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	16MHz
Connectivity	I²C, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	22
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	3.2V ~ 5.5V
Data Converters	A/D 6x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SO
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72c254g2m3

Table of Contents

3 REGISTER & MEMORY MAP

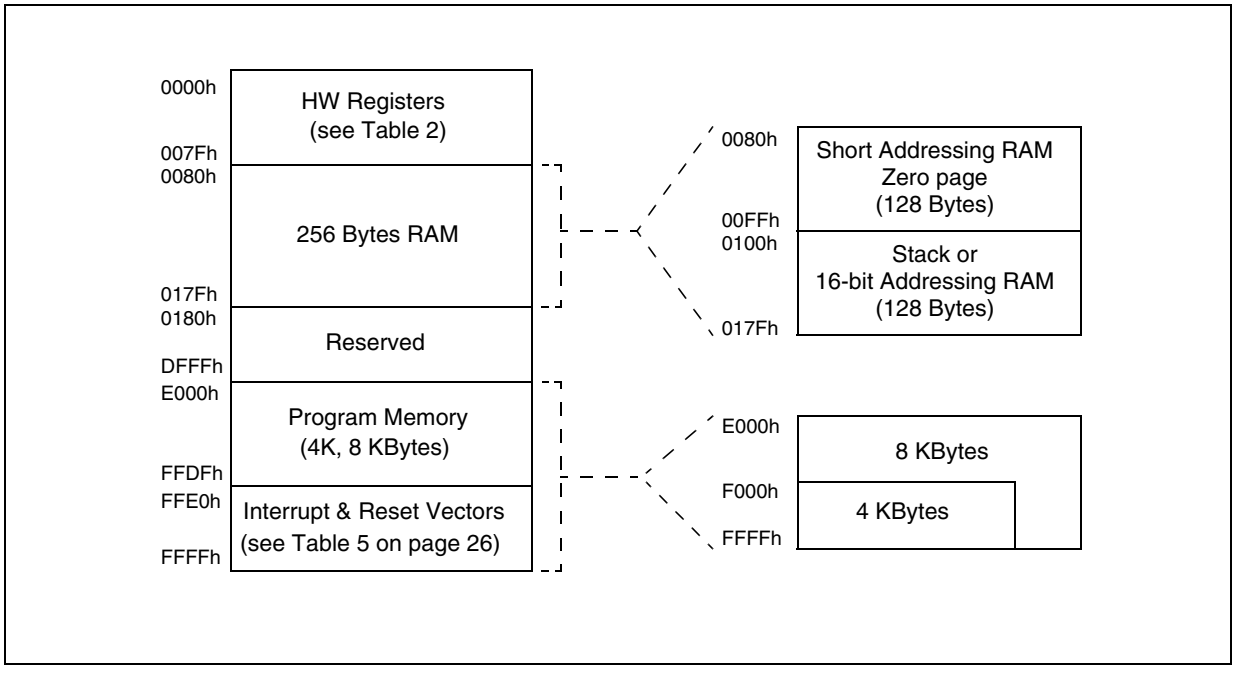
As shown in the Figure 4, the MCU is capable of addressing 64K bytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register location, 256 bytes of RAM and up to 8Kbytes of user program memory. The RAM space includes up to 128 bytes for the stack from 0100h to 017Fh.

The highest address bytes contain the user reset and interrupt vectors.

IMPORTANT: Memory locations marked as "Reserved" must never be accessed. Accessing a reserved area can have unpredictable effects on the device.

Figure 4. Memory Map



CPU REGISTERS (cont'd)

CONDITION CODE REGISTER (CC)

Read/Write

Reset Value: 111x1xxx

7							0
1	1	1	H	I	N	Z	C

The 8-bit Condition Code register contains the interrupt mask 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.

Bit 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 instruction. 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 3 = **I** *Interrupt mask*

This bit is set by hardware when entering in interrupt or by software to disable all interrupts except the TRAP software interrupt. This bit is cleared by software.
0: Interrupts are enabled.
1: Interrupts are disabled.

This bit is controlled by the RIM, SIM and IRET instructions and is tested by the JRM and JRNM instructions.

Note: Interrupts requested while I is set are latched and can be processed when I is cleared. By default an interrupt routine is not interruptible because the I bit is set by hardware at the start of the routine and reset by the IRET instruction at the end of the routine. If the I bit is cleared by software in the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine.

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 is a copy of the 7^{th} bit of the result.
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.

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.

7 INTERRUPTS

The ST7 core may be interrupted by one of two different methods: maskable hardware interrupts as listed in the Interrupt Mapping Table and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 1.

The maskable interrupts must be enabled by clearing the I bit in order to be serviced. However, disabled interrupts may be latched and processed when they are enabled (see external interrupts subsection).

Note: After reset, all interrupts are disabled.

When an interrupt 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.

– The I bit of the CC register is set to prevent additional interrupts.

– 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 the Interrupt Mapping Table for vector addresses).

The interrupt service routine should finish 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 I bit will be cleared and the main program will resume.

Priority Management

By default, a servicing interrupt cannot be interrupted because the I bit is set by hardware entering in interrupt routine.

In the case when several interrupts are simultaneously pending, an hardware priority defines which one will be serviced first (see the Interrupt Mapping Table).

Interrupts and Low Power Mode

All interrupts allow the processor to leave the WAIT low power mode. Only external and specifically mentioned interrupts allow the processor to leave the HALT low power mode (refer to the “Exit from HALT“ column in the Interrupt Mapping Table).

7.1 NON-MASKABLE SOFTWARE INTERRUPT

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It will be serviced according to the flowchart on Figure 1.

7.2 EXTERNAL INTERRUPTS

External interrupt vectors can be loaded into the PC register if the corresponding external interrupt occurred and if the I bit is cleared. These interrupts allow the processor to leave the Halt low power mode.

The external interrupt polarity is selected through the miscellaneous register or interrupt register (if available).

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

If several input pins, connected to the same interrupt vector, are configured as interrupts, their signals are logically NANDed before entering the edge/level detection block.

Caution: The type of sensitivity defined in the Miscellaneous or Interrupt register (if available) applies to the ei source. In case of a NANDed source (as described on the I/O ports section), a low level on an I/O pin configured as input with interrupt, masks the interrupt request even in case of rising-edge sensitivity.

7.3 PERIPHERAL INTERRUPTS

Different peripheral interrupt flags in the status register are able to cause an interrupt when they are active if both:

– The I bit of the CC register is cleared.

– The corresponding enable bit is set in the control register.

If any of these two conditions is false, the interrupt is latched and thus remains pending.

Clearing an interrupt request is done by:

– Writing “0” to the corresponding bit in the status register or

– Access to the status register while the flag is set followed by a read or write of an associated register.

Note: The clearing sequence resets the internal latch. A pending interrupt (that is, waiting to be enabled) will therefore be lost if the clear sequence is executed.

INTERRUPTS (Cont'd)

Figure 15. Interrupt Processing Flowchart

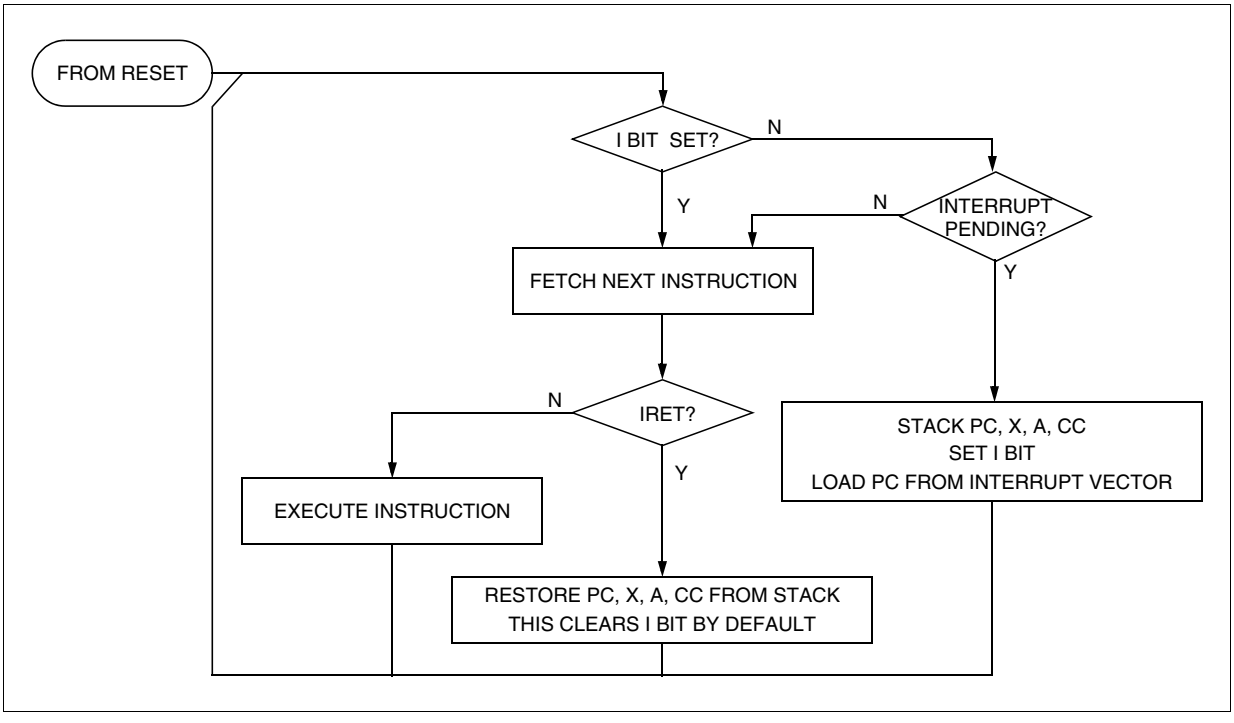


Table 5. Interrupt Mapping

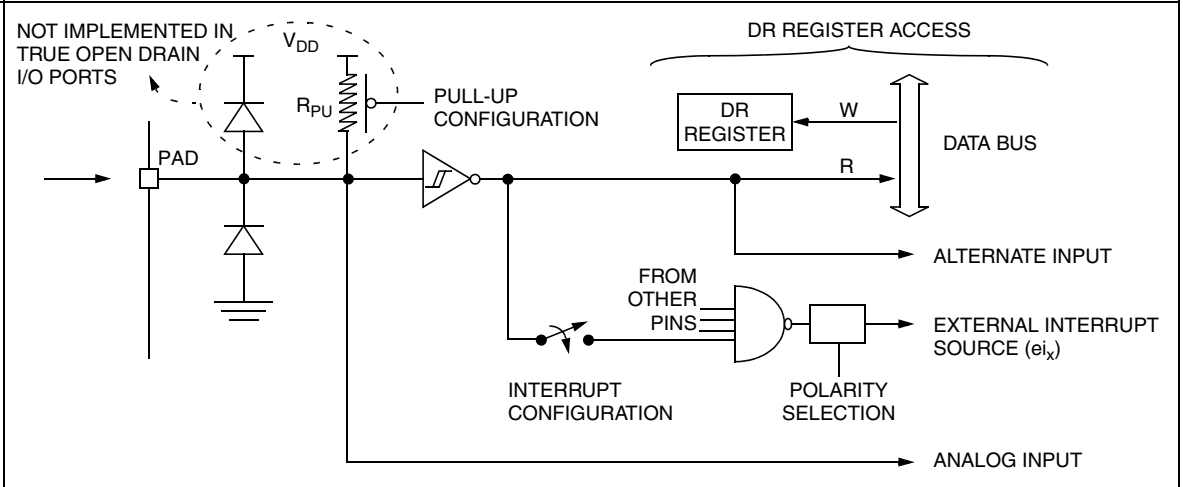
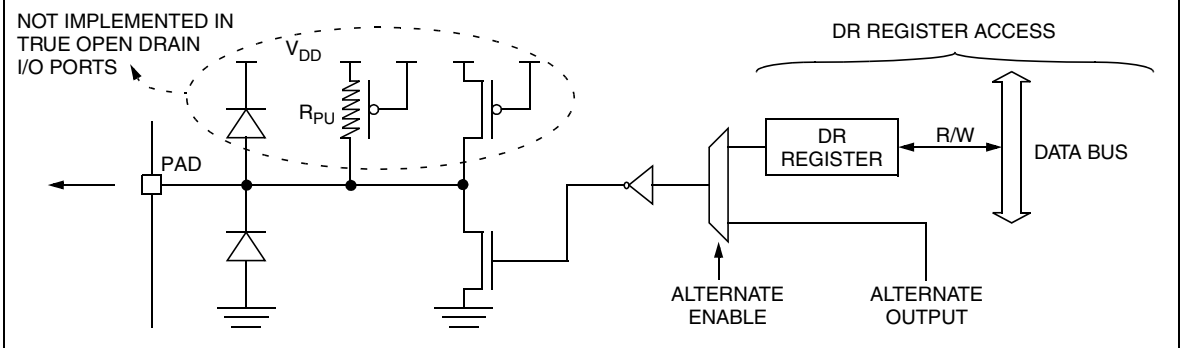
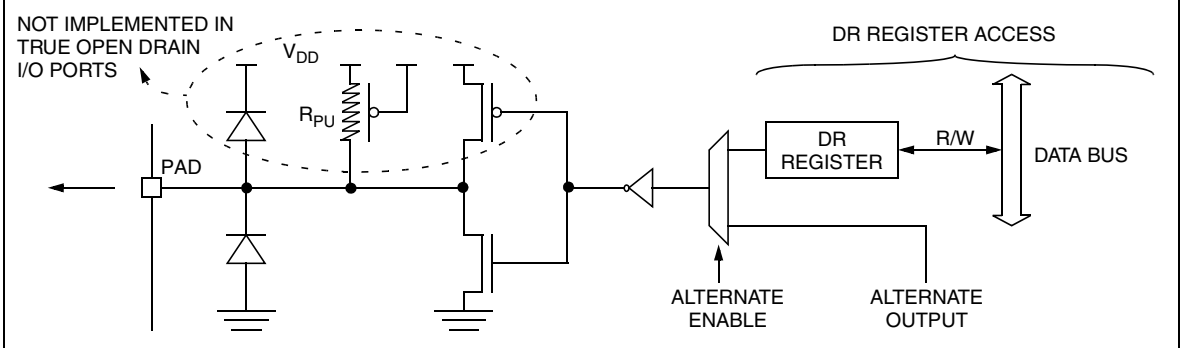
N°	Source Block	Description	Register Label	Priority Order	Exit from HALT	Address Vector
	RESET	Reset	N/A	Highest Priority	yes	FFFEh-FFFFh
	TRAP	Software Interrupt			no	FFFCh-FFFDh
0	ei0	External Interrupt Port A7..0 (C5..0[1])			yes	FFFAh-FFFBh
1	ei1	External Interrupt Port B7..0 (C5..0[1])				FFF8h-FFF9h
2	CSS	Clock Security System Interrupt	CRSR		no	FFF6h-FFF7h
3	SPI	SPI Peripheral Interrupts	SPISR			FFF4h-FFF5h
4	TIMER A	TIMER A Peripheral Interrupts	TASR			FFF2h-FFF3h
5		Not used				FFF0h-FFF1h
6	TIMER B	TIMER B Peripheral Interrupts	TBSR		no	FFEEh-FFEFh
7		Not used				FFECh-FFEDh
8		Not used				FFEAh-FFEBh
9		Not used				FFE8h-FFE9h
10		Not used				FFE6h-FFE7h
11	I²C	I²C Peripheral Interrupt	I2CSRx		no	FFE4h-FFE5h
12		Not Used		Lowest Priority		FFE2h-FFE3h
13		Not Used				FFE0h-FFE1h

Note

1. Configurable by option byte.

I/O PORTS (Cont'd)

Table 7. I/O Port Configurations

	Hardware Configuration
INPUT [1]	
OPEN-DRAIN OUTPUT [2]	
PUSH-PULL OUTPUT [2]	

Notes:

1. When the I/O port is in input configuration and the associated alternate function is enabled as an output, reading the DR register will read the alternate function output status.
2. When the I/O port is in output configuration and the associated alternate function is enabled as an input, the alternate function reads the pin status given by the DR register content.

16-BIT TIMER (Cont'd)

11.2.3.3 Input Capture

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

The two input capture 16-bit registers (IC1R and IC2R) are used to latch the value of the free running counter after a transition is detected by the ICAP*i* pin (see Figure 5).

	MS Byte	LS Byte
ICiR	IC*i*HR	IC*i*LR

The IC*i*R register is a read-only register.

The active transition is software programmable through the IEDG*i* bit of Control Registers (CR*i*).

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

Procedure:

To use the input capture function, select the following in the CR2 register:

– Select the timer clock (CC[1:0]) (see Table 1).

– Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit (the ICAP2 pin must be configured as a floating input or input with pull-up without interrupt if this configuration is available).

And select the following in the CR1 register:

– Set the ICIE bit to generate an interrupt after an input capture coming from either the ICAP1 pin or the ICAP2 pin

– Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin must be configured as a floating input or input with pull-up without interrupt if this configuration is available).

When an input capture occurs:

– The ICF*i* bit is set.

– The IC*i*R register contains the value of the free running counter on the active transition on the ICAP*i* pin (see Figure 6).

– A timer interrupt is generated if the ICIE bit is set and the I bit is cleared in the CC register. Otherwise, the interrupt remains pending until both conditions become true.

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

1. Reading the SR register while the ICF*i* bit is set.
2. An access (read or write) to the IC*i*LR register.

Notes:

1. After reading the IC*i*HR register, the transfer of input capture data is inhibited and ICF*i* will never be set until the IC*i*LR register is also read.
2. The IC*i*R register contains the free running counter value which corresponds to the most recent input capture.
3. The two input capture functions can be used together even if the timer also uses the two output compare functions.
4. In One Pulse mode and PWM mode only the input capture 2 function can be used.
5. The alternate inputs (ICAP1 and ICAP2) are always directly connected to the timer. So any transitions on these pins activate the input capture function.
   Moreover if one of the ICAP*i* pin is configured as an input and the second one as an output, an interrupt can be generated if the user toggles the output pin and if the ICIE bit is set.
   This can be avoided if the input capture function *i* is disabled by reading the IC*i*HR (see note 1).
6. The TOF bit can be used with an interrupt in order to measure events that exceed the timer range (FFFFh).

16-BIT TIMER (Cont'd)

OUTPUT COMPARE 2 HIGH REGISTER (OC2HR)

Read/Write
Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

7							0
MSB							LSB

OUTPUT COMPARE 2 LOW REGISTER (OC2LR)

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

7							0
MSB							LSB

COUNTER HIGH REGISTER (CHR)

Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

7							0
MSB							LSB

COUNTER LOW REGISTER (CLR)

Read Only
Reset Value: 1111 1100 (FCh)

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 accessing the SR register clears the TOF bit.

7							0
MSB							LSB

ALTERNATE COUNTER HIGH REGISTER (ACHR)

Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

7							0
MSB							LSB

ALTERNATE COUNTER LOW REGISTER (ACLR)

Read Only
Reset Value: 1111 1100 (FCh)

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 SR register does not clear the TOF bit in SR register.

7							0
MSB							LSB

INPUT CAPTURE 2 HIGH REGISTER (IC2HR)

Read Only
Reset Value: Undefined

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

7							0
MSB							LSB

INPUT CAPTURE 2 LOW REGISTER (IC2LR)

Read Only
Reset Value: Undefined

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

7							0
MSB							LSB

SERIAL PERIPHERAL INTERFACE (Cont'd)

11.3.4.2 Slave Configuration

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

The value of the SPR0 & SPR1 bits is not used for the data transfer.

Procedure

– For correct data transfer, the slave device must be in the same timing mode as the master device (CPOL and CPHA bits). See Figure 4.
– The $\overline{SS}$ pin must be connected to a low level signal during the complete byte transmit sequence.
– Clear the MSTR bit and set the SPE bit to assign the pins to alternate function.

In this configuration the MOSI pin is a data input and the MISO pin is a data output.

Transmit Sequence

The data byte is parallel loaded into the 8-bit shift register (from the internal bus) during a write cycle 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 is generated if SPIE bit is set and I bit in CCR register is cleared.

During the last clock cycle the SPIF bit is set, a copy of the data byte received in the shift register is moved to a buffer. When the DR register is read, the SPI peripheral returns this buffered value.

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

1. An access to the SR register while the SPIF bit is set.

2.A read to the DR register.

Notes: While the SPIF bit is set, all writes to the DR register are inhibited until the SR register is read.

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an overrun condition (see Section 0.1.4.6).

Depending on the CPHA bit, the $\overline{SS}$ pin has to be set to write to the DR register between each data byte transfer to avoid a write collision (see Section 0.1.4.4).

SERIAL PERIPHERAL INTERFACE (Cont'd)

11.3.7 Register Description

CONTROL REGISTER (CR)

Read/Write

Reset Value: 0000xxxx (0xh)

7							0
SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0

Bit 7 = **SPIE** *Serial peripheral interrupt enable.*
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SPI interrupt is generated whenever SPIF=1 or MODF=1 in the SR register

Bit 6 = **SPE** *Serial peripheral output enable.*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$=0 (see Section 0.1.4.5 Master Mode Fault).
0: I/O port connected to pins
1: SPI alternate functions connected to pins

The SPE bit is cleared by reset, so the SPI peripheral is not initially connected to the external pins.

Bit 5 = **SPR2** *Divider Enable*.

this bit is set and cleared by software and it is cleared by reset. It is used with the SPR[1:0] bits to set the baud rate. Refer to Table 1.
0: Divider by 2 enabled
1: Divider by 2 disabled

Bit 4 = **MSTR** *Master.*
This bit is set and cleared by software. It is also cleared by hardware when, in master mode, $\overline{SS}$=0 (see Section 0.1.4.5 Master Mode Fault).
0: Slave mode is selected
1: Master mode is selected, the function of the SCK pin changes from an input to an output and the functions of the MISO and MOSI pins are reversed.

Bit 3 = **CPOL** *Clock polarity.*
This bit is set and cleared by software. This bit determines the steady state of the serial Clock. The CPOL bit affects both the master and slave modes.
0: The steady state is a low value at the SCK pin.
1: The steady state is a high value at the SCK pin.

Bit 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.

Bit 1:0 = **SPR[1:0]** *Serial peripheral rate.*
These bits are set and cleared by software.Used with the SPR2 bit, they select one of six baud rates to be used as the serial clock when the device is a master.

These 2 bits have no effect in slave mode.

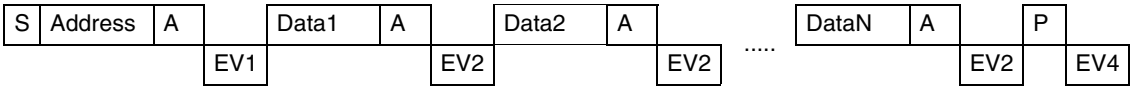
Table 15. Serial Peripheral Baud Rate

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

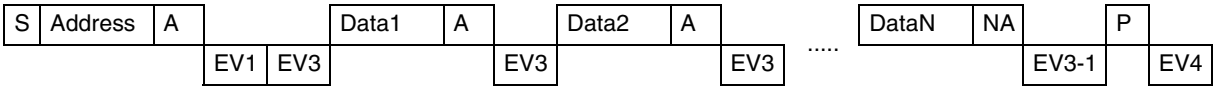
I²C BUS INTERFACE (Cont'd)

Figure 45. Transfer Sequencing

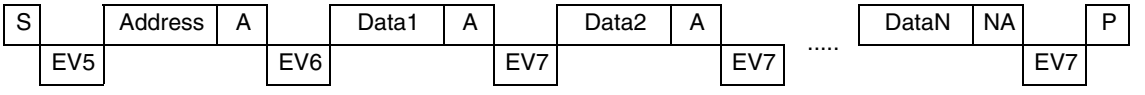
7-bit Slave receiver:



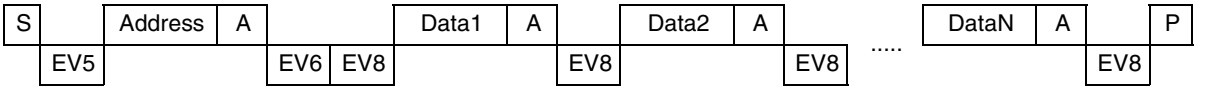
7-bit Slave transmitter:



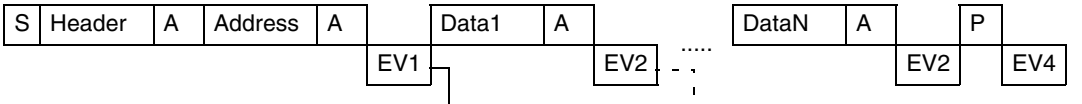
7-bit Master receiver:



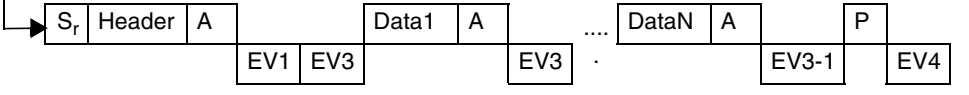
7-bit Master transmitter:



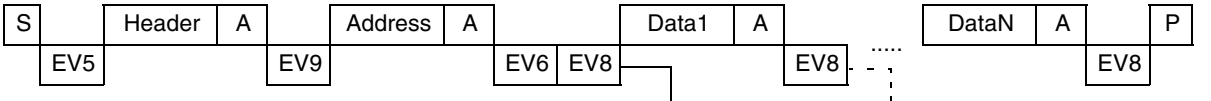
10-bit Slave receiver:



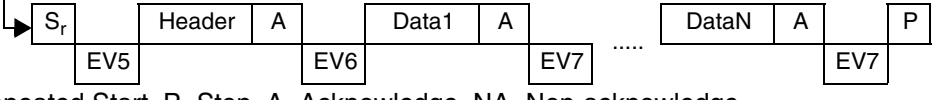
10-bit Slave transmitter:



10-bit Master transmitter



10-bit Master receiver:



Legend: S=Start, S_r = Repeated Start, P=Stop, A=Acknowledge, NA=Non-acknowledge, EVx=Event (with interrupt if ITE=1)

EV1: EVF=1, ADSL=1, cleared by reading SR1 register.

EV2: EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.

EV3: EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.

EV3-1: EVF=1, AF=1, BTF=1; AF is cleared by reading SR1 register. BTF is cleared by releasing the lines (STOP=1, STOP=0) or by writing DR register (DR=FFh). **Note:** If lines are released by STOP=1, STOP=0, the subsequent EV4 is not seen.

EV4: EVF=1, STOPF=1, cleared by reading SR2 register.

EV5: EVF=1, SB=1, cleared by reading SR1 register followed by writing DR register.

EV6: EVF=1, cleared by reading SR1 register followed by writing CR register (for example PE=1).

EV7: EVF=1, BTF=1, cleared by reading SR1 register followed by reading DR register.

EV8: EVF=1, BTF=1, cleared by reading SR1 register followed by writing DR register.

EV9: EVF=1, ADD10=1, cleared by reading SR1 register followed by writing DR register.

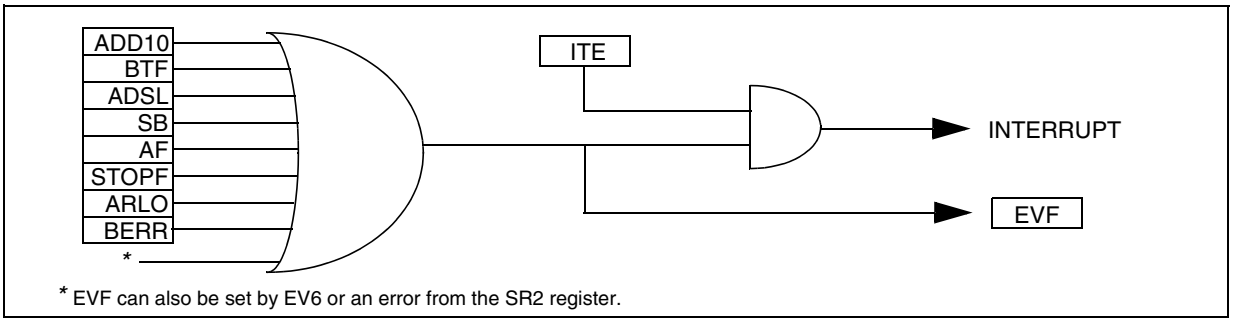
I²C BUS INTERFACE (Cont'd)

11.4.5 Low Power Modes

Mode	Description
WAIT	No effect on I²C interface. I²C interrupts cause the device to exit from WAIT mode.
HALT	I²C registers are frozen. In HALT mode, the I²C interface is inactive and does not acknowledge data on the bus. The I²C interface resumes operation when the MCU is woken up by an interrupt with "exit from HALT mode" capability.

11.4.6 Interrupts

Figure 46. Event Flags and Interrupt Generation



* EVF can also be set by EV6 or an error from the SR2 register.

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
10-bit Address Sent Event (Master mode)	ADD10	ITE	Yes	No
End of Byte Transfer Event	BTF		Yes	No
Address Matched Event (Slave mode)	ADSL		Yes	No
Start Bit Generation Event (Master mode)	SB		Yes	No
Acknowledge Failure Event	AF		Yes	No
Stop Detection Event (Slave mode)	STOPF		Yes	No
Arbitration Lost Event (Multimaster configuration)	ARLO		Yes	No
Bus Error Event	BERR		Yes	No

Note: The I²C interrupt events are connected to the same interrupt vector (see Interrupts chapter). They generate an interrupt if the corresponding Enable Control Bit is set and the I-bit in the CC register is reset (RIM instruction).

I²C BUS INTERFACE (Cont'd)

I²C CLOCK CONTROL REGISTER (CCR)

Read / Write
Reset Value: 0000 0000 (00h)

7							0
FM/SM	CC6	CC5	CC4	CC3	CC2	CC1	CC0

Bit 7 = **FM/SM** *Fast/Standard I²C mode.*
This bit is set and cleared by software. It is not cleared when the interface is disabled (PE=0).
0: Standard I²C mode
1: Fast I²C mode

Bit 6:0 = **CC[6:0]** *7-bit clock divider.*
These bits select the speed of the bus (F_{SCL}) depending on the I²C mode. They are not cleared when the interface is disabled (PE=0).

Refer to the Electrical Characteristics section for the table of values.

Note: The programmed F_{SCL} assumes no load on SCL and SDA lines.

I²C DATA REGISTER (DR)

Read / Write
Reset Value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0

Bit 7:0 = **D[7:0]** *8-bit Data Register.*
These bits contain the byte to be received or transmitted on the bus.

– Transmitter mode: Byte transmission start automatically when the software writes in the DR register.

– Receiver mode: the first data byte is received automatically in the DR register using the least significant bit of the address.
Then, the following data bytes are received one by one after reading the DR register.

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

Table 18. ADC Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0070h	**ADCDR**	D7	D6	D5	D4	D3	D2	D1	D0
	Reset Value	0	0	0	0	0	0	0	0
0071h	**ADCCSR**	COCO		ADON		CH3	CH2	CH1	CH0
	Reset Value	0	0	0	0	0	0	0	0

OPERATING CONDITIONS (Cont'd)

13.3.2 Operating Conditions with Low Voltage Detector (LVD)

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A.

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
V_{IT+}	Reset release threshold (V_{DD} rise)	High Threshold Med. Threshold Low Threshold	4.10 [2] 3.75 [2] 3.25 [2]	4.30 3.90 3.35	4.50 4.05 3.55	V
V_{IT-}	Reset generation threshold (V_{DD} fall)	High Threshold Med. Threshold Low Threshold[4]	3.85 [2] 3.50 [2] 3.00	4.05 3.65 3.10	4.30 3.95 3.35	
V_{hyst}	LVD voltage threshold hysteresis	V_{IT+}-V_{IT-}	200	250	300	mV
Vt_{POR}	V_{DD} rise time rate [3]		0.2		50	V/ms
$t_{g(VDD)}$	Filtered glitch delay on V_{DD} [2]	Not detected by the LVD			40	ns

Figure 53. High LVD Threshold Versus V_{DD} and f_{OSC} for FLASH devices [3]

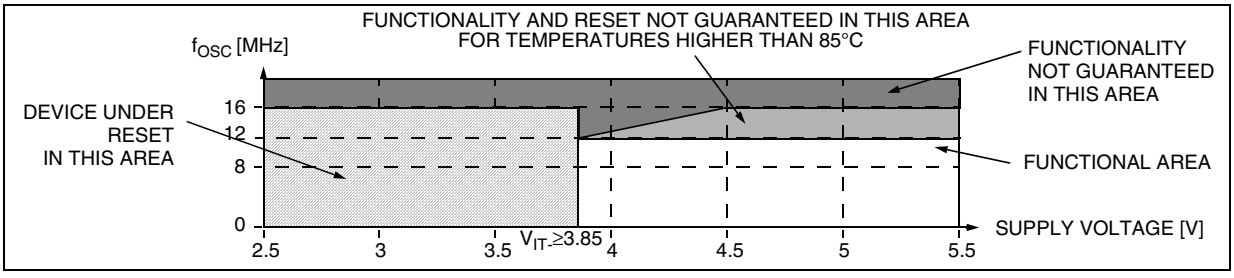


Figure 54. Medium LVD Threshold Versus V_{DD} and f_{OSC} for FLASH devices [3]

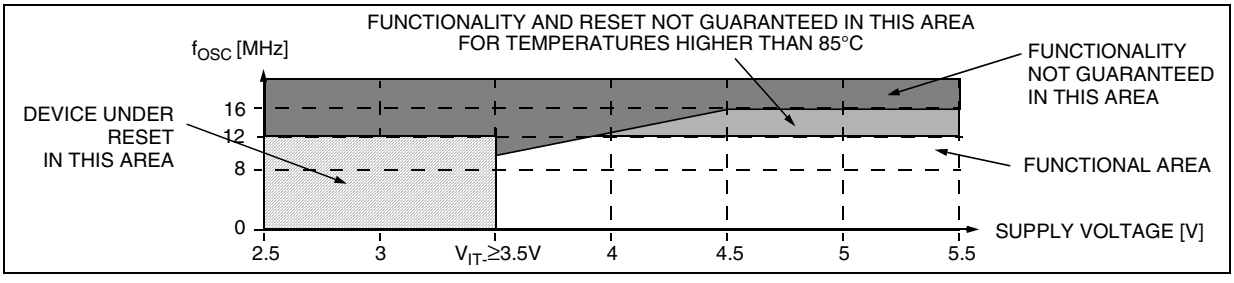
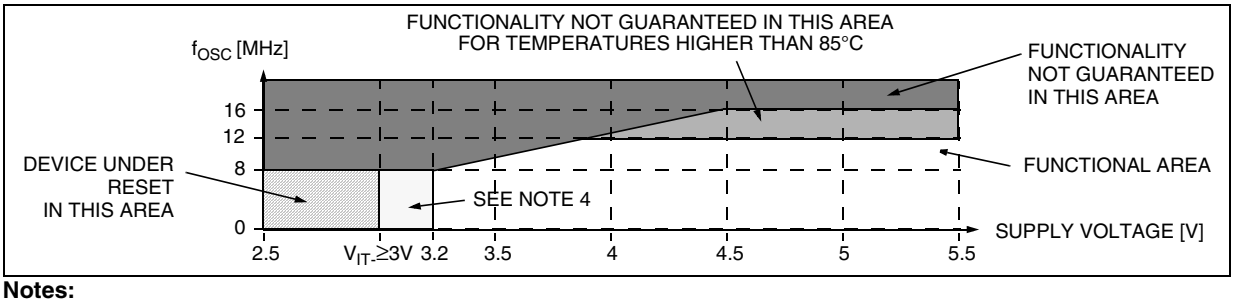


Figure 55. Low LVD Threshold Versus V_{DD} and f_{OSC} for FLASH devices [2][4]



Notes:

1. LVD typical data are based on T_A=25°C. They are given only as design guidelines and are not tested.
2. Data based on characterization results, not tested in production.
3. The V_{DD} rise time rate condition is needed to insure a correct device power-on and LVD reset. Not tested in production.
4. If the low LVD threshold is selected, when V_{DD} falls below 3.2V, (V_{DD} minimum operating voltage), the device is guaranteed to continue functioning until it goes into reset state. The specified V_{DD} min. value is necessary in the device power on phase, but during a power down phase or voltage drop the device will function below this min. level.

CLOCK AND TIMING CHARACTERISTICS (Cont'd)

13.5.3.2 Typical Ceramic Resonators

Symbol	Parameter	Conditions		Typ	Unit
$t_{SU(osc)}$	Ceramic resonator start-up time	LP	2MHz	4.2	ms
		MP	4MHz	2.1	
		MS	8MHz	1.1	
		HS	16MHz	0.7	

$t_{SU(OSC)}$ is the typical oscillator start-up time measured between V_{DD}=2.8V and the fetch of the first instruction (with a quick V_{DD} ramp-up from 0 to 5V (<50µs).

Table 21. Typical Ceramic Resonators for General Purpose Applications

Option Byte Config.	f_{OSC} (MHz)	Resonator Part Number[1]	C_{L1} [pF][3]	C_{L2} [pF][3]	R_{FEXT} [kΩ]	R_D [kΩ]
LP	1	CSB1000J	100	100	Open	3.3
		CSBF1000J				
	2	CSTS0200MG06	(47)	(47)		0
		CSTCC2.00MG0H6				
MP	2	CSTS0200MG06				
		CSTCC2.00MG0H6				
	4	CSTS0400MG06				
		CSTCC4.00MG0H6				
MS	4	CSTS0400MG06				
		CSTCC4.00MG0H6				
	8	CSTS0800MG06				
		CSTCC8.00MG0H6				
HS	8	CSTS0800MG06				
		CSTCC8.00MG0H6				
	10	CSTS1000MG03	(15)	(15)		
		CSTCC10.0MG				
	12	CST12.0MTW	30	30		
		CSTCV12.0MTJ0C4	(22)	(22)		
	16[2]	CSA16.00MXZ040	15	15		
		CST16.00MXW0C3	(15)	(15)		
		CSACV16.00MXJ040	15	15	10	
		CSTCV16.00MXJ0C3	(15)	(15)		
		CSACW1600MX03	10	10		

Notes:

1. Murata Ceralock (refer to Table 22 for correlation factor)
2. V_{DD} 4.5 to 5.5V
3. Values in parentheses refer to the capacitors integrated in the resonator

CLOCK AND TIMING CHARACTERISTICS (Cont'd)

Table 22. Ceramic Resonator Frequency Correlation Factor[1]

Option Byte Config.	Resonator[1)]	Correlation %	Reference IC
LP	CSB1000J	+0.03	4069UBE
	CSTS0200MG06	-0.16	74HCU04
	CSTCC2.00MG0H6	-0.10	
MP	CSTS0200MG06	-0.15	
	CSTCC2.00MG0H6	-0.14	
	CSTS0400MG06	0.00	
	CSTS0400MGA06	-0.01	
	CSTCC4.00MG0H6	-0.02	
MS	CSTS0200MG06	-0.15	74HCU04
	CSTCC2.00MG0H6	-0.14	
	CSTS0400MG06	0.00	
	CSTS0400MGA06	-0.01	
	CSTCC4.00MG0H6	-0.02	
	CSTS0200MG06	-0.15	

Option Byte Config.	Resonator[1)]	Correlation %	Reference IC
HS	CSTS0800MG06	+0.10	74HCU04
	CSTS0800MGA06	+0.07	
	CSTCC8.00MG0H6	+0.09	
	CSTS1000MG03	+0.34	4069UBP
	CSTCC10.0MG	+0.75	4069UBE
	CST12.0MTW	+0.45	4069UBE
	CSTCV12.0MTJ0C4	+0.30	40H004
	CSTCS12.0MTA	+0.50	4069UBE
	CSA16.00MXZ040	+0.10	74HCU04
	CSACV16.00MXJ040	+0.09	
	CSACW1600MX03	+0.03	
	CSACV16.00MXA040Q	+0.09	

Notes:
1. See Table 21 for ceramic resonator values.

13.6 MEMORY CHARACTERISTICS

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

13.6.1 RAM and Hardware Registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode [1]	HALT mode (or RESET)	1.6			V

13.6.2 FLASH Program Memory

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$T_{A(prog)}$	Programming temperature range [2]		0	25	70	°C
t_{prog}	Programming time for 1~16 bytes [3]	T_A=+25°C		8	25	ms
	Programming time for 4 or 8kBytes	T_A=+25°C		2.1	6.4	sec
t_{ret}	Data retention [5]	T_A=+55°C [4]	20			years
N_{RW}	Write erase cycles [5]	T_A=+25°C	100			cycles

Notes:

1. Minimum V_{DD} supply voltage without losing data stored in RAM (in HALT mode or under RESET) or in hardware registers (only in HALT mode). Guaranteed by construction, not tested in production.
2. Data based on characterization results, tested in production at T_A=25°C.
3. Up to 16 bytes can be programmed at a time for a 4kBytes FLASH block (then up to 32 bytes at a time for an 8k device)
4. The data retention time increases when the T_A decreases.
5. Data based on reliability test results and monitored in production.

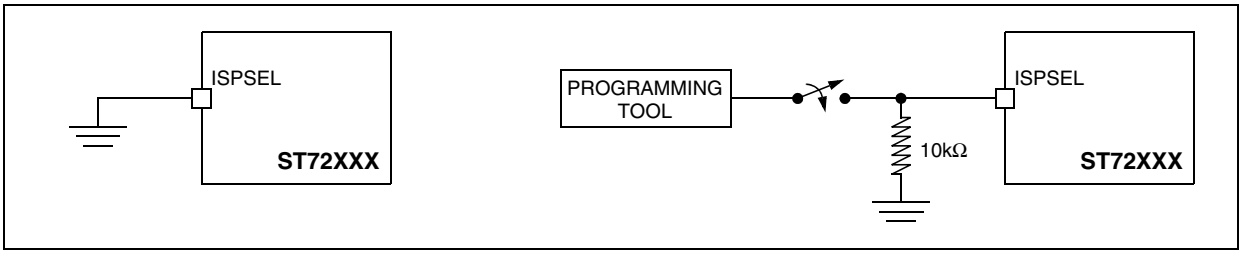
CONTROL PIN CHARACTERISTICS (Cont'd)

13.9.2 ISPSEL Pin

Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IL}	Input low level voltage [1]		V_{SS}	0.2	V
V_{IH}	Input high level voltage [1]		V_{DD}-0.1	12.6	
I_L	Input leakage current	$V_{IN}=V_{SS}$		±1	µA

Figure 90. Two typical Applications with ISPSEL Pin [2]



Notes:

1. Data based on design simulation and/or technology characteristics, not tested in production.
2. When the ISP Remote mode is not required by the application ISPSEL pin must be tied to V_{SS}.

15.3 DEVELOPMENT TOOLS

STmicroelectronics offers a range of hardware and software development tools for the ST7 microcontroller family. Full details of tools available for the ST7 from third party manufacturers can be obtain from the STMicroelectronics Internet site:
➟ http//mcu.st.com.

Third Party Tools

- ACTUM
- BP
- COSMIC
- CMX
- DATA I/O
- HITEX
- HIWARE
- ISYSTEM
- KANDA
- LEAP

Tools from these manufacturers include C compliers, emulators and gang programmers.

STMicroelectronics Tools

Three types of development tool are offered by ST, all of them connect to a PC via a parallel (LPT) port: see Table 26 and Table 27 for more details.

Table 26. STMicroelectronics Tool Features

	In-Circuit Emulation	Programming Capability[1)]	Software Included
ST7 Development Kit	Yes. (Same features as HDS2 emulator but without logic analyzer)	Yes (DIP packages only)	ST7 CD ROM with: – ST7 Assembly toolchain – STVD7 and WGDB7 powerful Source Level Debugger for Win 3.1, Win 95 and NT – C compiler demo versions – ST Realizer for Win 3.1 and Win 95. – Windows Programming Tools for Win 3.1, Win 95 and NT
ST7 HDS2 Emulator	Yes, powerful emulation features including trace/ logic analyzer	No	
ST7 Programming Board	No	Yes (All packages)	

Table 27. Dedicated STMicroelectronics Development Tools

Supported Products	ST7 Development Kit	ST7 HDS2 Emulator	ST7 Programming Board
ST72254G1, ST72C254G1 ST72254G2, ST72C254G2 ST72215G2, ST72C215G2 ST72216G1, ST72C216G1 ST72104G1, ST72C104G1, ST72104G2, ST72C104G2	ST7MDT1-DVP2	ST7MDT1-EMU2B	ST7MDT1-EPB2/EU ST7MDT1-EPB2/US ST7MDT1-EPB2/UK

Note:
1. In-Situ Programming (ISP) interface for FLASH devices.