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	Not For New Designs
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	32
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 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f324bj6t3

List of tables

2 Pin description

Figure 2. 44-pin LQFP package pinout

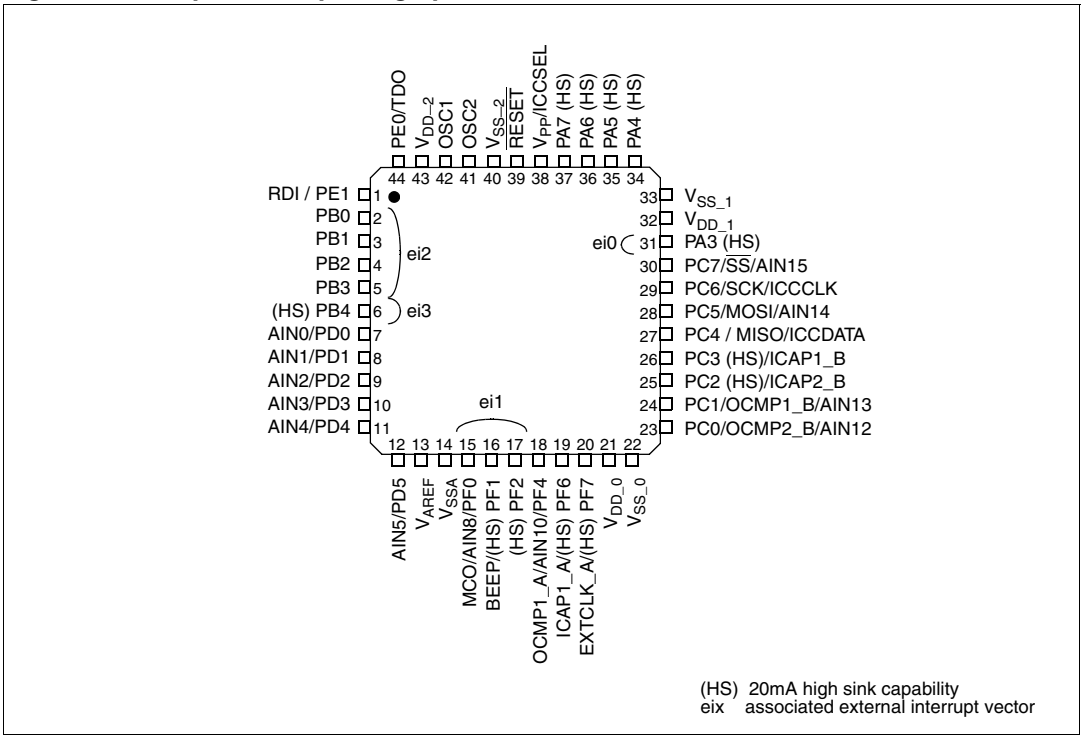
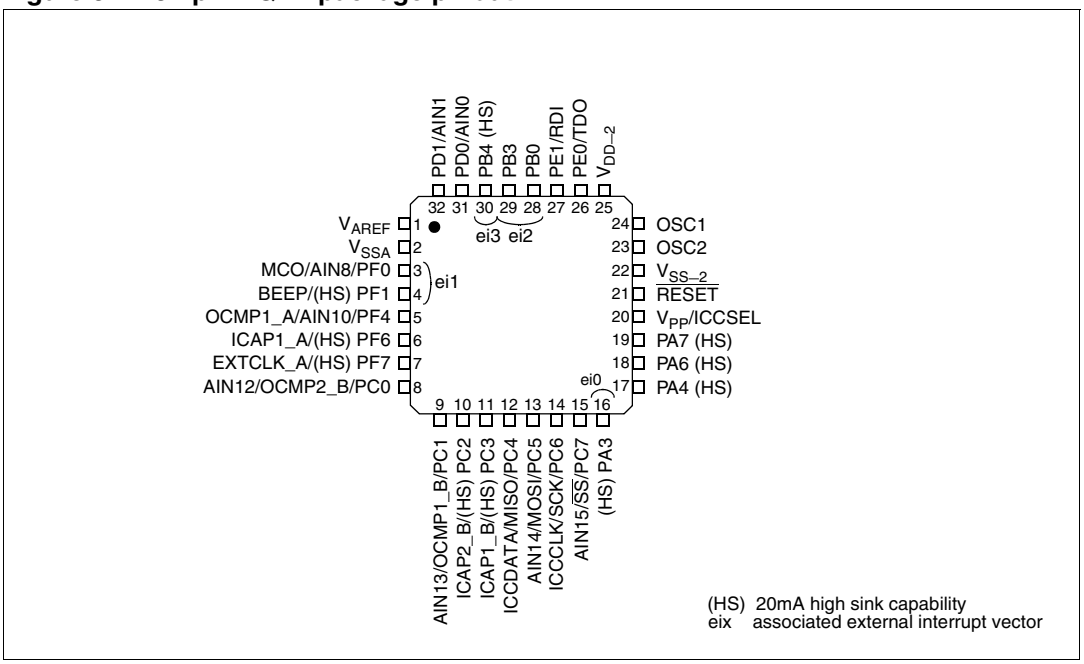


Figure 3. 32-pin LQFP package pinout



See *Section 12: Electrical characteristics on page 145* for external pin connection guidelines.

Refer to *Section 9: I/O ports on page 58* for more details on the software configuration of the I/O ports.

The reset configuration of each pin is shown in bold. This configuration is valid as long as the device is in reset state.

Table 2. Device pin description

Pin			Type	Level		Port						Main function (after reset)	Alternate function	
No.		Name		Input	Output	Input				Output				
LQFP44	LQFP32					float	wpu	int	ana	OD	PP			
6	30	PB4 (HS)	I/O	C_T	HS	**X**	ei3			X	X	Port B4		
7	31	PD0/AIN0	I/O	C_T		**X**	X		X	X	X	Port D0	ADC analog input 0	
8	32	PD1/AIN1	I/O	C_T		**X**	X		X	X	X	Port D1	ADC analog input 1	
9	-	PD2/AIN2	I/O	C_T		**X**	X		X	X	X	Port D2	ADC analog input 2	
10	-	PD3/AIN3	I/O	C_T		**X**	X		X	X	X	Port D3	ADC analog input 3	
11	-	PD4/AIN4	I/O	C_T		**X**	X		X	X	X	Port D4	ADC analog input 4	
12	-	PD5/AIN5	I/O	C_T		**X**	X		X	X	X	Port D5	ADC analog input 5	
13	1	V_{AREF}(1)	S									Analog reference voltage for ADC		
14	2	V_{SSA}(1)	S									Analog ground voltage		
15	3	PF0/MCO/AIN8	I/O	C_T		**X**	ei1		X	X	X	Port F0	Main clock out (f_{CPU})	ADC analog input 8
16	4	PF1 (HS)/BEEP	I/O	C_T	HS	**X**	ei1			X	X	Port F1	Beep signal output	
17	-	PF2 (HS)	I/O	C_T	HS	**X**	ei1			X	X	Port F2		
18	5	PF4/OCMP1_A /AIN10	I/O	C_T		**X**	X		X	X	X	Port F4	Timer A output compare 1	ADC analog Input 10
19	6	PF6 (HS)/ICAP1_A	I/O	C_T	HS	**X**	X			X	X	Port F6	Timer A input capture 1	
20	7	PF7 (HS)/EXTCLK_A	I/O	C_T	HS	**X**	X			X	X	Port F7	Timer A external clock source	
21	-	V_{DD_0}(1)	S									Digital main supply voltage		
22	-	V_{SS_0}(1)	S									Digital ground voltage		
23	8	PC0/OCMP2_B /AIN12	I/O	C_T		**X**	X		X	X	X	Port C0	Timer B output compare 2	ADC analog input 12
24	9	PC1/OCMP1_B /AIN13	I/O	C_T		**X**	X		X	X	X	Port C1	Timer B output compare 1	ADC analog input 13
25	10	PC2 (HS)/ICAP2_B	I/O	C_T	HS	**X**	X			X	X	Port C2	Timer B input capture 2	

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.

2 Reset and TRAP can be considered as having the highest software priority in the decision process.

7.2.2 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).

7.2.3 Non-maskable sources

These sources are processed regardless of the state of the I1 and I0 bits of the CC register (see *Figure 16*). After stacking the PC, X, A and CC registers (except for reset), the corresponding vector is loaded in the PC register and the I1 and I0 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 16*.

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.

7.2.4 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.

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 *Table 25: Interrupt mapping*. 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

They can be also set/cleared by software with the RIM, SIM, HALT, WFI, IRET and PUSH/POP instructions (see *Table 18: Dedicated interrupt instruction set*).

7.5.2 Interrupt software priority registers (ISPRx)

ISPRx Reset value: 1111 1111 (FFh)

	7	6	5	4	3	2	1	0
ISPR0	I1_3	I0_3	I1_2	I0_2	I1_1	I0_1	I1_0	I0_0
ISPR1	I1_7	I0_7	I1_6	I0_6	I1_5	I0_5	I1_4	I0_4
ISPR2	I1_11	I0_11	I1_10	I0_10	I1_9	I0_9	I1_8	I0_8
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ISPR3	1	1	1	1	I1_13	I0_13	I1_12	I0_12
	RO	RO	RO	RO	R/W	R/W	R/W	R/W

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 correspondence is shown in the following *Table 17*.

Table 17. ISPRx interrupt vector correspondence

Vector address	ISPRx bits
FFFBh-FFFAh	I1_0 and I0_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 cannot be written (I1_x = 1, I0_x = 0). In this case, the previously stored value is kept (for example, previous value = 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 behavior 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).

Table 18. Dedicated interrupt instruction set(1)

Instruction	New description	Function/example	I1	H	I0	N	Z	C
HALT	Entering HALT mode		1		0			

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

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

Where:

Δt = Output compare period (in seconds)
f_{EXT} = External timer clock frequency (in hertz)

Clearing the output compare interrupt request (that is, 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 OC*i*LR 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).

Note: 1 After a processor write cycle to the OCiHR register, the output compare function is inhibited until the OCiLR register is also written.

2 If the OCiE bit is not set, the OCMPi pin is a general I/O port and the OLVLi bit will not appear when a match is found but an interrupt could be generated if the OCIE bit is set.

3 In both internal and external clock modes, OCFi and OCMPi are set while the counter value equals the OCiR register value (see Figure 42 on page 83 for an example with $f_{CPU}/2$ and Figure 43 on page 83 for an example with $f_{CPU}/4$). This behavior is the same in OPM or PWM mode.

4 The output compare functions can be used both for generating external events on the OCMPi pins even if the input capture mode is also used.

5 The value in the 16-bit OCiR register and the OLVi bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

Forced output compare capability

When the FOLV*i* bit is set by software, the OLVL*i* bit is copied to the OCMP*i* pin. The OLV*i* bit has to be toggled in order to toggle the OCMP*i* pin when it is enabled (OC*i*E bit = 1). The OCF*i* bit is then not set by hardware, and thus no interrupt request is generated.

The FOLVL*i* bits have no effect in both one pulse mode and PWM mode.

Alternate Counter Low Register (ACLR)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to CSR register does not clear the TOF bit in the CSR register.

ACLR — Reset value: 1111 1100 (FCh)

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Input Capture 2 High Register (IC2HR)

This is an 8-bit register that contains the high part of the counter value (transferred by the Input Capture 2 event).

1C2HR — Reset value: undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Input Capture 2 Low Register (IC2LR)

This is an 8-bit register that contains the low part of the counter value (transferred by the Input Capture 2 event).

1C2LR — Reset value: undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

Table 52. 16-bit timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
Timer A: 32 Timer B: 42	CR1 Reset value	ICIE 0	OCIE 0	TOIE 0	FOLV2 0	FOLV1 0	OLVL2 0	IEDG1 0	OLVL1 0
Timer A: 31 Timer B: 41	CR2 Reset value	OC1E 0	OC2E 0	OPM 0	PWM 0	CC1 0	CC0 0	IEDG2 0	EXEDG 0
Timer A: 33 Timer B: 43	CSR Reset value	ICF1 x	OCF1 x	TOF x	ICF2 x	OCF2 x	TIMD 0	- x	- x
Timer A: 34 Timer B: 44	IC1HR Reset value	MSB x	x	x	x	x	x	x	LSB x

Master mode transmit sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt request is generated if the SPIE bit is set and the interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SPICSR register while the SPIF bit is set.
2. A read to the SPIDR register.

Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

Slave mode operation

In slave mode, the serial clock is received on the SCK pin from the master device.

To operate the SPI in slave mode:

1. Write to the SPICSR register to perform the following actions:
   - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see *Figure 52*). The slave must have the same CPOL and CPHA settings as the master.
   - Manage the $\overline{SS}$ pin as described in *Slave Select management on page 99* and *Figure 50*. If CPHA = 1, $\overline{SS}$ must be held low continuously. If CPHA = 0, $\overline{SS}$ must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.
2. Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

Slave mode transmit sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MISO pin most significant bit first.

The transmit sequence begins when the slave device receives the clock signal and the most significant bit of the data on its MOSI pin.

When data transfer is complete:

- The SPIF bit is set by hardware
- An interrupt request is generated if SPIE bit is set and interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SPICSR register while the SPIF bit is set.
2. A write or a read to the SPIDR register.

Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

Table 55. SPICR register description (continued)

Bit	Name	Function
2	CPHA	Clock Phase This bit is set and cleared by software. 0: The first clock transition is the first data capture edge. 1: The second clock transition is the first capture edge. *Note: The slave must have the same CPOL and CPHA settings as the master.*
1:0	SPR[1:0]	Serial clock frequency These bits are set and cleared by software. Used with the SPR2 bit, they select the baud rate of the SPI serial clock SCK output by the SPI in master mode (see *Table 56*). *Note: These 2 bits have no effect in slave mode.*

Table 56. SPI master mode SCK frequency

Serial clock	SPR2	SPR1	SPR0
$f_{CPU}/4$	1	0	0
$f_{CPU}/8$	0	0	0
$f_{CPU}/16$	0	0	1
$f_{CPU}/32$	1	1	0
$f_{CPU}/64$	0	1	0
$f_{CPU}/128$	0	1	1

SPI Control/Status Register (SPICSR)

SPICSR Reset value: 0000 0000 (00h)

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

Table 64. SCICR2 register description (continued)

Bit	Name	Function
5	RIE	Receiver interrupt Enable This bit is set and cleared by software. 0: Interrupt is inhibited 1: An SCI interrupt is generated whenever OR = 1 or RDRF = 1 in the SCISR register.
4	ILIE	Idle Line Interrupt Enable This bit is set and cleared by software. 0: Interrupt is inhibited 1: An SCI interrupt is generated whenever IDLE = 1 in the SCISR register.
3	TE	Transmitter Enable This bit enables the transmitter. It is set and cleared by software. 0: Transmitter is disabled 1: Transmitter is enabled *Notes:* *- During transmission, a '0' pulse on the TE bit ('0' followed by '1') sends a preamble (Idle line) after the current word.* *- When TE is set there is a 1 bit-time delay before the transmission starts.* **Caution:** The TDO pin is free for general purpose I/O only when the TE and RE bits are both cleared (or if TE is never set).
2	RE	Receiver Enable This bit enables the receiver. It is set and cleared by software. 0: Receiver is disabled 1: Receiver is enabled and begins searching for a start bit *Note: Before selecting Mute mode (setting the RWU bit), the SCI must first receive some data, otherwise it cannot function in Mute mode with Wake-Up by Idle line detection.*
1	RWU	Receiver Wake-Up This bit determines if the SCI is in mute mode or not. It is set and cleared by software and can be cleared by hardware when a wake-up sequence is recognized. 0: Receiver in Active mode 1: Receiver in Mute mode
0	SBK	Send Break This bit set is used to send break characters. It is set and cleared by software. 0: No break character is transmitted. 1: Break characters are transmitted. *Note: If the SBK bit is set to '1' and then to '0', the transmitter will send a Break word at the end of the current word.*

SCI Data Register (SCIDR)

This register contains the received or transmitted data character, depending on whether it is read from or written to.

SCIDR Reset value: undefined

7	6	5	4	3	2	1	0
DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

The TDR register provides the parallel interface between the internal bus and the output shift register (see *Figure 55*). The RDR register provides the parallel interface between the input shift register and the internal bus (see *Figure 55*).

SCI Baud Rate Register (SCIBRR)

SCIBRR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
SCP[1:0]		SCT[2:0]			SCR[2:0]		
R/W		R/W			R/W		

Table 65. SCIBRR register description

Bit	Name	Function
7:6	SCP[1:0]	First SCI Prescaler These 2 prescaling bits allow several standard clock division ranges. 00: PR prescaling factor = 1 01: PR prescaling factor = 3 10: PR prescaling factor = 4 11: PR prescaling factor = 13

10.6 10-bit A/D converter (ADC)

10.6.1 Introduction

The on-chip analog-to-digital converter (ADC) peripheral is a 10-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 16 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 16 different sources.

The result of the conversion is stored in a 10-bit Data Register. The A/D converter is controlled through a Control/Status Register.

10.6.2 Main features

- 10-bit conversion
- Up to 16 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in *Figure 59*.

Figure 59. ADC block diagram

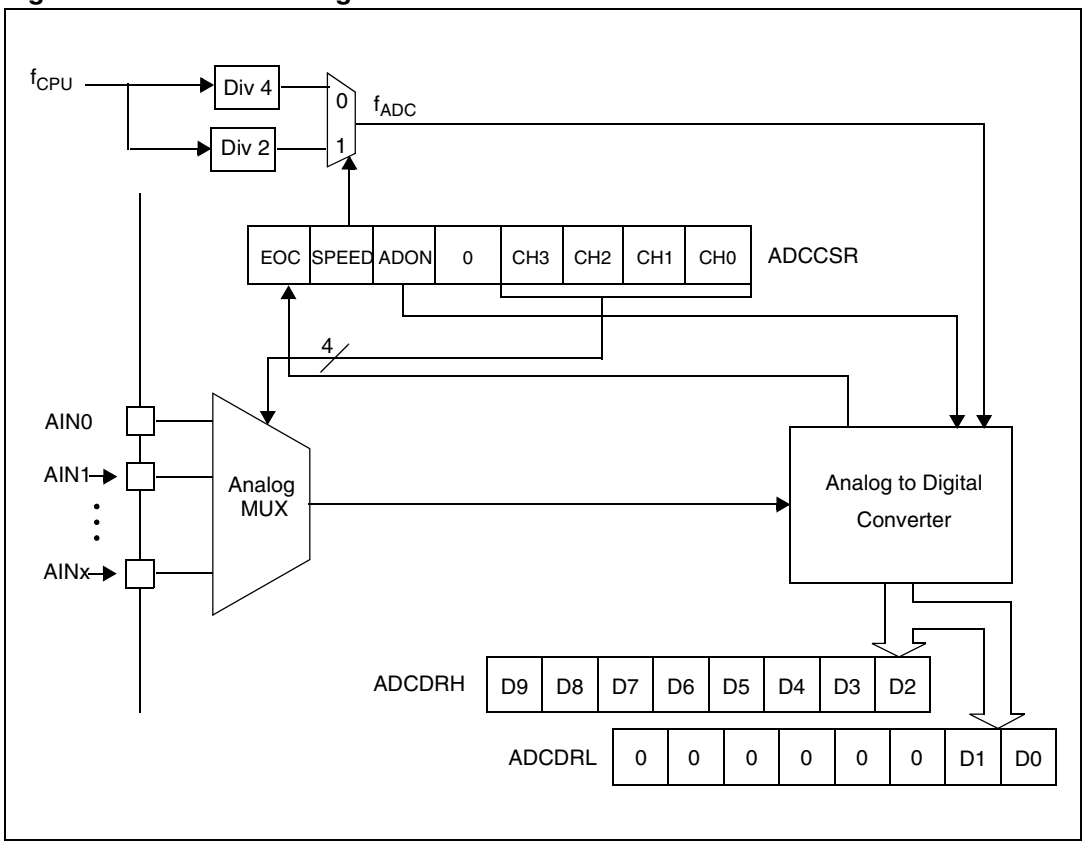


Figure 65. Typical application with a crystal or ceramic resonator (32 Kbyte Flash and ROM devices)

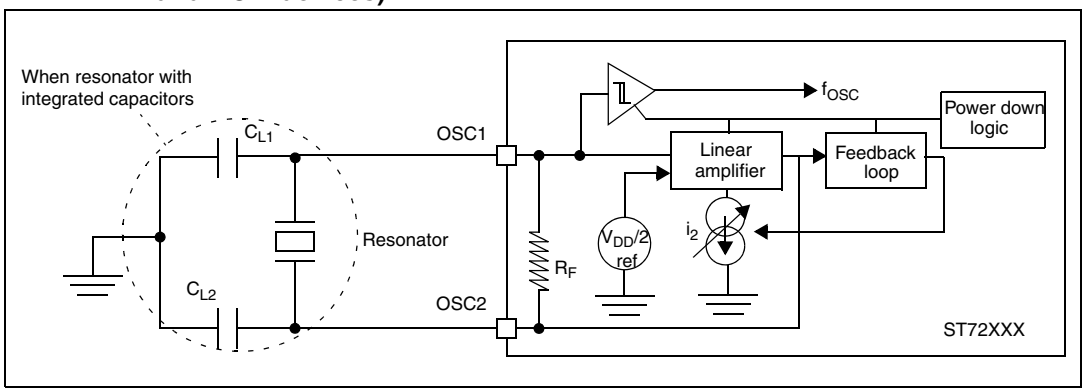


Table 96. OSCRANGE selection for typical resonators

Supplier	f_{OSC} (MHz)	Typical ceramic resonators(1)	
		Reference	Recommended OSCRANGE option bit configuration
Murata	2	CSTCC2M00G56A-R0	MP mode(2)
	4	CSTCR4M00G55B-R0	MS mode
	8	CSTCE8M00G52A-R0	HS mode
	16	CSTCE16M0V51A-R0	HS mode

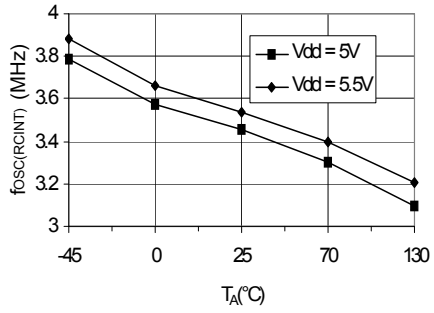
1. Resonator characteristics given by the ceramic resonator manufacturer.
2. LP mode is not recommended for 2 MHz resonator because the peak to peak amplitude is too small (>0.8V). For more information on these resonators, please consult www.murata.com.

12.6.4 RC oscillators

Table 97. RC oscillators

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$f_{OSC\ (RCINT)}$	Internal RC oscillator frequency (see *Figure 66)*	T_A = 25°C, V_{DD} = 5V	2	3.5	5.6	MHz

Figure 66. Typical $f_{OSC(RCINT)}$ vs T_A



12.8 EMC characteristics

Susceptibility tests are performed on a sample basis during product characterization.

12.8.1 Functional electromagnetic susceptibility (EMS)

Based on a simple running application on the product (toggling two LEDs through I/O ports), the product is stressed by two electromagnetic events until a failure occurs (indicated by the LEDs).

- **ESD**: Electrostatic discharge (positive and negative) is applied on all pins of the device until a functional disturbance occurs. This test conforms with the IEC 1000-4-2 standard.
- **FTB**: A burst of fast transient voltage (positive and negative) is applied to V_{DD} and V_{SS} through a 100pF capacitor, until a functional disturbance occurs. This test conforms with the IEC 1000-4-4 standard.

A device reset allows normal operations to be resumed. The test results given in *Table 101 on page 159* are based on the EMS levels and classes defined in application note AN1709.

Designing hardened software to avoid noise problems

EMC characterization and optimization are performed at component level with a typical application environment and simplified MCU software. It should be noted that good EMC performance is highly dependent on the user application and the software in particular.

Therefore it is recommended that the user applies EMC software optimization and prequalification tests in relation with the EMC level requested for his application.

Software recommendations

The software flowchart must include the management of runaway conditions such as:

- corrupted program counter
- unexpected reset
- critical data corruption (control registers...)

Prequalification trials

Most of the common failures (unexpected reset and program counter corruption) can be reproduced by manually forcing a low state on the reset pin or the oscillator pins for 1 second.

To complete these trials, ESD stress can be applied directly on the device, over the range of specification values. When unexpected behavior is detected, the software can be hardened to prevent unrecoverable errors occurring (see application note AN1015).

12.12 Communication interface characteristics

12.12.1 Serial peripheral interface (SPI)

The following characteristics are ubject to general operating conditions for V_{DD}, f_{CPU}, and T_A unless otherwise specified. The data is based on design simulation and/or characterization results, not tested in production.

When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration. Refer to the I/O port characteristics for more details on the input/output alternate function characteristics ($\overline{SS}$, SCK, MOSI, MISO).

Table 110. SPI characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master f_{CPU} = 8 MHz	$f_{CPU}/128 = 0.0625$	$f_{CPU}/4 = 2$	MHz
		Slave f_{CPU} = 8 MHz	0	$f_{CPU}/2 = 4$	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time		see I/O port pin description		
$t_{su(\overline{SS})}$ (1)	$\overline{SS}$ setup time(2)	Slave	t_{CPU} + 50		ns
$t_{h(\overline{SS})}$ (1)	$\overline{SS}$ hold time	Slave	120		
$t_{w(SCKH)}$ (1) $t_{w(SCKL)}$ (1)	SCK high and low time	Master Slave	100 90		
$t_{su(MI)}$ (1) $t_{su(SI)}$ (1)	Data input setup time	Master Slave	100 100		
$t_{h(MI)}$ (1) $t_{h(SI)}$ (1)	Data input hold time	Master Slave	100 100		
$t_{a(SO)}$ (1)	Data output access time	Slave	0	120	
$t_{dis(SO)}$ (1)	Data output disable time	Slave		240	
$t_{v(SO)}$ (1)	Data output valid time	Slave (after enable edge)		120	
$t_{h(SO)}$ (1)	Data output hold time		0		
$t_{v(MO)}$ (1)	Data output valid time	Master (after enable edge)		120	
$t_{h(MO)}$ (1)	Data output hold time		0		

1. Data based on design simulation and/or characterization results, not tested in production.
2. Depends on f_{CPU}. For example, if f_{CPU} = 8 MHz, then $t_{CPU} = 1 / f_{CPU}$ = 125ns and $t_{su(SS)}$ = 175ns.

13 Package characteristics

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: *www.st.com*. ECOPACK® is an ST trademark.

13.1 LQFP44 package characteristics

Figure 87. 44-pin low profile quad flat package outline

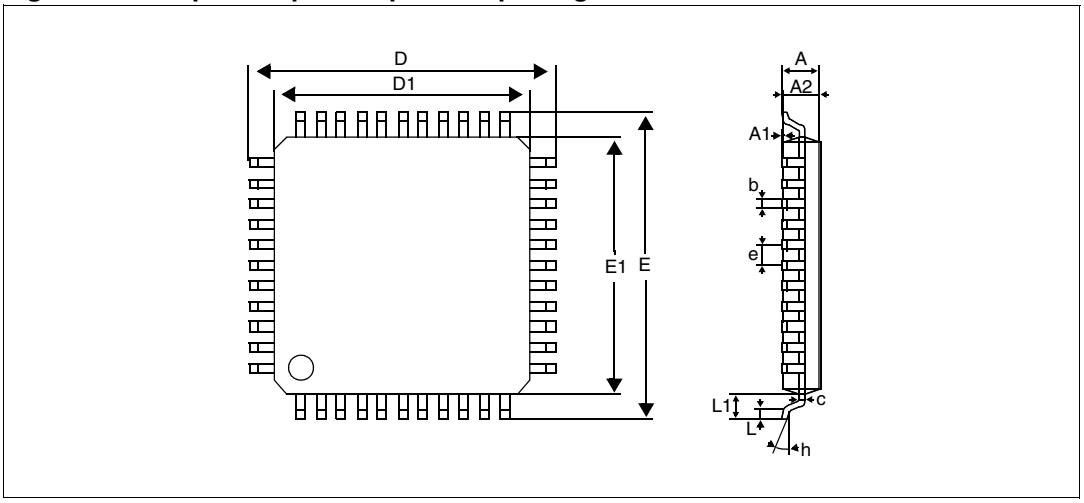


Table 113. 44-pin low profile quad flat package mechanical data

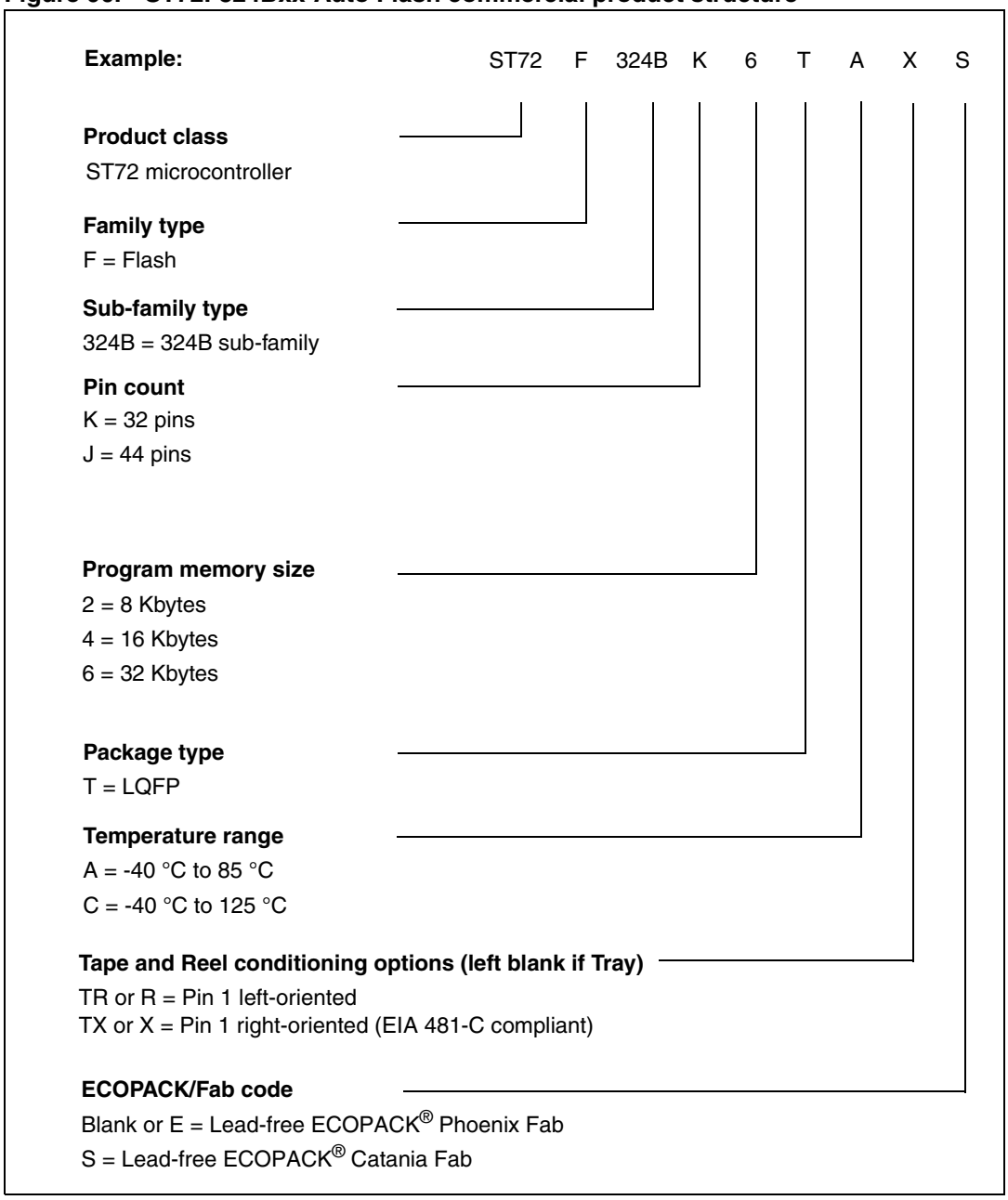
Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.30	0.37	0.45	0.012	0.015	0.018
C	0.09		0.20	0.004	0.000	0.008
D		12.00			0.472	
D1		10.00			0.394	
E		12.00			0.472	
E1		10.00			0.394	
e		0.80			0.031	
θ	0°	3.5°	7°	0°	3.5°	7°

Table 114. 32-pin low profile quad flat package mechanical data (continued)

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
L1		1.00			0.039	
Number of pins						
N	32					

14.1.2 Flash ordering information

Figure 90. ST72F324Bxx-Auto Flash commercial product structure



1. For a list of available options (e.g. memory size, package) and orderable part numbers or for further information on any aspect of this device, please go to *www.st.com* or contact the ST Sales Office nearest to you.

14.3 Development tools

14.3.1 Introduction

Development tools for the ST7 microcontrollers include a complete range of hardware systems and software tools from STMicroelectronics and third-party tool suppliers. The range of tools includes solutions to help you evaluate microcontroller peripherals, develop and debug your application, and program your microcontrollers.

14.3.2 Evaluation tools and starter kits

ST offers complete, affordable **starter kits** and full-featured **evaluation boards** that allow you to evaluate microcontroller features and quickly start developing ST7 applications. Starter kits are complete, affordable hardware/software tool packages that include features and samples to help you quickly start developing your application. ST evaluation boards are open-design, embedded systems, which are developed and documented to serve as references for your application design. They include sample application software to help you demonstrate, learn about and implement your ST7's features.

14.3.3 Development and debugging tools

Application development for ST7 is supported by fully optimizing **C Compilers** and the **ST7 Assembler-Linker** toolchain, which are all seamlessly integrated in the ST7 integrated development environments in order to facilitate the debugging and fine-tuning of your application. The Cosmic C Compiler is available in a free version that outputs up to 16 Kbytes of code.

The range of hardware tools includes cost effective ST7-DVP3 series emulators. These tools are supported by the ST7 Toolset from STMicroelectronics, which includes the STVD7 integrated development environment (IDE) with high-level language debugger, editor, project manager and integrated programming interface.

14.3.4 Programming tools

During the development cycle, the ST7-DVP3 and ST7-EMU3 series emulators and the RLink provide in-circuit programming capability for programming the Flash microcontroller on your application board.

ST also provides dedicated a low-cost dedicated in-circuit programmer, the ST7-STICK, as well as ST7 socket boards which provide all the sockets required for programming any of the devices in a specific ST7 subfamily on a platform that can be used with any tool with in-circuit programming capability for ST7.

For production programming of ST7 devices, ST's third-party tool partners also provide a complete range of gang and automated programming solutions, which are ready to integrate into your production environment.

For additional ordering codes for spare parts, accessories and tools available for the ST7 (including from third party manufacturers), refer to the online product selector at www.st.com/mcu.