E·XFL



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	SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
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/st72f32aj1t6

Email: info@E-XFL.COM

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

Table of Contents

9 I,	/O P	ORTS	42
	9.1		42
	9.2	FUNCTIONAL DESCRIPTION	42
		9.2.1 Input Modes	42
		9.2.2 Output Modes	42
		9.2.3 Alternate Functions	42
	9.3	I/O PORT IMPLEMENTATION	45
	9.4	LOW POWER MODES	45
	9.5	INTERRUPTS	45
		9.5.1 I/O Port Implementation	46
10	ON-	CHIP PERIPHERALS	48
-	10.1	WATCHDOG TIMER (WDG)	48
		10.1.1 Introduction	48
		10.1.2 Main Features	48
		10.1.3 Functional Description	48
		10.1.4 How to Program the Watchdog Timeout	49
		10.1.5 Low Power Modes	51
		10.1.6 Hardware Watchdog Option	51
		10.1.7 Using Halt Mode with the WDG (WDGHALT option)	51
		10.1.8 Interrupts	51
		10.1.9 Register Description	51
	10.2	2 MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK AND BEEPER (MCC/RTC) .	53
		10.2.1 Programmable CPU Clock Prescaler	53
		10.2.2 Clock-out Capability	53
		10.2.3 Real Time Clock Timer (RTC)	53
		10.2.4 Beeper	53
		10.2.5 Low Power Modes	54
		10.2.6 Interrupts	54
		10.2.7 Register Description	54
	10.3	3 16-BIT TIMER	56
		10.3.1 Introduction	56
		10.3.2 Main Features	56
		10.3.3 Functional Description	56
		10.3.4 Low Power Modes	68
		10.3.5 Interrupts	68
		10.3.6 Summary of Timer modes	68
		10.3.7 Register Description	69
	10.4	SERIAL PERIPHERAL INTERFACE (SPI)	76
		10.4.1 Introduction	76
		10.4.2 Main Features	76
		10.4.3 General Description	76
		10.4.4 Clock Phase and Clock Polarity	80
		10.4.5 Error Flags	81
		10.4.6 Low Power Modes	83
		10.4.7 Interrupts	83
	10 -		84
	10.5	SERIAL COMMUNICATIONS INTERFACE (SCI)	87

Pin n°					Level		Port						Main			
944	42	32	32	Pin Name	ype	Ŧ	out		Inp	out		Out	tput	function	Alternate	Function
TQFF	SDIP	TQFF	SDIP		F	Inpi	Outp	float	ndw	int	ana	OD	РР	reset)	reset)	
24	17	9	12	PC1/OCMP1_B/ AIN13	I/O	С _т		x	х		х	х	x	Port C1	Timer B Out- put Com- pare 1	ADC Analog Input 13
25	18	10	13	PC2 (HS)/ICAP2_B	I/O	C_{T}	HS	Х	Х			Х	Х	Port C2	Timer B Input	t Capture 2
26	19	11	14	PC3 (HS)/ICAP1_B	I/O	C_T	HS	X	Х			Х	Х	Port C3	Timer B Input	t Capture 1
27	20	12	15	PC4/MISO/ICCDA- TA	I/O	CT		x	х			х	x	Port C4	SPI Master In / Slave Out Data	ICC Data In- put
28	21	13	16	PC5/MOSI/AIN14	I/O	CT		x	х		х	х	х	Port C5	SPI Master Out / Slave In Data	ADC Analog Input 14
29	22	14	17	PC6/SCK/ICCCLK	I/O	CT		x	Х			х	х	Port C6	SPI Serial Clock	ICC Clock Output
30	23	15	18	PC7/SS/AIN15	I/O	CT		x	х		х	х	х	Port C7	SPI Slave Select (ac- tive low)	ADC Analog Input 15
31	24	16	19	PA3 (HS)	I/O	C_T	HS	Х		ei0		Х	Х	Port A3		
32	25			V _{DD_1}	S									Digital M	ain Supply Vol	tage
33	26			V _{SS_1}	S									Digital G	round Voltage	
34	27	17	20	PA4 (HS)	I/O	C_T	HS	X	Х			Х	Х	Port A4		
35	28			PA5 (HS)	I/O	C_T	HS	Х	Х			Х	Х	Port A5		
36	29	18	21	PA6 (HS)	I/O	C_T	HS	Х				Т		Port A6 ¹)	
37	30	19	22	PA7 (HS)	I/O	C_T	HS	Х				Т		Port A7 ¹)	
38	31	20	23	V _{PP} /ICCSEL	I									Must be tied low. In the flash pro- gramming mode, this pin acts as the programming voltage input V_{PP} . See Section 12.9.2 for more details. High voltage must not be applied to ROM devices.		
39	32	21	24	RESET	I/O	C_T								Top prior	ity non maskal	ole interrupt.
40	33	22	25	V _{SS_2}	S									Digital G	round Voltage	
41	34	23	26	OSC2	0									Resonate	or oscillator inv	erter output
42	35	24	27	OSC1	I									External cillator in	clock input or I verter input	Resonator os-
43	36	25	28	V _{DD_2}	S									Digital M	ain Supply Vol	tage
44	37	26	29	PE0/TDO	I/O	C_T		Х	Х			Х	Х	Port E0	SCI Transmit	Data Out
1	38	27	30	PE1/RDI	I/O	C_{T}		Х	Х			Х	Х	Port E1	SCI Receive	Data In
2	39	28	31	PB0	I/O	CT		x	е	i2		х	x	Port B0	B0 Caution: Negative cur- rent injection not al- lowed on this pin ⁵⁾	
3	40			PB1	I/O	C_T		X	е	i2		Х	Х	Port B1		
4	41			PB2	I/O	C_{T}		Х	е	i2		Х	Х	Port B2		
5	42	29	32	PB3	I/O	C_{T}		Х		ei2		Х	Х	Port B3		

CENTRAL PROCESSING UNIT (Cont'd)

Condition Code Register (CC)

Read/Write

Reset Value: 111x1xxx



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 Bits

Bit $4 = \mathbf{H}$ 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 7^{th} bit.

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

- 1: The result of the last operation is negative
- (i.e. 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 Bits

Bit 5,3 = 11, 10 Interrupt

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

Interrupt Software Priority	11	10
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.

RESET SEQUENCE MANAGER (Cont'd)

The RESET pin is an asynchronous signal which plays a major role in EMS performance. In a noisy environment, it is recommended to follow the guidelines mentioned in the electrical characteristics section.

6.3.3 External Power-On RESET

To start up the microcontroller correctly, the user must ensure by means of an external reset circuit that the reset signal is held low until V_{DD} is over the minimum level specified for the selected f_{OSC} frequency.

A proper reset signal for a slow rising V_{DD} supply can generally be provided by an external RC network connected to the RESET pin.

6.3.4 Internal Watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow is shown in Figure 14.

Starting from the Watchdog counter underflow, the device RESET pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.



Figure 14. RESET Sequences

6.4 SYSTEM INTEGRITY MANAGEMENT

6.4.1 Register Description

SYSTEM INTEGRITY (SI) CONTROL/STATUS REGISTER (SICSR)

Read/Write

Reset Value: 0000 000x (00h)

7							0
0	0	0	0	0	0	0	WDG RF

Bits 7:1 = 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).

7 INTERRUPTS

7.1 INTRODUCTION

The ST7 enhanced interrupt management provides the following features:

- Hardware interrupts
- Software interrupt (TRAP)
- Nested or concurrent interrupt management with flexible interrupt priority and level management:
 - Up to 4 software programmable nesting levels
 - Up to 16 interrupt vectors fixed by hardware
- 2 non maskable events: RESET, TRAP

This interrupt management is based on:

- Bit 5 and bit 3 of the CPU CC register (I1:0),
- Interrupt software priority registers (ISPRx),
- Fixed interrupt vector addresses located at the high addresses of the memory map (FFE0h to FFFFh) sorted by hardware priority order.

This enhanced interrupt controller guarantees full upward compatibility with the standard (not nested) ST7 interrupt controller.

7.2 MASKING AND PROCESSING FLOW

The interrupt masking is managed by the I1 and I0 bits of the CC register and the ISPRx registers which give the interrupt software priority level of each interrupt vector (see Table 6). The processing flow is shown in Figure 15

Figure 15. Interrupt Processing Flowchart

When an interrupt request has to be serviced:

- Normal processing is suspended at the end of the current instruction execution.
- The PC, X, A and CC registers are saved onto the stack.
- I1 and I0 bits of CC register are set according to the corresponding values in the ISPRx registers of the serviced interrupt vector.
- The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to "Interrupt Mapping" table for vector addresses).

The interrupt service routine should end with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

Note: As a consequence of the IRET instruction, the I1 and I0 bits will be restored from the stack and the program in the previous level will resume.

Table 6. Interrupt Software Priority Levels

Interrupt software priority	Level	l1	10
Level 0 (main)	Low	1	0
Level 1		0	1
Level 2	•	0	0
Level 3 (= interrupt disable)	High	1	1



INTERRUPTS (Cont'd)

7.5 INTERRUPT REGISTER DESCRIPTION

CPU CC REGISTER INTERRUPT BITS

Read/Write

Reset Value: 111x 1010 (xAh)

7							0
1	1	11	н	10	Ν	Z	С

Bit 5, 3 = 11, 10 Software Interrupt Priority

These two bits indicate the current interrupt software priority.

Interrupt Software Priority	Level	1	10
Level 0 (main)	Low	1	0
Level 1		0	1
Level 2	🔸	0	0
Level 3 (= interrupt disable*)	High	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 (ISPRx).

They can be also set/cleared by software with the RIM, SIM, HALT, WFI, IRET and PUSH/POP instructions (see "Interrupt Dedicated Instruction Set" table).

*Note: TRAP and RESET events can interrupt a level 3 program.

INTERRUPT SOFTWARE PRIORITY REGISTERS (ISPRX)

Read/Write (bit 7:4 of **ISPR3** are read only) Reset Value: 1111 1111 (FFh)

	1							0
ISPR0	l1_3	10_3	l1_2	10_2	11_1	10_1	l1_0	10_0
ISPR1	11_7	10_7	l1_6	I0_6	l1_5	10_5	11_4	I0_4
ISPR2	11_11	10_11	11_10	10_10	l1_9	10_9	l1_8	10_8
ISPR3	1	1	1	1	11_13	10_13	11_12	10_12

These four registers contain the interrupt software priority of each interrupt vector.

 Each interrupt vector (except RESET and TRAP) has corresponding bits in these registers where its own software priority is stored. This correspondance is shown in the following table.

Vector address	ISPRx bits
FFFBh-FFFAh	11_0 and 10_0 bits*
FFF9h-FFF8h	I1_1 and I0_1 bits
FFE1h-FFE0h	I1_13 and I0_13 bits

Each I1_x and I0_x bit value in the ISPRx registers has the same meaning as the I1 and I0 bits in the CC register.

 Level 0 can not be written (l1_x=1, l0_x=0). In this case, the previously stored value is kept. (example: previous=CFh, write=64h, result=44h)

The RESET, and TRAP vectors have no software priorities. When one is serviced, the I1 and I0 bits of the CC register are both set.

Caution: If the $I1_x$ and $I0_x$ bits are modified while the interrupt x is executed the following behaviour has to be considered: If the interrupt x is still pending (new interrupt or flag not cleared) and the new software priority is higher than the previous one, the interrupt x is re-entered. Otherwise, the software priority stays unchanged up to the next interrupt request (after the IRET of the interrupt x).

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 MCCSR register).

MCCSR 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 (MCCSR) is set (see Section 10.2 on page 53 for more details on the MCCSR register).

The MCU can exit ACTIVE-HALT mode on reception of either an MCC/RTC interrupt, a specific interrupt (see Table 8, "Interrupt Mapping," on page 33) 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 24). 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 (MCCSR.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 interrupt, OIE bit of MCCSR register must not be cleared before t_{DELAY} after the interrupt occurs (t_{DELAY} = 256 or 4096 t_{CPU} delay de-

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

Figure 23. ACTIVE-HALT Timing Overview



Figure 24. 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. Only the MCC/RTC interrupt and some specific interrupts can exit the MCU from ACTIVE-HALT mode (such as external interrupt). Refer to Table 8, "Interrupt Mapping," on page 33 for more details.

4. 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.

57

16-BIT TIMER (Cont'd)

10.3.3.4 Output Compare

In this section, the index, *i*, may be 1 or 2 because there are 2 output compare functions in the 16-bit timer.

This function can be used to control an output waveform or indicate when a period of time has elapsed.

When a match is found between the Output Compare register and the free running counter, the output compare function:

- Assigns pins with a programmable value if the OCiE bit is set
- Sets a flag in the status register
- Generates an interrupt if enabled

Two 16-bit registers Output Compare Register 1 (OC1R) and Output Compare Register 2 (OC2R) contain the value to be compared to the counter register each timer clock cycle.

	MS Byte	LS Byte
OC <i>i</i> R	OC <i>i</i> HR	OC <i>i</i> LR

These registers are readable and writable and are not affected by the timer hardware. A reset event changes the OC_iR value to 8000h.

Timing resolution is one count of the free running counter: $(f_{CPU/CC[1:0]})$.

Procedure:

To use the output compare function, select the following in the CR2 register:

- Set the OC*i*E bit if an output is needed then the OCMP*i* pin is dedicated to the output compare *i* signal.
- Select the timer clock (CC[1:0]) (see Table 16 Clock Control Bits).

And select the following in the CR1 register:

- Select the OLVL*i* bit to applied to the OCMP*i* pins after the match occurs.
- Set the OCIE bit to generate an interrupt if it is needed.

When a match is found between OCRi register and CR register:

- OCFi bit is set.

- The OCMP*i* pin takes OLVL*i* bit value (OCMP*i* pin latch is forced low during reset).
- A timer interrupt is generated if the OCIE bit is set in the CR1 register and the I bit is cleared in the CC register (CC).

The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$\Delta \text{ OC} i \text{R} = \frac{\Delta t * f_{\text{CPU}}}{\text{PRESC}}$$

Where:

- Δt = Output compare period (in seconds)
- $f_{CPU} = CPU$ clock frequency (in hertz)
- PRESC = Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits, see Table 16 Clock Control Bits)

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

$$\Delta \text{ OC} i \mathbb{R} = \Delta t \star f_{\text{EXT}}$$

Where:

 Δt = Output compare period (in seconds)

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

Clearing the output compare interrupt request (i.e. clearing the OCF*i* bit) is done by:

- 1. Reading the SR register while the OCF*i* bit is set.
- 2. An access (read or write) to the OCiLR register.

The following procedure is recommended to prevent the OCF*i* bit from being set between the time it is read and the write to the OC*i*R register:

- Write to the OC*i*HR register (further compares are inhibited).
- Read the SR register (first step of the clearance of the OCF*i* bit, which may be already set).
- Write to the OC*i*LR register (enables the output compare function and clears the OCF*i* bit).



16-BIT TIMER (Cont'd)

Figure 40. Output Compare Timing Diagram, f_{TIMER} =f_{CPU}/2

INTERNAL CPU CLOCK TIMER CLOCK COUNTER REGISTER OUTPUT COMPARE REGISTER <i>i</i> (OCR <i>i</i>)	2ECF 2ED0 2ED1 2ED2 2ED3 2ED4 2ECF 2ED0 2ED1 2ED2 2ED3 2ED4
OUTPUT COMPARE FLAG i (OCFi)	
OCMP <i>i</i> PIN (OLVL <i>i</i> =1)	

Figure 41. Output Compare Timing Diagram, $f_{TIMER} = f_{CPU}/4$

INTERNAL CPU CLOCK TIMER CLOCK COUNTER REGISTER OUTPUT COMPARE REGISTER <i>i</i> (OCR <i>i</i>) COMPARE REGISTER <i>i</i> LATCH OUTPUT COMPARE FLAG <i>i</i> (OCF <i>i</i>)	
OCMP <i>i</i> PIN (OLVL <i>i</i> =1)	



SERIAL COMMUNICATIONS INTERFACE (Cont'd)

Figure 51. SCI Block Diagram



Δ7/

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.5.4 Functional Description

The block diagram of the Serial Control Interface, is shown in Figure 51. It contains 6 dedicated registers:

- Two control registers (SCICR1 & SCICR2)
- A status register (SCISR)

5/

- A baud rate register (SCIBRR)
- An extended prescaler receiver register (SCIER-PR)
- An extended prescaler transmitter register (SCI-ETPR)

Refer to the register descriptions in Section 10.5.7for the definitions of each bit.

10.5.4.1 Serial Data Format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the SCICR1 register (see Figure 51).

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

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

An Idle character is interpreted as an entire frame of "1"s followed by the start bit of the next frame which contains data.

A Break character is interpreted on receiving "0"s for some multiple of the frame period. At the end of the last break frame the transmitter inserts an extra "1" bit to acknowledge the start bit.

Transmission and reception are driven by their own baud rate generator.



Figure 52. Word Length Programming

SERIAL COMMUNICATIONS INTERFACE (Cont'd) CONTROL REGISTER 1 (SCICR1)

Read/Write

Reset Value: x000 0000 (x0h)

7							0
R8	Т8	SCID	М	WAKE	PCE	PS	PIE

Bit 7 = **R8** Receive data bit 8.

This bit is used to store the 9th bit of the received word when M=1.

Bit 6 = T8 Transmit data bit 8.

This bit is used to store the 9th bit of the transmitted word when M=1.

Bit 5 = **SCID** *Disabled for low power consumption* When this bit is set the SCI prescalers and outputs are stopped and the end of the current byte transfer in order to reduce power consumption. This bit is set and cleared by software.

0: SCI enabled

1: SCI prescaler and outputs disabled

Bit $4 = \mathbf{M}$ Word length. This bit determines the word length. It is set or cleared by software.

0: 1 Start bit, 8 Data bits, 1 Stop bit

1: 1 Start bit, 9 Data bits, 1 Stop bit

Note: The M bit must not be modified during a data transfer (both transmission and reception).

Bit 3 = WAKE Wake-Up method.

This bit determines the SCI Wake-Up method, it is set or cleared by software. 0: Idle Line 1: Address Mark

Bit 2 = **PCE** Parity control enable.

This bit selects the hardware parity control (generation and detection). When the parity control is enabled, the computed parity is inserted at the MSB position (9th bit if M=1; 8th bit if M=0) and parity is checked on the received data. This bit is set and cleared by software. Once it is set, PCE is active after the current byte (in reception and in transmission).

0: Parity control disabled

1: Parity control enabled

Bit 1 = **PS** Parity selection.

This bit selects the odd or even parity when the parity generation/detection is enabled (PCE bit set). It is set and cleared by software. The parity will be selected after the current byte.

0: Even parity

1: Odd parity

Bit 0 = **PIE** *Parity interrupt enable.*

This bit enables the interrupt capability of the hardware parity control when a parity error is detected (PE bit set). It is set and cleared by software.

0: Parity error interrupt disabled

1: Parity error interrupt enabled.

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

10.6.6 Register Description

CONTROL/STATUS REGISTER (ADCCSR)

Read/Write (Except bit 7 read only)

Reset Value: 0000 0000 (00h)

7							0
EOC	SPEED	ADON	0	СНЗ	CH2	CH1	CH0

Bit 7 = **EOC** End of Conversion This bit is set by hardware. It is cleared by hardware when software reads the ADCDRH register or writes to any bit of the ADCCSR register. 0: Conversion is not complete 1: Conversion complete

Bit 6 = **SPEED** ADC clock selection This bit is set and cleared by software. 0: $f_{ADC} = f_{CPU}/4$ 1: $f_{ADC} = f_{CPU}/2$

Bit 5 = **ADON** A/D Converter on This bit is set and cleared by software. 0: Disable ADC and stop conversion 1: Enable ADC and start conversion

Bit 4 = **Reserved.** Must be kept cleared.

Bit 3:0 = CH[3:0] Channel Selection

These bits are set and cleared by software. They select the analog input to convert.

Channel Pin*	CH3	CH2	CH1	CH0
AIN0	0	0	0	0
AIN1	0	0	0	1
AIN2	0	0	1	0
AIN3	0	0	1	1
AIN4	0	1	0	0
AIN5	0	1	0	1
AIN6	0	1	1	0
AIN7	0	1	1	1
AIN8	1	0	0	0
AIN9	1	0	0	1
AIN10	1	0	1	0
AIN11	1	0	1	1
AIN12	1	1	0	0
AIN13	1	1	0	1
AIN14	1	1	1	0
AIN15	1	1	1	1

*The number of channels is device dependent. Refer to the device pinout description.

DATA REGISTER (ADCDRH)

Read Only Reset Value: 0000 0000 (00h)

7

D9	D8	D7	D6	D5	D4	D3	D2

Bit 7:0 = D[9:2] MSB of Converted Analog Value

DATA REGISTER (ADCDRL)

Read Only Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	D1	D0

Bit 7:2 = Reserved. Forced by hardware to 0.

Bit 1:0 = **D[1:0]** LSB of Converted Analog Value

0

INSTRUCTION SET OVERVIEW (Cont'd)

11.1.6 Indirect Indexed (Short, Long)

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

The indirect indexed addressing mode consists of two sub-modes:

Indirect Indexed (Short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - 1FE addressing space, and requires 1 byte after the opcode.

Indirect Indexed (Long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

Table25. InstructionsSupportingDirect,Indexed,IndirectandIndirectIndexedAddressing Modes

Long and Short Instructions	Function		
LD	Load		
СР	Compare		
AND, OR, XOR	Logical Operations		
ADC, ADD, SUB, SBC	Arithmetic Additions/Sub- stractions operations		
BCP	Bit Compare		

Short Instructions Only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
BSET, BRES	Bit Operations
BTJT, BTJF	Bit Test and Jump Opera- tions
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Opera- tions
SWAP	Swap Nibbles
CALL, JP	Call or Jump subroutine

57/

11.1.7 Relative mode (Direct, Indirect)

This addressing mode is used to modify the PC register value, by adding an 8-bit signed offset to it.

Available Relative Direct/Indirect Instructions	Function		
JRxx	Conditional Jump		
CALLR	Call Relative		

The relative addressing mode consists of two submodes:

Relative (Direct)

The offset is following the opcode.

Relative (Indirect)

The offset is defined in memory, which address follows the opcode.

INSTRUCTION SET OVERVIEW (Cont'd)

57

Mnemo	Description	Function/Example	Dst	Src		11	н	10	Ν	Z	С
ADC	Add with Carry	A=A+M+C	А	М			Н		Ν	Ζ	С
ADD	Addition	A = A + M	А	М			Н		Ν	Ζ	С
AND	Logical And	A = A . M	А	М					Ν	Ζ	
BCP	Bit compare A, Memory	tst (A . M)	А	М					Ν	Ζ	
BRES	Bit Reset	bres Byte, #3	М								
BSET	Bit Set	bset Byte, #3	М								
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	М								С
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	М								С
CALL	Call subroutine										
CALLR	Call subroutine relative										
CLR	Clear		reg, M						0	1	
СР	Arithmetic Compare	tst(Reg - M)	reg	М					Ν	Ζ	С
CPL	One Complement	A = FFH-A	reg, M						Ν	Ζ	1
DEC	Decrement	dec Y	reg, M						Ν	Ζ	
HALT	Halt					1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC				11	Н	10	Ν	Ζ	С
INC	Increment	inc X	reg, M						Ν	Ζ	
JP	Absolute Jump	jp [TBL.w]									
JRA	Jump relative always										
JRT	Jump relative										
JRF	Never jump	jrf *									
JRIH	Jump if ext. INT pin = 1	(ext. INT pin high)									
JRIL	Jump if ext. INT pin = 0	(ext. INT pin low)									
JRH	Jump if H = 1	H = 1 ?									
JRNH	Jump if H = 0	H = 0 ?									
JRM	Jump if I1:0 = 11	l1:0 = 11 ?									
JRNM	Jump if I1:0 <> 11	11:0 <> 11 ?									
JRMI	Jump if N = 1 (minus)	N = 1 ?									
JRPL	Jump if N = 0 (plus)	N = 0 ?									
JREQ	Jump if Z = 1 (equal)	Z = 1 ?									
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?									
JRC	Jump if C = 1	C = 1 ?									
JRNC	Jump if C = 0	C = 0 ?									
JRULT	Jump if C = 1	Unsigned <									
JRUGE	Jump if $C = 0$	Jmp if unsigned >=									
JRUGT	Jump if $(C + Z = 0)$	Unsigned >									

12 ELECTRICAL CHARACTERISTICS

12.1 PARAMETER CONDITIONS

Unless otherwise specified, all voltages are referred to $\mathrm{V}_{\mathrm{SS}}.$

12.1.1 Minimum and Maximum values

Unless otherwise specified the minimum and maximum values are guaranteed in the worst conditions of ambient temperature, supply voltage and frequencies by tests in production on 100% of the devices with an ambient temperature at $T_A=25^{\circ}C$ and $T_A=T_Amax$ (given by the selected temperature range).

Data based on characterization results, design simulation and/or technology characteristics are indicated in the table footnotes and are not tested in production. Based on characterization, the minimum and maximum values refer to sample tests and represent the mean value plus or minus three times the standard deviation (mean $\pm 3\Sigma$).

12.1.2 Typical values

Unless otherwise specified, typical data are based on $T_A=25^{\circ}$ C, $V_{DD}=5$ V. They are given only as design guidelines and are not tested.

12.1.3 Typical curves

57/

Unless otherwise specified, all typical curves are given only as design guidelines and are not tested.

12.1.4 Loading capacitor

The loading conditions used for pin parameter measurement are shown in Figure 56.

Figure 56. Pin loading conditions



12.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in Figure 57.

Figure 57. Pin input voltage



13 PACKAGE CHARACTERISTICS

13.1 PACKAGE MECHANICAL DATA

5/

Figure 84. 32-Pin Thin Quad Flat Package



Figure 85. 32-Pin Plastic Dual In-Line Package, Shrink 400-mil Width



ST7232A DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

14.2 ROM DEVICE ORDERING INFORMATION AND TRANSFER OF CUSTOMER CODE

ROM devices can be ordered in any combination of memory size and temperature range with the types given in Figure 89 and by completing the option list on the next page.

ROM customer code is made up of the ROM contents and the list of the selected options (if any). The ROM contents are to be sent with the S19 hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

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.



Figure 89. ROM Factory Coded Device Types

DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

14.3 VERSION-SPECIFIC SALES CONDITIONS

To satisfy the different customer requirements and to ensure that ST Standard Microcontrollers will consistently meet or exceed the expectations of each Market Segment, the Codification System for Standard Microcontrollers clearly distinguishes products intended for use in automotive environments, from products intended for use in non-automotive environments.

It is the responsibility of the Customer to select the appropriate product for his application.

14.4 FLASH DEVICE ORDERING INFORMATION

Part Number	Version	Package	Flash Memory (KBytes)	Temp. Range
ST72F32AK2TA		TQFP32		10°C + 95°C
ST72F32AJ2TA		TQFP44		-40 C +85 C
ST72F32AK2TB	Automotivo	TQFP32	o	40°C + 105°C
ST72F32AJ2TB	Automotive	TQFP44	- O	-40 C +105 C
ST72F32AK2TC	1	TQFP32		40°C + 105°C
ST72F32AJ2TC	1	TQFP44		-40 0 +125 0
ST72F32AK1B5			4	10%0 + 85%0
ST72F32AK2B5		6DID22	8	-10 C +05 C
ST72F32AK1B6		5DIF32	4	
ST72F32AK2B6			8	-40°C +85°C
ST72F32AJ1B5			4	10%0 + 85%0
ST72F32AJ2B5			8	-10 C +05 C
ST72F32AJ1B6		SDIF42	4	
ST72F32AJ2B6	Standard/		8	-40 C +65 C
ST72F32AK1T5	Industrial		4	10%0 + 85%0
ST72F32AK2T5		TOED20	8	-10°C +85°C
ST72F32AK1T6		IQFP32	4	40%0 + 85%0
ST72F32AK2T6			8	-40°C +85°C
ST72F32AJ1T5			4	10%0 + 85%0
ST72F32AJ2T5	1		8	-1010 +0010
ST72F32AJ1T6	1		4	
ST72F32AJ2T6			8	-40 0 +00 0

Table 27. ST72F32A Flash Order Codes

57/