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	CANbus, I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	48
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f521ar9tctr

2 Package pinout and pin description

2.1 Package pinout

Figure 2. 80-pin LQFP 14x14 package pinout

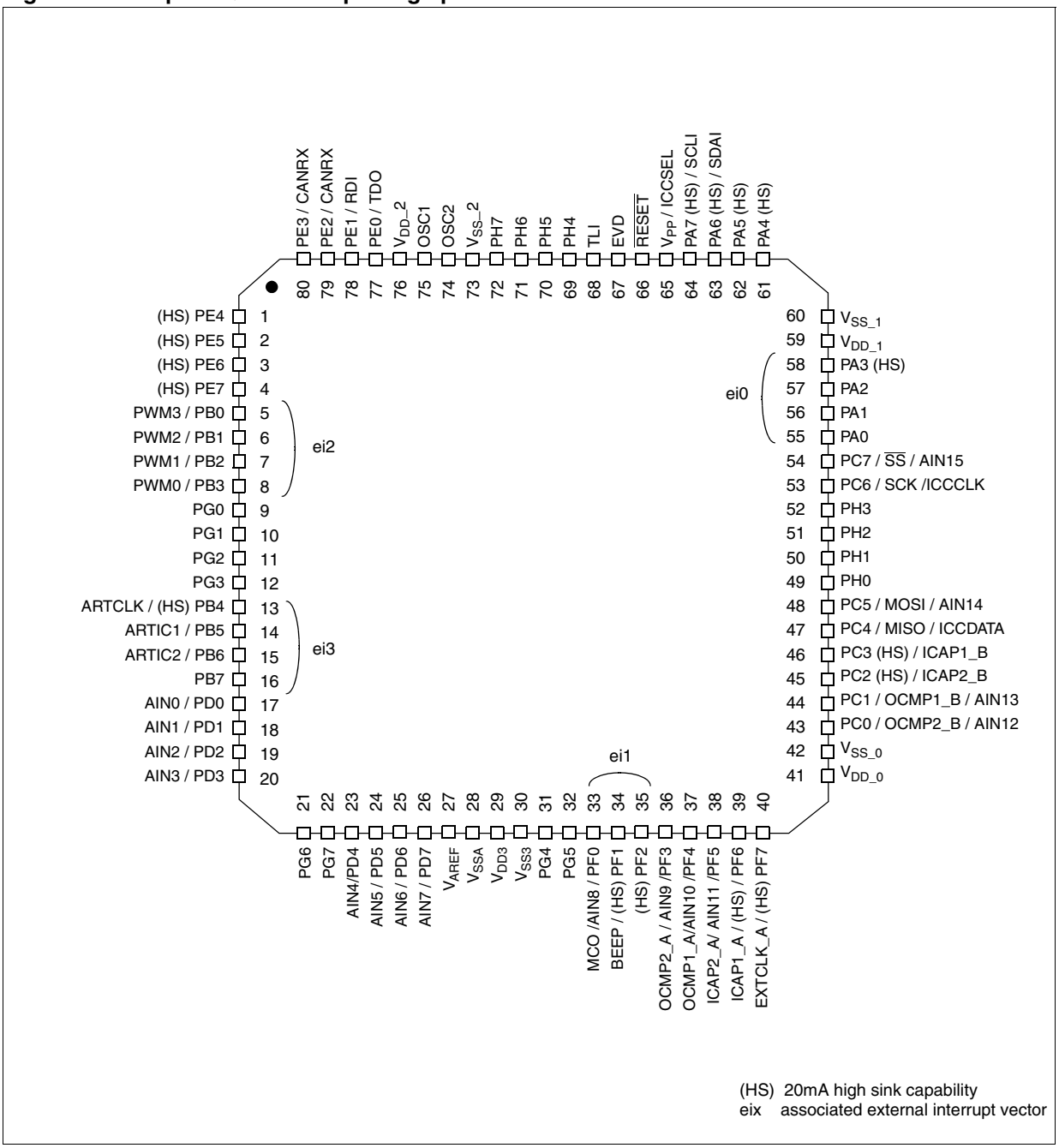
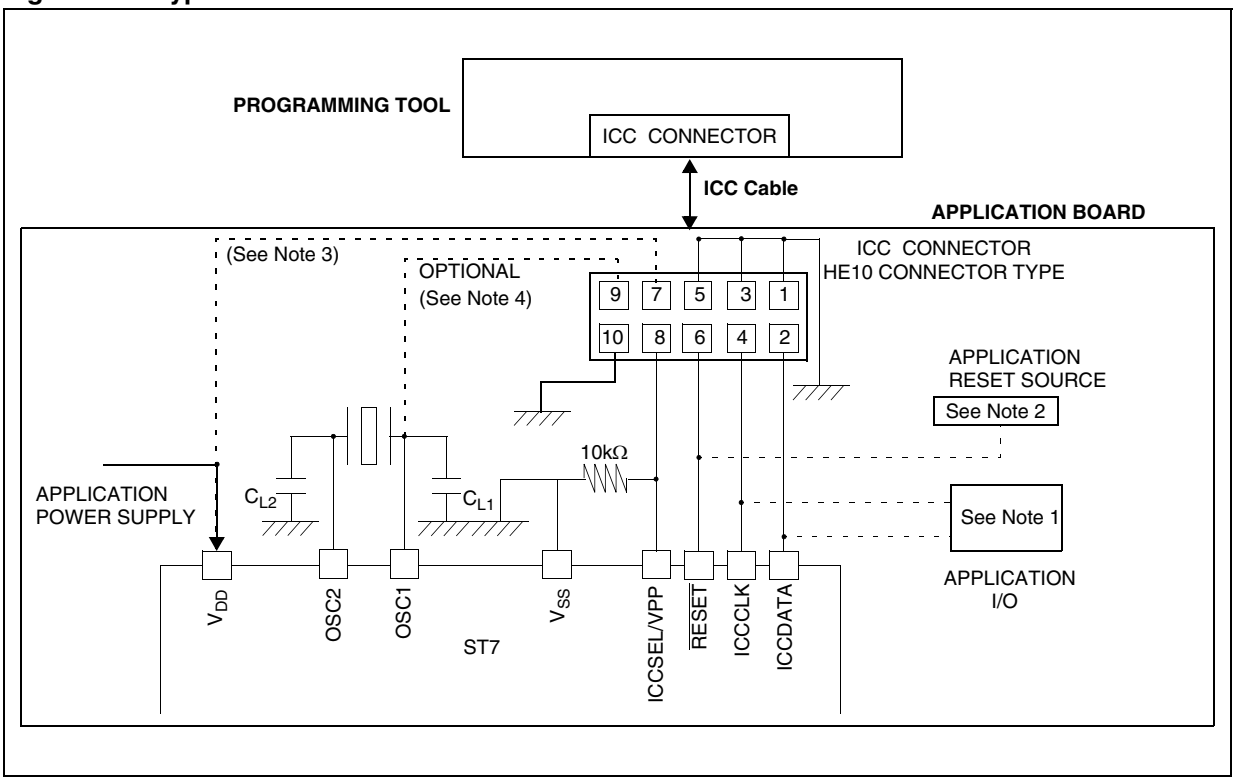


Figure 6. Typical ICC interface



1. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the programming tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to implemented in case another device forces the signal. Refer to the programming tool documentation for recommended resistor values.
2. During the ICC session, the programming tool must control the $\overline{\text{RESET}}$ pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5mA at high level (push-pull output or pull-up resistor < 1K). A schottky diode can be used to isolate the application RESET circuit in this case. When using a classical RC network with R > 1K or a reset management IC with open-drain output and pull-up resistor > 1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.
3. The use of Pin 7 of the ICC connector depends on the programming tool architecture. This pin must be connected when using most ST programming tools (it is used to monitor the application power supply). Please refer to the programming tool manual.
4. Pin 9 has to be connected to the OSC1 or OSCIN pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte. ST7 devices with multi-oscillator capability need to have OSC2 grounded in this case.

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 6*). For more details on the pin locations, refer to the device pinout description.

5.3.1 Accumulator (A)

The accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations as well as data manipulations.

5.3.2 Index registers (X and Y)

These 8-bit registers are used to create effective addresses or as temporary storage areas for data manipulation (the Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures.

5.3.3 Program counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

5.3.4 Condition code (CC) register

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.

CC Reset value: 111x1xxx

7	6	5	4	3	2	1	0
1	1	I1	H	I0	N	Z	C
		RW	RW	RW	RW	RW	RW

Table 7. Arithmetic management bits

Bit	Name	Function
4	H	*Half carry* This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instructions. It is reset by hardware during the same instructions. 0: No half carry has occurred. 1: A half carry has occurred. This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.
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 is a copy of the result 7th bit. 0: The result of the last operation is positive or null. 1: The result of the last operation is negative (that is, the most significant bit is a logic 1). This bit is accessed by the JRMI and JRPL instructions.

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
  - 1 maskable Top Level event: TLI

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 15*). The processing flow is shown in *Figure 17*.

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 *Table 20: Interrupt mapping* 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.

7.6 External interrupts

7.6.1 I/O port interrupt sensitivity

The external interrupt sensitivity is controlled by the IPA, IPB and ISxx bits of the EICR register (*Figure 21*). This control allows to have up to four fully independent external interrupt source sensitivities.

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

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

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

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

Figure 29. I/O port general block diagram

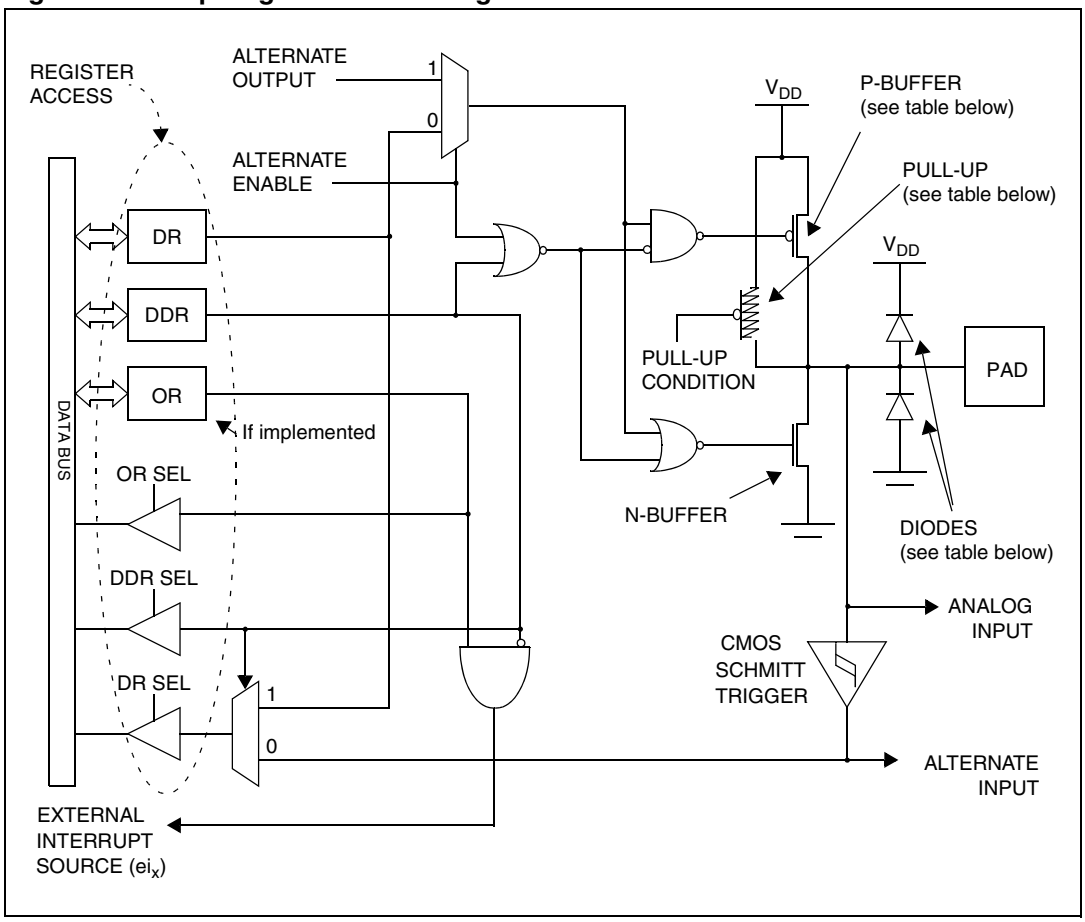


Table 29. I/O port mode options

Configuration mode		Pull-up	P-buffer	Diodes	
				to V_{DD}	to V_{SS}
Input	Floating with/without Interrupt	Off	Off	On	On
	Pull-up with/without Interrupt	On			
Output	Push-pull	Off	On		
	Open-drain (logic level)		Off		
	True open-drain	NI	NI	NI(1)	

1. The diode to V_{DD} is not implemented in the true open-drain pads. A local protection between the pad and V_{SS} is implemented to protect the device against positive stress.

Legend:

Off - Implemented not activated

On - Implemented and activated

NI - Not implemented

Table 34. I/O port register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
Reset value of all I/O port registers		0	0	0	0	0	0	0	0
0000h	PADR	MSB							LSB
0001h	PADDR								
0002h	PAOR								
0003h	PBDR	MSB							LSB
0004h	PBDDR								
0005h	PBOR								
0006h	PCDR	MSB							LSB
0007h	PCDDR								
0008h	PCOR								
0009h	PDDR	MSB							LSB
000Ah	PDDDR								
000Bh	PDOR								
000Ch	PEDR	MSB							LSB
000Dh	PEDDR								
000Eh	PEOR								
000Fh	PFDR	MSB							LSB
0010h	PFDDR								
0011h	PFOR								
0012h	PGDR	MSB							LSB
0013h	PGDDR								
0014h	PGOR								
0015h	PHDR	MSB							LSB
0016h	PHDDR								
0017h	PHOR								

Related documentation

SPI Communication between ST7 and EEPROM (AN 970)

S/W implementation of I2C bus master (AN1045)

Software LCD driver (AN1048)

Figure 37. PWM auto-reload timer function

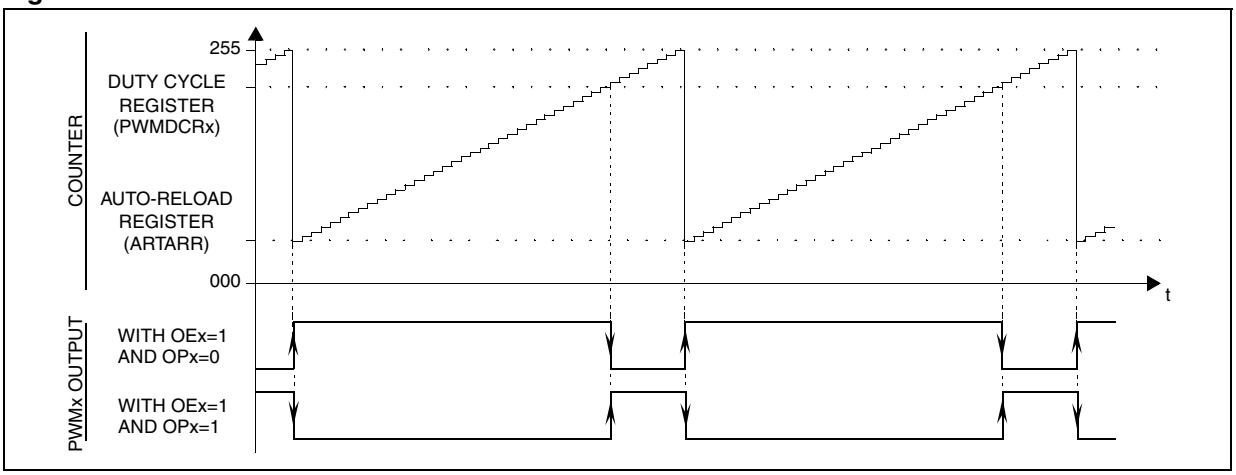
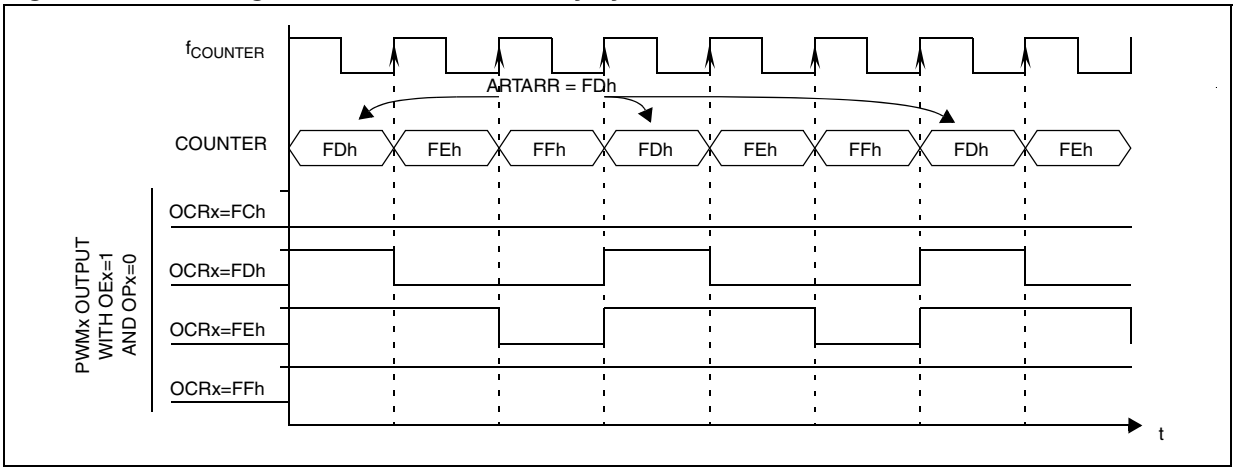


Figure 38. PWM signal from 0% to 100% duty cycle



12.2.6 Output compare and time base interrupt

On overflow, the OVF flag of the ARTCSR register is set and an overflow interrupt request is generated if the overflow interrupt enable bit, OIE, in the ARTCSR register, is set. The OVF flag must be reset by the user software. This interrupt can be used as a time base in the application.

12.2.7 External clock and event detector mode

Using the f_{EXT} external prescaler input clock, the auto-reload timer can be used as an external clock event detector. In this mode, the ARTARR register is used to select the n_{EVENT} number of events to be counted before setting the OVF flag.

$$n_{EVENT} = 256 - ARTARR$$

Caution: The external clock function is not available in Halt mode. If Halt mode is used in the application, prior to executing the HALT instruction, the counter must be disabled by clearing the TCE bit in the ARTCSR register to avoid spurious counter increments.

Table 58. Timer modes

Modes	Timer resources			
	Input Capture 1	Input Capture 2	Output Compare 1	Output Compare 2
One Pulse mode	No	Not recommended(1)	No	Partially(2)
PWM mode		Not recommended(3)		No

1. See *Note 4* in *Section 13.3.6 One Pulse mode*
2. See *Note 5* in *Section 13.3.6 One Pulse mode*
3. See *Note 4* in *Section 13.3.7 Pulse width modulation mode*

13.7 16-bit timer registers

Each timer is associated with 3 control and status registers, and with 6 pairs of data registers (16-bit values) relating to the 2 input captures, the 2 output compares, the counter and the alternate counter.

13.7.1 Control register 1 (CR1)

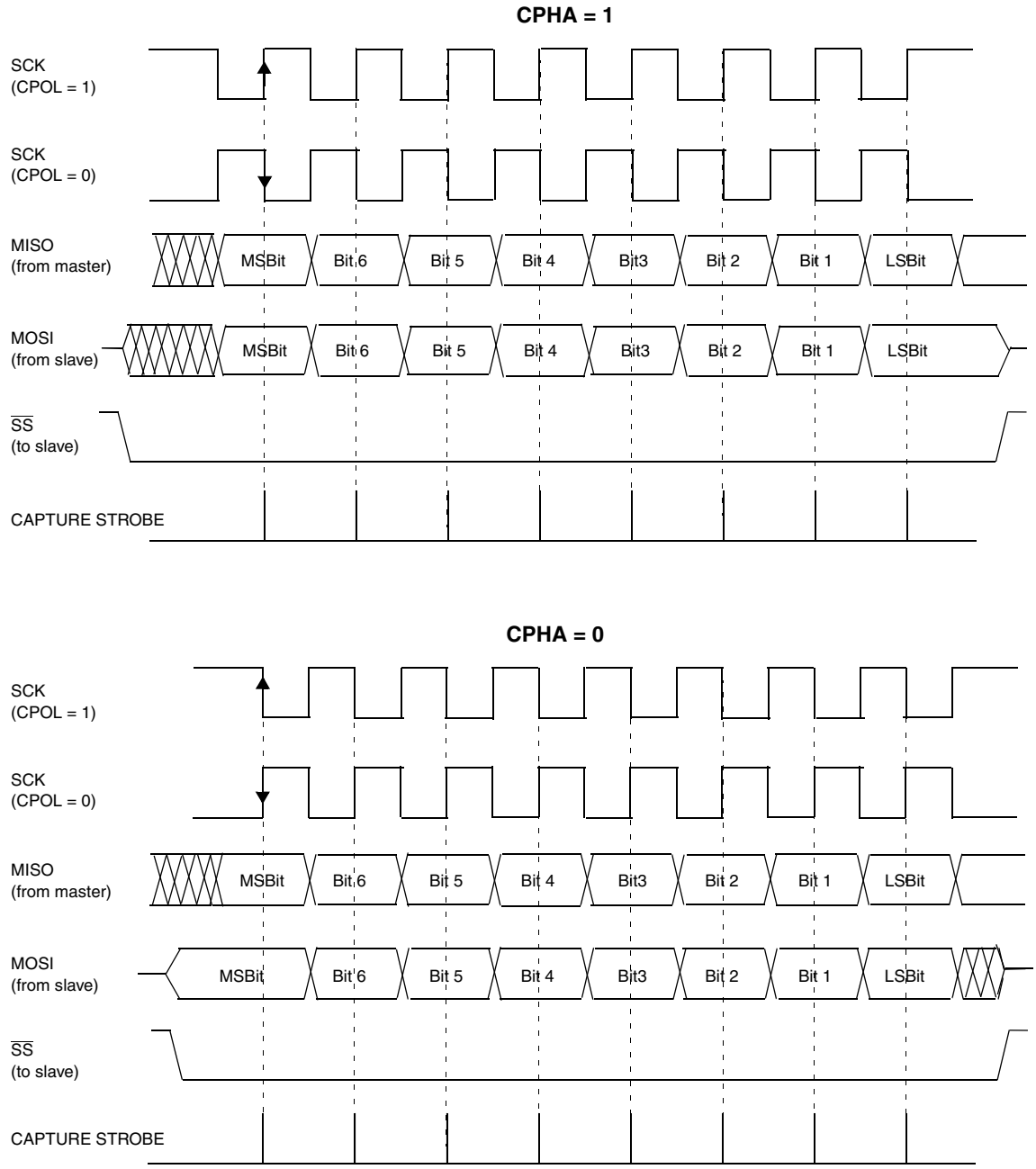
CR1 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
RW	RW	RW	RW	RW	RW	RW	RW

Table 59. CR1 register description

Bit	Name	Function
7	ICIE	*Input Capture Interrupt Enable* 0: Interrupt is inhibited 1: A timer interrupt is generated whenever the ICF1 or ICF2 bit of the SR register is set.
6	OCIE	*Output Compare Interrupt Enable* 0: Interrupt is inhibited 1: A timer interrupt is generated whenever the OCF1 or OCF2 bit of the SR register is set.
5	TOIE	*Timer Overflow Interrupt Enable* 0: Interrupt is inhibited 1: A timer interrupt is enabled whenever the TOF bit of the SR register is set.
4	FOLV2	*Forced Output Compare 2* This bit is set and cleared by software. 0: No effect on the OCMP2 pin 1: Forces the OLVL2 bit to be copied to the OCMP2 pin, if the OC2E bit is set and even if there is no successful comparison

Figure 59. Data clock timing diagram



Note: This figure should not be used as a replacement for parametric information.
Refer to Section 20: Electrical characteristics.

Table 77. SCIERPR register description

Bit	Name	Function
7:0	ERPR[7:0]	*8-bit Extended Receive Prescaler Register* The extended baud rate generator is activated when a value different from 00h is stored in this register. Therefore the clock frequency issued from the 16 divider (see *Figure 64*) is divided by the binary factor set in the SCIERPR register (in the range 1 to 255). The extended baud rate generator is not used after a reset.

15.7.7 Extended transmit prescaler division register (SCIETPR)

This register allows setting of the external prescaler rate division factor for the transmit circuit.

SCIETPR — Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ETPR[7:0]							
RW							

Table 78. SCIETPR register description

Bit	Name	Function
7:0	ETPR[7:0]	*8-bit Extended Transmit Prescaler Register* The extended baud rate generator is activated when a value different from 00h is stored in this register. Therefore the clock frequency issued from the 16 divider (see *Figure 64*) is divided by the binary factor set in the SCIETPR register (in the range 1 to 255). The extended baud rate generator is not used after a reset.

Table 79. Baud rate selection

Symbol	Parameter	Conditions			Standard	Baud rate	Unit
		f_{CPU}	Accuracy versus standard	Prescaler			
f_{Tx} f_{Rx}	Communication frequency	8 MHz	~0.16%	Conventional mode TR (or RR) = 128, PR = 13 TR (or RR) = 32, PR = 13 TR (or RR) = 16, PR = 13 TR (or RR) = 8, PR = 13 TR (or RR) = 4, PR = 13 TR (or RR) = 16, PR = 3 TR (or RR) = 2, PR = 13 TR (or RR) = 1, PR = 13	300 1200 2400 4800 9600 10400 19200 38400	~300.48 ~1201.92 ~2403.84 ~4807.69 ~9615.38 ~10416.67 ~19230.77 ~38461.54	Hz
			~0.79%	Extended mode ETPR (or ERPR) = 35, TR (or RR) = 1, PR = 1	14400	~14285.71	

Receive error counter register (RECR)

RECR Reset value: 00h

7	6	5	4	3	2	1	0
REC[7:0]							
RO							

Table 100. RECR register description

Bit	Name	Function
7:0	REC[7:0]	*Receive Error Counter* This is the Receive Error Counter implementing part of the fault confinement mechanism of the CAN protocol. In case of an error during reception, this counter is incremented by 1 or by 8 depending on the error condition as defined by the CAN standard. After every successful reception the counter is decremented by 1 or reset to 120 if its value was higher than 128. When the counter value exceeds 127, the CAN controller enters the error passive state.

Identifier high registers (IDHRx)

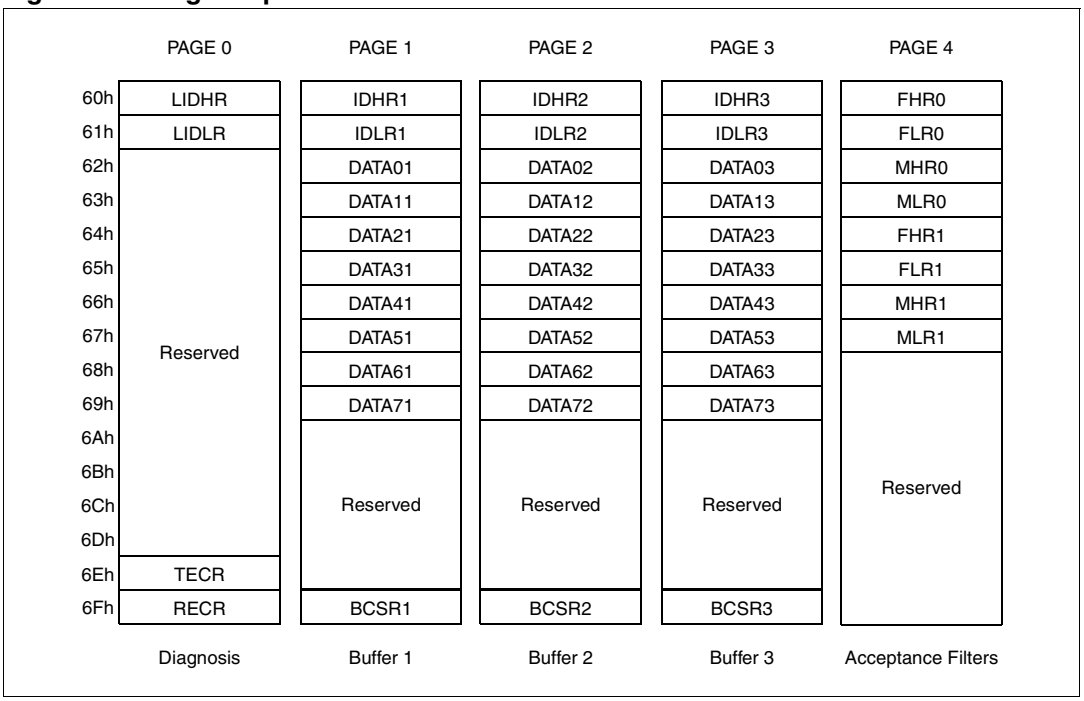
IDHRx Reset value: Undefined

7	6	5	4	3	2	1	0
ID[10:3]							
R/W							

Table 101. IDHRx register description

Bit	Name	Function
7:0	ID[10:3]	*Message Identifier (MSB)* These are the most significant 8 bits of the 11-bit message identifier. The identifier acts as the message's name, used for bus access arbitration and acceptance filtering.

Figure 76. Page maps



20.3 Operating conditions

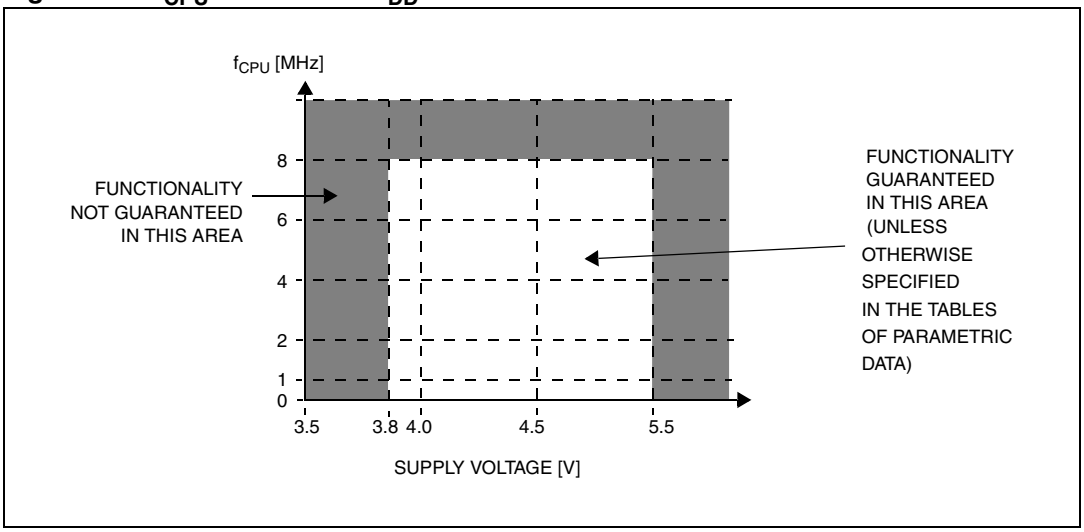
20.3.1 General operating conditions

Table 127. General operating conditions

Symbol	Parameter	Conditions	Min	Max	Unit
f_{CPU}	Internal clock frequency		0	8	MHz
V_{DD}	Standard voltage range (except Flash Write/Erase)		3.8	5.5	V
	Operating voltage for Flash Write/Erase	V_{PP} = 11.4 to 12.6V	4.5	5.5	
T_A	Ambient temperature range	A suffix version	-40	85	°C
		B suffix version		105	
		C suffix version		125	

Note: Some temperature ranges are only available with a specific package and memory size. Refer to Section 22: Device configuration and ordering information on page 257.

Figure 88. f_{CPU} max versus V_{DD}



20.5.3 Crystal and ceramic resonator oscillators

The ST7 internal clock can be supplied with four different crystal/ceramic resonator oscillators. All the information given in this paragraph is based on characterization results with specified typical external components. In the application, the resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time. Refer to the crystal/ceramic resonator manufacturer for more details (such as frequency, package or accuracy).

Table 136. Crystal and ceramic resonator oscillators

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
f_{OSC}	Oscillator frequency(1)	LP: Low power oscillator MP: Medium power oscillator MS: Medium speed oscillator HS: High speed oscillator		1 >2 >4 >8	-	2 4 8 16	MHz
R_F	Feedback resistor(2)	-		20	-	40	kΩ
C_{L1} C_{L2}	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator (R_S)(3)	R_S = 200Ω R_S = 200Ω R_S = 200Ω R_S = 100Ω	LP oscillator MP oscillator MS oscillator HS oscillator	22 22 18 15	-	56 46 33 33	pF
i_2	OSC2 driving current	V_{DD} = 5V, V_{IN} = V_{SS}	LP oscillator MP oscillator MS oscillator HS oscillator	-	80 160 310 610	150 250 460 910	µA

1. The oscillator selection can be optimized in terms of supply current using a high-quality resonator with small R_S value. Refer to crystal/ceramic resonator manufacturer for more details.
2. Data based on characterization results, not tested in production. The relatively low value of the RF resistor offers a good protection against issues resulting from use in a humid environment, due to the induced leakage and the bias condition change. However, it is recommended to take this point into account if the microcontroller is used in tough humidity conditions.
3. For C_{L1} and C_{L2} it is recommended to use high-quality ceramic capacitors in the 5pF to 25pF range (typ.) designed for high-frequency applications and selected to match the requirements of the crystal or resonator. C_{L1} and C_{L2}, are usually the same size. The crystal manufacturer typically specifies a load capacitance which is the series combination of C_{L1} and C_{L2}. PCB and MCU pin capacitance must be included when sizing C_{L1} and C_{L2} (10pF can be used as a rough estimate of the combined pin and board capacitance).

Figure 94. Typical application with a crystal or ceramic resonator

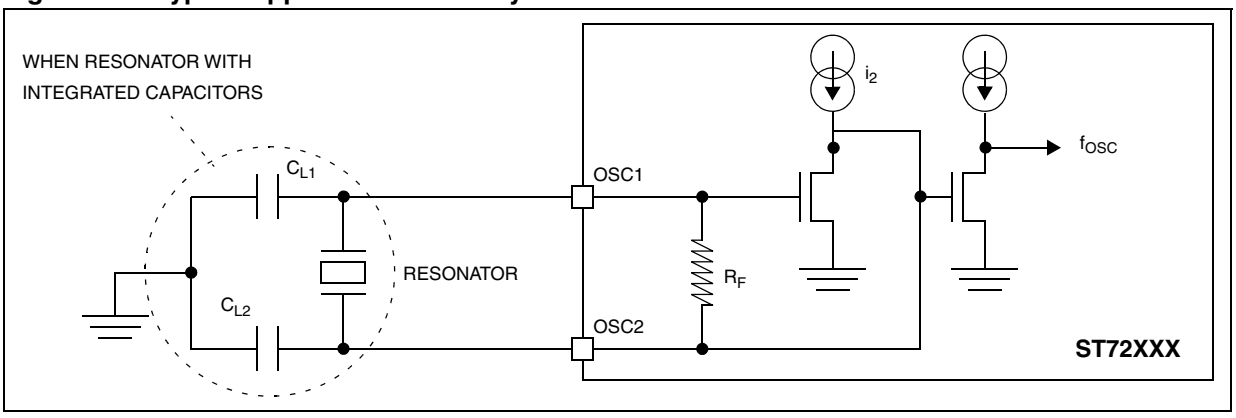


Figure 118. 64-pin (14x14) low profile quad flat package outline

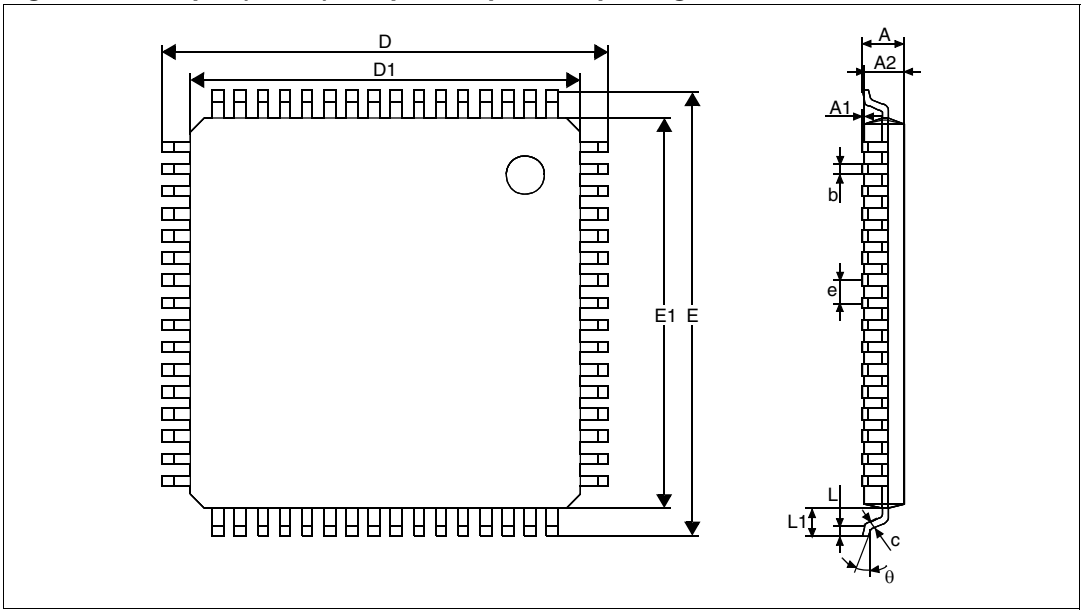


Table 159. 64-pin (14x14) low profile quad flat package mechanical data

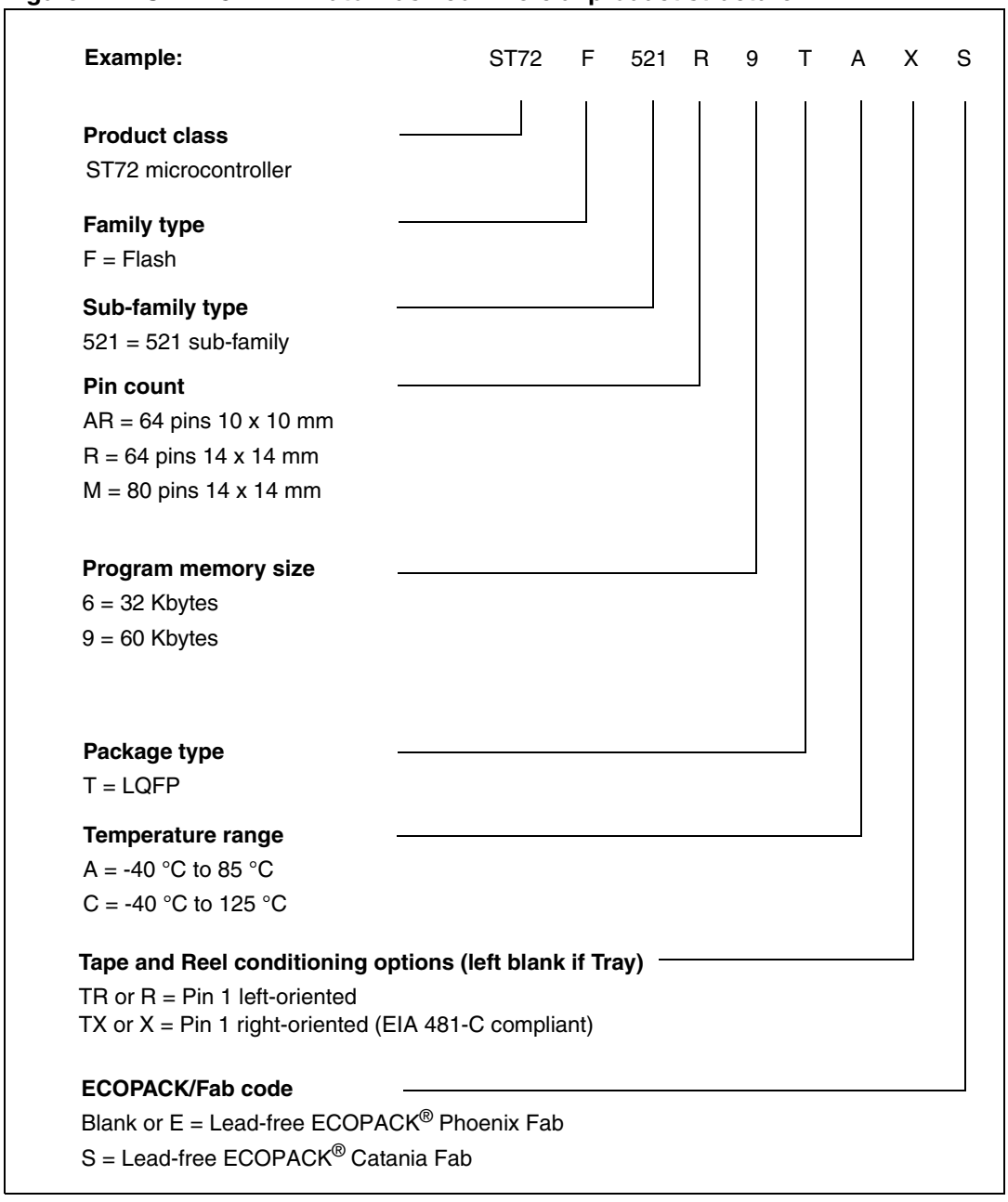
Dimension	mm			inches[1]		
	Min	Typ	Max	Min	Typ	Max
A			1.600			0.0630
A1	0.050		0.150	0.0020		0.0059
A2	1.350	1.400	1.450	0.0531	0.0551	0.0571
b	0.300	0.370	0.450	0.0118	0.0146	0.0177
c	0.090		0.200	0.0035		0.0079
D		16.000			0.6299	
D1		14.000			0.5512	
E		16.000			0.6299	
E1		14.000			0.5512	
e		0.800			0.0315	
θ	0°	3.5°	7°	0°	3.5°	7°
L	0.450	0.600	0.750	0.0177	0.0236	0.0295
L1		1.000			0.0394	

1. Values in inches are converted from mm and rounded to 4 decimal digits.

22.1.2 Flash ordering information

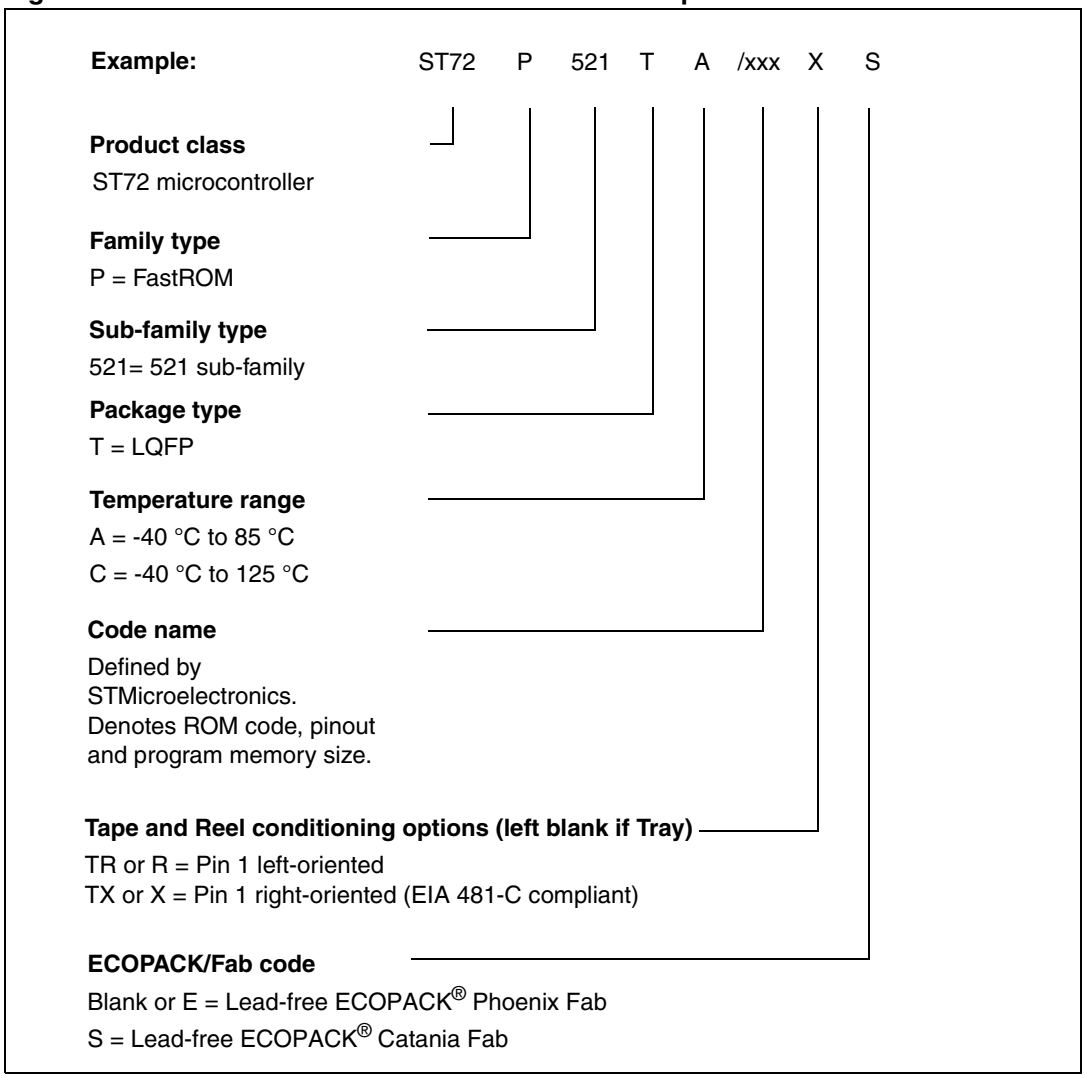
The following *Figure 121* serves as a guide for ordering.

Figure 121. ST72F521xxx-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.

Figure 122. ST72P521xxx-Auto FastROM commercial product structure



23.1.7 SCI wrong break duration

Description

A single break character is sent by setting and resetting the SBK bit in the SCICR2 register. In some cases, the break character may have a longer duration than expected:

- 20 bits instead of 10 bits if M = 0
- 22 bits instead of 11 bits if M = 1

In the same way, as long as the SBK bit is set, break characters are sent to the TDO pin. This may lead to generating one break more than expected.

Occurrence

The occurrence of the problem is random and proportional to the baud rate. With a transmit frequency of 19200 baud (f_{CPU} = 8 MHz and SCIBRR = 0xC9), the wrong break duration occurrence is around 1%.

Workaround

If this wrong duration is not compliant with the communication protocol in the application, software can request that an Idle line be generated before the break character. In this case, the break duration is always correct assuming the application is not doing anything between the idle and the break. This can be ensured by temporarily disabling interrupts.

The exact sequence is:

- Disable interrupts
- Reset and Set TE (IDLE request)
- Set and Reset SBK (Break Request)
- Re-enable interrupts

23.1.8 16-bit timer PWM mode

In PWM mode, the first PWM pulse is missed after writing the value FFFCh in the OC1R register (OC1HR, OC1LR). It leads to either full or no PWM during a period, depending on the OLVL1 and OLVL2 settings.