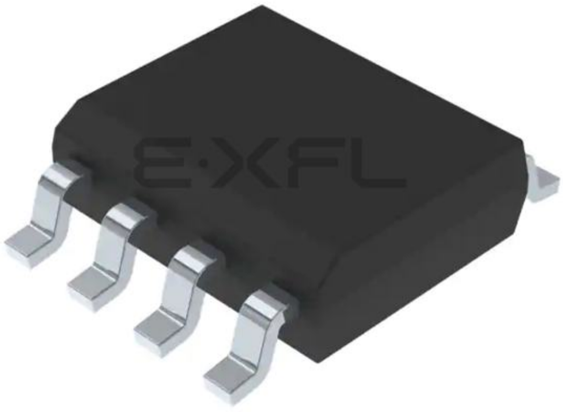


Welcome to E-XFL.COM

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	-
Peripherals	LVD, POR, PWM, WDT
Number of I/O	5
Program Memory Size	1KB (1K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.154", 3.90mm Width)
Supplier Device Package	8-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fliteus2m6tr

1 Introduction

The ST7LITEUS2 and ST7LITEUS5 are members of the ST7 microcontroller family. All ST7 devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set.

The ST7LITEUS2 and ST7LITEUS5 feature FLASH memory with byte-by-byte In-Circuit Programming (ICP) and In-Application Programming (IAP) capability.

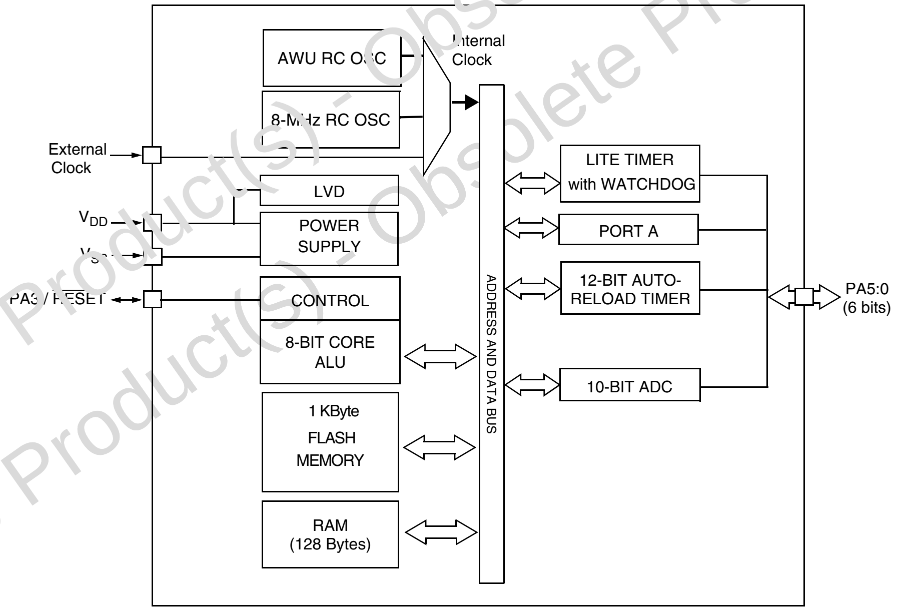
Under software control, the ST7LITEUS2 and ST7LITEUS5 can be placed in Wait, Slow, or Halt mode, reducing power consumption when the application is in idle or standby state.

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 12 on page 92*.

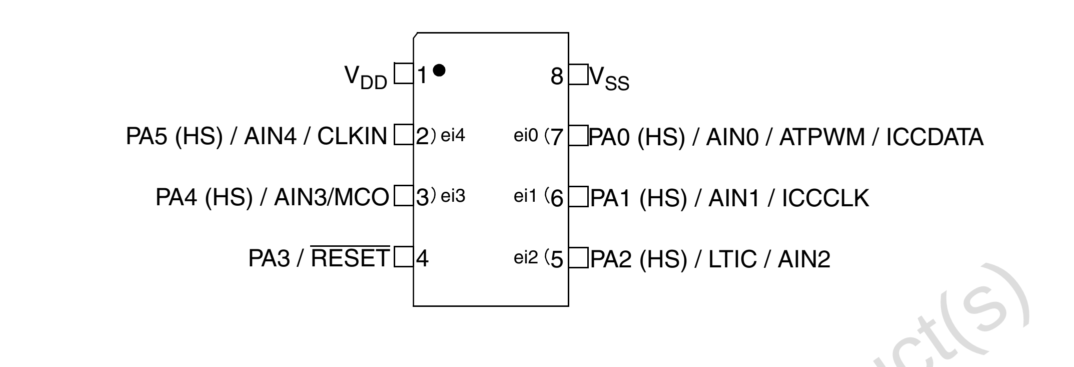
The devices feature an on-chip debug module (DM) to support in-circuit debugging (ICD). For a description of the DM registers, refer to the ST7 I^2C protocol reference manual.

Figure 1. General block diagram



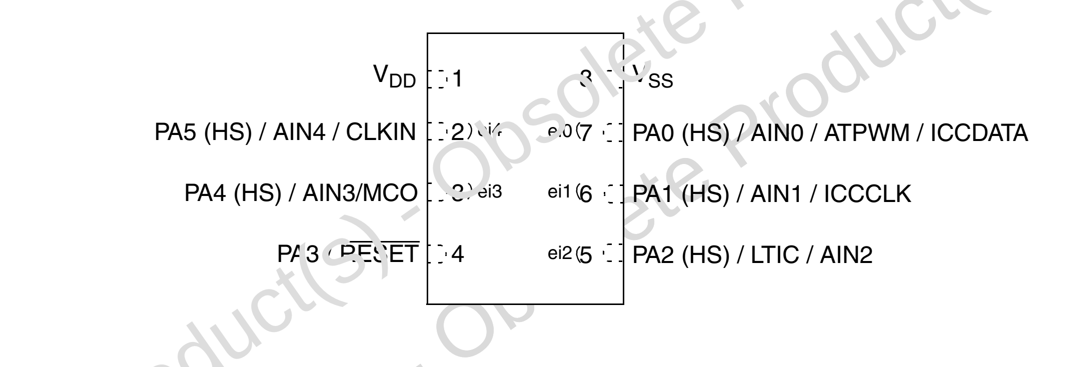
2 Pin description

Figure 2. 8-pin SO and Plastic DIP package pinout



1. HS: High sink capability.
2. eix : associated external interrupt vector

Figure 3. 8-pin DFN package pinout



1. HS: High sink capability.
2. eix : associated external interrupt vector

Table 3. Hardware register map (continued)[1]

Address	Block	Register label	Register name	Reset status	Remarks
0049h	AWU	AWUPR	AWU Prescaler register	FFh	R/W
004Ah		AWUCSR	AWU Control/Status register	00h	R/W
004Bh	DM[4]	DMCR	DM Control register	00h	R/W
004Ch		DMSR	DM Status register	00h	R/W
004Dh		DMBK1H	DM Breakpoint register 1 High	00h	R/W
004Eh		DMBK1L	DM Breakpoint register 1 Low	00h	R/W
004Fh		DMBK2H	DM Breakpoint register 2 High	00h	R/W
0050h		DMBK2L	DM Breakpoint register 2 Low	00h	R/W
0051h to 007Fh	Reserved area (47 bytes)				

1. Legend: x=undefined, R/W=read/write
2. The contents of the I/O port DR registers are readable only in output configuration. In input configuration, the values of the I/O pins are returned instead of the DR register contents.
3. The bits associated with unavailable pins must always keep their reset value.
4. For a description of the DM registers, see the ST7 I^2C Protocol Reference Manual.

the I²C protocol routine. This routine enables the ST7 to receive bytes from the I²C interface.

- Download ICP driver code in RAM from the ICCDATA pin
- Execute ICP driver code in RAM to program the FLASH memory

Depending on the ICP driver code downloaded in RAM, FLASH memory programming can be fully customized (number of bytes to program, program locations, or selection of the serial communication interface for downloading).

4.3.2 In application programming (IAP)

This mode uses an IAP driver program previously programmed in Sector 0 by the user (in ICP mode).

This mode is fully controlled by user software. This allows it to be adapted to the user application, (user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored etc).

IAP mode can be used to program any memory areas except Sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

4.4 I²C interface

ICP needs a minimum of 4 and up to 6 pins to be connected to the programming tool. These pins are:

- $\overline{\text{RESET}}$: device reset
- V_{SS}: device power supply ground
- ICCCLK: I²C output serial clock pin
- ICCDATA: I²C input serial data pin
- CLKIN: main clock input for external source
- V_{DD}: application board power supply

Refer to *Figure 6* for a description of the I²C interface.

If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the programming tool is plugged to the board, even if an I²C session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to be implemented in case another device forces the signal. Refer to the programming tool documentation for recommended resistor values.

During the ICP session, the programming tool must control the $\overline{\text{RESET}}$ pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5 mA at high level (push pull output or pull-up resistor<1 kΩ). A schottky diode can be used to isolate the application RESET circuit in this case. When using a classical RC network with R>1 kΩ or a reset management IC with open drain output and pull-up resistor>1 kΩ, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the I²C session.

The use of Pin 7 of the I²C connector depends on the programming tool architecture. This pin must be connected when using most ST programming tools (it is used to monitor the application power supply). Please refer to the programming tool manual.

4.7 Register description

4.7.1 Flash Control/Status register (FCSR)

This register controls the XFlash erasing and programming using ICP, IAP or other programming methods.

1st RASS Key: 0101 0110 (56h)

2nd RASS Key: 1010 1110 (AEh)

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

Reset value: 000 0000 (00h)

7							0
0	0	0	0	0	OPT	LAT	PGM
Read/write							

Table 4. FLASH register map and reset values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
002Fh	FCSR Reset value	0	0	0	0	0	OPT 0	LAT 0	PGM 0

6.3.2 RC Control register (RCCR)

Reset value: 1111 1111 (FFh)

7							0
CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2
Read / Write							

Bits 7:0 **CR[9:2]** *RC Oscillator Frequency Adjustment Bits*
These bits, as well as CR[1:0] bits in the SICSR register must be written immediately after reset to adjust the RC oscillator frequency and to obtain the required accuracy. The application can store the correct value for each voltage range in Flash memory and write it to this register at startup.
00h = maximum available frequency
FFh = lowest available frequency

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

6.3.3 System Integrity (SI) Control/status register (SICSR)

Reset value: 0000 0x00 (0xh)

7							0
0	CR1	CR0	0	0	LVDRF	AVDF	AVDIE
Read / Write							

Bit 7 Reserved, must be kept cleared.

Bits 6:5 **CR[1:0]** *RC Oscillator Frequency Adjustment bits*
These bits, as well as CR[9:2] bits in the RCCR register must be written immediately after reset to adjust the RC oscillator frequency and to obtain the required accuracy. Refer to *Section 6.2 on page 28*.

Bits 4:3 Reserved, must be kept cleared.

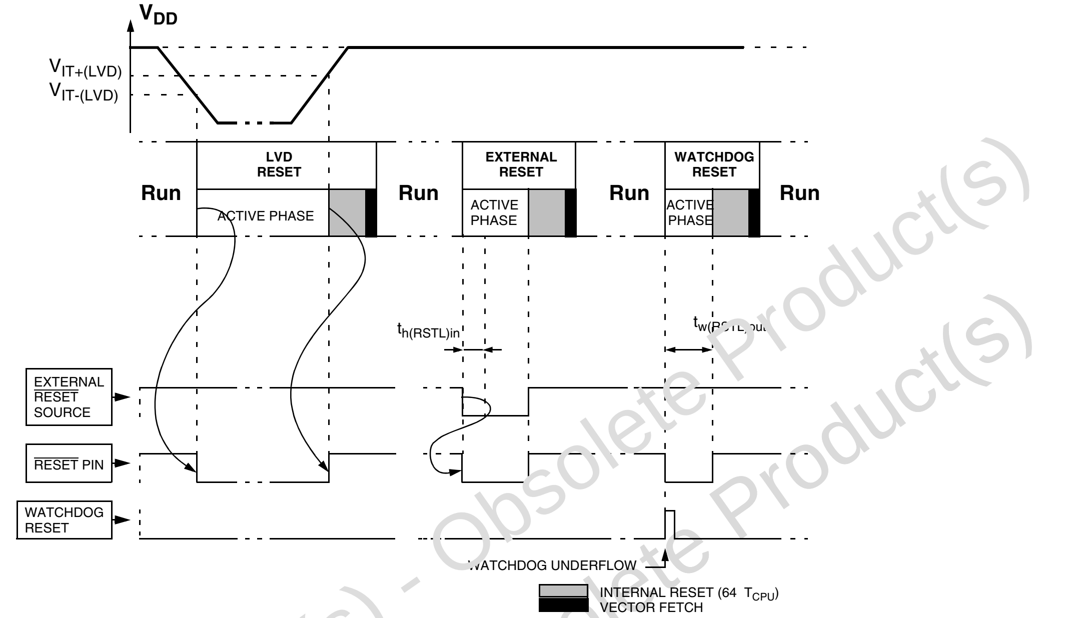
Bits 2:0 System Integrity bits. Refer to *Section 7.4 on page 43*.

6.4.5 Internal watchdog reset

The reset sequence generated by a internal watchdog counter overflow is shown in *Figure 13*.

Starting from the watchdog counter underflow, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.

Figure 13. Reset sequences



6.5 Register description

6.5.1 Multiplexed I/O Reset Control register 1 (MUXCR1)

Reset value: 0000 0000 (00h)

7							0
MIR15	MIR14	MIR13	MIR12	MIR11	MIR10	MIR9	MIR8
Read / Write once							

6.5.2 Multiplexed I/O Reset Control register 0 (MUXCR0)

Reset value: 0000 0000 (00h)

7							0
MIR7	MIR6	MIR5	MIR4	MIR3	MIR2	MIR1	MIR0
Read / Write once							

Bits 15:0 **MIR[15:0]**

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.

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 in 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.

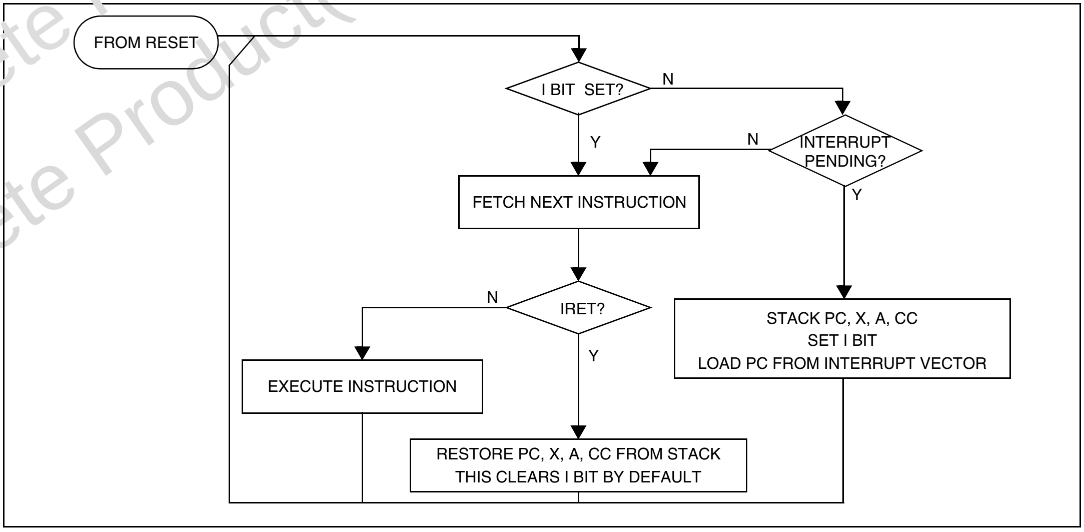
If any of these two conditions is false, the interrupt is latched and thus remains pending.

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 for being enabled) will therefore be lost if the clear sequence is executed.

Figure 14. Interrupt processing flowchart



Bit 2 **LVDRF** *LVD reset flag*
This bit indicates that the last Reset was generated by the LVD block. It is set by hardware (LVD reset) and cleared when read. See WDGRF flag description in *Section 10.1.6 on page 69* for more details. When the LVD is disabled by OPTION BYTE, the LVDRF bit value is undefined.

Note: If the selected clock source is one of the two internal ones, and if V_{DD} remains below the selected LVD threshold during less than T_{AWU} (33us typ.), the LVDRF flag cannot be set even if the device is reset by the LVD. If the selected clock source is the external clock (CLKIN), the flag is never set if the reset occurs during Halt mode. In run mode the flag is set only if f_{CLKIN} is greater than 10 MHz.

Bit 1 **AVDF** *Voltage Detector flag*
This read-only bit is set and cleared by hardware. If the AVDIE bit is set, an interrupt request is generated when the AVDF bit is set. Refer to *Figure 17* for additional details
0: V_{DD} over AVD threshold
1: V_{DD} under AVD threshold

Bit 0 **AVDIE** *Voltage Detector interrupt enable*
This bit is set and cleared by software. It enables an interrupt to be generated when the AVDF flag is set. The pending interrupt information is automatically cleared when software enters the AVD interrupt routine.
0: AVD interrupt disabled
1: AVD interrupt enabled

AVD Threshold Selection register (AVDTHCR)

Refer to *Section 6.3.4: AVD Threshold Selection register (AVDTHCR)* for a full description of this register.

Application notes

The LVDRF flag is not cleared when another reset type occurs (external or watchdog), the LVDRF flag remains set to keep trace of the original failure.
In this case, a watchdog reset can be detected by software while an external reset can not.

Table 13. System integrity register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
003Ah	SICSR reset value	0	1	1	0	0	LVDRF x	AVDF 0	AVDIE 0
003Eh	AVDTHCR reset value	CK2 0	CK1 0	CK0 0	0	0	0	AVD1 1	AVD2 1

8 Power saving modes

8.1 Introduction

To give a large measure of flexibility to the application in terms of power consumption, four main power saving modes are implemented in the ST7 (see *Figure 18*):

- Slow
- Wait (and Slow-wait)
- Active-halt
- Auto-wakeup from Halt (AWUFH)
- Halt

After a reset the normal operating mode is selected by default (Run mode). This mode drives the device (CPU and embedded peripherals) by means of a master clock which is based on the main oscillator frequency (f_{OSC}).

From Run mode, the different power saving modes may be selected by setting the relevant register bits or by calling the specific ST7 software instruction whose action depends on the oscillator status.

Figure 18. Power saving mode transitions

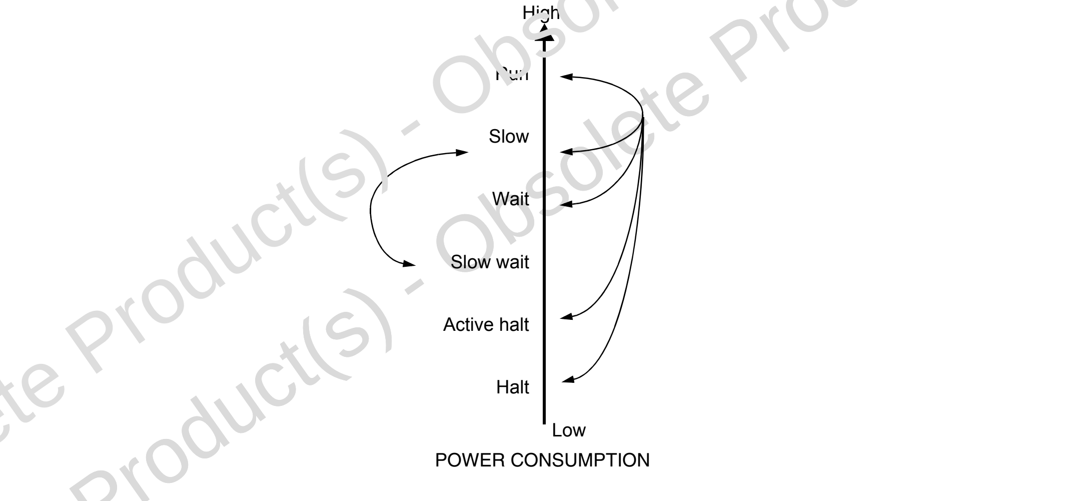


Table 15. **Configuring the dividing factor**

AWUPR[7:0]	Dividing factor
00h	Forbidden
01h	1
...	...
FEh	254
FFh	255

In AWU mode, the period that the MCU stays in Halt Mode (t_{AWU} in *Figure 26*) is defined by

$$t_{AWU} = 64 \times AWUPR \times \frac{1}{f_{AWURC}} + t_{RCSTRT}$$

This prescaler register can be programmed to modify the time that the MCU stays in Halt mode before waking up automatically.

Note: *If 00h is written to AWUPR, depending on the product, an interrupt is generated immediately after a HALT instruction, or the AWUPR remains unchanged.*

Table 16. **AWU register map and reset values**

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0049h	**AWUPR** Reset value	AWUPR7 1	AWUPR6 1	AWUPR5 1	AWUPR4 1	AWUPR3 1	AWUPR2 1	AWUPR1 1	AWUPR0 1
004Ah	**AWUCSR** Reset value	0	0	0	0	0	AWUF	AWUM	AWUEN

10.1.6 Register description

Lite timer control/status register (LTCSR)

Reset value: 0000 0x00 (0xh)

7							0
ICIE	ICF	TB	TBIE	TBF	WDGR	WDGE	WDGD
Read / Write							

Bit 7 **ICIE** *Interrupt Enable.*

This bit is set and cleared by software.

0: Input Capture (IC) interrupt disabled

1: Input Capture (IC) interrupt enabled

Bit 6 **ICF** *Input Capture Flag.*

This bit is set by hardware and cleared by software by reading the LTICR register. Writing to this bit does not change the bit value.

0: No input capture

1: An input capture has occurred

Note: After an MCU reset, software must initialise the ICF bit by reading the LTICR register

Bit 5 **TB** *Timebase period selection.*

This bit is set and cleared by software.

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

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

Bit 4 **TBIE** *Timebase Interrupt enable.*

This bit is set and cleared by software.

0: Timebase (TB) interrupt disabled

1: Timebase (TB) interrupt enabled

Bit 3 **TBF** *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

Table 36. ST7 addressing mode overview (continued)[1]

Mode			Syntax	Destination/ source	Pointer address	Pointer size	Length (bytes)
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.

11.1.1 Inherent mode

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

Table 37. Instructions supporting inherent addressing mode

Inherent instruction	Function
NOP	No operation
TRAP	S/W Interrupt
WFI	Wait For Interrupt (low power mode)
HALT	Halt Oscillator (lowest power mode)
RET	Subroutine return
IRET	Interrupt subroutine return
SIM	Set interrupt mask
RIM	Reset interrupt mask
SCF	Set carry flag
RCF	Reset carry flag
RSP	Reset stack pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 complement
MUL	Byte multiplication
SLL, SRL, SRA, RLC, RRC	Shift and rotate operations
SWAP	Swap nibbles

11.1.2 Immediate

Immediate instructions have 2 bytes, the first byte contains the opcode, the second byte contains the operand value.

Table 38. Instructions supporting inherent immediate addressing mode

Immediate instruction	Function
LD	Load
CP	Compare
BCP	Bit compare
AND, OR, XOR	Logical operations
ADC, ADD, SUB, SBC	Arithmetic operations

11.1.3 Direct

In Direct instructions, the operands are referenced by their memory address.

The direct addressing mode consists of two submodes:

Direct (short) addressing mode

the address is a byte, thus requires only 1 byte after the opcode, but only allows 00 - FF addressing space.

Direct (long) addressing mode

The address is a word, thus allowing 64 Kbyte addressing space, but requires 2 bytes after the opcode.

11.1.4 Indexed mode (no offset, short, long)

In this mode, the operand is referenced by its memory address, which is defined by the unsigned addition of an index register (X or Y) with an offset.

The indirect addressing mode consists of three submodes:

Indexed mode (no offset)

There is no offset (no extra byte after the opcode), and allows 00 - FF addressing space.

Indexed mode (short)

The offset is a byte, thus requires only 1 byte after the opcode and allows 00 - 1FE addressing space.

Indexed mode (long)

The offset is a word, thus allowing 64 Kbyte addressing space and requires 2 bytes after the opcode.

Table 41. ST7 instruction set (continued)

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.

11.2.1 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.*

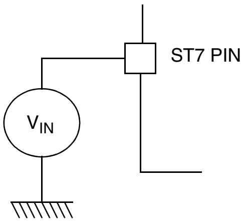
Table 42. Illegal opcode detection

Mnemo	Description	Function/example	Dst	Src	H	I	N	Z	C
ADC	Add with carry	A = A + M + C	A	M	H	-	N	Z	C
ADD	Addition	A = A + M	A	M	H	-	N	Z	C
AND	Logical and	A = A . M	A	M	-	-	N	Z	-
BCP	Bit compare A, Memory	tst (A . M)	A	M	-	-	N	Z	-

12.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in *Figure 38*.

Figure 38. Pin input voltage



12.2 Absolute maximum ratings

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

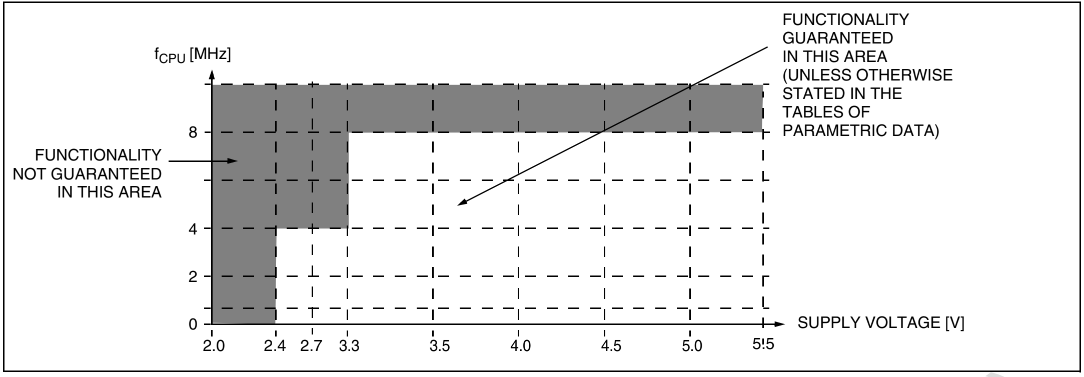
1. Directly connecting the I/O pins to V_{DD} or V_{SS} could damage the device if an unexpected change of the I/O configuration occurs (for example, due to a corrupted program counter). To guarantee safe operation, this connection has to be done through a pull-up or pull-down resistor (typical: 10 kΩ for I/Os). Unused I/O pins must be tied in the same way to V_{DD} or V_{SS} according to their reset configuration.

Table 43. Voltage characteristics

Symbol	Ratings	Maximum value	Unit
V_{DD} - V_{SS}	Supply voltage	7.0	V
V_{IN}	Input voltage on any pin[1]	V_{SS}-0.3 to V_{DD}+0.3	
$V_{ESD(HBM)}$	Electrostatic discharge voltage (Human Body Model)	see *Section 12.7.2*	
$V_{ESD(MM)}$	Electrostatic discharge voltage (Machine Model)	see *Section 12.7.2*	

[1] $I_{INJ(PIN)}$ must never be exceeded. This is implicitly insured if V_{IN} maximum is respected. If V_{IN} maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by V_{IN}>V_{DD} while a negative injection is induced by V_{IN}<V_{SS}.

Figure 39. f_{CPU} maximum operating frequency versus V_{DD} supply voltage



12.3.2 Operating conditions with low voltage detector (LVD)

T_A = -40 to 125 °C, unless otherwise specified

Table 47. Operating characteristics with LVD

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(LVD)}$	Reset release threshold (V_{DD} rise)	High threshold Med. threshold Low threshold	3.9 3.2 2.5	4.2 3.5 2.7	4.5 3.8 3.0	V
$V_{IT-(LVD)}$	Reset generation threshold (V_{DD} fall)	High threshold Med. threshold Low threshold	3.7 3.0 2.4	4.0 3.3 2.6	4.3 3.6 2.9	
V_{hys}	LVD voltage threshold hysteresis	$V_{IT+(LVD)}$-$V_{IT-(LVD)}$		150		mV
V_{tPOR}	V_{DD} rise time rate [1][2]		20			µs/V
$I_{DD(LVD)}$ [3]	LVD/AVD current consumption	V_{DD} = 5 V		220		µA

1. Not tested in production. The V_{DD} rise time rate condition is needed to ensure a correct device power-on and LVD reset release. When the V_{DD} slope is outside these values, the LVD may not release properly the reset of the MCU

2. 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 0V to ensure optimum restart conditions. Refer to circuit example in *Figure 61 on page 114*.

3. Not tested in production.

Figure 45. I_{DD} vs temp @V_{DD} 5 V & int RC = 8 MHz

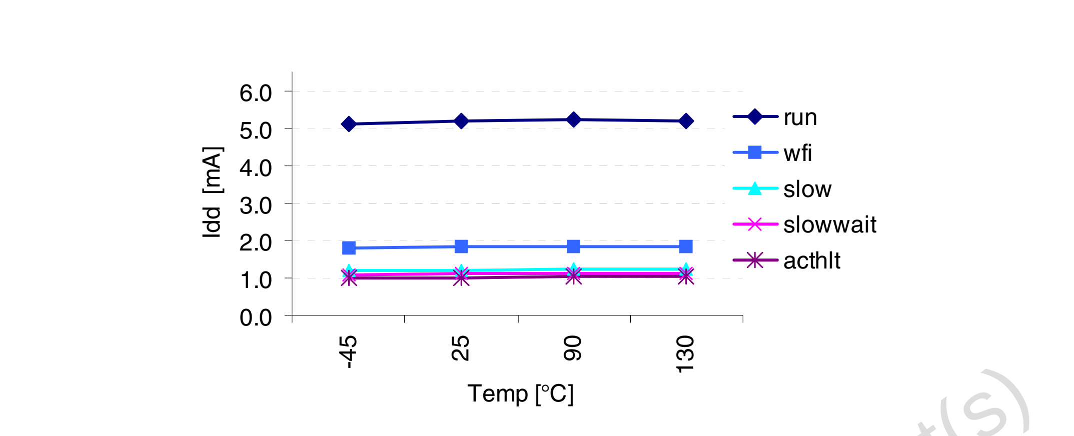


Figure 46. I_{DD} vs temp @V_{DD} 5 V & int RC = 4 MHz

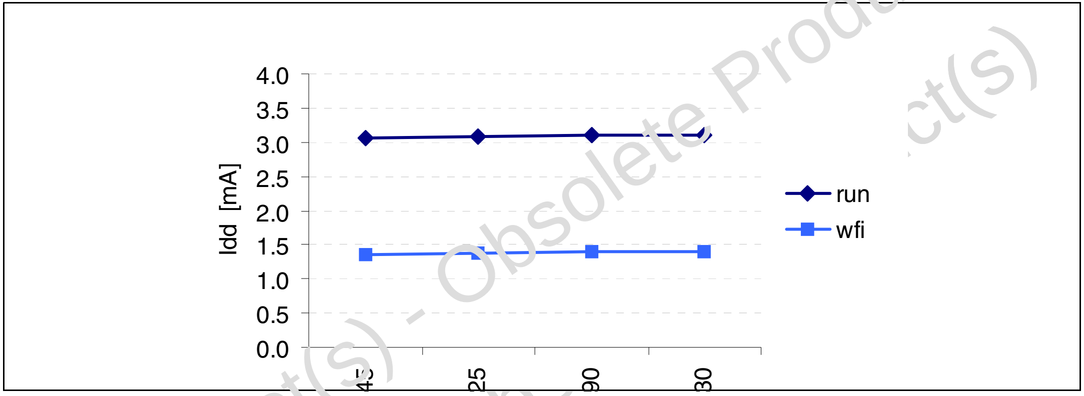


Figure 47. I_{DD} vs temp @V_{DD} 5 V & int RC = 2 MHz

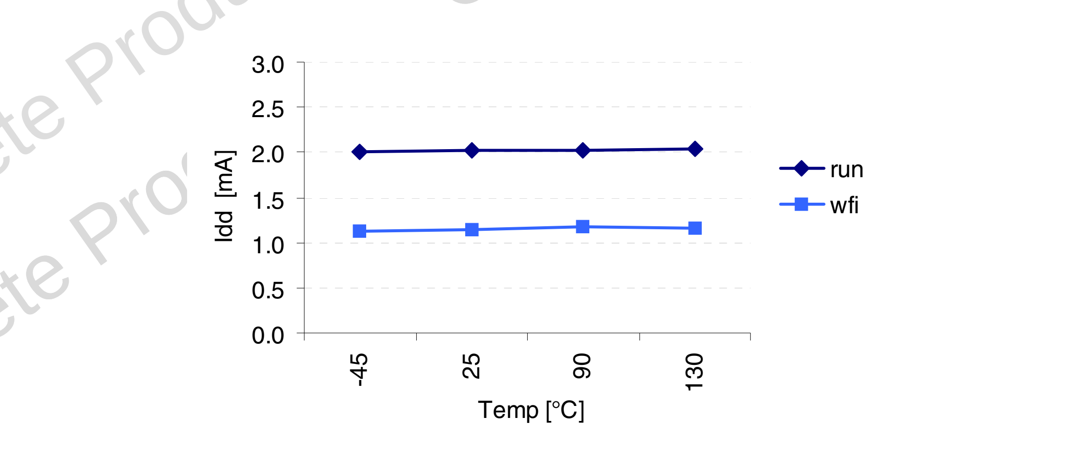
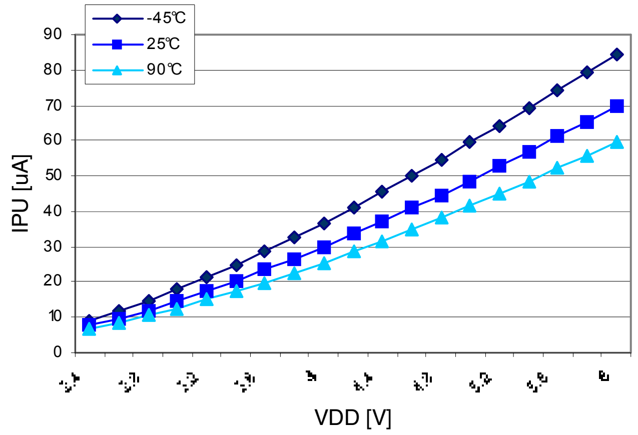


Figure 49. Typical I_{PU} vs. V_{DD} with V_{IN}=V_{SS}I



12.8.2 Output driving current characteristics

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

Table 64. Output driving current characteristics

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL} [1]	Output low level voltage for PA3/RESET standard I/O pin (see *Figure 52*)	V_{DD}=5 V	I_{IO}= +5 mA, T_A≤ 125 °C		1200	mV
			I_{IO} = +2 mA, T_A≤ 125 °C		400	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see *Figure 55*)		I_{IO}=+20 mA, T_A≤ 125 °C		1300	
			I_{IO} = +8 mA, T_A≤ 125 °C		750	
V_{OH} [2]	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see *Figure 58*)		I_{IO}= -5 mA, T_A≤ 125 °C	V_{DD}-1500		
			I_{IO} = -2 mA, T_A≤ 125 °C	V_{DD}-800		
V_{OL} [1][3]	Output low level voltage for PA3/RESET standard I/O pin (see *Figure 51*)	V_{DD}=3 V	I_{IO} = +2 mA, T_A≤ 125 °C		500	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see *Figure 54*)		I_{IO} = +2 mA, T_A≤ 125 °C		180	
			I_{IO} = +8 mA, T_A≤ 125 °C		600	
V_{OH} [2][3]	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see *Figure 57*)		I_{IO} = -2 mA, T_A≤ 125 °C	V_{DD}-800		
V_{OL} [1][3]	Output low level voltage for PA3/RESET standard I/O pin (see *Figure 53*)	V_{DD}=2.4 V	I_{IO} = +2 mA, T_A≤ 125 °C		700	
	Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see *Figure 53*)		I_{IO} = +2 mA, T_A≤ 125 °C		200	
			I_{IO}=+8 mA, T_A≤ 125 °C		800	
V_{OH} [2][3]	Output high level voltage for an I/O pin when 4 pins are sourced at same time (see *Figure 56*)		I_{IO}=-2 mA, T_A≤ 125 °C	V_{DD}-900		

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Table 52* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
2. The I_{IO} current sourced must always respect the absolute maximum rating specified in *Table 52* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}. True open drain I/O pins do not have V_{OH}.
3. Not tested in production, based on characterization results.

Table 81. ST7 application notes (continued)

Identification	Description
AN1044	Multiple interrupt sources management for ST7 MCUs
AN1045	ST7 S/W implementation of I²C bus master
AN1046	UART emulation software
AN1047	Managing reception errors with the ST7 SCI peripherals
AN1048	ST7 software LCD driver
AN1078	PWM duty cycle switch implementing true 0% & 100% duty cycle
AN1082	Description of the ST72141 motor control peripherals registers
AN1083	ST72141 BLDC motor control software and flowchart example
AN1105	ST7 pCAN peripheral driver
AN1129	PWM management for BLDC motor drives using the ST72141
AN1130	An introduction to sensorless brushless DC motor drive applications with the ST72141
AN1148	Using the ST7263 for designing a USB mouse
AN1149	Handling Suspend mode on a USB mouse
AN1180	Using the ST7263 Kit to implement a USB game pad
AN1276	BLDC motor start routine for the ST72141 microcontroller
AN1321	Using the ST72141 motor control MCU in sensor mode
AN1325	Using the ST7 USB low-speed firmware V4.x
AN1445	Emulated 16-bit slave SPI
AN1475	Developing an ST7265X mass storage application
AN1504	Starting a PWM signal directly at high level using the ST7 16-bit timer
AN1602	16-bit timing operations using ST7262 or ST7263B ST7 USB MCUs
AN1633	Device firmware upgrade (DFU) implementation in ST7 non-USB applications
AN1712	Generating a high resolution sinewave using ST7 PWMART
AN1713	SMBus slave driver for ST7 I2C peripherals
AN1753	Software UART using 12-bit ART
AN1947	ST7MC PMAC sine wave motor control software library
General purpose	
AN1476	Low cost power supply for home appliances
AN1526	ST7FLITE0 quick reference note
AN1709	EMC design for ST Microcontrollers
AN1752	ST72324 quick reference note
Product evaluation	
AN 910	Performance benchmarking
AN 990	ST7 benefits vs industry standard