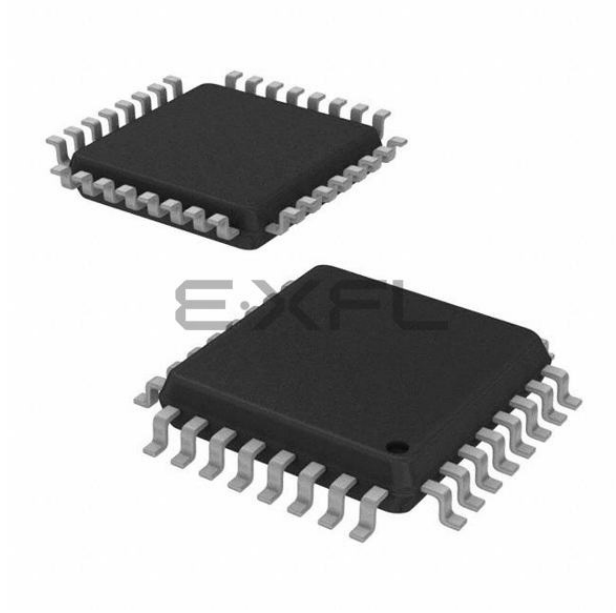


Welcome to E-XFL.COM

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.

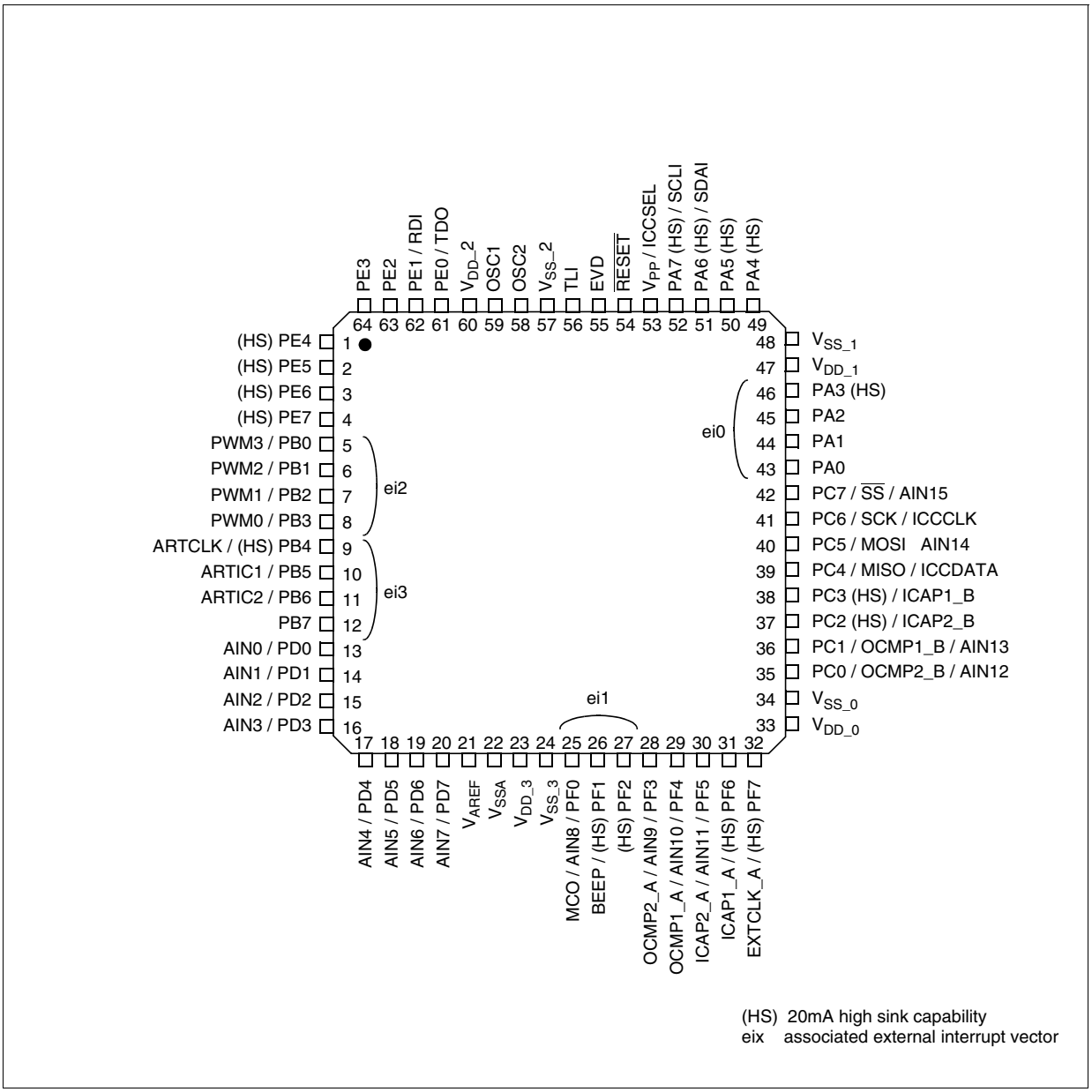
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f321bk6t6

Table of Contents

2 PIN DESCRIPTION

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



FLASH PROGRAM MEMORY (Cont'd)

4.5 ICP (In-Circuit Programming)

To perform ICP the microcontroller must be switched to ICC (In-Circuit Communication) mode by an external controller or programming tool.

Depending on the ICP code downloaded in RAM, Flash memory programming can be fully customized (number of bytes to program, program locations, or selection serial communication interface for downloading).

When using an STMicroelectronics or third-party programming tool that supports ICP and the specific microcontroller device, the user needs only to implement the ICP hardware interface on the application board (see Figure 7). For more details on the pin locations, refer to the device pinout description.

4.6 IAP (In-Application Programming)

This mode uses a BootLoader program previously stored in Sector 0 by the user (in ICP mode or by plugging the device in a programming tool).

This mode is fully controlled by user software. This allows it to be adapted to the user application, (user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored, etc.). For example, it is possible to download code from the SPI, SCI, USB or CAN interface and program it in the Flash. IAP mode can be used to program any of the Flash sectors except Sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

4.7 Related Documentation

For details on Flash programming and ICC protocol, refer to the ST7 Flash Programming Reference Manual and to the ST7 ICC Protocol Reference Manual.

4.7.1 Register Description

FLASH CONTROL/STATUS REGISTER (FCSR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	0	0

This register is reserved for use by Programming Tool software. It controls the Flash programming and erasing operations.

Figure 8. Flash Control/Status Register Address and Reset Value

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0029h	**FCSR** Reset Value	0	0	0	0	0	0	0	0

RESET SEQUENCE MANAGER (Cont'd)

The $\overline{\text{RESET}}$ 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.

If the external $\overline{\text{RESET}}$ pulse is shorter than $t_{w(RSTL)out}$ (see short ext. Reset in Figure 15), the signal on the $\overline{\text{RESET}}$ pin may be stretched. Otherwise the delay will not be applied (see long ext. Reset in Figure 15). Starting from the external $\overline{\text{RESET}}$ pulse recognition, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.

6.3.3 External Power-On RESET

If the LVD is disabled by option byte, 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. (see "OPERATING CONDITIONS" on page 140)

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

6.3.4 Internal Low Voltage Detector (LVD) RESET

Two different RESET sequences caused by the internal LVD circuitry can be distinguished:

- Power-On RESET
- Voltage Drop RESET

The device $\overline{\text{RESET}}$ pin acts as an output that is pulled low when $V_{DD}<V_{IT+}$ (rising edge) or $V_{DD}<V_{IT-}$ (falling edge) as shown in Figure 15.

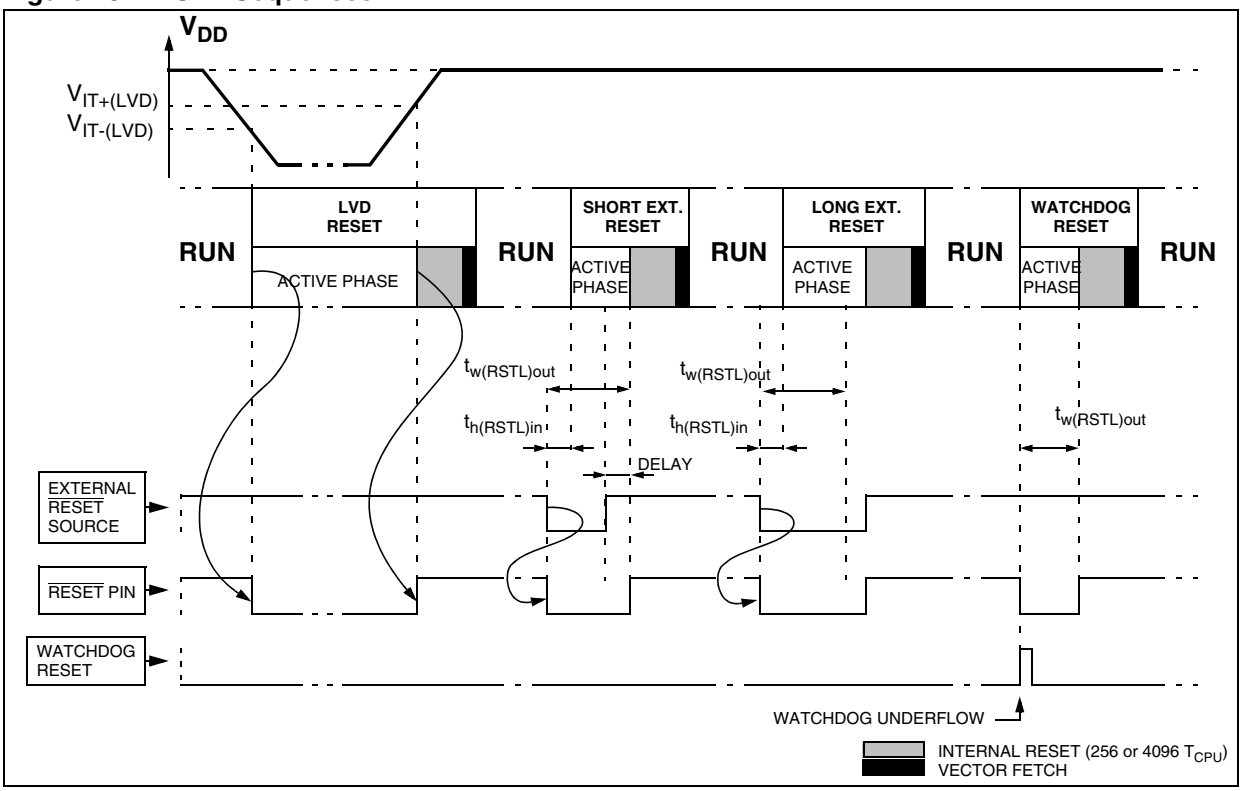
The LVD filters spikes on V_{DD} larger than $t_{g(VDD)}$ to avoid parasitic resets.

6.3.5 Internal Watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow is shown in Figure 15.

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

Figure 15. RESET Sequences



SYSTEM INTEGRITY MANAGEMENT (Cont'd)

6.4.2 Auxiliary Voltage Detector (AVD)

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

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

Caution: The AVD function is active only if the LVD is enabled through the option byte.

6.4.2.1 Monitoring the V_{DD} Main Supply

This mode is selected by clearing the AVDS bit in the SICSR register.

The AVD voltage threshold value is relative to the selected LVD threshold configured by option byte (see section 14.1 on page 174).

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

In the case of a drop in voltage, the AVD interrupt acts as an early warning, allowing software to shut down safely before the LVD resets the microcontroller. See Figure 17.

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

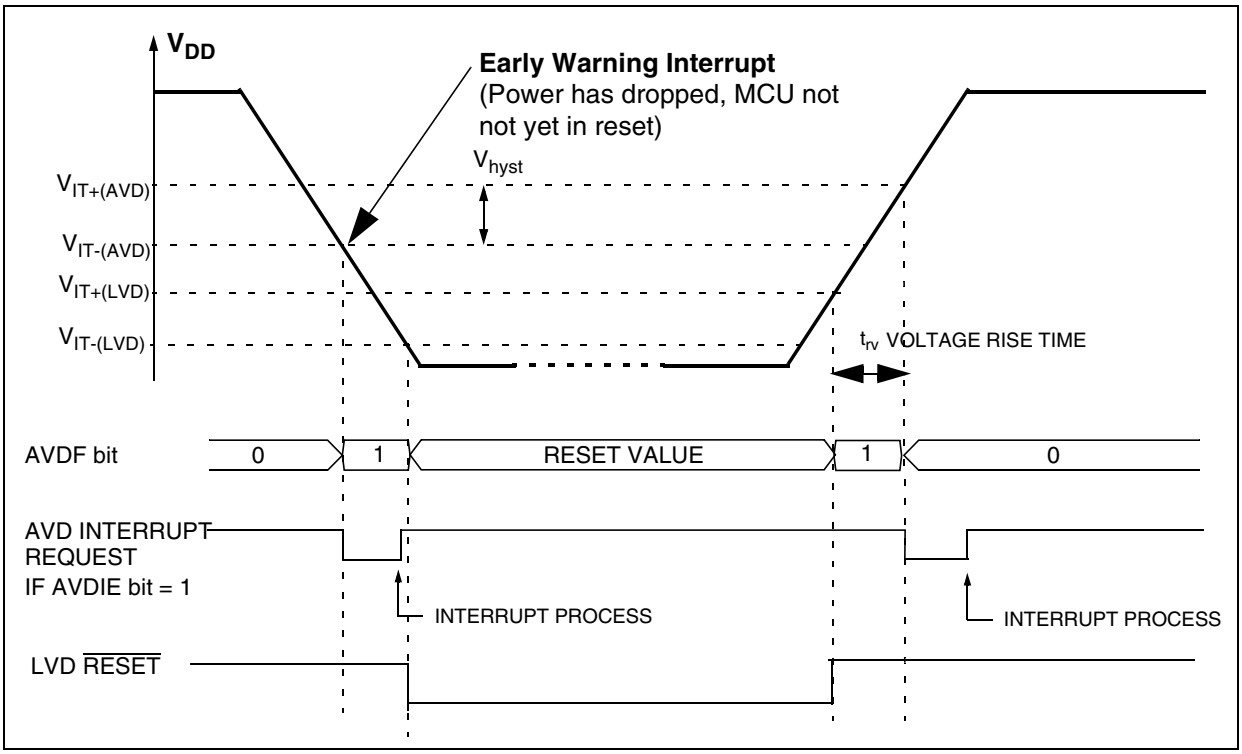
If the voltage rise time t_{rv} is less than 256 or 4096 CPU cycles (depending on the reset delay selected by option byte), no AVD interrupt will be generated when $V_{IT+(AVD)}$ is reached.

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

– If the AVD interrupt is enabled before the $V_{IT+(AVD)}$ threshold is reached, then 2 AVD interrupts will be received: the first when the AVDIE bit is set, and the second when the threshold is reached.

– If the AVD interrupt is enabled after the $V_{IT+(AVD)}$ threshold is reached then only one AVD interrupt will occur.

Figure 17. Using the AVD to Monitor V_{DD} (AVDS bit=0)



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 10 in I1:0 of CC	1		0			
SIM	Disable interrupt (level 3 set)	Load 11 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.

Table 14. Watchdog Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
002Ah	**WDGCR**	WDGA	T6	T5	T4	T3	T2	T1	T0
	Reset Value	0	1	1	1	1	1	1	1

10.2 MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK AND BEEPER (MCC/RTC)

The Main Clock Controller consists of three different functions:

- a programmable CPU clock prescaler
- a clock-out signal to supply external devices
- a real time clock timer with interrupt capability

Each function can be used independently and simultaneously.

10.2.1 Programmable CPU Clock Prescaler

The programmable CPU clock prescaler supplies the clock for the ST7 CPU and its internal peripherals. It manages SLOW power saving mode (See Section 8.2 SLOW MODE for more details).

The prescaler selects the f_{CPU} main clock frequency and is controlled by three bits in the MCCSR register: CP[1:0] and SMS.

10.2.2 Clock-out Capability

The clock-out capability is an alternate function of an I/O port pin that outputs a f_{CPU} clock to drive external devices. It is controlled by the MCO bit in the MCCSR register.

CAUTION: When selected, the clock out pin suspends the clock during ACTIVE-HALT mode.

10.2.3 Real Time Clock Timer (RTC)

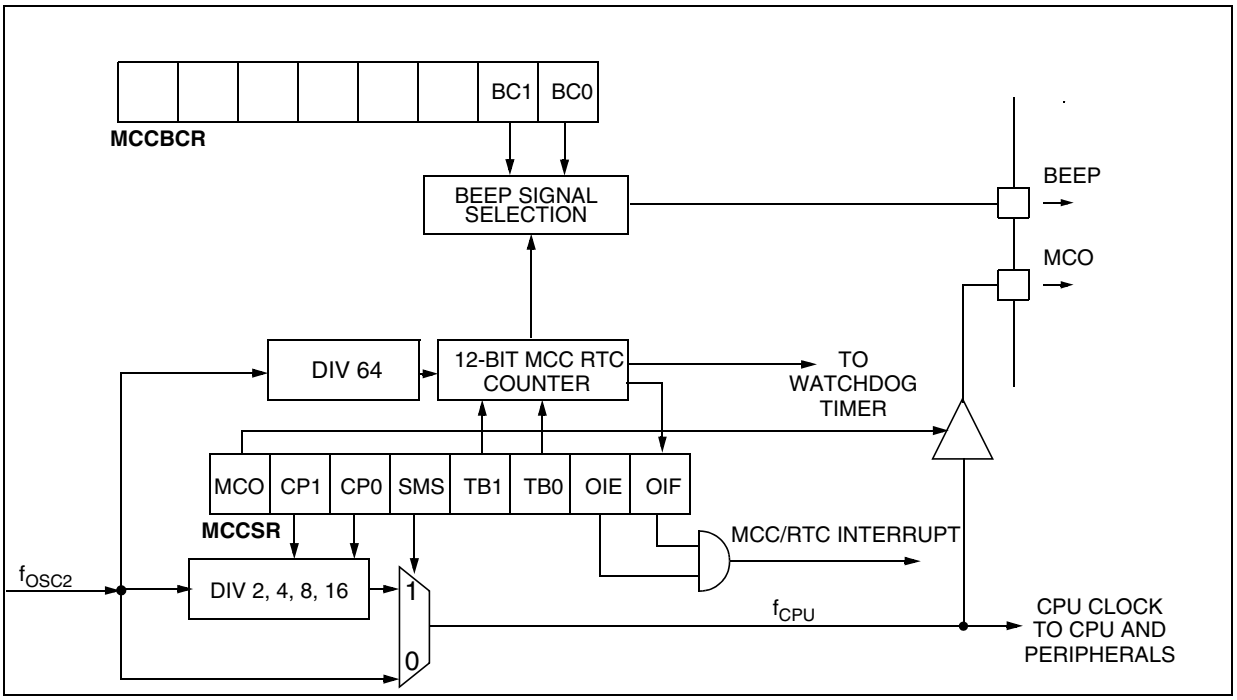
The counter of the real time clock timer allows an interrupt to be generated based on an accurate real time clock. Four different time bases depending directly on f_{OSC2} are available. The whole functionality is controlled by four bits of the MCCSR register: TB[1:0], OIE and OIF.

When the RTC interrupt is enabled (OIE bit set), the ST7 enters ACTIVE-HALT mode when the HALT instruction is executed. See Section 8.4 ACTIVE-HALT AND HALT MODES for more details.

10.2.4 Beeper

The beep function is controlled by the MCCBCR register. It can output three selectable frequencies on the BEEP pin (I/O port alternate function).

Figure 36. Main Clock Controller (MCC/RTC) Block Diagram



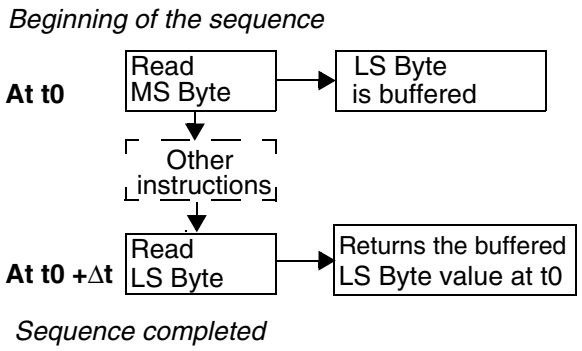
PWM AUTO-RELOAD TIMER (Cont'd)

Table 16. PWM Auto-Reload Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0073h	**PWMDCR3**	DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0
	Reset Value	0	0	0	0	0	0	0	0
0074h	**PWMDCR2**	DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0
	Reset Value	0	0	0	0	0	0	0	0
0075h	**PWMDCR1**	DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0
	Reset Value	0	0	0	0	0	0	0	0
0076h	**PWMDCR0**	DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0
	Reset Value	0	0	0	0	0	0	0	0
0077h	**PWMCR**	OE3	OE2	OE1	OE0	OP3	OP2	OP1	OP0
	Reset Value	0	0	0	0	0	0	0	0
0078h	**ARTCSR**	EXCL	CC2	CC1	CC0	TCE	FCRL	RIE	OVF
	Reset Value	0	0	0	0	0	0	0	0
0079h	**ARTCAR**	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
	Reset Value	0	0	0	0	0	0	0	0
007Ah	**ARTARR**	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
	Reset Value	0	0	0	0	0	0	0	0
007Bh	**ARTICCSR**			CS2	CS1	CIE2	CIE1	CF2	CF1
	Reset Value	0	0	0	0	0	0	0	0
007Ch	**ARTICR1**	IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0
	Reset Value	0	0	0	0	0	0	0	0
007Dh	**ARTICR2**	IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0
	Reset Value	0	0	0	0	0	0	0	0

16-BIT TIMER (Cont'd)

16-bit read sequence: (from either the Counter Register or the Alternate Counter Register).



The user must read the MS Byte first, then the LS Byte value is buffered automatically.

This buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MS Byte several times.

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LS Byte of the count value at the time of the read.

Whatever the timer mode used (input capture, output compare, One Pulse mode or PWM mode) an overflow occurs when the counter rolls over from FFFFh to 0000h then:

– The TOF bit of the SR register is set.

– A timer interrupt is generated if:

  – TOIE bit of the CR1 register is set and

  – I bit of the CC register is cleared.

If one of these conditions is false, the interrupt remains pending to be issued as soon as they are both true.

Clearing the overflow interrupt request is done in two steps:

1. Reading the SR register while the TOF bit is set.
2. An access (read or write) to the CLR register.

Notes: The TOF bit is not cleared by accesses to ACLR register. The advantage of accessing the ACLR register rather than the CLR register is that it allows simultaneous use of the overflow function and reading the free running counter at random times (for example, to measure elapsed time) without the risk of clearing the TOF bit erroneously.

The timer is not affected by WAIT mode.

In HALT mode, the counter stops counting until the mode is exited. Counting then resumes from the previous count (MCU awakened by an interrupt) or from the reset count (MCU awakened by a Reset).

10.4.3.2 External Clock

The external clock (where available) is selected if CC0 = 1 and CC1 = 1 in the CR2 register.

The status of the EXEDG bit in the CR2 register determines the type of level transition on the external clock pin EXTCLK that will trigger the free running counter.

The counter is synchronized with the falling edge of the internal CPU clock.

A minimum of four falling edges of the CPU clock must occur between two consecutive active edges of the external clock; thus the external clock frequency must be less than a quarter of the CPU clock frequency.

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.6.5 Low Power Modes

Mode	Description
WAIT	No effect on SCI. SCI interrupts cause the device to exit from Wait mode.
HALT	SCI registers are frozen. In Halt mode, the SCI stops transmitting/receiving until Halt mode is exited.

10.6.6 Interrupts

The SCI interrupt events are connected to the same interrupt vector.

These events generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Transmit Data Register Empty	TDRE	TIE	Yes	No
Transmission Complete	TC	TCIE	Yes	No
Received Data Ready to be Read	RDRF	RIE	Yes	No
Overrun Error Detected	OR		Yes	No
Idle Line Detected	IDLE	ILIE	Yes	No
Parity Error	PE	PIE	Yes	No

I²C INTERFACE (Cont'd)

How to release the SDA / SCL lines

Set and subsequently clear the STOP bit while BTF is set. The SDA/SCL lines are released after the transfer of the current byte.

SMBus Compatibility

ST7 I²C is compatible with SMBus V1.1 protocol. It supports all SMBus adressing modes, SMBus bus protocols and CRC-8 packet error checking. Refer to AN1713: SMBus Slave Driver For ST7 I²C Peripheral.

10.7.4.2 Master Mode

To switch from default Slave mode to Master mode a Start condition generation is needed.

Start condition

Setting the START bit while the BUSY bit is cleared causes the interface to switch to Master mode (M/SL bit set) and generates a Start condition.

Once the Start condition is sent:

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

Then the master waits for a read of the SR1 register followed by a write in the DR register with the Slave address, **holding the SCL line low** (see Figure 67 Transfer sequencing EV5).

Slave address transmission

Then the slave address is sent to the SDA line via the internal shift register.

In 7-bit addressing mode, one address byte is sent.

In 10-bit addressing mode, sending the first byte including the header sequence causes the following event:

– The EVF bit is set by hardware with interrupt generation if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see Figure 67 Transfer sequencing EV9).

Then the second address byte is sent by the interface.

After completion of this transfer (and acknowledge from the slave if the ACK bit is set):

– The EVF bit is set by hardware with interrupt generation if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the CR register (for example set PE bit), **holding the SCL line low** (see Figure 67 Transfer sequencing EV6).

Next the master must enter Receiver or Transmitter mode.

Note: In 10-bit addressing mode, to switch the master to Receiver mode, software must generate a repeated Start condition and resend the header sequence with the least significant bit set (11110xx1).

Master Receiver

Following the address transmission and after SR1 and CR registers have been accessed, the master 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 by hardware 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 67 Transfer sequencing EV7).

To close the communication: before reading the last byte from the DR register, set the STOP bit to generate the Stop condition. The interface goes automatically back to slave mode (M/SL bit cleared).

Note: In order to generate the non-acknowledge pulse after the last received data byte, the ACK bit must be cleared just before reading the second last data byte.

I²C BUS INTERFACE (Cont'd)

Table 24. I²C Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0018h	**I2CCR** Reset Value	0	0	PE 0	ENGC 0	START 0	ACK 0	STOP 0	ITE 0
0019h	**I2CSR1** Reset Value	EVF 0	ADD10 0	TRA 0	BUSY 0	BTF 0	ADSL 0	M/SL 0	SB 0
001Ah	**I2CSR2** Reset Value	0	0	0	AF 0	STOPF 0	ARLO 0	BERR 0	GCAL 0
001Bh	**I2CCCR** Reset Value	FM/SM 0	CC6 0	CC5 0	CC4 0	CC3 0	CC2 0	CC1 0	CC0 0
001Ch	**I2COAR1** Reset Value	ADD7 0	ADD6 0	ADD5 0	ADD4 0	ADD3 0	ADD2 0	ADD1 0	ADD0 0
001Dh	**I2COAR2** Reset Value	FR1 0	FR0 1	0	0	0	ADD9 0	ADD8 0	0
001Eh	**I2CDR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0

INSTRUCTION SET OVERVIEW (Cont'd)

11.1.1 Inherent

All Inherent instructions consist of a single byte. The opcode fully specifies all the required information for the CPU to process the operation.

Inherent Instruction	Function
NOP	No operation
TRAP	S/W Interrupt
WFI	Wait For Interrupt (Low Power Mode)
HALT	Halt Oscillator (Lowest Power Mode)
RET	Sub-routine Return
IRET	Interrupt Sub-routine Return
SIM	Set Interrupt Mask (level 3)
RIM	Reset Interrupt Mask (level 0)
SCF	Set Carry Flag
RCF	Reset Carry Flag
RSP	Reset Stack Pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
MUL	Byte Multiplication
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles

11.1.2 Immediate

Immediate instructions have 2 bytes, the first byte contains the opcode, the second byte contains the operand value.

Immediate Instruction	Function
LD	Load
CP	Compare
BCP	Bit Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Operations

11.1.3 Direct

In Direct instructions, the operands are referenced by their memory address.

The direct addressing mode consists of two submodes:

Direct (short)

The address is a byte, thus requires only one byte after the opcode, but only allows 00 - FF addressing space.

Direct (long)

The address is a word, thus allowing 64 Kbyte addressing space, but requires 2 bytes after the opcode.

11.1.4 Indexed (No Offset, Short, Long)

In this mode, the operand is referenced by its memory address, which is defined by the unsigned addition of an index register (X or Y) with an offset.

The indirect addressing mode consists of three submodes:

Indexed (No Offset)

There is no offset, (no extra byte after the opcode), and allows 00 - FF addressing space.

Indexed (Short)

The offset is a byte, thus requires only one byte after the opcode and allows 00 - 1FE addressing space.

Indexed (long)

The offset is a word, thus allowing 64 Kbyte addressing space and requires 2 bytes after the opcode.

11.1.5 Indirect (Short, Long)

The required data byte to do the operation is found by its memory address, located in memory (pointer).

The pointer address follows the opcode. The indirect addressing mode consists of two submodes:

Indirect (short)

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

Indirect (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.

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							
Interruption 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.

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	Parameter	Conditions	Flash Devices		ROM Devices		Unit
			Typ	Max [1]	Typ	Max [1]	
I_{DD}	Supply current in RUN mode [2]	f_{OSC}=2MHz, f_{CPU}=1MHz	1.3	3.0	0.5	1.0	mA
		f_{OSC}=4MHz, f_{CPU}=2MHz	2.0	5.0	1.2	2.0	
		f_{OSC}=8MHz, f_{CPU}=4MHz	3.6	8.0	2.2	4.0	
		f_{OSC}=16MHz, f_{CPU}=8MHz	7.1	15.0	4.8	8.0	
	Supply current in SLOW mode [2]	f_{OSC}=2MHz, f_{CPU}=62.5kHz	600	2700	100	600	µA
		f_{OSC}=4MHz, f_{CPU}=125kHz	700	3000	200	700	
		f_{OSC}=8MHz, f_{CPU}=250kHz	800	3600	300	800	
		f_{OSC}=16MHz, f_{CPU}=500kHz	1100	4000	500	950	
	Supply current in WAIT mode [2]	f_{OSC}=2MHz, f_{CPU}=1MHz	0.8	3.0	0.5	1.0	mA
		f_{OSC}=4MHz, f_{CPU}=2MHz	1.2	4.0	0.8	1.3	
		f_{OSC}=8MHz, f_{CPU}=4MHz	2.0	5.0	1.5	2.2	
		f_{OSC}=16MHz, f_{CPU}=8MHz	3.5	7.0	3.0	4.0	
	Supply current in SLOW WAIT mode [2]	f_{OSC}=2MHz, f_{CPU}=62.5kHz	580	1200	50	100	µA
		f_{OSC}=4MHz, f_{CPU}=125kHz	650	1300	90	150	
		f_{OSC}=8MHz, f_{CPU}=250kHz	770	1800	180	300	
		f_{OSC}=16MHz, f_{CPU}=500kHz	1050	2000	350	600	
	Supply current in HALT mode [3]	-40°C ≤T_A≤+85°C	<1	10	<1	10	µA
		-40°C ≤T_A≤+125°C	5	50	<1	50	
I_{DD}	Supply current in ACTIVE-HALT mode [4]	f_{OSC}=2MHz	450	No max. guaranteed	15	25	µA
		f_{OSC}=4MHz	465		30	50	
		f_{OSC}=8MHz	530		60	100	
		f_{OSC} =16MHz	650		120	200	

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.
- 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).

ADC CHARACTERISTICS (Cont'd)

Figure 93. R_{AIN} max. vs f_{ADC} with C_{AIN}=0pF[1)]

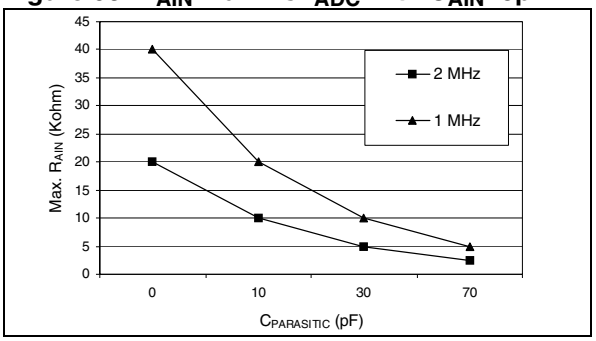


Figure 94. Recommended C_{AIN} & R_{AIN} values.[2)]

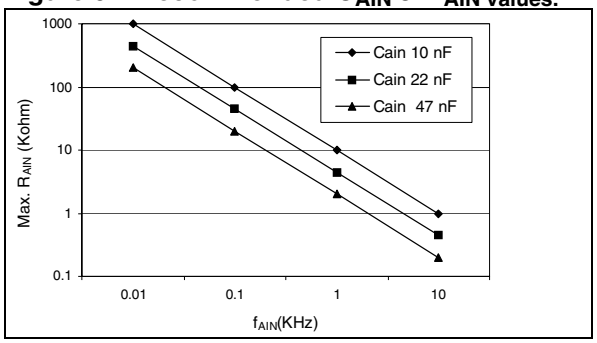
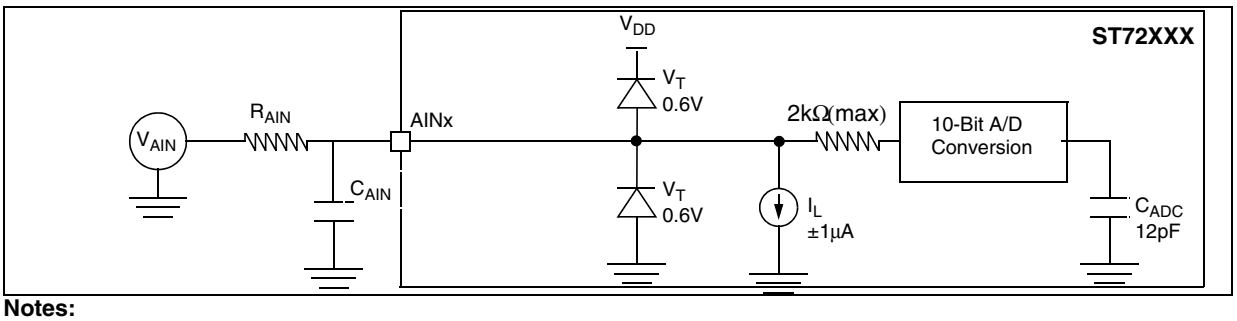


Figure 95. Typical A/D Converter Application



Notes:

1. $C_{PARASITIC}$ represents the capacitance of the PCB (dependent on soldering and PCB layout quality) plus the pad capacitance (3pF). A high $C_{PARASITIC}$ value will downgrade conversion accuracy. To remedy this, f_{ADC} should be reduced.

2. This graph shows that depending on the input signal variation (f_{AIN}), C_{AIN} can be increased for stabilization time and decreased to allow the use of a larger serial resistor (R_{AIN}).

13.2 THERMAL CHARACTERISTICS

Symbol	Ratings	Value	Unit
R_{thJA}	Package thermal resistance (junction to ambient) LQFP64 14x14 LQFP64 10x10 LQFP44 10x10 LQFP32 7x7	 47 50 52 70	°C/W
P_D	Power dissipation [1]	500	mW
T_{Jmax}	Maximum junction temperature [2]	150	°C

Notes:

1. The maximum chip-junction temperature is based on technology characteristics.

2. The maximum power dissipation is obtained from the formula PD = (TJ -TA) / RthJA.

The power dissipation of an application can be defined by the user with the formula: PD=PINT+PPORT where PINT is the chip internal power (IDDxVDD) and PPORT is the port power dissipation depending on the ports used in the application.

14 ST72321B DEVICE CONFIGURATION AND ORDERING INFORMATION

Each device is available for production in user programmable versions (FLASH) as well as in factory coded versions (ROM/FASTROM).

ST72321B devices are ROM versions. ST72P321B devices are Factory Advanced Service Technique ROM (FASTROM) versions: they are factory-programmed HDFlash devices. FLASH devices are shipped to customers with a default content, while ROM/FASTROM factory coded parts contain the code supplied by the customer. This implies that FLASH devices have to be configured by the customer using the Option Bytes while the ROM/FASTROM devices are factory-configured.

14.1 FLASH OPTION BYTES

	STATIC OPTION BYTE 0								STATIC OPTION BYTE							
	7							0	17							0
	WDG		Reserved	VD		Reserved	PKG0	FMP_R	PKG1	RSTC	OSCTYPE		OSCRANGE			PLLOFF
	HALT	SW		1	0						1	0	2	1	0	
Default	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1

The option bytes allow the hardware configuration of the microcontroller to be selected. They have no address in the memory map and can be accessed only in programming mode (for example using a standard ST7 programming tool). The default content of the FLASH is fixed to FFh. To program the FLASH devices directly using ICP, FLASH devices are shipped to customers with the internal RC clock source enabled. In masked ROM devices, the option bytes are fixed in hardware by the ROM code (see option list).

OPTION BYTE 0

OPT7= **WDG HALT** *Watchdog and HALT mode*
This option bit determines if a RESET is generated when entering HALT mode while the Watchdog is active.
0: No Reset generation when entering Halt mode
1: Reset generation when entering Halt mode

OPT6= **WDG SW** *Hardware or software watchdog*
This option bit selects the watchdog type.
0: Hardware (watchdog always enabled)
1: Software (watchdog to be enabled by software)

OPT5 = Reserved, must be kept at default value.

OPT4:3= **VD[1:0]** *Voltage detection*
These option bits enable the voltage detection block (LVD, and AVD) with a selected threshold for the LVD and AVD (EVD+AVD).

Selected Low Voltage Detector	VD1	VD0
LVD and AVD Off	1	1
Lowest Threshold: (V_{DD}~3V)	1	0
Med. Threshold (V_{DD}~3.5V)	0	1
Highest Threshold (V_{DD}~4V)	0	0

Caution: If the medium or low thresholds are selected, the detection may occur outside the specified operating voltage range. Below 3.8V, device operation is not guaranteed. For details on the AVD and LVD threshold levels refer to section 12.3.2 on page 141

OPT2 = Reserved, must be kept at default value.

OPT1= **PKG0** *Package selection bit 0*
This option bit is not used.

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
LQFP32 7 X 7	IRONWOOD SF-QFE32SA-L-01	IRONWOOD SK-UGA06/32A-01

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