	х	х
Timer A: 34	IC1HR	MSB							LSB
Timer B: 44	Reset Value	х	х	х	х	х	х	х	х
Timer A: 35	IC1LR	MSB							LSB
Timer B: 45	Reset Value	х	х	х	х	х	х	х	х
Timer A: 36	OC1HR	MSB							LSB
Timer B: 46	Reset Value	1	0	0	0	0	0	0	0
Timer A: 37	OC1LR	MSB							LSB
Timer B: 47	Reset Value	0	0	0	0	0	0	0	0
Timer A: 3E	OC2HR	MSB							LSB
Timer B: 4E	Reset Value	1	0	0	0	0	0	0	0
Timer A: 3F	OC2LR	MSB							LSB
Timer B: 4F	Reset Value	0	0	0	0	0	0	0	0
Timer A: 38	CHR	MSB							LSB
Timer B: 48	Reset Value	1	1	1	1	1	1	1	1
Timer A: 39	CLR	MSB							LSB
Timer B: 49	Reset Value	1	1	1	1	1	1	0	0
Timer A: 3A	ACHR	MSB							LSB
Timer B: 4A	Reset Value	1	1	1	1	1	1	1	1
Timer A: 3B	ACLR	MSB							LSB
Timer B: 4B	Reset Value	1	1	1	1	1	1	0	0
Timer A: 3C	IC2HR	MSB							LSB
Timer B: 4C	Reset Value	х	х	х	Х	Х	Х	Х	Х
Timer A: 3D	IC2LR	MSB							LSB
Timer B: 4D	Reset Value	х	х	х	х	х	х	х	х

Related Documentation

<u>ل</u>رک

AN 973: SCI software communications using 16bit timer

AN 974: Real Time Clock with ST7 Timer Output Compare

AN 976: Driving a buzzer through the ST7 Timer PWM function

AN1041: Using ST7 PWM signal to generate analog input (sinusoid)

AN1046: UART emulation software

AN1078: PWM duty cycle switch implementing true 0 or 100 per cent duty cycle

AN1504: Starting a PWM signal directly at high level using the ST7 16-Bit timer

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 60. SCI Block Diagram



Δ7/

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.6.4.7 Parity Control

Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the PCE bit in the SCICR1 register. Depending on the frame length defined by the M bit, the possible SCI frame formats are as listed in Table 1.

Table 21. Frame Formats	Table	21.	Frame	Formats
-------------------------	-------	-----	-------	---------

M bit	PCE bit	SCI frame
0	0	SB 8 bit data STB
0	1	SB 7-bit data PB STB
1	0	SB 9-bit data STB
1	1	SB 8-bit data PB STB

Legend: SB = Start Bit, STB = Stop Bit,

PB = Parity Bit

Note: In case of wake up by an address mark, the MSB bit of the data is taken into account and not the parity bit

Even parity: the parity bit is calculated to obtain an even number of "1s" inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit is 0 if even parity is selected (PS bit = 0).

Odd parity: the parity bit is calculated to obtain an odd number of "1s" inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit is 1 if odd parity is selected (PS bit = 1).

Transmission mode: If the PCE bit is set then the MSB bit of the data written in the data register is not transmitted but is changed by the parity bit.

Reception mode: If the PCE bit is set then the interface checks if the received data byte has an

even number of "1s" if even parity is selected (PS = 0) or an odd number of "1s" if odd parity is selected (PS = 1). If the parity check fails, the PE flag is set in the SCISR register and an interrupt is generated if PIE is set in the SCICR1 register.

10.6.4.8 SCI Clock Tolerance

During reception, each bit is sampled 16 times. The majority of the 8th, 9th and 10th samples is considered as the bit value. For a valid bit detection, all the three samples should have the same value otherwise the noise flag (NF) is set. For example: If the 8th, 9th and 10th samples are 0, 1 and 1 respectively, then the bit value is "1", but the Noise Flag bit is set because the three samples values are not the same.

Consequently, the bit length must be long enough so that the 8th, 9th and 10th samples have the desired bit value. This means the clock frequency should not vary more than 6/16 (37.5%) within one bit. The sampling clock is resynchronized at each start bit, so that when receiving 10 bits (one start bit, 1 data byte, 1 stop bit), the clock deviation must not exceed 3.75%.

Note: The internal sampling clock of the microcontroller samples the pin value on every falling edge. Therefore, the internal sampling clock and the time the application expects the sampling to take place may be out of sync. For example: If the baud rate is 15.625 Kbaud (bit length is 64µs), then the 8th, 9th and 10th samples are at 28µs, 32µs and 36µs respectively (the first sample starting ideally at 0µs). But if the falling edge of the internal clock occurs just before the pin value changes, the samples would then be out of sync by ~4us. This means the entire bit length must be at least 40µs (36µs for the 10th sample + 4µs for synchronization with the internal sampling clock).

I²C BUS INTERFACE (Cont'd)

10.7.4 Functional Description

Refer to the CR, SR1 and SR2 registers in Section 10.7.7. for the bit definitions.

By default the I²C interface operates in Slave mode (M/SL bit is cleared) except when it initiates a transmit or receive sequence.

First the interface frequency must be configured using the FRi bits in the OAR2 register.

10.7.4.1 Slave Mode

As soon as a start condition is detected, the address is received from the SDA line and sent to the shift register; then it is compared with the address of the interface or the General Call address (if selected by software).

Note: In 10-bit addressing mode, the comparison includes the header sequence (11110xx0) and the two most significant bits of the address.

Header matched (10-bit mode only): the interface generates an acknowledge pulse if the ACK bit is set.

Address not matched: the interface ignores it and waits for another Start condition.

Address matched: the interface generates in sequence:

- Acknowledge pulse if the ACK bit is set.
- EVF and ADSL bits are set with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register, **holding the SCL line low** (see Figure 66 Transfer sequencing EV1).

Next, in 7-bit mode read the DR register to determine from the least significant bit (Data Direction Bit) if the slave must enter Receiver or Transmitter mode.

In 10-bit mode, after receiving the address sequence the slave is always in receive mode. It will enter transmit mode on receiving a repeated Start condition followed by the header sequence with matching address bits and the least significant bit set (11110xx1).

Slave Receiver

57

Following the address reception and after SR1 register has been read, the slave receives bytes from the SDA line into the DR register via the internal shift register. After each byte the interface generates in sequence:

- Acknowledge pulse if the ACK bit is set
- EVF and BTF bits are set with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR1 register followed by a read of the DR register, **holding the SCL line low** (see Figure 66 Transfer sequencing EV2).

Slave Transmitter

Following the address reception and after SR1 register has been read, the slave sends bytes from the DR register to the SDA line via the internal shift register.

The slave waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see Figure 66 Transfer sequencing EV3).

When the acknowledge pulse is received:

 The EVF and BTF bits are set by hardware with an interrupt if the ITE bit is set.

Closing slave communication

After the last data byte is transferred a Stop Condition is generated by the master. The interface detects this condition and sets:

EVF and STOPF bits with an interrupt if the ITE bit is set.

Then the interface waits for a read of the SR2 register (see Figure 66 Transfer sequencing EV4). Error Cases

 BERR: Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and the BERR bits are set with an interrupt if the ITE bit is set.

If it is a Stop then the interface discards the data, released the lines and waits for another Start condition.

If it is a Start then the interface discards the data and waits for the next slave address on the bus.

 AF: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set with an interrupt if the ITE bit is set.

The AF bit is cleared by reading the I2CSR2 register. However, if read before the completion of the transmission, the AF flag will be set again, thus possibly generating a new interrupt. Software must ensure either that the SCL line is back at 0 before reading the SR2 register, or be able to correctly handle a second interrupt during the 9th pulse of a transmitted byte.

Note: In case of errors, SCL line is not held low; however, the SDA line can remain low if the last bits transmitted are all 0. While AF=1, the SCL line may be held low due to SB or BTF flags that are set at the same time. It is then necessary to release both lines by software.

117/193

11 INSTRUCTION SET

11.1 CPU ADDRESSING MODES

The CPU features 17 different addressing modes which can be classified in seven main groups:

Addressing Mode	Example
Inherent	nop
Immediate	ld A,#\$55
Direct	ld A,\$55
Indexed	ld A,(\$55,X)
Indirect	ld A,([\$55],X)
Relative	jrne loop
Bit operation	bset byte,#5

The CPU Instruction set is designed to minimize the number of bytes required per instruction: To do

Table 26. CPU Addressing Mode Overview

so, most of the addressing modes may be subdivided in two submodes called long and short:

- Long addressing mode is more powerful because it can use the full 64 Kbyte address space, however it uses more bytes and more CPU cycles.
- Short addressing mode is less powerful because it can generally only access page zero (0000h -00FFh range), but the instruction size is more compact, and faster. All memory to memory instructions use short addressing modes only (CLR, CPL, NEG, BSET, BRES, BTJT, BTJF, INC, DEC, RLC, RRC, SLL, SRL, SRA, SWAP)

The ST7 Assembler optimizes the use of long and short addressing modes.

Mode		Syntax	Destination	Pointer Address (Hex.)	Pointer Size (Hex.)	Length (Bytes)	
Inherent			nop				+ 0
Immediate			ld A,#\$55				+ 1
Short	Direct		ld A,\$10	00FF			+ 1
Long	Direct		ld A,\$1000	0000FFFF			+ 2
No Offset	Direct	Indexed	ld A,(X)	00FF			+ 0
Short	Direct	Indexed	ld A,(\$10,X)	001FE			+ 1
Long	Direct	Indexed	ld A,(\$1000,X)	0000FFFF			+ 2
Short	Indirect		ld A,[\$10]	00FF	00FF	byte	+ 2
Long	Indirect		ld A,[\$10.w]	0000FFFF	00FF	word	+ 2
Short	Indirect	Indexed	ld A,([\$10],X)	001FE	00FF	byte	+ 2
Long	Indirect	Indexed	ld A,([\$10.w],X)	0000FFFF	00FF	word	+ 2
Relative	Direct		jrne loop	PC+/-127			+ 1
Relative	Indirect		jrne [\$10]	PC+/-127	00FF	byte	+ 2
Bit	Direct		bset \$10,#7	00FF			+ 1
Bit	Indirect		bset [\$10],#7	00FF	00FF	byte	+ 2
Bit	Direct	Relative	btjt \$10,#7,skip	00FF			+ 2
Bit	Indirect	Relative	btjt [\$10],#7,skip	00FF	00FF	byte	+ 3

12.4 SUPPLY CURRENT CHARACTERISTICS

The following current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To get the total device consumption, the two current values must be added (except for HALT mode for which the clock is stopped).

12.4.1 CURRENT CONSUMPTION

Symbol	Deremeter	Conditiono	Flash I	Devices	ROM Devices		llmit	
Symbol	Farameter	Conditions	Тур	Max ¹⁾	Тур	Max ¹⁾	Unit	
	Supply current in RUN mode ²⁾	$\begin{array}{l} f_{OSC}{=}2MHz, f_{CPU}{=}1MHz\\ f_{OSC}{=}4MHz, f_{CPU}{=}2MHz\\ f_{OSC}{=}8MHz, f_{CPU}{=}4MHz\\ f_{OSC}{=}16MHz, f_{CPU}{=}8MHz \end{array}$	1.3 2.0 3.6 7.1	3.0 5.0 8.0 15.0	1.3 2.0 3.6 7.1	2.0 3.0 5.0 10.0	mA	
	Supply current in SLOW mode ²⁾	$\begin{array}{l} f_{OSC}{=}2MHz, f_{CPU}{=}62.5 \text{kHz} \\ f_{OSC}{=}4MHz, f_{CPU}{=}125 \text{kHz} \\ f_{OSC}{=}8MHz, f_{CPU}{=}250 \text{kHz} \\ f_{OSC}{=}16MHz, f_{CPU}{=}500 \text{kHz} \end{array}$	600 700 800 1100	2700 3000 3600 4000	600 700 800 1100	1800 2100 2400 3000	μA	
I _{DD}	Supply current in WAIT mode ²⁾	$\begin{array}{l} f_{OSC}{=}2MHz, f_{CPU}{=}1MHz\\ f_{OSC}{=}4MHz, f_{CPU}{=}2MHz\\ f_{OSC}{=}8MHz, f_{CPU}{=}4MHz\\ f_{OSC}{=}16MHz, f_{CPU}{=}8MHz \end{array}$	1.0 1.5 2.5 4.5	3.0 4.0 5.0 7.0	1.0 1.5 2.5 4.5	1.3 2.0 3.3 6.0	mA	
	Supply current in SLOW WAIT mode ²⁾	$\begin{array}{l} f_{OSC}{=}2MHz, f_{CPU}{=}62.5 \text{kHz} \\ f_{OSC}{=}4MHz, f_{CPU}{=}125 \text{kHz} \\ f_{OSC}{=}8MHz, f_{CPU}{=}250 \text{kHz} \\ f_{OSC}{=}16MHz, f_{CPU}{=}500 \text{kHz} \end{array}$	580 650 770 1050	1200 1300 1800 2000	70 100 200 350	200 300 600 1200	μA	
	Supply current in HALT mode ³⁾	-40°C≤T _A ≤+85°C	<1	10	<1	10	uА	
		-40°C≤T _A ≤+125°C	<1	50	<1	50	μΛ	
I _{DD}	Supply current in ACTIVE-HALT mode	$f_{OSC}=2MHz$ $f_{OSC}=4MHz$ $f_{OSC}=8MHz$ $f_{OSC}=16MHz$	80 160 325 650	No max. guaran- teed	80 160 325 650	No max. guar- anteed	μA	

Notes:

1. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.

2. Measurements are done in the following conditions:

- Program executed from RAM, CPU running with RAM access. The increase in consumption when executing from Flash is 50%.

- All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load)

- All peripherals in reset state.
- LVD disabled.
- Clock input (OSC1) driven by external square wave.
- In SLOW and SLOW WAIT mode, f_{CPU} is based on f_{OSC} divided by 32.
- To obtain the total current consumption of the device, add the clock source (Section 12.4.2) and the peripheral power consumption (Section 12.4.3).
- 3. All I/O pins in push-pull 0 mode (when applicable) with a static value at V_{DD} or VSS (no load), LVD disabled. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
- 4. Data based on characterisation results, not tested in production. All I/O pins in push-pull 0 mode (when applicable) with a static value at V_{DD} or V_{SS} (no load); clock input (OSC1) driven by external square wave, LVD disabled. To obtain the total current consumption of the device, add the clock source consumption (Section 12.4.2).



ST72321 DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

14.2 DEVICE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE

Customer code is made up of the ROM/FAS-TROM contents and the list of the selected options (if any). The ROM/FASTROM contents are to be sent on diskette, or by electronic means, with the S19 hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected options are communicated to STMicroelectronics using the correctly completed OPTION LIST appended.

57

Refer to application note AN1635 for information on the counter listing returned by ST after code has been transferred. The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.

Caution: The Readout Protection binary value is inverted between ROM and FLASH products. The option byte checksum will differ between ROM and FLASH.

- The interrupt flag is cleared within any interrupt routine with higher or identical priority level
- The interrupt flag is cleared in any part of the code while this interrupt is disabled

47/

If these conditions are not met, the symptom can be avoided by implementing the following sequence: PUSH CC SIM reset interrupt flag POP CC