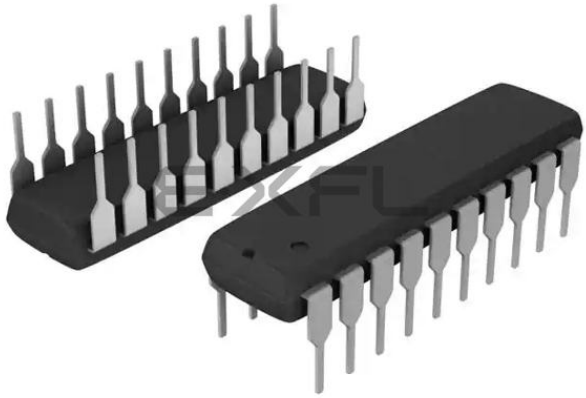


Welcome to E-XFL.COM

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	SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	15
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-DIP
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7flite29f2b6

1 Description

ST7LITE20F2, ST7LITE25F2 and ST7LITE29F2 are referred to as ST7LITE2. The ST7LITE2 is a member of the ST7 microcontroller family. All ST7 devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set.

The ST7LITE2 features FLASH memory with byte-by-byte In-Circuit Programming (ICP) and In-Application Programming (IAP) capability.

Under software control, the ST7LITE2 device can be placed in WAIT, SLOW, or HALT mode, reducing power consumption when the application is in idle or standby state.

The enhanced instruction set and addressing modes of the ST7 offer both power and flexibility to software developers, enabling the design of highly efficient and compact application code. In addition to standard 8-bit data management, all ST7 microcontrollers feature true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

For easy reference, all parametric data are located in *Section 15: Device configuration*.

The devices feature an on-chip Debug Module (DM) to support in-circuit debugging (ICD). For a description of the DM registers, refer to the ST7 ICC Protocol Reference Manual.

Table 1. Device summary

Features	ST7LITE20F2	ST7LITE25F2	ST7LITE29F2
Program memory - bytes	8 Kbyte		
RAM (stack) - bytes	384 (128)		
Data EEPROM - bytes	–	–	256
Peripherals	Lite timer with Watchdog, autoreload timer, SPI, 10-bit ADC with Op-Amp	Lite timer with watchdog, autoreload timer with 32-MHz input clock, SPI, 10-bit ADC with op-amp	
Operating supply	2.4V to 5.5V		
CPU frequency	Up to 8 MHz (w/ ext OSC up to 16 MHz)	Up to 8 MHz (w/ ext OSC up to 16 MHz and int 1MHz RC 1% PLLx8/4 MHz)	
Operating temperature	–40 °C to +85 °C	–40 °C to +85 °C	–40 °C to +85 °C –40 °C to +105 °C
Packages	SO20 300”, DIP20		

Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

Index registers (X and Y)

In indexed addressing modes, these 8-bit registers are used to create either effective addresses or 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 (not pushed to and popped from the stack).

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).

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 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.

- 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 7th bit of the result.
  0: The result of the last operation is positive or null
  1: The result of the last operation is negative
  (i.e. the most significant bit is a logic 1)

  This bit is accessed by the JRMI and JRPL instructions.

- Bit 1 = **Z** *Zero*

  This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.
  0: The result of the last operation is different from zero
  1: The result of the last operation is zero

  This bit is accessed by the JREQ and JRNE test instructions.

- Bit 0 = **C** *Carry/borrow*

  This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.
  0: No overflow or underflow has occurred
  1: An overflow or underflow has occurred

  This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the “bit test and branch”, shift and rotate instructions.

Stack pointer register (SP)

Read/Write

Reset value: 01FFh

15							**8**
0	0	0	0	0	0	0	1
7							**0**
1	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see *Figure 11*).

Since the stack is 128 bytes deep, the 9 most significant bits are forced by hardware. Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP6 to SP0 bits are set) which is the stack higher address.

The least significant byte of the Stack Pointer (called S) can be directly accessed by a LD instruction.

Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location

8 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 nonmaskable software interrupt (TRAP). The Interrupt processing flowchart is shown in *Figure 20: Interrupt processing flowchart*.

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 interrupt function on page 64*).

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 *Table 12: Interrupt mapping* 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 *Table 12: Interrupt mapping*).

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 *Table 12: Interrupt mapping*).

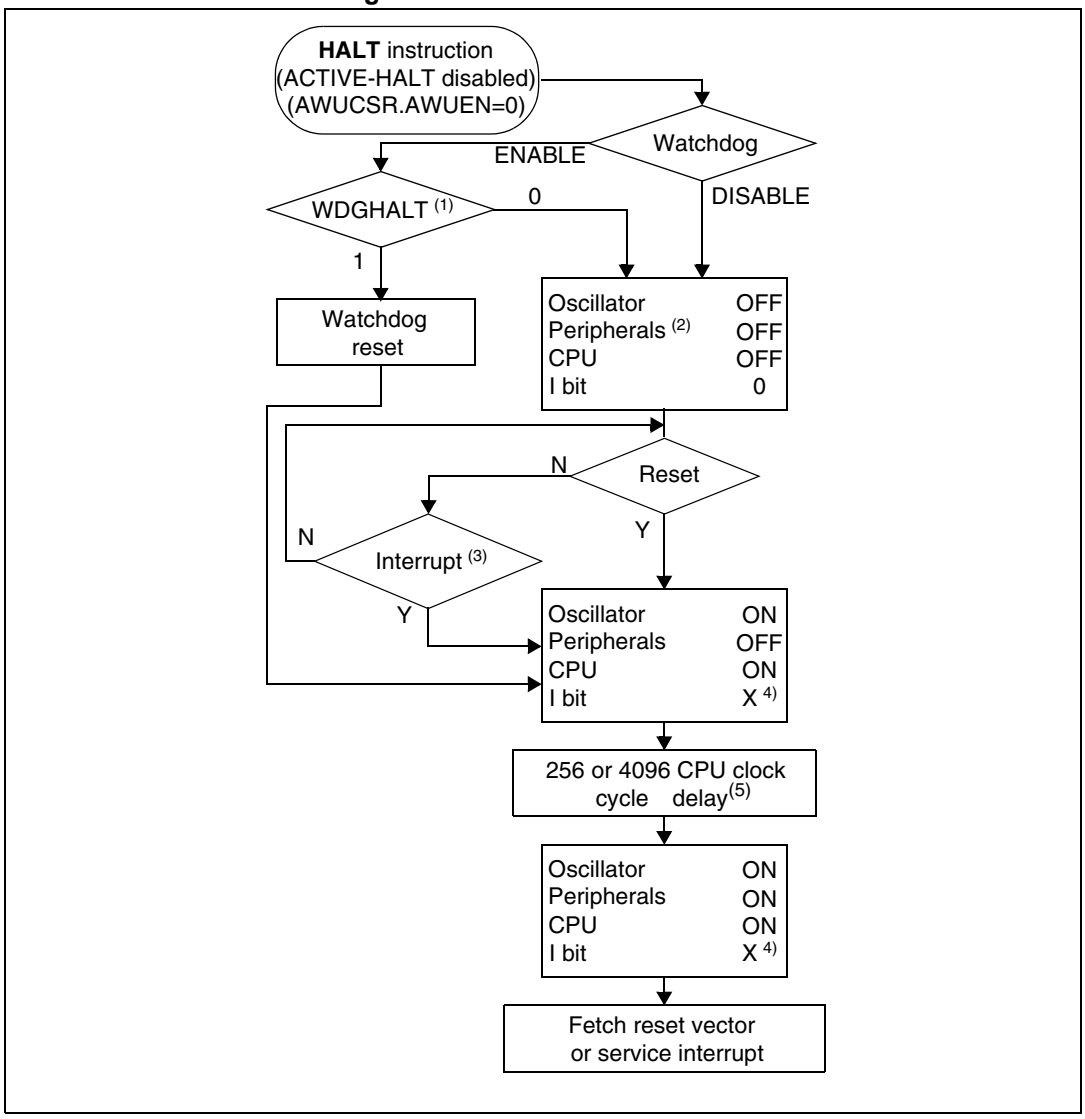
8.1 Non maskable software interrupt

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 20: Interrupt processing flowchart*.

8.2 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.

Figure 25. HALT mode flowchart



1. WDGHALT is an option bit (see *Section 15.1: Option bytes* for more details).
2. Peripheral clocked with an external clock source can still be active.
3. Only some specific interrupts can exit the MCU from HALT mode (such as external interrupt). Refer to *Table 12: Interrupt mapping* for more details.
4. Before servicing an interrupt, the CC register is pushed on the stack. The I bit of the CC register is set during the interrupt routine and cleared when the CC register is popped.
5. If the PLL is enabled by option byte, it outputs the clock after a delay of $t_{STARTUP}$ (see *Figure 12: PLL output frequency timing diagram*).

9.4.1 HALT mode recommendations

- Make sure that an external event is available to wake up the microcontroller from Halt mode.
- When using an external interrupt to wake up the microcontroller, reinitialize the corresponding I/O as “Input Pull-up with Interrupt” before executing the HALT

Spurious interrupts

When enabling/disabling an external interrupt by setting/resetting the related OR register bit, a spurious interrupt is generated if the pin level is low and its edge sensitivity includes falling/rising edge. This is due to the edge detector input which is switched to '1' when the external interrupt is disabled by the OR register.

To avoid this unwanted interrupt, a "safe" edge sensitivity (rising edge for enabling and falling edge for disabling) has to be selected before changing the OR register bit and configuring the appropriate sensitivity again.

Caution: In case a pin level change occurs during these operations (asynchronous signal input), as interrupts are generated according to the current sensitivity, it is advised to disable all interrupts before and to reenable them after the complete previous sequence in order to avoid an external interrupt occurring on the unwanted edge.

This corresponds to the following steps:

1. To enable an external interrupt:
   - set the interrupt mask with the SIM instruction (in cases where a pin level change could occur)
   - select rising edge
   - enable the external interrupt through the OR register
   - select the desired sensitivity if different from rising edge
   - reset the interrupt mask with the RIM instruction (in cases where a pin level change could occur).
2. To disable an external interrupt:
   - set the interrupt mask with the SIM instruction SIM (in cases where a pin level change could occur)
   - select falling edge
   - disable the external interrupt through the OR register
   - select rising edge
   - reset the interrupt mask with the RIM instruction (in cases where a pin level change could occur).

10.2.2 Output modes

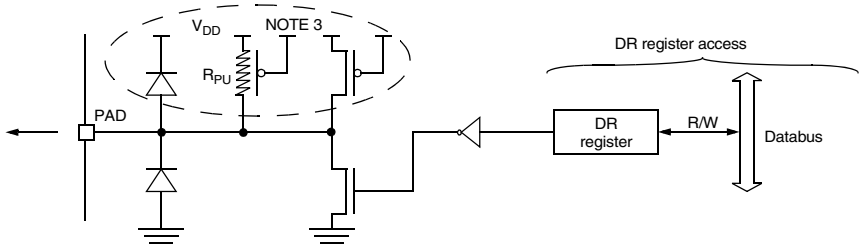
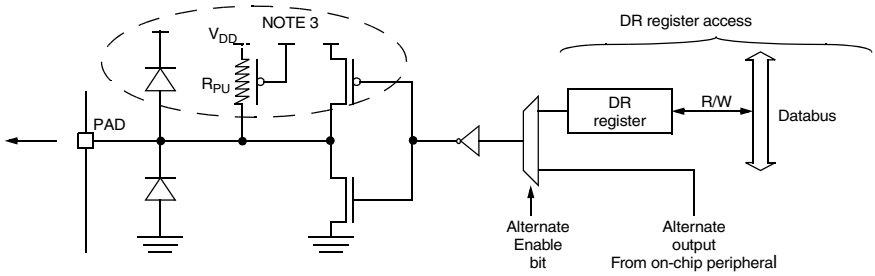
Setting the DDRx bit selects output mode. Writing to the DR bits applies a digital value to the I/O through the latch. Reading the DR bits returns the previously stored value.

If an OR bit is available, different output modes can be selected by software: push-pull or opendrain. Refer to I/O Port Implementation section for configuration.

Table 21. DR value and output pin status

DR	Push-pull	Open-drain
0	V_{OL}	V_{OL}
1	V_{OH}	Floating

Table 23. I/O configurations (continued)

I/O port	Hardware configuration
Open-drain output(2)	
Push-pull output(3)	

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.
3. For true open drain, these elements are not implemented.

Analog alternate function

Configure the I/O as floating input to use an ADC input. The analog multiplexer (controlled by the ADC registers) switches the analog voltage present on the selected pin to the common analog rail, connected to the ADC input.

Analog recommendations

Do not change the voltage level or loading on any I/O while conversion is in progress. Do not have clocking pins located close to a selected analog pin.

Warning: The analog input voltage level must be within the limits stated in the absolute maximum ratings.

10.3 I/O port implementation

The hardware implementation on each I/O port depends on the settings in the DDR and OR registers and specific I/O port features such as ADC input or open drain.

Switching these I/O ports from one state to another should be done in a sequence that prevents unwanted side effects. Recommended safe transitions are illustrated in *Figure 32*. Other transitions are potentially risky and should be avoided, since they may present unwanted side-effects such as spurious interrupt generation.

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. This downcounter is freerunning: it counts down even if the watchdog is disabled. The value to be stored in the CR register must be between FFh and C0h (see *Table 31: Watchdog timing*):

- The WDGA bit is set (watchdog enabled)
- The T6 bit is set to prevent generating an immediate reset
- The T[5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

If the watchdog is activated, the HALT instruction will generate a Reset.

Table 31. Watchdog timing(1)

f_{CPU} = 8 MHz		
WDG counter code	**min [ms]**	**max [ms]**
C0h	1	2
FFh	127	128

1. The timing variation is due to the unknown status of the prescaler when writing to the CR register.

Note: The number of CPU clock cycles applied during the reset phase (256 or 4096) must be taken into account in addition to these timings.

11.1.4 Hardware watchdog option

If Hardware Watchdog is selected by option byte, the watchdog is always active and the WDGA bit in the CR is not used.

Refer to the Option Byte description in *Section 15: Device configuration*.

Using HALT mode or ACTIVE-HALT mode with the WDG (WDGHALT option)

If Halt mode with Watchdog is enabled by option byte (No watchdog reset on HALT instruction), it is recommended before executing the HALT instruction to refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller. Same behavior in active-halt mode.

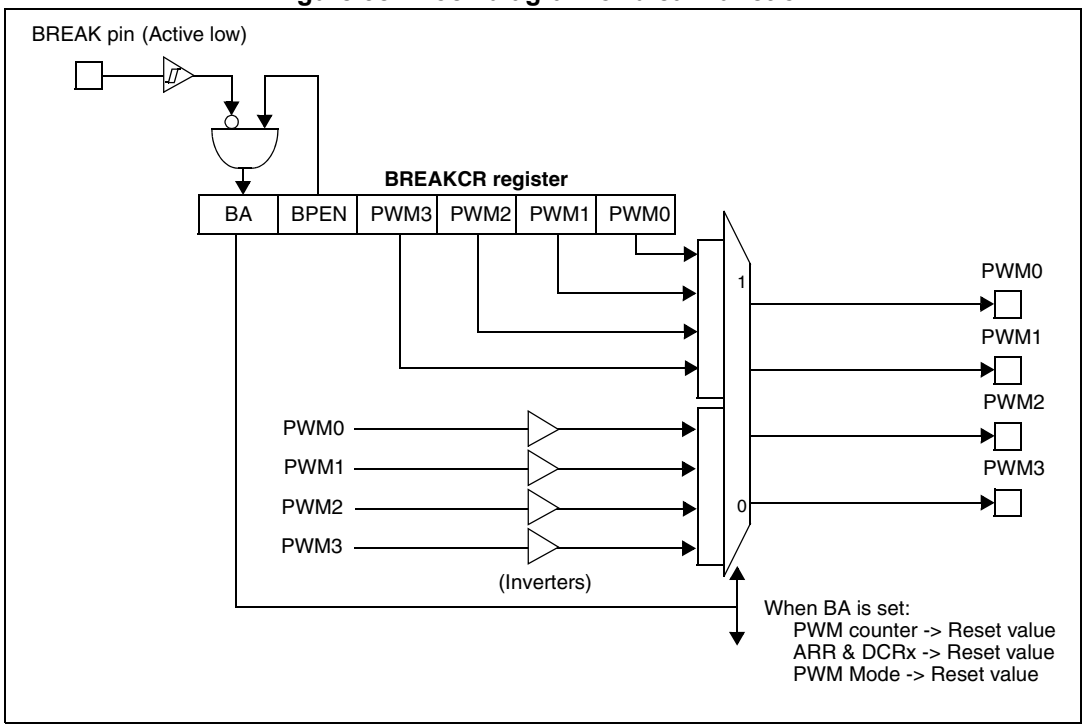
11.1.5 Interrupts

None.

Software can set the BA bit to activate the break function without using the BREAK pin.

- When the break function is activated (BA bit =1):
  - the break pattern (PWM[3:0] bits in the BREAKCR) is forced directly on the PWMx output pins (after the inverter),
  - the 12-bit PWM counter is set to its reset value,
  - the ARR, DCRx and the corresponding shadow registers are set to their reset values,
  - the PWMCR register is reset.
- When the break function is deactivated after applying the break (BA bit goes from 1 to 0 by software):
  - the control of PWM outputs is transferred to the port registers.

Figure 38. Block diagram of break function



Note: The BREAK pin value is latched by the BA bit.

Input capture

The 12-bit ATICR register is used to latch the value of the 12-bit free running upcounter after a rising or falling edge is detected on the ATIC pin.

When an input capture occurs, the ICF bit is set and the ATICR register contains the value of the free running upcounter. An IC interrupt is generated if the ICIE bit is set. The ICF bit is reset by reading the ATICR register when the ICF bit is set. The ATICR is a read only register and always contains the free running upcounter value which corresponds to the most recent input capture. Any further input capture is inhibited while the ICF bit is set.

11.3.3 Functional description

Timebase counter 1

The 8-bit value of Counter 1 cannot be read or written by software. After an MCU reset, it starts incrementing from 0 at a frequency of $f_{OSC}/32$. An overflow event occurs when the counter rolls over from F9h to 00h. If f_{OSC} = 8 MHz, then the time period between two counter overflow events is 1 ms. This period can be doubled by setting the TB bit in the LTCSR1 register.

When Counter 1 overflows, the TB1F bit is set by hardware and an interrupt request is generated if the TB1IE bit is set. The TB1F bit is cleared by software reading the LTCSR1 register.

Input capture

The 8-bit input capture register is used to latch the free-running upcounter (Counter 1) 1 after a rising or falling edge is detected on the LTIC pin. When an input capture occurs, the ICF bit is set and the LTICR1 register contains the MSB of Counter 1. An interrupt is generated if the ICIE bit is set. The ICF bit is cleared by reading the LTICR register.

The LTICR is a read-only register and always contains the data from the last input capture. Input capture is inhibited if the ICF bit is set.

Timebase counter 2

Counter 2 is an 8-bit autoreload upcounter. It can be read by accessing the LTCNTR register. After an MCU reset, it increments at a frequency of $f_{OSC}/32$ starting from the value stored in the LTARR register. A counter overflow event occurs when the counter rolls over from FFh to the LTARR reload value. Software can write a new value at anytime in the LTARR register, this value will be automatically loaded in the counter when the next overflow occurs.

When Counter 2 overflows, the TB2F bit in the LTCSR2 register is set by hardware and an interrupt request is generated if the TB2IE bit is set. The TB2F bit is cleared by software reading the LTCSR2 register.

Figure 41. Input capture timing diagram

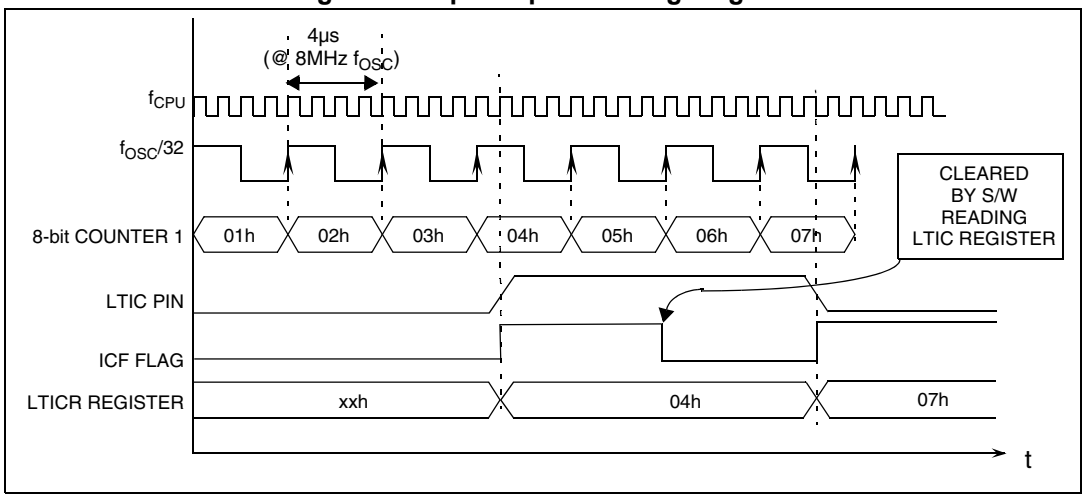


Table 45. Channel selection bits (continued)

Channel pin(1)	CH2	CH1	CH0
AIN3	0	1	1
AIN4	1	0	0
AIN5	1	0	1
AIN6	1	1	0

1. The number of channels is device dependent. Refer to *Table 2: Device pin description*.

Table 53. Short instructions supporting direct, indexed, indirect and indirect indexed addressing modes

Short instructions only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
BSET, BRES	Bit operations
BTJT, BTJF	Bit Test and Jump operations
SLL, SRL, SRA, RLC, RRC	Shift and Rotate operations
SWAP	Swap nibbles
CALL, JP	Call or Jump sub-routine

12.1.7 Relative mode (direct, indirect)

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

Table 54. Relative direct and indirect instructions and functions

Available relative direct/indirect instructions	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two submodes:

- Relative (direct)
  The offset follows the opcode.
- Relative (indirect)
  The offset is defined in the memory, the address of which follows the opcode.

12.2 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 *Table 55*:

Table 55. Instruction groups

Group	Instructions							
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	–	–	–

13 Electrical characteristics

13.1 Parameter conditions

Unless otherwise specified, all voltages are referred to V_{SS}.

13.1.1 Minimum and maximum values

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

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

13.1.2 Typical values

Unless otherwise specified, typical data are based on T_A=25 °C, V_{DD}=5 V (for the 4.5V≤V_{DD}≤ 5.5 V voltage range) and V_{DD}=3.3 V (for the 3 V≤V_{DD}≤4 V voltage range). They are given only as design guidelines and are not tested.

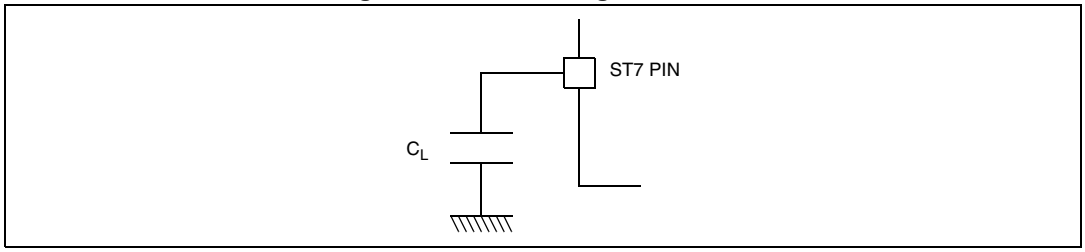
13.1.3 Typical curves

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

13.1.4 Loading capacitor

The loading conditions used for pin parameter measurement are shown in *Figure 50*.

Figure 50. Pin loading conditions



13.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in *Figure 51*.

1. Not tested in production.
2. Not tested in production. The V_{DD} rise time rate condition is needed to insure a correct device power-on and LVD reset. When the V_{DD} slope is outside these values, the LVD may not ensure a proper reset of the MCU.
   Use of LVD with capacitive power supply: with this type of power supply, if power cuts occur in the application, it is recommended to pull V_{DD} down to 0 V to ensure optimum restart conditions. Refer to circuit example in *Figure 84: RESET pin protection when LVD is enabled*

Table 70. Resonator characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{CrOSC}	Crystal Oscillator Frequency(1)	–	2	–	16	MHz
C_{L1} C_{L2}	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator (R_S)	–	See *Table 72: Resonator performances*			pF

1. When PLL is used, please refer to the *Section 13.3.4: Internal RC oscillator and PLL* and *Section 7: Supply, reset and clock management* (f_{CrOSC} min. is 8 Mhz with PLL).

Table 71: Resonator performances

Supplier	f_{CrOSC} [MHz]	Typical ceramic resonators(1)		CL1(2) [pF]	CL2(2) [pF]	Rd [Ω]	Supply voltage range [V]	Temperature range [°C]
		Type (3)	Reference					
Murata	2	SMD	CSTCC2M00G56-R0	(47)	(47)	0	2.4V to 5.5V	-40 to 85
	4	SMD	CSTCR4M00G53-R0	(15)	(15)	0		
		LEAD	CSTLS4M00G53-B0	(15)	(15)	0		
	8	SMD	CSTCE8M00G52-R0	(10)	(10)	0		
		LEAD	CSTLS8M00G53-B0	(15)	(15)	0		
	16	SMD	CSTCE16M0V51-R0	(5)	(5)	0	3.3V to 5.5V	
		LEAD	CSTLS16M0X53-B0	(15)	(15)	0	4.5V to 5.5V	
		LEAD	CSALS16M0X55-B0	7	7	1.5k	3.8V to 5.5V	

1. Resonator characteristics given by the ceramic resonator manufacturer. For more information on these resonators, please consult www.murata.com
2. () means load capacitor built in resonator.
3. SMD = -R0: Plastic tape package (∅=180mm).
   LEAD = -B0: Bulk

Figure 82. Typical V_{OL} vs. V_{DD} (high-sink I/Os)

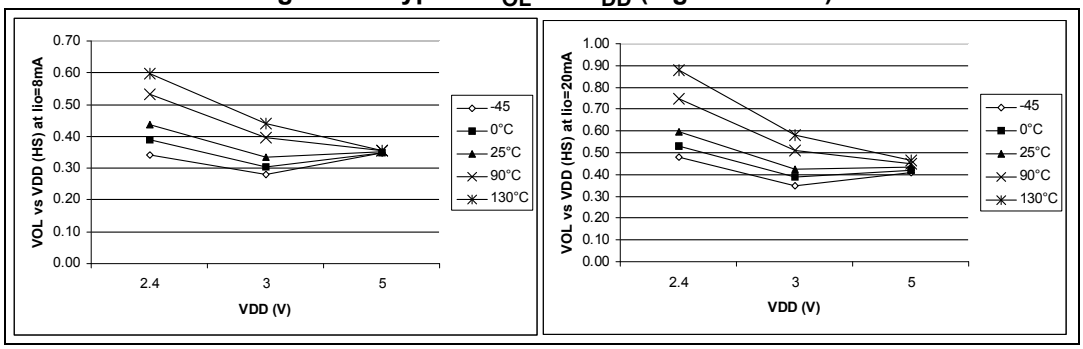
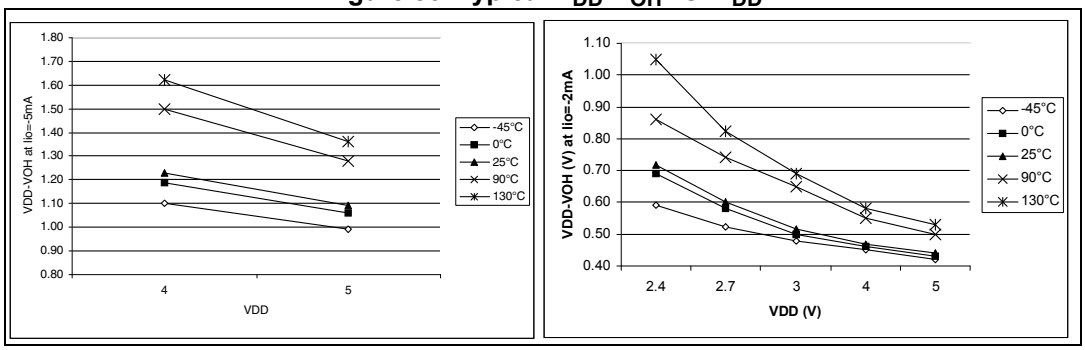


Figure 83. Typical V_{DD}-V_{OH} vs. V_{DD}



13.9 Control pin characteristics

Table 82. Asynchronous $\overline{\text{RESET}}$ Pin(1)

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IL}	Input low level voltage	–		V_{SS} - 0.3	–	0.3xV_{DD}	V
V_{IH}	Input high level voltage	–		0.7xV_{DD}	–	V_{DD} + 0.3	
V_{hys}	Schmitt trigger voltage hysteresis(2)	–		–	2	–	V
V_{OL}	Output low level voltage(3)	V_{DD}=5V	I_{IO}=+5mA $T_A \leq 85°C$ $T_A \geq 85°C$	–	0.5	1.0 1.2	V
			I_{IO}=+2mA $T_A \leq 85°C$ $T_A \geq 85°C$	–	0.2	0.4 0.5	
R_{ON}	Pull-up equivalent resistor(2)(4)	V_{DD}=5V		20	40	80	kΩ
		V_{DD}=3V		40	70	120	
$t_{w(RSTL)out}$	Generated reset pulse duration	Internal reset sources		–	30	–	µs
$t_{h(RSTL)in}$	External reset pulse hold time(5)	–		20	–	–	µs
$t_{g(RSTL)in}$	Filtered glitch duration	–		–	200	–	ns

1. TA = -40°C to 85°C, unless otherwise specified.
2. Data based on characterization results, not tested in production.

Table 93. Size definition

Sector 0 size	SEC1	SEC0
0.5k	0	0
1k	0	1
2k	1	0
4k	1	1

- OPT1 = **FMP_R** *Read-out protection*
  Read-out protection, when selected provides a protection against program memory content extraction and against write access to Flash memory. Erasing the option bytes when the FMP_R option is selected will cause the whole memory to be erased first and the device can be reprogrammed. Refer to the ST7 Flash Programming Reference Manual and *Section 4.5: Memory protection* for more details
  0: Read-out protection off
  1: Read-out protection on
- OPT0 = **FMP_W** *Flash write protection*
  This option indicates if the Flash program memory is write protected.
  0: Write protection off
  1: Write protection on

Warning: When this option is selected, the program memory (and the option bit itself) can never be erased or programmed again.

Table 94. Option byte default values

	Option byte 0								Option byte 1							
	7							0	7							0
	Res.	OSCRANGE 2:0			SEC 1	SEC 0	FMP R	FMP W	PLL x4x8	PLL OFF	PLL32 OFF	OSC	LVD 1	LVD 0	WDG SW	WDG HALT
Default Value	1	1	1	1	1	1	0	0	1	1	1	0	1	1	1	1

Table 100. ST7 application notes (continued)

Identification	Description
AN1151	Performance Comparison Between ST72254 & PC16F876
AN1278	LIN (Local Interconnect Network) Solutions
Product Migration	
AN1131	Migrating applications from ST72511/311/214/124 to ST72521/321/324
AN1322	Migrating an application from ST7263 Rev.B to ST7263B
AN1365	Guidelines for migrating ST72C254 applications to ST72F264
AN1604	How to use ST7MDT1-TRAIN with ST72F264
AN2200	guidelines for migrating st7lite1x applications to st7flite1xb
Product Optimization	
AN 982	Using ST7 with Ceramic Resonator
AN1014	How to Minimize the ST7 Power Consumption
AN1015	SOFTWARE TECHNIQUES FOR IMPROVING MICROCONTROLLER EMC PERFORMANCE
AN1040	Monitoring the Vbus Signal for USB Self-Powered Devices
AN1070	ST7 Checksum Self-Checking Capability
AN1181	Electrostatic Discharge Sensitive Measurement
AN1324	Calibrating the RC Oscillator of the ST7FLITE0 MCU using the MAINS
AN1502	Emulated Data EEPROM with ST7 HDFlash Memory
AN1529	Extending the current & voltage capability on the ST7265 VDDF Supply
AN1530	Accurate timebase for low-cost ST7 applications with internal RC oscillator
AN1605	Using an active RC to wakeup the ST7LITE0 from power saving mode
AN1636	UNDERSTANDING AND MINIMIZING ADC CONVERSION ERRORS
AN1828	PIR (Passive Infrared) Detector using the ST7FLITE05/09/SUPERLITE
AN1946	SENSORLESS BLDC MOTOR CONTROL AND BEMF SAMPLING METHODS WITH ST7MC
AN1953	PFC FOR ST7MC STARTER KIT
AN1971	ST7LITE0 MICROCONTROLLED BALLAST
Programming and Tools	
AN 978	ST7 Visual DeVELOP Software Key Debugging Features
AN 983	Key Features of the Cosmic ST7 C-Compiler Package
AN 985	Executing Code In ST7 RAM
AN 986	Using the Indirect Addressing Mode with ST7
AN 987	ST7 Serial Test Controller Programming
AN 988	Starting with ST7 Assembly Tool Chain
AN 989	Getting Started with the ST7 Hiware C Toolchain

Table 101. Revision history (continued)

Date	Revision	Description of changes
07-Jul-2006	4	Added 300K read/write cycles for EEPROM on first page Updated *Section 4.4 on page 22* and modified note 5 and *Figure 5* Added note 2 in *External interrupt control register (EICR) on page 41* and changed *External interrupt function on page 64* Modified read operation section in *Memory access on page 25* Added note to *Section 7.1 on page 34* Modified one note in *Section 7.1 on page 34* Modified *Table on page 38* Added note on illegal opcode reset to *Section 7.5.1 on page 39* Added note to *Section 7.6.1 on page 41* Changed note below *Figure 8 on page 27* and the last paragraph of *Access error handling on page 27* In *Section 11.2.6 on page 80*, modified description of OE bits in the PWMCR register (added “after an overflow event”). Added important note to *Section on page 94* Changed *Section 13.2.1 on page 117* (f_{OSC} or f_{CLKIN} replaced by f_{CPU} and frequency values changed accordingly) Added note 1 and modified note 3 in *Section on page 113* and *Section on page 122* and changed table titles Added *Crystal and Ceramic Resonator Oscillators on page 120* Changed I_S value and note 2 in *Section 12.7.1 on page 127* Updated *Section 14.2 on page 153* Added note to *Figure 67 on page 138* Changed notes 1 and 2 to *Table 89 on page 152* and added R_{thJA} value for DIP20 package Changed *Figure 84*, *Figure 85 on page 144* (and notes) and removed EMC protection circuitry in *Figure 85 on page 144* (device works correctly without these components) Added note 2 to opt 4 (option byte 2) in *Section 15.1 on page 154* Modified *Section 14.2 on page 153* Changed *Section 15.3 on page 160* Added *Section 16.7 on page 166* Changed LTICR reset value in *Table 3 on page 19* Modified “caution” in *Section 8.2 on page 47* Replaced bit1 by bit2 for AWUF bit in AWUCSR description in *Section 9.6.1 on page 62* Modified *Section on page 64* Changed order of *Section* and *Section on page 87* and removed two paragraphs before *Section 11.3.4 on page 88* Added note 3 to *Section 13.3.2 on page 121* Modified *Section 12.9 on page 137*: changed $t_{h(MO)}$ and $t_{v(MO)}$, as well as $t_{su(\overline{SS})}$ and $t_{h\ (\overline{SS})}$ values and added note 4. The change made to $t_{su(\overline{SS})}$ and $t_{h\ (\overline{SS})}$ values applies from silicon rev B of this product. Changed $t_{w(JIT)}$ value in *Section on page 113* and *Section on page 122* Added note to *Section 12.4.2 on page 120* Changed LTCSR2 reset value in *Section 11.3.6 on page 88* Modified *Figure 84* and *Figure 85 on page 144* Added note 1 to the max column in table *on page 148* and modified the content of this note.