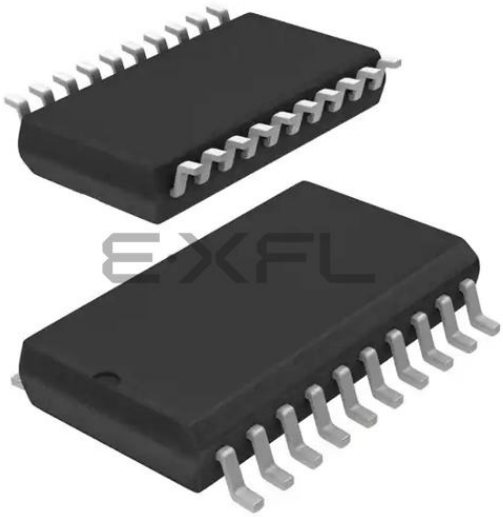


Welcome to E-XFL.COM

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	-
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	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SO
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7flite20f2m6

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		

Table 2. Device pin description (continued)

Pin No.		Pin name	Type	Level		Port / Control						Main function (after reset)	Alternate function
						Input				Output			
SO20	DIP20			Input	Output	float	wpu	int	ana	OD	PP		
15	10	PA3/ATPWM1	I/O	C_T	HS	**X**	ei0		-	X	X	**Port A3**	Auto-reload timer PWM1
16	11	PA2/ATPWM0	I/O	C_T	HS	**X**			-	X	X	**Port A2**	Auto-reload timer PWM0
17	12	PA1/ATIC	I/O	C_T	HS	**X**			-	X	X	**Port A1**	Auto-reload timer input capture
18	13	PA0/LTIC	I/O	C_T	HS	**X**			-	X	X	**Port A0**	Lite timer input capture
19	14	OSC2	O	–	–	-	-	-	-	-	-	Resonator oscillator inverter output	
20	15	OSC1/CLKIN	I	–	–	-	-	-	-	-	-	Resonator oscillator inverter input or external clock input	

1. No negative current injection allowed on this pin. For details (refer to *Table 58: Current characteristics*).
2. During normal operation this pin must be pulled- up, internally or externally (external pull-up of 10k mandatory in noisy environment). This is to avoid entering ICC mode unexpectedly during a reset. In the application, even if the pin is configured as output, any reset will put it back in input pull-up.

3 Register & memory map

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

The available memory locations consist of 128 bytes of register locations, 384 bytes of RAM, 256 bytes of data EEPROM and 8 Kbytes of user program memory. The RAM space includes up to 128 bytes for the stack from 180h to 1FFh.

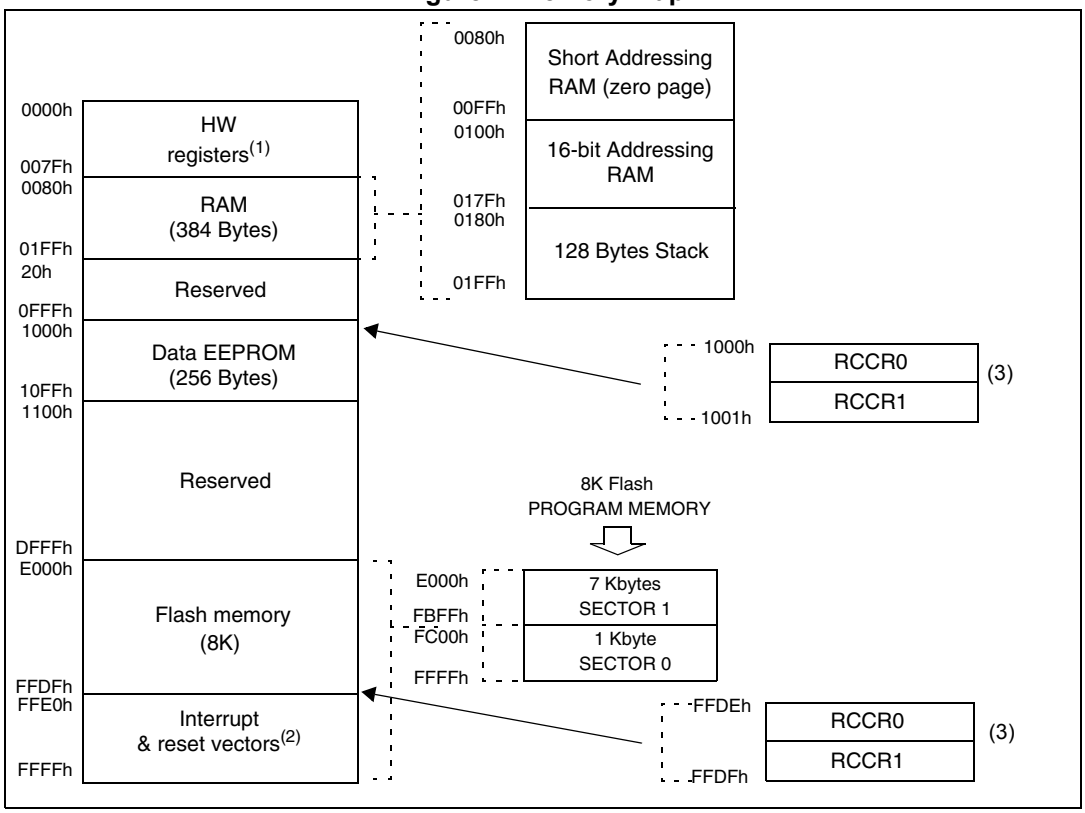
The highest address bytes contain the user reset and interrupt vectors.

The Flash memory contains two sectors (see *Figure 4*) mapped in the upper part of the ST7 addressing space so the reset and interrupt vectors are located in Sector 0 (F000h-FFFFh).

The size of Flash Sector 0 and other device options are configurable by Option byte (refer to *Section 15: Device configuration*).

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

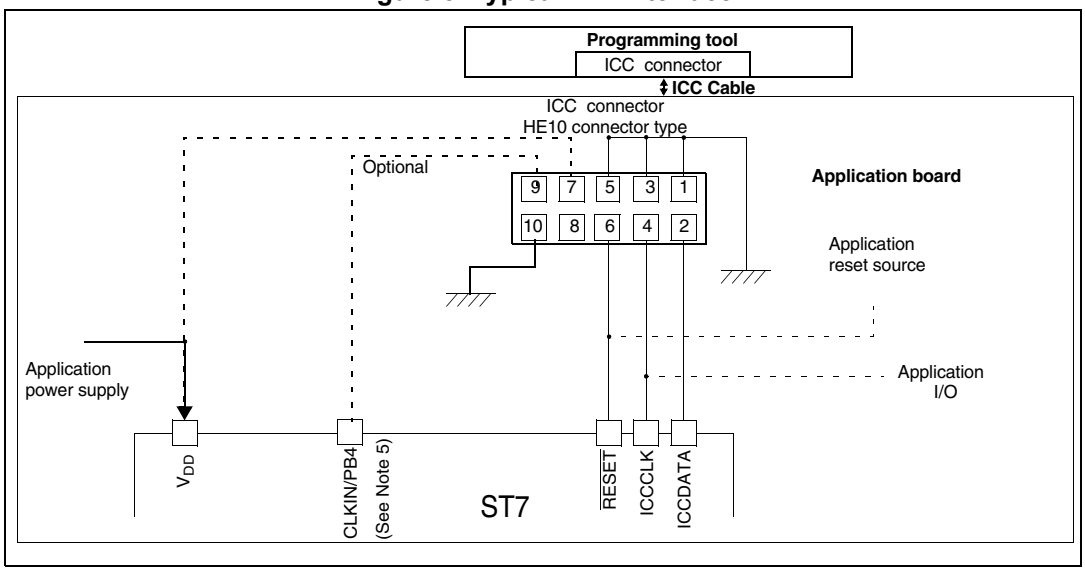
Figure 4. Memory map



1. See *Table 3: Hardware register map*
2. See *Table 12: Interrupt mapping*
3. See *Section 7.1: Internal RC oscillator adjustment*

Caution: During normal operation the ICCCLK pin must be pulled up, internally or externally (external pull-up of 10k mandatory in noisy environment). This is to avoid entering ICC mode unexpectedly during a reset. In the application, even if the pin is configured as output, any reset will put it back in input pull-up.

Figure 5. Typical ICC interface



4.5 Memory protection

There are two different types of memory protection: Read-Out Protection and Write/Erase Protection which can be applied individually.

4.5.1 Read-out protection

Read-out protection, when selected provides a protection against program memory content extraction and against write access to Flash memory. Even if no protection can be considered as totally unbreakable, the feature provides a very high level of protection for a general purpose microcontroller. Both program and data E^2 memory are protected.

In flash devices, this protection is removed by reprogramming the option. In this case, both program and data E^2 memory are automatically erased and the device can be reprogrammed.

Read-out protection selection depends on the device type:

- In Flash devices it is enabled and removed through the FMP_R bit in the option byte.
- In ROM devices it is enabled by mask option specified in the Option List.

4.5.2 Flash write/erase protection

Write/erase protection, when set, makes it impossible to both overwrite and erase program memory. It does not apply to E^2 data. Its purpose is to provide advanced security to applications and prevent any change being made to the memory content.

Caution: Once set, Write/erase protection can never be removed. A write-protected flash device is no longer reprogrammable.

Write/erase protection is enabled through the FMP_W bit in the option byte.

4.6 Related documentation

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

4.7 Register description

Flash control/status register (FCSR)

Read / Write

Reset value: 0000 0000 (00h)

1st RASS Key: 0101 0110 (56h)

2nd RASS Key: 1010 1110 (AEh)

7							0
0	0	0	0	0	OPT	LAT	PGM

Note: This register is reserved for programming using ICP, IAP or other programming methods. It controls the XFlash programming and erasing operations.

When an EPB or another programming tool is used (in socket or ICP mode), the RASS keys are sent automatically.

Read operation (E2LAT=0)

The EEPROM can be read as a normal ROM location when the E2LAT bit of the EECSR register is cleared.

On this device, Data EEPROM can also be used to execute machine code. Take care not to write to the Data EEPROM while executing from it. This would result in an unexpected code being executed.

Write operation (E2LAT=1)

To access the write mode, the E2LAT bit has to be set by software (the E2PGM bit remains cleared). When a write access to the EEPROM area occurs, the value is latched inside the 32 data latches according to its address.

When PGM bit is set by the software, all the previous bytes written in the data latches (up to 32) are programmed in the EEPROM cells. The effective high address (row) is determined by the last EEPROM write sequence. To avoid wrong programming, the user must take care that all the bytes written between two programming sequences have the same high address: only the five Least Significant Bits of the address can change.

At the end of the programming cycle, the PGM and LAT bits are cleared simultaneously.

Note: Care should be taken during the programming cycle. Writing to the same memory location will over-program the memory (logical AND between the two write access data result) because the data latches are only cleared at the end of the programming cycle and by the falling edge of the E2LAT bit.
It is not possible to read the latched data.
This note is illustrated by the Figure 9: Data EEPROM programming cycle.

Figure 7. Data EEPROM programming flowchart

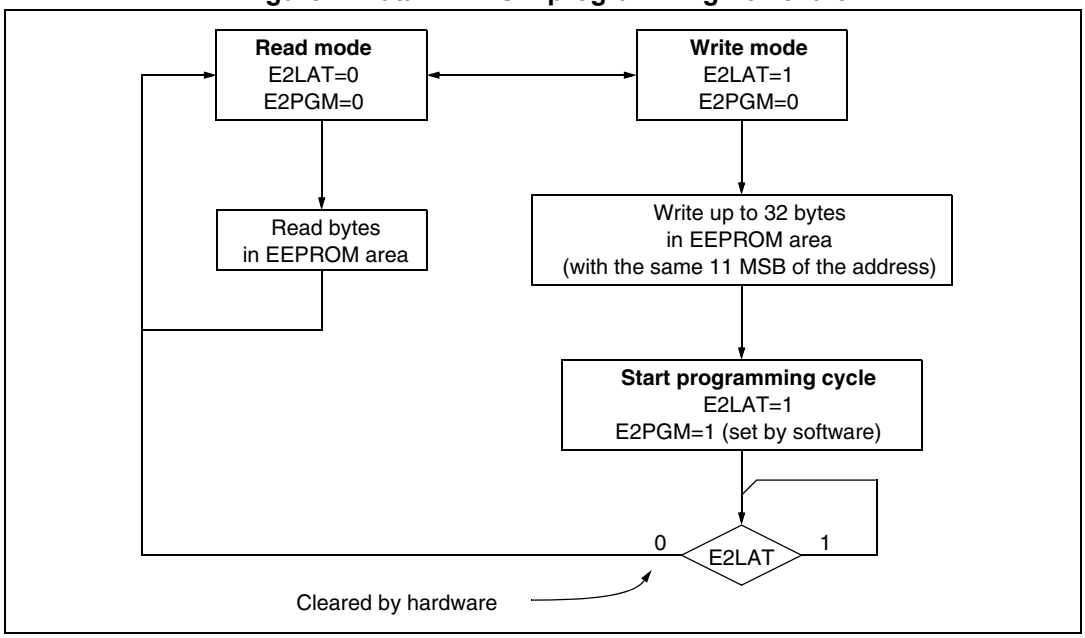


Table 5. DATA EEPROM register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0030h	**EECSR** Reset Value	0	0	0	0	0	0	E2LAT 0	E2PGM 0

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

for TA = -40 to +85°C) @ VDD = 4.5 to 5.5 V).

Refer to *Section 7.6.4: Register description* for a description of the LOCKED bit in the SICSR register.

7.3 Register description

Main clock control/status register (MCCSR)

Read / Write

Reset value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	MCO	SMS

- Bits 7:2 = Reserved, must be kept cleared
- Bit 1 = **MCO** *Main Clock Out enable*
  This bit is read/write by software and cleared by hardware after a reset. This bit allows to enable the MCO output clock.
  0: MCO clock disabled, I/O port free for general purpose I/O.
  1: MCO clock enabled.
- Bit 0 = **SMS** *Slow Mode select*
  This bit is read/write by software and cleared by hardware after a reset. This bit selects the input clock f_{OSC2} or $f_{OSC2}/32$.
  0: Normal mode ($f_{CPU} = f_{OSC2}$
  1: Slow mode ($f_{CPU} = f_{OSC2}/32$)

RC control register (RCCR)

Read / Write

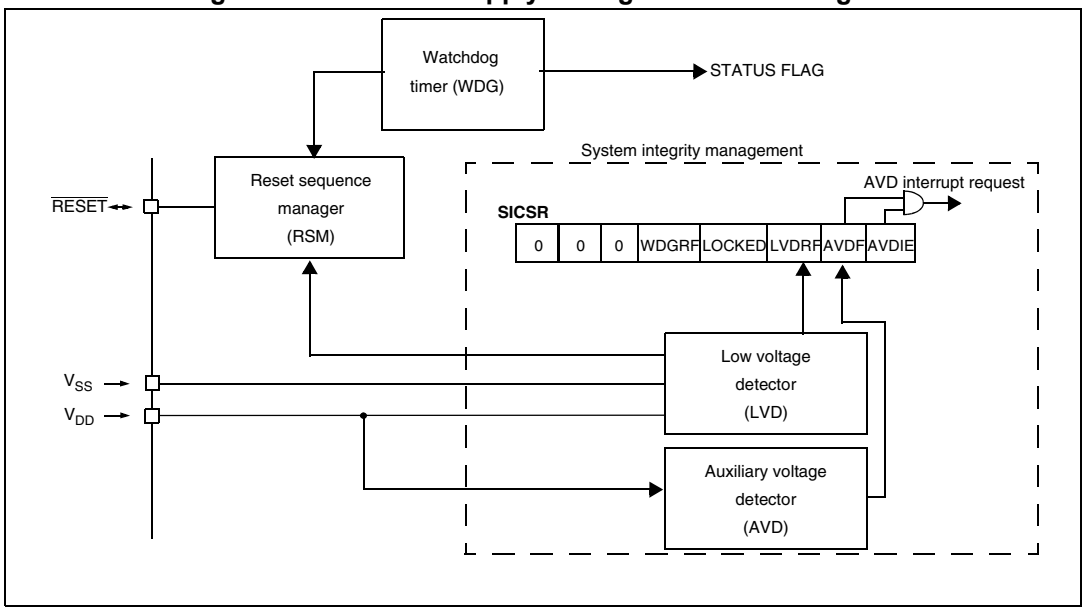
Reset value: 1111 1111 (FFh)

7							0
CR70	CR60	CR50	CR40	CR30	CR20	CR10	CR0

- Bits 7:0 = **CR[7:0]** *RC oscillator frequency adjustment bits*
  These bits must be written immediately after reset to adjust the RC oscillator frequency and to obtain an accuracy of 1%. The application can store the correct value for each voltage range in EEPROM and write it to this register at startup.
  00h = maximum available frequency
  FFh = lowest available frequency

Note: To tune the oscillator, write a serie of different values in the register until the correct frequency is reached. The fastest method is to use a dichotomy starting with 80h.

Figure 18. Reset and supply management block diagram



7.6.2 Auxiliary Voltage Detector (AVD)

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

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

Note: The AVD functions only if the LVD is enabled through the option byte.

Monitoring the V_{DD} main supply

The AVD voltage threshold value is relative to the selected LVD threshold configured by option byte (see *Section 15.1: Option bytes*).

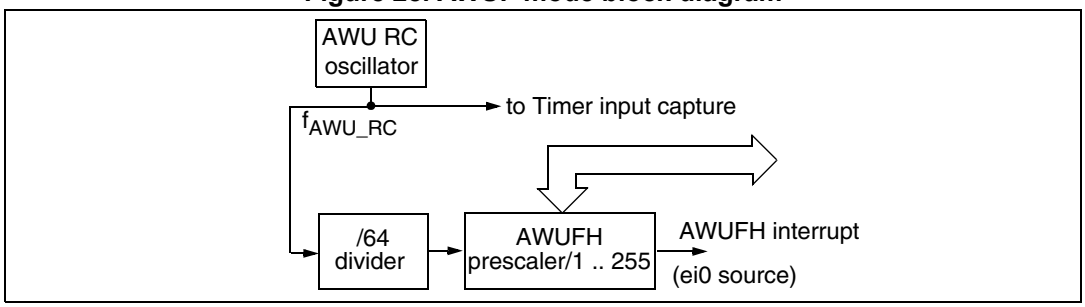
If the AVD interrupt is enabled, an interrupt is generated when the voltage crosses the $V_{IT+(LVD)}$ or $V_{IT-(AVD)}$ threshold (AVDF bit is set).

In the case of a drop in voltage, the AVD interrupt acts as an early warning, allowing software to shut down safely before the LVD resets the microcontroller (See *Figure 19: Using the AVD to monitor VDD*).

9.6 Auto-wakeup from HALT mode

Auto Wake Up From Halt (AWUF) mode is similar to Halt mode with the addition of a specific internal RC oscillator for wake-up (Auto Wake Up from Halt Oscillator). Compared to ACTIVE-HALT mode, AWUF has lower power consumption (the main clock is not kept running, but there is no accurate realtime clock available. It is entered by executing the HALT instruction when the AWUEN bit in the AWUCSR register has been set.

Figure 28. AWUF mode block diagram



As soon as HALT mode is entered, and if the AWUEN bit has been set in the AWUCSR register, the AWU RC oscillator provides a clock signal (f_{AWU_RC}). Its frequency is divided by a fixed divider and a programmable prescaler controlled by the AWUPR register. The output of this prescaler provides the delay time. When the delay has elapsed the AWUF flag is set by hardware and an interrupt wakes-up the MCU from Halt mode. At the same time the main oscillator is immediately turned on and a 256 or 4096 cycle delay is used to stabilize it. After this start-up delay, the CPU resumes operation by servicing the AWUF interrupt. The AWU flag and its associated interrupt are cleared by software reading the AWUCSR register.

To compensate for any frequency dispersion of the AWU RC oscillator, it can be calibrated by measuring the clock frequency f_{AWU_RC} and then calculating the right prescaler value. Measurement mode is enabled by setting the AWUM bit in the AWUCSR register in Run mode. This connects f_{AWU_RC} to the input capture of the 12-bit Auto-Reload timer, allowing the f_{AWU_RC} to be measured using the main oscillator clock as a reference timebase.

Similarities with HALT mode:

The following AWUF mode behavior is the same as normal Halt mode:

- The MCU can exit AWUF mode by means of any interrupt with exit from Halt capability or a reset (see *Section 9.4: HALT mode*).
- When entering AWUF mode, the I bit in the CC register is forced to 0 to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.
- In AWUF mode, the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. None of the peripherals are clocked except those which get their clock supply from another clock generator (such as an external or auxiliary oscillator like the AWU oscillator).
- The compatibility of Watchdog operation with AWUF mode is configured by the WDGHALT option bit in the option byte. Depending on this setting, the HALT instruction when executed while the Watchdog system is enabled, can generate a Watchdog RESET.

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

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.

11.3.4 Low power modes

Table 37. Effect of low power modes on Lite timer

Mode	Description
SLOW	No effect on Lite timer (this peripheral is driven directly by $f_{OSC}/32$)
WAIT	No effect on Lite timer
ACTIVE-HALT	No effect on Lite timer
HALT	Lite timer stops counting

11.3.5 Interrupts

Table 38. TBxF and ICF interrupt events

Interrupt event	Event flag	Enable Control bit	Exit from WAIT	Exit from ACTIVE-HALT	Exit from HALT
Timebase 1 event	TB1F	TB1IE	Yes	Yes	No
Timebase 2 event	TB2F	TB2IE	Yes	No	No
IC event	ICF	ICIE	Yes	No	No

Note: The TBxF and ICF interrupt events are connected to separate interrupt vectors (see Interrupts chapter).

They generate an interrupt if the enable bit is set in the LTCSR1 or LTCSR2 register and the interrupt mask in the CC register is reset (RIM instruction).

11.3.6 Register description

Lite timer control/status register 2 (LTCSR2)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	TB2IE	TB2F

- Bits 7:2 = Reserved, must be kept cleared.
- Bit 1 = **TB2IE** *Timebase 2 Interrupt enable*
  This bit is set and cleared by software.
  0: Timebase (TB2) interrupt disabled
  1: Timebase (TB2) interrupt enabled
- Bit 0 = **TB2F** *Timebase 2 Interrupt Flag*
  This bit is set by hardware and cleared by software reading the LTCSR register. Writing to this bit has no effect.
  0: No Counter 2 overflow

0: Timebase period = t_{OSC} * 8000 (1 ms @ 8 MHz)
1: Timebase period = t_{OSC} * 16000 (2 ms @ 8 MHz)

- Bit 4 = **TB1IE** *Timebase interrupt enable*
  This bit is set and cleared by software.
  0: Timebase (TB1) interrupt disabled
  1: Timebase (TB1) interrupt enabled
- Bit 3 = **TB1F** *Timebase interrupt flag*
  This bit is set by hardware and cleared by software reading the LTCSR register. Writing to this bit has no effect.
  0: No counter overflow
  1: A counter overflow has occurred
- Bit 2:0 = reserved.

Lite timer input capture register (LTICR)

Read only

Reset Value: 0000 0000 (00h)

7							0
ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0

- Bits 7:0 = **ICR[7:0]** *Input capture value*
  These bits are read by software and cleared by hardware after a reset. If the ICF bit in the LTCSR is cleared, the value of the 8-bit up-counter will be captured when a rising or falling edge occurs on the LTIC pin.

Table 39. Lite timer register map and reset values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
08	**LTCSR2**							TB2IE	TB2F
	Reset Value	0	0	0	0	0	0	0	0
09	**LTARR**	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
	Reset Value	0	0	0	0	0	0	0	0
0A	**LTCNTR**	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
	Reset Value	0	0	0	0	0	0	0	0
0B	**LTCSR1**	ICIE	ICF	TB	TB1IE	TB1F			
	Reset Value	0	x	0	0	0	0	0	0
0C	**LTICR**	ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0
	Reset Value	0	0	0	0	0	0	0	0

Table 49. ST7 addressing mode overview

Mode			Syntax	Destination/ source	Pointer address (Hex.)	Pointer size (Hex.)	Length (bytes)
Long	Direct	Indexed	ld A,($1000,X)	0000..FFFF	–	–	+ 2
Short	Indirect	–	ld A,[$10]	00..FF	00..FF	byte	+ 2
Long	Indirect	–	ld A,[$10.w]	0000..FFFF	00..FF	word	+ 2
Short	Indirect	Indexed	ld A,([$10],X)	00..1FE	00..FF	byte	+ 2
Long	Indirect	Indexed	ld A,([$10.w],X)	0000..FFFF	00..FF	word	+ 2
Relative	Direct	–	jrne loop	PC-128/ PC+127(1)	–	–	+ 1
Relative	Indirect	–	jrne [$10]	PC-128/ PC+127(1)	00..FF	byte	+ 2
Bit	Direct	–	bset $10,#7	00..FF	–	–	+ 1
Bit	Indirect	–	bset [$10],#7	00..FF	00..FF	byte	+ 2
Bit	Direct	Relative	btjt $10,#7,skip	00..FF	–	–	+ 2
Bit	Indirect	Relative	btjt [$10],#7,skip	00..FF	00..FF	byte	+ 3

1. At the time the instruction is executed, the Program Counter (PC) points to the instruction following JRxx.

12.1.1 Inherent

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

Table 50. Inherent instructions

Instruction	Function
NOP	No Operation
TRAP	S/W Interrupt
WFI	WAIT for Interrupt (low power mode)
HALT	HALT oscillator (lowest power mode)
RET	Sub-routine Return
IRET	Interrupt sub-routine Return
SIM	Set Interrupt Mask (level 3)
RIM	Reset Interrupt Mask (level 0)
SCF	Set Carry Flag
RCF	Reset Carry Flag
RSP	Reset Stack Pointer
LD	Load
CLR	Clear

Table 55. Instruction groups (continued)

Group	Instructions							
Bit operation	BSET	BRES	–	–	–	–	–	–
Conditional Bit Test and Branch	BTJT	BTJF	–	–	–	–	–	–
Arithmetic operations	ADC	ADD	SUB	SBC	MUL	–	–	–
Shift and Rotates	SLL	SRL	SRA	RLC	RRC	SWAP	SLA	–
Unconditional Jump or Call	JRA	JRT	JRF	JP	CALL	CALLR	NOP	RET
Conditional Branch	JRxx	–	–	–	–	–	–	–
Interruption management	TRAP	WFI	HALT	IRET	–	–	–	–
Condition Code Flag modification	SIM	RIM	SCF	RCF	–	–	–	–

Using a prebyte

The instructions are described with 1 to 4 bytes.

In order to extend the number of available opcodes for an 8-bit CPU (256 opcodes), three different prebyte opcodes are defined. These prebytes modify the meaning of the instruction they precede.

The whole instruction becomes:

- PC-2: End of previous instruction
- PC-1: Prebyte
- PC: Opcode
- PC+1: Additional word (0 to 2) according to the number of bytes required to compute the effective address.

These prebytes enable instruction in Y as well as indirect addressing modes to be implemented.

They precede the opcode of the instruction in X or the instruction using direct addressing mode. The prebytes are:

- PDY 90: Replace an X based instruction using immediate, direct, indexed, or inherent addressing mode by a Y one.
- PIX 92: Replace an instruction using direct, direct bit or direct relative addressing mode to an instruction using the corresponding indirect addressing mode.
  It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.
- PIY 91: Replace an instruction using X indirect indexed addressing mode by a Y one.

12.2.1 Illegal opcode reset

In order to provide enhanced robustness to the device against unexpected behavior, a system of illegal opcode detection is implemented. If a code to be executed does not correspond to any opcode or prebyte value, a reset is generated. This, combined with the Watchdog, allows the detection and recovery from an unexpected fault or interference.

Note: A valid prebyte associated with a valid opcode forming an unauthorized combination does not generate a reset.

Table 79. Electrical sensitivities

Symbol	Parameter	Conditions	Class(1)
LU	Static latch-up class	T_A=+25°C	A
DLU	Dynamic latch-up class	V_{DD}=5.5V, f_{OSC}=4MHz, T_A=+25°C	A

1. Class description: A Class is an STMicroelectronics internal specification. All its limits are higher than the JEDEC specifications, that means when a device belongs to Class A it exceeds the JEDEC standard. B Class strictly covers all the JEDEC criteria (international standard).

13.8 I/O port pin characteristics

Table 80. General characteristics(1)

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IL}	Input low level voltage	–		V_{SS} - 0.3	–	0.3 x V_{DD}	V
V_{IH}	Input high level voltage	–		0.7 x V_{DD}	–	V_{DD} + 0.3	
V_{hys}	Schmitt trigger voltage hysteresis(2)	–		–	400	–	mV
I_L	Input leakage current	V_{SS}≤V_{IN}≤V_{DD}		–	–	±1	µA
I_S	Static current consumption induced by each floating input pin(3)	Floating input mode		–	400	–	
R_{PU}	Weak pull-up equivalent resistor(4)	V_{IN}= V_{SS}	V_{DD}=5V	50	120	250	kΩ
			V_{DD}=3V	–	160	–	
C_{IO}	I/O pin capacitance	–		–	5	–	pF
$t_{f(IO)out}$	Output high to low level fall time(2)	C_L=50pF Between 10% and 90%		–	25	–	ns
$t_{r(IO)out}$	Output low to high level rise time(2)			–	25	–	
$t_{w(IT)in}$	External interrupt pulse time(5)	–		1	–	–	t_{CPU}

1. Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.
2. Data based on characterization results, not tested in production.
3. Configuration not recommended, all unused pins must be kept at a fixed voltage: using the output mode of the I/O for example or an external pull-up or pull-down resistor (see*Figure 67*). Static peak current value taken at a fixed V_{IN} value, based on design simulation and technology characteristics, not tested in production. This value depends on V_{DD} and temperature values.
4. The R_{PU} pull-up equivalent resistor is based on a resistive transistor (corresponding I_{PU} current characteristics described in *Figure 68*).
5. To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

15 Device configuration

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

ST7FLITE2 devices are FLASH versions. ST7PLITE2 devices are Factory Advanced Service Technique ROM (FASTROM) versions: they are factory programmed FLASH devices.

ST7FLITE2 devices are shipped to customers with a default program memory content (FFh), while FASTROM factory coded parts contain the code supplied by the customer. This implies that FLASH devices have to be configured by the customer using the Option Bytes while the FASTROM devices are factory configured.

15.1 Option bytes

The two option bytes allow the hardware configuration of the microcontroller to be selected.

The option bytes can be accessed only in programming mode (for example using a standard ST7 programming tool).

15.1.1 Option byte 0

- OPT7 = Reserved, must always be 1
- OPT6:4 = **OSCRANGE[2:0]** *Oscillator range*
  When the internal RC oscillator is not selected (Option OSC=1), these option bits select the range of the resonator oscillator current source or the external clock source.

Table 92. Option bytes values

			OSCRANGE		
			2	**1**	**0**
Typ. frequency range with Resonator	LP	1~2MHz	0	0	0
	MP	2~4MHz	0	0	1
	MS	4~8MHz	0	1	0
	HS	8~16MHz	0	1	1
	VLP	32.768kHz	1	0	0
External Clock source: CLKIN	on OSC1		1	0	1
	on PB4		1	1	1
Reserved			1	1	0

Note: When the internal RC oscillator is selected, the OSCRANGE option bits must be kept at their default value in order to select the 256 clock cycle delay (see Section 7.5: Reset sequence manager (RSM)).

- OPT3:2 = **SEC[1:0]** *Sector 0 size definition*
  These option bits indicate the size of sector 0 according to *Table 93*.