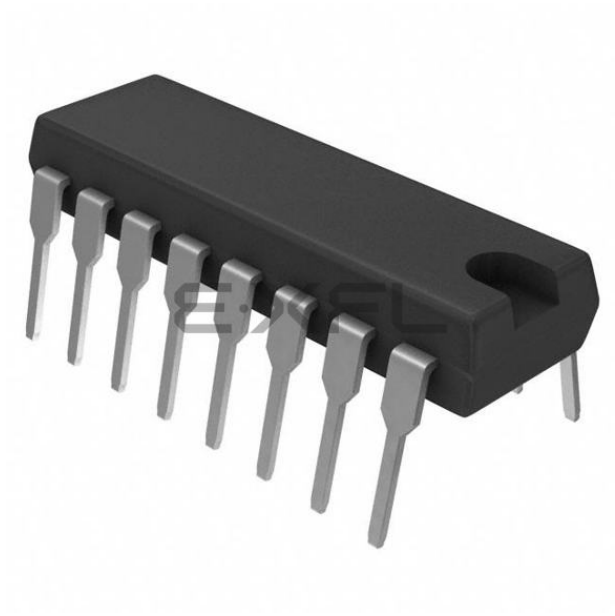


**Welcome to E-XFL.COM**

## 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 | 11 |
| Program Memory Size | 2KB (2K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 128 x 8 |
| RAM Size | 256 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 5.5V |
| Data Converters | A/D 5x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 16-DIP (0.300", 7.62mm) |
| Supplier Device Package | - |
| Purchase URL | https://www.e-xfl.com/product-detail/stmicroelectronics/st7flit19by0b6 |

# 1 INTRODUCTION

The ST7LITE1xB 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.

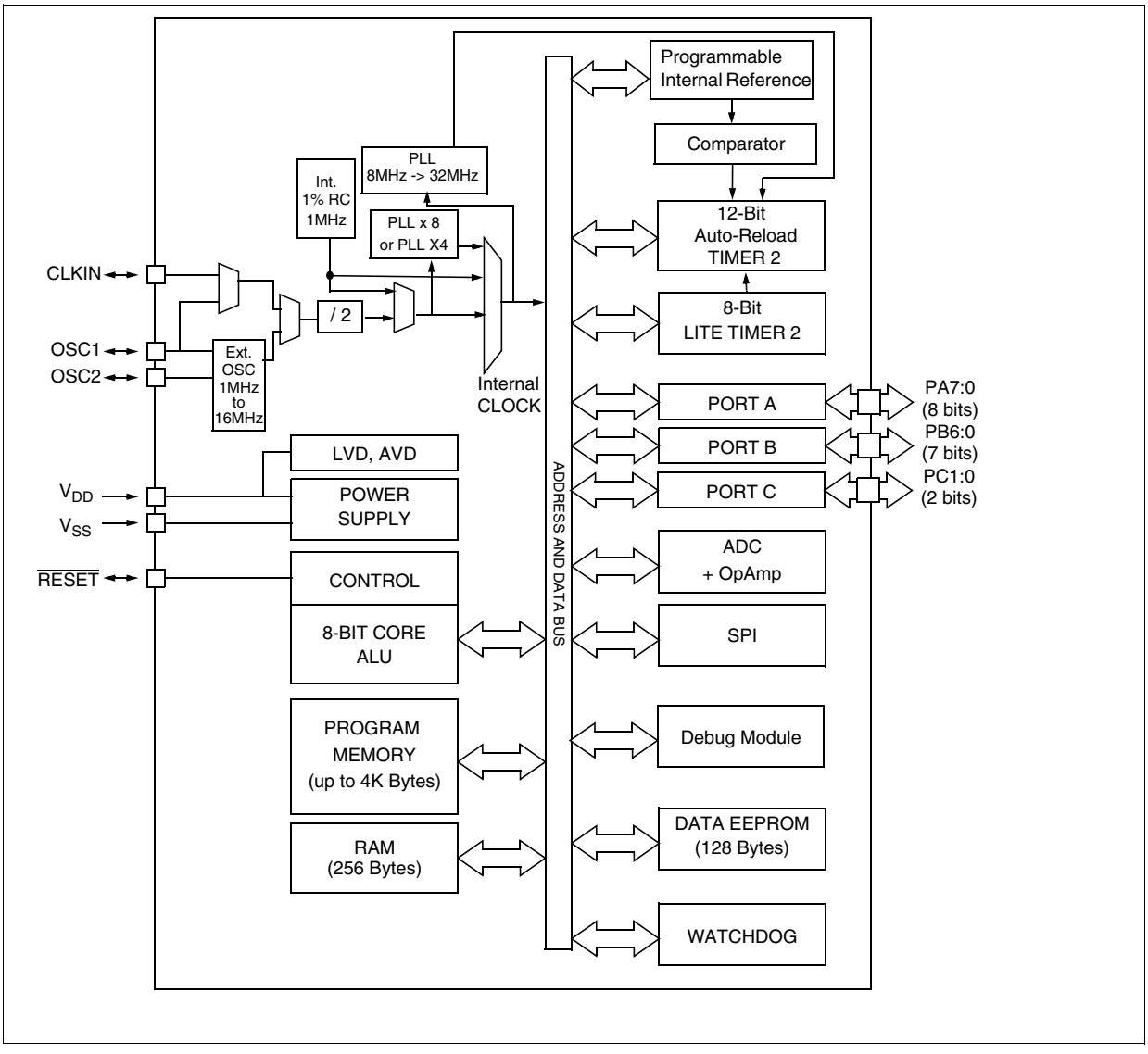
The ST7LITE1xB features FLASH memory with byte-by-byte In-Circuit Programming (ICP) and In-Application Programming (IAP) capability.

Under software control, the ST7LITE1xB 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 13 on page 110. The ST7LITE1xB features 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.

**Figure 1. General Block Diagram**



# 2 PIN DESCRIPTION

**Figure 2. 20-Pin SO and DIP Package Pinout**

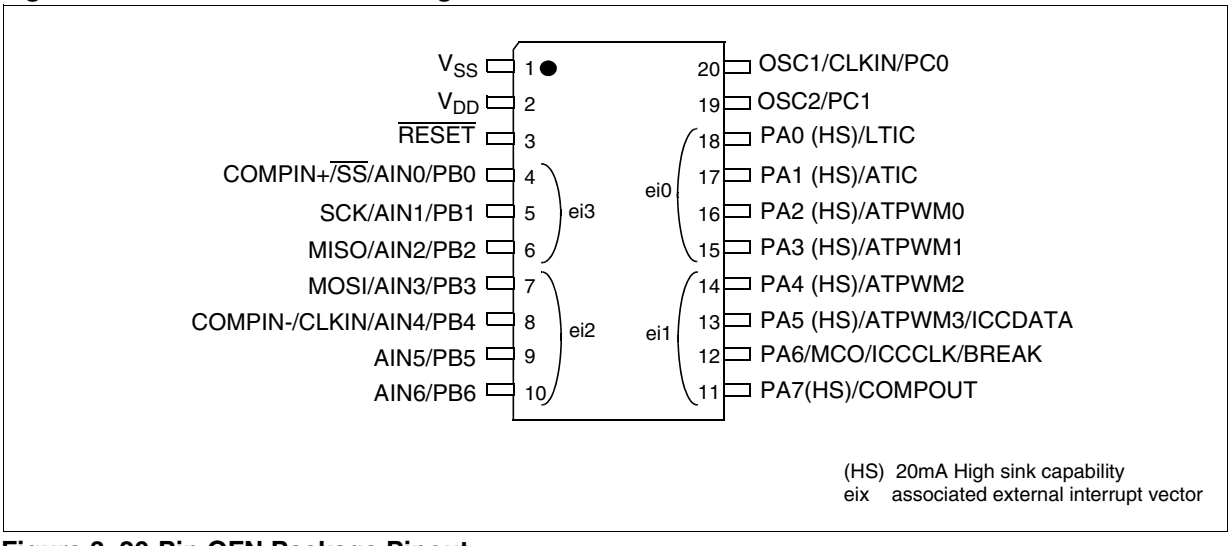
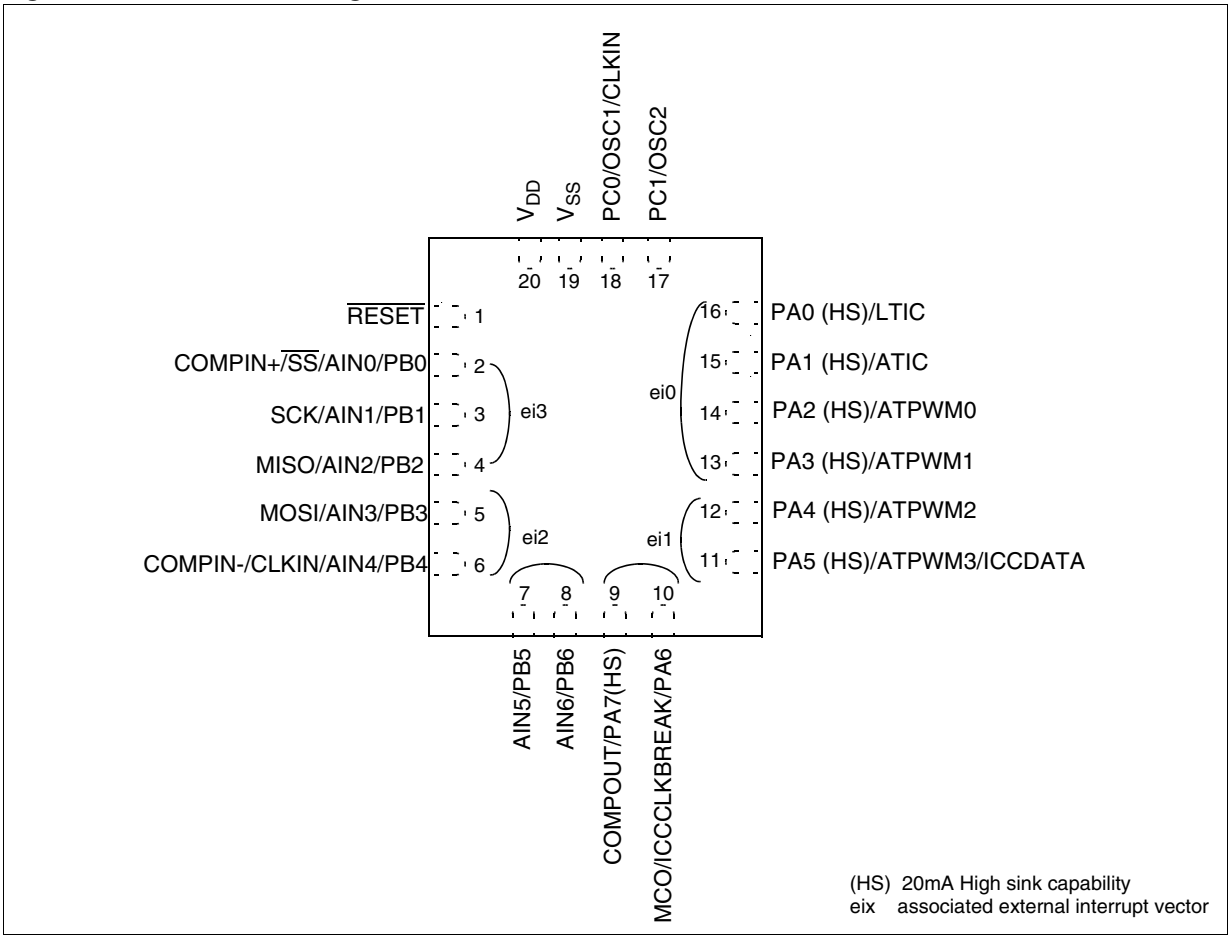



**Figure 3. 20-Pin QFN Package Pinout**



**RESET SEQUENCE MANAGER** (Cont'd)

The $\overline{\text{RESET}}$ pin is an asynchronous signal which plays a major role in EMS performance. In a noisy environment, it is recommended to follow the guidelines mentioned in the electrical characteristics section.

### 7.5.3 External Power-On RESET

If the LVD is disabled by option byte, to start up the microcontroller correctly, the user must ensure by means of an external reset circuit that the reset signal is held low until $V_{DD}$ is over the minimum level specified for the selected $f_{OSC}$ frequency.

A proper reset signal for a slow rising $V_{DD}$ supply can generally be provided by an external RC network connected to the $\overline{\text{RESET}}$ pin.

### 7.5.4 Internal Low Voltage Detector (LVD) RESET

Two different RESET sequences caused by the internal LVD circuitry can be distinguished:

- Power-On RESET
- Voltage Drop RESET

The device $\overline{\text{RESET}}$ pin acts as an output that is pulled low when $V_{DD}<V_{IT+}$ (rising edge) or $V_{DD}<V_{IT-}$ (falling edge) as shown in Figure 17.

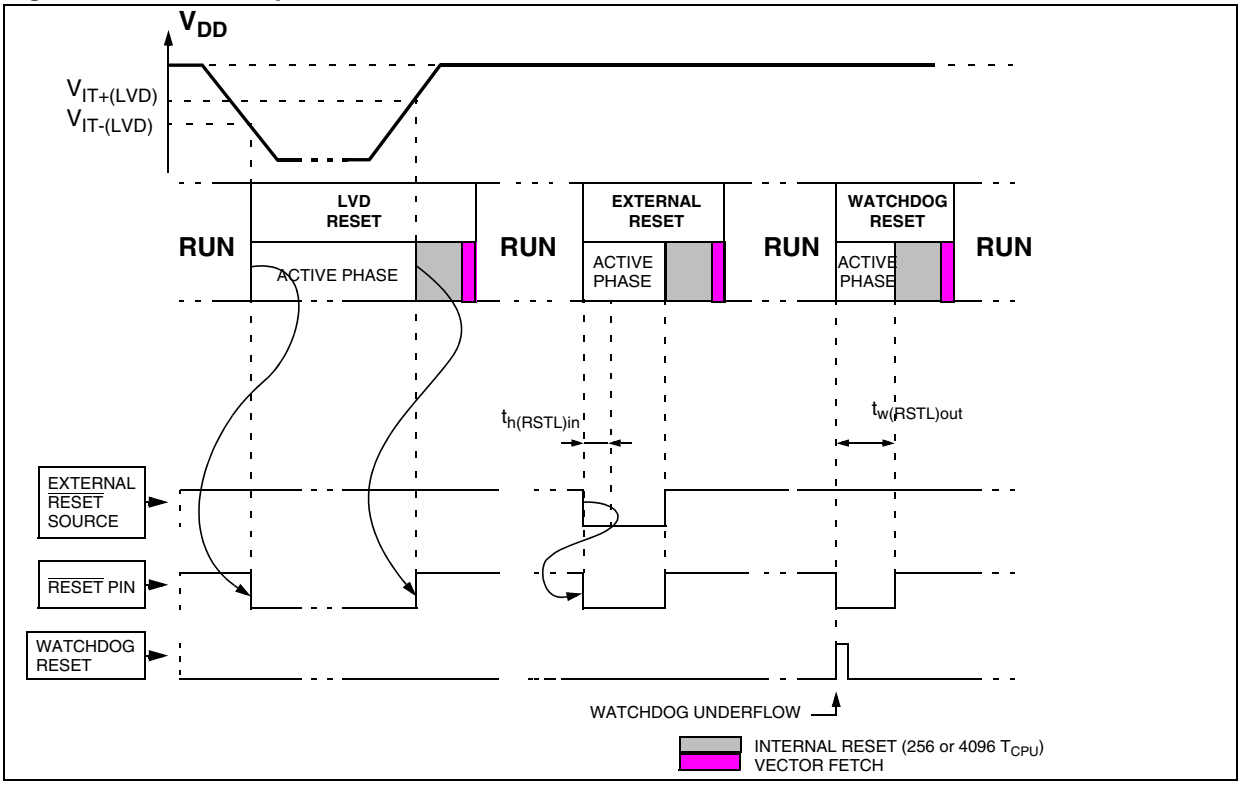
The LVD filters spikes on $V_{DD}$ larger than $t_{g(VDD)}$ to avoid parasitic resets.

### 7.5.5 Internal Watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow is shown in Figure 17.

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 17. RESET Sequences**



# 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 non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 1.

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

**Note:** After reset, all interrupts are disabled.

When an interrupt has to be serviced:

– Normal processing is suspended at the end of the current instruction execution.

– The PC, X, A and CC registers are saved onto the stack.

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

– The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to the Interrupt Mapping table for vector addresses).

The interrupt service routine should finish with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

**Note:** As a consequence of the IRET instruction, the I bit is cleared and the main program resumes.

**Priority Management**

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

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

**Interrupts and Low Power Mode**

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

## 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 is serviced according to the flowchart in Figure 1.

## 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.

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.

## 8.3 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.

# 9 POWER SAVING MODES

## 9.1 INTRODUCTION

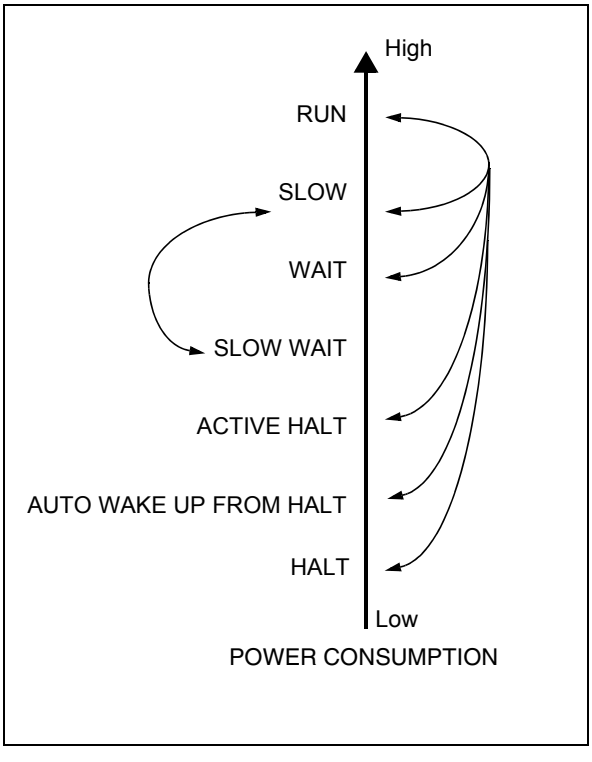
To give a large measure of flexibility to the application in terms of power consumption, five main power saving modes are implemented in the ST7 (see Figure 22):

- Slow
- Wait (and Slow-Wait)
- Active Halt
- Auto Wake up 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 divided or multiplied by 2 ($f_{OSC2}$).

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 22. Power Saving Mode Transitions**




## 9.2 SLOW MODE

This mode has two targets:

– To reduce power consumption by decreasing the internal clock in the device,

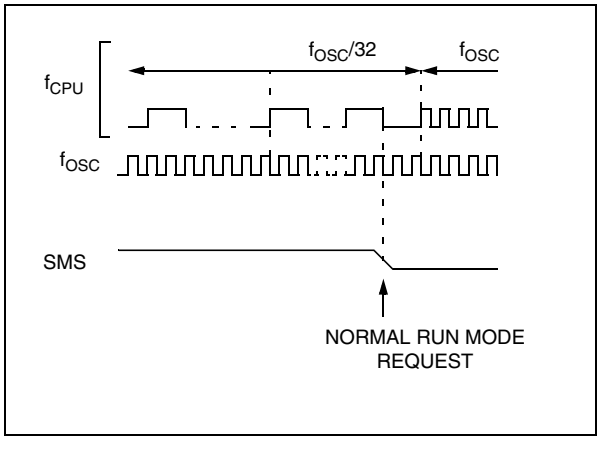
– To adapt the internal clock frequency ($f_{CPU}$) to the available supply voltage.

SLOW mode is controlled by the SMS bit in the MCCSR register which enables or disables Slow mode.

In this mode, the oscillator frequency is divided by 32. The CPU and peripherals are clocked at this lower frequency.

**Note**: SLOW-WAIT mode is activated when entering WAIT mode while the device is already in SLOW mode.

**Figure 23. SLOW Mode Clock Transition**



**POWER SAVING MODES** (Cont'd)

### 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, re-initialize the corresponding I/O as "Input Pull-up with Interrupt" before executing the HALT instruction. The main reason for this is that the I/O may be wrongly configured due to external interference or by an unforeseen logical condition.

– For the same reason, re-initialize the level sensitiveness of each external interrupt as a precautionary measure.

– The opcode for the HALT instruction is 0x8E. To avoid an unexpected HALT instruction due to a program counter failure, it is advised to clear all occurrences of the data value 0x8E from memory. For example, avoid defining a constant in program memory with the value 0x8E.

– As the HALT instruction clears the interrupt mask in the CC register to allow interrupts, the user may choose to clear all pending interrupt bits before executing the HALT instruction. This avoids entering other peripheral interrupt routines after executing the external interrupt routine corresponding to the wake-up event (reset or external interrupt).

## 9.5 ACTIVE-HALT MODE

ACTIVE-HALT mode is the lowest power consumption mode of the MCU with a real time clock available. It is entered by executing the 'HALT' instruction. The decision to enter either in ACTIVE-HALT or HALT mode is given by the LTCSR/ATCSR register status as shown in the following table:

| LTCSR1 TB1IE bit | ATCSR OVFIE bit | ATCSR CK1 bit | ATCSR CK0 bit | Meaning |
|---|---|---|---|---|
| 0 | x | x | 0 | ACTIVE-HALT mode disabled |
| 0 | 0 | x | x | |
| 1 | x | x | x | ACTIVE-HALT mode enabled |
| x | 1 | 0 | 1 | |

The MCU can exit ACTIVE-HALT mode on reception of a specific interrupt (see Table 5, "Interrupt Mapping," on page 37) or a RESET.

– When exiting ACTIVE-HALT mode by means of a RESET, a 256 or 4096 CPU cycle delay occurs. After the start up delay, the CPU resumes operation by fetching the reset vector which woke it up (see Figure 28).

– When exiting ACTIVE-HALT mode by means of an interrupt, the CPU immediately resumes operation by servicing the interrupt vector which woke it up (see Figure 28).

When entering ACTIVE-HALT mode, the I bit in the CC register is cleared to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately (see Note 3).

In ACTIVE-HALT mode, only the main oscillator and the selected timer counter (LT/AT) are running to keep a wake-up time base. All other peripherals are not clocked except those which get their clock supply from another clock generator (such as external or auxiliary oscillator).

**Note:** As soon as ACTIVE-HALT is enabled, executing a HALT instruction while the Watchdog is active does not generate a RESET.
This means that the device cannot spend more than a defined delay in this power saving mode.

**I/O PORTS** (Cont'd)

## 10.7 DEVICE-SPECIFIC I/O PORT CONFIGURATION

The I/O port register configurations are summarised as follows.

### Standard Ports

#### PA7:0, PB6:0

| MODE | DDR | OR |
|---|---|---|
| floating input | 0 | 0 |
| pull-up input | 0 | 1 |
| open drain output | 1 | 0 |
| push-pull output | 1 | 1 |

#### PC1:0 (multiplexed with OSC1,OSC2)

| MODE | DDR |
|---|---|
| floating input | 0 |
| push-pull output | 1 |

The selection between OSC1 or PC0 and OSC2 or PC1 is done by option byte. Refer to section 15.1 on page 149. Interrupt capability is not available on PC1:0.

**Note:** PCOR not implemented but p-transistor always active in output mode (refer to Figure 32 on page 50)

### Interrupt Ports

#### Ports where the external interrupt capability is selected using the EISR register

| MODE | DDR | OR |
|---|---|---|
| floating input | 0 | 0 |
| pull-up interrupt input | 0 | 1 |
| open drain output | 1 | 0 |
| push-pull output | 1 | 1 |

**Table 10. Port Configuration (Standard ports)**

| Port | Pin name | Input | | Output | |
|---|---|---|---|---|---|
| | | OR = 0 | OR = 1 | OR = 0 | OR = 1 |
| Port A | PA7:0 | floating | pull-up | open drain | push-pull |
| Port B | PB6:0 | floating | pull-up | open drain | push-pull |

**Note:** On ports where the external interrupt capability is selected using the EISR register, the configuration will be as follows:

| Port | Pin name | Input | | Output | |
|---|---|---|---|---|---|
| | | OR = 0 | OR = 1 | OR = 0 | OR = 1 |
| Port A | PA7:0 | floating | pull-up interrupt | open drain | push-pull |
| Port B | PB6:0 | floating | pull-up interrupt | open drain | push-pull |

**Table 11. I/O Port Register Map and Reset Values**

| Address (Hex.) | Register Label | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0000h | **PADR** | MSB | | | | | | | LSB |
| | Reset Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0001h | **PADDR** | MSB | | | | | | | LSB |
| | Reset Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**DUAL 12-BIT AUTORELOAD TIMER 4 (Cont'd)**

### 11.2.3.6 One Pulse Mode

One Pulse Mode can be used to control PWM2/3 signal with an external LTIC pin. This mode is available only in dual timer mode i.e. only for CNTR2, when the OP_EN bit in PWM3CSR register is set.

One Pulse Mode is activated by the external LTIC input. The active edge of the LTIC pin is selected by the OPEDGE bit in the PWM3CSR register.

After getting the active edge of the LTIC pin, CNTR2 is reset (000h) and PWM3 is set to high. CNTR2 starts counting from 000h, when it reaches the active DCR3 value then the PWM3 output goes low. Till this time, any further transitions on the LTIC signal will have no effect. If there are LTIC transitions after CNTR2 reaches the DCR3 value, CNTR2 is reset again and the PWM3 output goes high.

If there is no LTIC active edge then CNTR2 will count till it reaches the ATR2 value, and then it will be reset again and the PWM3 output is set to high. The counter again starts counting from 000h, when it reaches the active DCR3 value the PWM3 output goes low, the counter counts till it reaches the ATR2 value, it resets and the PWM3 output is set to high and it goes on the same way.

The same operation applies for the PWM2 output, but in this case the comparison is done on the DCR2 value.

The OP_EN and OPEDGE bits take effect on the fly and are not synchronized with the CNTR2 overflow.

The OP2/3 bits can be used to inverse the polarity of the PWM2/3 outputs in one-pulse mode. The update of these bits (OP2/3) is synchronized with the CNTR2 overflow, they will be updated if the TRAN2 bit is set.

**Notes**:

1. If CNTR2 is running at 32 MHz, the time taken from activation of LTIC input and CNTR2 reset is between 2 and 3 $t_{CNTR2}$ cycles, i.e. 66 ns to 99 ns (with 8 MHz $f_{cpu}$).

2. The Lite Timer input capture interrupt must be disabled while 12-bit ARTimer is in One Pulse Mode. This is to avoid spurious interrupts.

3. The priority of various events affecting PWM3 is as follows:

– Break (Highest priority)

– One-pulse mode with active LTIC edge

– Forced overflow (by FORCE2 bit)

– One-pulse mode without active LTIC edge

– Normal PWM operation. (Lowest priority)

4. It is possible to synchronize the update of DCR2/3 registers and OP2/3 bits with the CNTR2 reset. This is managed by the overflow interrupt which is generated if CNTR2 is reset either due to an ATR match or an active pulse on the LTIC pin.

5. Updating the DCR2/3 registers and OP2/3 bits in one-pulse mode is done dynamically by software using force update (FORCE2 bit in the ATCSR2 register).

6. DCR3 update in this mode is not synchronized with any event. Consequently the next PWM3 cycle just after the change may be longer than expected (refer to Figure 15).

7. In One Pulse Mode the ATR2 value must be greater than the DCR2/3 value for the PWM2/3 outputs. (contrary to normal PWM mode)

8. If there is an active edge on the LTIC pin after the CNTR2 has reset due to an ATR2 match, then the timer gets reset again. The duty cycle may be modified depending on whether the new DCR value is less than or more than the previous value.

9. The TRAN2 bit must be set simultaneously with the FORCE2 bit in the same instruction after a write to the DCR register.

10. The ATR2 value should be changed after an overflow in one pulse mode to avoid an irregular PWM cycle.

11. When exiting from one pulse mode, the OP_EN bit in the PWM3CSR register must be reset first and then the ENCNTR2 bit (if CNTR2 is to be stopped).

**How to Enter One Pulse Mode:**

1. Load the ATR2H/ATR2L registers with required value.

2. Load the DCR3H/DCR3L registers for PWM3 output. The ATR2 value must be greater than DCR3.

3. Set the OP3 bit in the PWM3CSR register if polarity change is required.

4. Start the CNTR2 counter by setting the ENCNTR2 bit in the ATCSR2 register.

5. Set TRAN2 bit in ATCSR2 to enable transfer.

6. Wait for an overflow event by polling the OVF2 flag in the ATCSR2 register.

7. Select the counter clock using the CK[1:0] bits in the ATCSR register.

**DUAL 12-BIT AUTORELOAD TIMER 4** (Cont'd)

### 11.2.6 Register Description

**TIMER CONTROL STATUS REGISTER (ATCSR)**
Read / Write
Reset Value: 0x00 0000 (x0h)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| 0 | ICF | ICIE | CK1 | CK0 | OVF1 | OVFIE1 | CMPIE |

Bit 7 = Reserved.

Bit 6 = **ICF** *Input Capture Flag.*
This bit is set by hardware and cleared by software by reading the ATICR register (a read access to ATICRH or ATICRL will clear this flag). Writing to this bit does not change the bit value.
0: No input capture
1: An input capture has occurred

Bit 5 = **ICIE** *IC Interrupt Enable.*
This bit is set and cleared by software.
0: Input capture interrupt disabled
1: Input capture interrupt enabled

Bits 4:3 = **CK[1:0]** *Counter Clock Selection.*
These bits are set and cleared by software and cleared by hardware after a reset. They select the clock frequency of the counter.

| Counter Clock Selection | CK1 | CK0 |
|---|---|---|
| OFF | 0 | 0 |
| 32 MHz | 1 | 1 |
| $f_{LTIMER}$ (1 ms timebase @ 8 MHz) | 0 | 1 |
| $f_{CPU}$ | 1 | 0 |

Bit 2 = **OVF1** *Overflow Flag.*
This bit is set by hardware and cleared by software by reading the ATCSR register. It indicates the transition of the counter1 CNTR1 from FFFh to ATR1 value.
0: No counter overflow occurred
1: Counter overflow occurred

Bit 1 = **OVFIE1** *Overflow Interrupt Enable.*
This bit is read/write by software and cleared by hardware after a reset.
0: Overflow interrupt disabled.
1: Overflow interrupt enabled.

Bit 0 = **CMPIE** *Compare Interrupt Enable*.
This bit is read/write by software and cleared by hardware after a reset. It can be used to mask the interrupt generated when any of the CMPFx bit is set.
0: Output compare interrupt disabled.
1: Output Compare interrupt enabled.

**COUNTER REGISTER 1 HIGH (CNTR1H)**
Read only
Reset Value: 0000 0000 (00h)

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | CNTR1_11 | CNTR1_10 | CNTR1_9 | CNTR1_8 |

**COUNTER REGISTER 1 LOW (CNTR1L)**
Read only
Reset Value: 0000 0000 (00h)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| CNTR1_7 | CNTR1_6 | CNTR1_5 | CNTR1_4 | CNTR1_3 | CNTR1_2 | CNTR1_1 | CNTR1_0 |

Bits 15:12 = Reserved.

Bits 11:0 = **CNTR1[11:0]** *Counter Value*.
This 12-bit register is read by software and cleared by hardware after a reset. The counter CNTR1 increments continuously as soon as a counter clock is selected. To obtain the 12-bit value, software should read the counter value in two consecutive read operations. As there is no latch, it is recommended to read LSB first. In this case, CNTR1H can be incremented between the two read operations and to have an accurate result when $f_{timer}=f_{CPU}$, special care must be taken when CNTR1L values close to FFh are read.
When a counter overflow occurs, the counter restarts from the value specified in the ATR1 register.

**DUAL 12-BIT AUTORELOAD TIMER 4** (Cont'd)

Bits 11:0 = **ICR[11:0]** *Input Capture Data*.
This is a 12-bit register which is readable by software and cleared by hardware after a reset. The ATICR register contains captured the value of the 12-bit CNTR1 register when a rising or falling edge occurs on the ATIC or LTIC pin (depending on

ICS). Capture will only be performed when the ICF flag is cleared.

**BREAK ENABLE REGISTER (BREAKEN)**

Read/Write
Reset Value: 0000 0011 (03h)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | BREN2 | BREN1 |

Bits 7:2 = Reserved, must be kept cleared.

Bit 1 = **BREN2** *Break Enable for Counter 2*
This bit is read/write by software. It enables the break functionality for Counter2 if BA bit is set in BREAKCR. It controls PWM2/3 if ENCNTR2 bit is set.
0: No Break applied for CNTR2
1: Break applied for CNTR2

Bit 0 = **BREN1** *Break Enable for Counter 1*
This bit is read/write by software. It enables the break functionality for Counter1. If BA bit is set, it controls PWM0/1 by default, and controls PWM2/3 also if ENCNTR2 bit is reset.
0: No Break applied for CNTR1
1: Break applied for CNTR1

**TIMER CONTROL REGISTER2 (ATCSR2)**
Read/Write
Reset Value: 0000 0011 (03h)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| FORCE 2 | FORCE 1 | ICS | OVFIE2 | OVF2 | ENCNT R2 | TRAN2 | TRAN1 |

Bit 7 = **FORCE2** *Force Counter 2 Overflow*
This bit is read/set by software. When set, it loads FFFh in the CNTR2 register. It is reset by hardware one CPU clock cycle after counter 2 overflow has occurred.

0 : No effect on CNTR2
1 : Loads FFFh in CNTR2

**Note:** This bit must not be reset by software

Bit 6 = **FORCE1** *Force Counter 1 Overflow*
This bit is read/set by software. When set, it loads FFFh in CNTR1 register. It is reset by hardware one CPU clock cycle after counter 1 overflow has occurred.

0 : No effect on CNTR1
1 : Loads FFFh in CNTR1

**Note:** This bit must not be reset by software

Bit 5 = **ICS** *Input Capture Shorted*
This bit is read/write by software. It allows the AT-timer CNTR1 to use the LTIC pin for long input capture.
0 : ATIC for CNTR1 input capture
1 : LTIC for CNTR1 input capture

Bit 4 = **OVFIE2** *Overflow interrupt 2 enable*
This bit is read/write by software and controls the overflow interrupt of counter2.
0: Overflow interrupt disabled.
1: Overflow interrupt enabled.

Bit 3 = **OVF2** *Overflow Flag.*
This bit is set by hardware and cleared by software by reading the ATCSR2 register. It indicates the transition of the counter2 from FFFh to ATR2 value.
0: No counter overflow occurred
1: Counter overflow occurred

Bit 2 = **ENCNTR2** *Enable counter2* for PWM2/3

This bit is read/write by software and switches the PWM2/3 operation to the CNTR2 counter. If this bit is set, PWM2/3 will be generated using CNTR2.
0: PWM2/3 is generated using CNTR1.
1: PWM2/3 is generated using CNTR2.

**Note:** Counter 2 gets frozen when the ENCNTR2 bit is reset. When ENCNTR2 is set again, the counter will restart from the last value.

**LITE TIMER** (Cont'd)

### 11.3.3 Functional Description

#### 11.3.3.1 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.

#### 11.3.3.2 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 LTICR register contains the counter 1 value. An interrupt is generated if the ICIE bit is set. The ICF bit is cleared by reading the LTICR register.

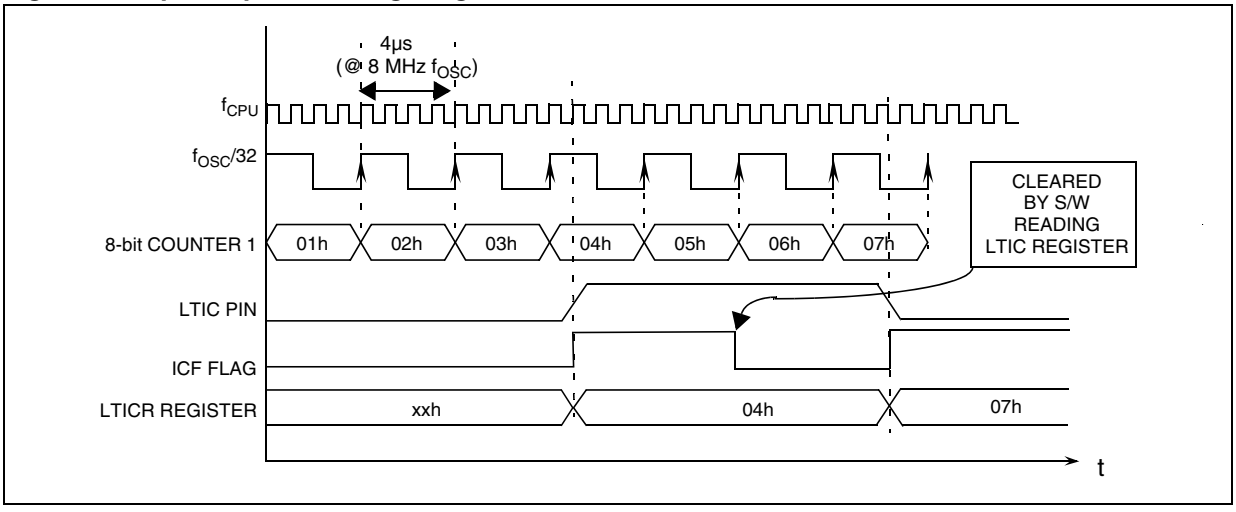
The LTICR is a read-only register and always contains the data from the last input capture. Input capture is inhibited if the ICF bit is set.

#### 11.3.3.3 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 any time 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 52. Input Capture Timing Diagram.**



**ST7 ADDRESSING MODES** (cont'd)

### 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.

| 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 |

### 12.1.2 Immediate

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

| Immediate Instruction | Function |
|---|---|
| LD | Load |
| CP | Compare |
| BCP | Bit Compare |
| AND, OR, XOR | Logical Operations |
| ADC, ADD, SUB, SBC | Arithmetic Operations |

### 12.1.3 Direct

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

The direct addressing mode consists of two submodes:

**Direct (Short)**

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

**Direct (Long)**

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

### 12.1.4 Indexed (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 (No Offset)**

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

**Indexed (Short)**

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

**Indexed (Long)**

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

### 12.1.5 Indirect (Short, Long)

The required data byte to do the operation is found by its memory address, located in memory (pointer).

The pointer address follows the opcode. The indirect addressing mode consists of two submodes:

**Indirect (Short)**

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - FF addressing space, and requires 1 byte after the opcode.

**Indirect (Long)**

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

# 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, $V_{DD}$=5V (for the 4.5V≤$V_{DD}$≤5.5V voltage range) and $V_{DD}$=3.3V (for the 3V≤$V_{DD}$≤3.6V voltage range). They are given only as design guidelines and are not tested.

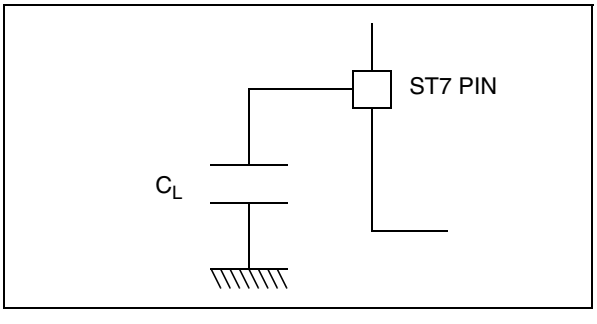
### 13.1.3 Typical curves

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 63.

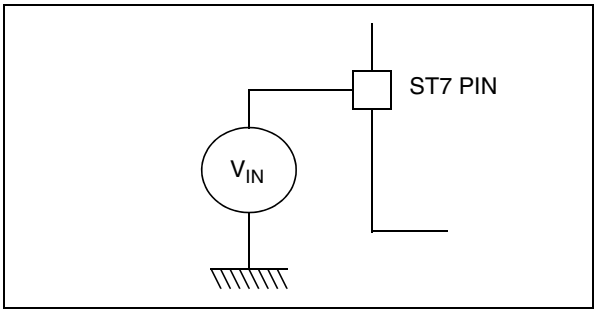
**Figure 63. Pin loading conditions**




### 13.1.5 Pin input voltage

The input voltage measurement on a pin of the device is described in Figure 64.

**Figure 64. Pin input voltage**



**OPERATING CONDITIONS** (Cont'd)

The RC oscillator and PLL characteristics are temperature-dependent and are grouped in four tables.

### 13.3.5.1 Devices with '"6" or "3"order code suffix (tested for $T_A$ = -40 to +125°C) @ $V_{DD}$ = 5V

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $f_{RC}$ | Internal RC oscillator frequency [1] | RCCR = FF (reset value), $T_A$=25°C,$V_{DD}$=5V | | 700 | | kHz |
| | | RCCR = RCCR0[2],$T_A$=25°C,$V_{DD}$=5V | 992 | 1000 | 1008 | |
| $ACC_{RC}$ | Accuracy of Internal RC oscillator with RCCR=RCCR0[2] | $T_A$=25°C,$V_{DD}$=5V | -0.8 | | +0.8 | % |
| | | $T_A$=25°C, $V_{DD}$=4.5 to 5.5V[3] | -1 | | +1 | % |
| | | $T_A$=25°C to +85°C,$V_{DD}$=5V | -3 | | +3 | % |
| | | $T_A$=25°C to +85°C,$V_{DD}$=4.5 to 5.5V[3] | -3.5 | | +3.5 | % |
| | | $T_A$=85°C to +125°C,$V_{DD}$=5V | -3.5 | | +5 | % |
| | | $T_A$=85°C to +125°C,$V_{DD}$=4.5 to 5.5V[3] | -3.5 | | +6 | % |
| | | $T_A$=-40 to +25°C, $V_{DD}$=5V[3] | -3 | | +7 | % |
| $I_{DD(RC)}$ | RC oscillator current consumption | $T_A$=25°C,$V_{DD}$=5V | | 600[3] | | µA |
| $t_{su(RC)}$ | RC oscillator setup time | $T_A$=25°C,$V_{DD}$=5V | | | 10[2] | µs |
| $f_{PLL}$ | x8 PLL input clock | | | 1[3] | | MHz |
| $t_{LOCK}$ | PLL Lock time[5] | | | 2 | | ms |
| $t_{STAB}$ | PLL Stabilization time[5] | | | 4 | | ms |
| $ACC_{PLL}$ | x8 PLL Accuracy | $f_{RC}$ = 1MHz@$T_A$=25°C,$V_{DD}$=4.5 to 5.5V | | 0.1[4] | | % |
| | | $f_{RC}$ = 1MHz@$T_A$=-40 to +85°C,$V_{DD}$=5V | | 0.1[4] | | % |
| $t_{w(JIT)}$ | PLL jitter period [6] | $f_{RC}$ = 1MHz | | 120 | | µs |
| $JIT_{PLL}$ | PLL jitter ($\Delta f_{CPU}/f_{CPU}$) | | | 1[7] | | % |
| $I_{DD(PLL)}$ | PLL current consumption | $T_A$=25°C | | 600[3] | | µA |

**Notes:**

**1.** If the RC oscillator clock is selected, to improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100nF, between the $V_{DD}$ and $V_{SS}$ pins as close as possible to the ST7 device.

**2.** See "INTERNAL RC OSCILLATOR ADJUSTMENT" on page 23

**3.** Data based on characterization results, not tested in production

**4.** Averaged over a 4ms period. After the LOCKED bit is set, a period of $t_{STAB}$ is required to reach $ACC_{PLL}$ accuracy.

**5.** After the LOCKED bit is set $ACC_{PLL}$ is max. 10% until $t_{STAB}$ has elapsed. See Figure 13 on page 24.

**6.** This period is the phase servo loop period. During this period, the frequency remains unchanged.

**7.** Guaranteed by design.

**Figure 66. Typical accuracy with RCCR=RCCR0 vs $V_{DD}$= 4.5-5.5V and Temperature**

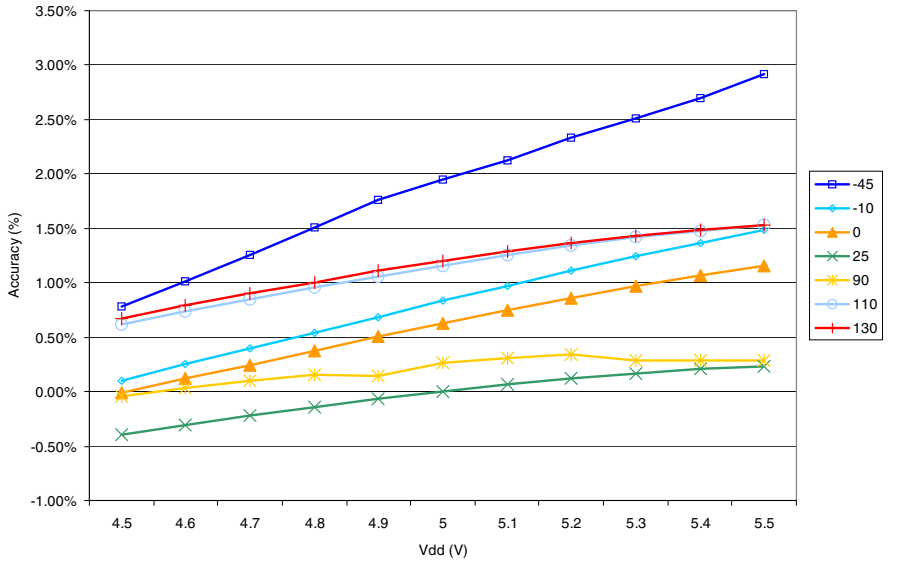
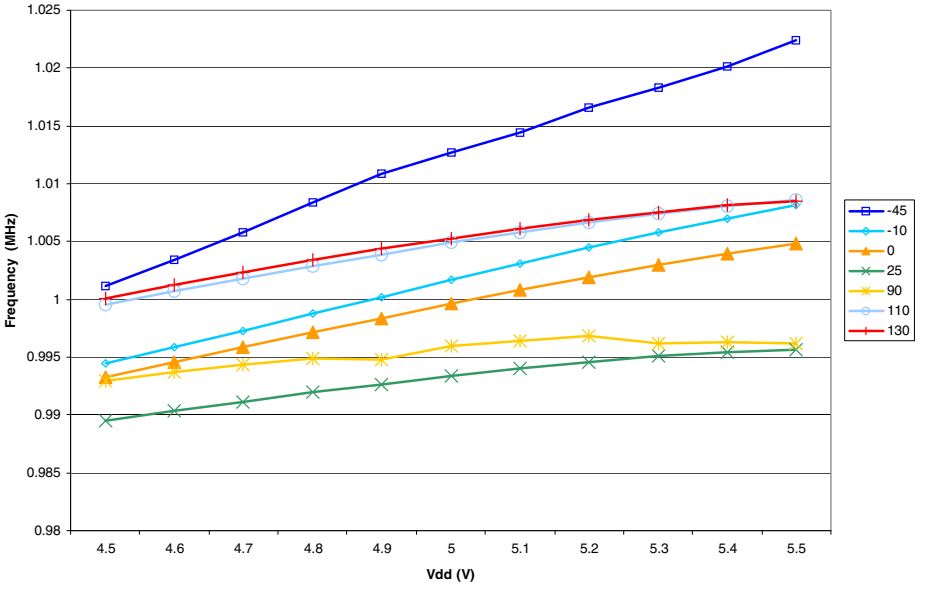



**Figure 67. Typical RCCR0 vs $V_{DD}$ and Temperature**



## I/O PORT PIN CHARACTERISTICS (Cont'd)

### 13.8.2 Output Driving Current

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

| Symbol | Parameter | Conditions | | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{OL}$ 1) | Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see Figure 83) | $V_{DD}$=5V | $I_{IO}$=+5mA $T_A \leq 125°C$ | | 1.0 | V |
| | | | $I_{IO}$=+2mA $T_A \leq 125°C$ | | 0.4 | |
| | Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time (see Figure 89) | | $I_{IO}$=+20mA,$T_A \leq 125°C$ | | 1.3 | |
| | | | $I_{IO}$=+8mA $T_A \leq 125°C$ | | 0.75 | |
| $V_{OH}$ 2) | Output high level voltage for an I/O pin when 4 pins are sourced at same time (see Figure 95) | | $I_{IO}$=-5mA, $T_A \leq 125°C$ | $V_{DD}$-1.5 | | |
| | | | $I_{IO}$=-2mA $T_A \leq 125°C$ | $V_{DD}$-0.8 | | |
| $V_{OL}$ 1)3) | Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see Figure 82) | $V_{DD}$=3.3V | $I_{IO}$=+2mA $T_A \leq 125°C$ | | 0.5 | |
| | Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time | | $I_{IO}$=+8mA $T_A \leq 125°C$ | | 0.5 | |
| $V_{OH}$ 2)3) | Output high level voltage for an I/O pin when 4 pins are sourced at same time (Figure 94) | | $I_{IO}$=-2mA $T_A \leq 125°C$ | $V_{DD}$-0.8 | | |
| $V_{OL}$ 1)3) | Output low level voltage for a standard I/O pin when 8 pins are sunk at same time (see Figure 87) | $V_{DD}$=2.7V | $I_{IO}$=+2mA $T_A \leq 125°C$ | | 0.6 | |
| | Output low level voltage for a high sink I/O pin when 4 pins are sunk at same time | | $I_{IO}$=+8mA $T_A \leq 125°C$ | | 0.6 | |
| $V_{OH}$ 2)3) | Output high level voltage for an I/O pin when 4 pins are sourced at same time (see Figure 101) | | $I_{IO}$=-2mA $T_A \leq 125°C$ | $V_{DD}$-0.9 | | |

**Notes:**

**1.** The $I_{IO}$ current sunk must always respect the absolute maximum rating specified in Section 13.2.2 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 Section 13.2.2 and the sum of $I_{IO}$ (I/O ports and control pins) must not exceed $I_{VDD}$.

**3.** Not tested in production, based on characterization results.

## 13.10 COMMUNICATION INTERFACE CHARACTERISTICS

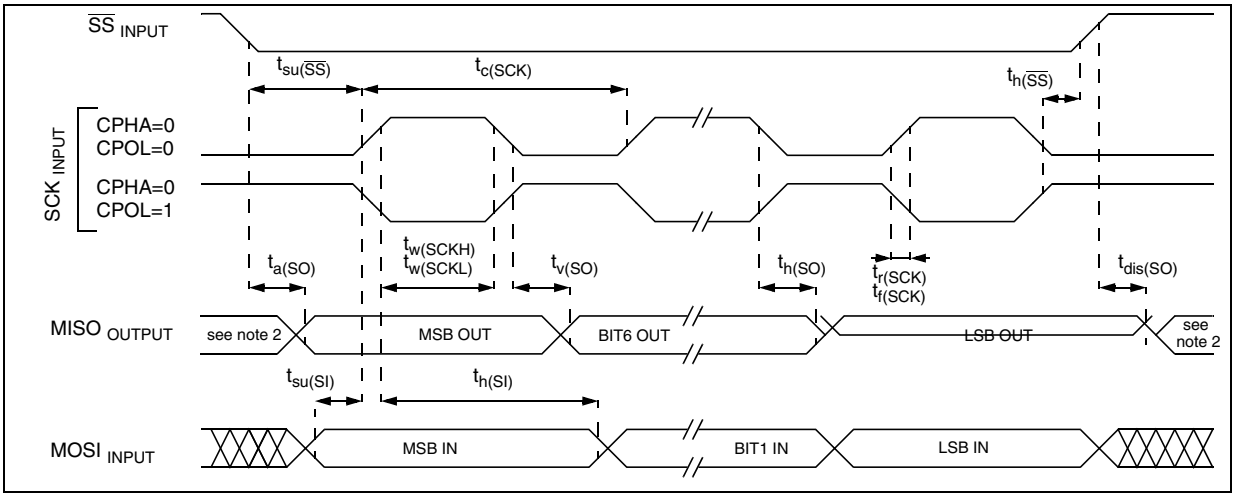
### 13.10.1 SPI - Serial Peripheral Interface

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

Refer to I/O port characteristics for more details on the input/output alternate function characteristics ($\overline{SS}$, SCK, MOSI, MISO).

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $f_{SCK}$<br>$1/t_{c(SCK)}$ | SPI clock frequency | Master<br>$f_{CPU}$=8MHz | $f_{CPU}/128$<br>0.0625 | $f_{CPU}/4$<br>2 | MHz |
| | | Slave<br>$f_{CPU}$=8MHz | 0 | $f_{CPU}/2$<br>4 | |
| $t_{r(SCK)}$<br>$t_{f(SCK)}$ | SPI clock rise and fall time | | see I/O port pin description | | |
| $t_{su(\overline{SS})}$ [1] | $\overline{SS}$ setup time [4] | Slave | $(4 \times T_{CPU}) + 50$ | | ns |
| $t_{h(\overline{SS})}$ [1] | $\overline{SS}$ hold time | Slave | 120 | | |
| $t_{w(SCKH)}$ [1]<br>$t_{w(SCKL)}$ [1] | SCK high and low time | Master<br>Slave | 100<br>90 | | |
| $t_{su(MI)}$ [1]<br>$t_{su(SI)}$ [1] | Data input setup time | Master<br>Slave | 100<br>100 | | |
| $t_{h(MI)}$ [1]<br>$t_{h(SI)}$ [1] | Data input hold time | Master<br>Slave | 100<br>100 | | |
| $t_{a(SO)}$ [1] | Data output access time | Slave | 0 | 120 | |
| $t_{dis(SO)}$ [1] | Data output disable time | Slave | | 240 | |
| $t_{v(SO)}$ [1] | Data output valid time | Slave (after enable edge) | | 120 | |
| $t_{h(SO)}$ [1] | Data output hold time | | 0 | | |
| $t_{v(MO)}$ [1] | Data output valid time | Master (after enable edge) | | 120 | |
| $t_{h(MO)}$ [1] | Data output hold time | | 0 | | |

**Figure 107. SPI Slave Timing Diagram with CPHA=0** [3]




**Notes:**

**1.** Data based on design simulation, not tested in production.

**2.** When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.

**3.** Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.

**4.** Depends on $f_{CPU}$. For example, if $f_{CPU}$=8MHz, then $T_{CPU} = 1/f_{CPU}$ =125ns and $t_{su(\overline{SS})}$=550ns

## 13.12 ANALOG COMPARATOR CHARACTERISTICS

| Symbol | Parameter | Conditions | Min | Typ [1] | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{DDA}$ | Supply range | | 4.5 | | 5.5 | V |
| $V_{IN}$ | Comparator input voltage range | | 0 | | $V_{DDA}$ | V |
| Temp | Temperature range | | -40 | | 125 | °C |
| $V_{offset}$ | Comparator offset error | | | 20 | | mV |
| $I_{DD(CMP)}$ | Analog Comparator Consumption | | | 120 | | µA |
| | Analog Comparator Consumption during power-down | | | 200 | | pA |
| $t_{propag}$ | Comparator propagation delay | | | 40 | | ns |
| $t_{startup}$ | Startup filter duration | | | 500[2] | | ns |
| $t_{stab}$ | Stabilisation time | | | 500 | | ns |

## 13.13 PROGRAMMABLE INTERNAL VOLTAGE REFERENCE CHARACTERISTICS

| Symbol | Parameter | Conditions | Min | Typ [1] | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{DDA}$ | Supply range | | 4 | 5 | 5.5 | V |
| Temp | Temperature range | | -40 | 27 | 125 | °C |
| $I_{DD(VOLTREF)}$ | Internal Voltage Reference Consumption | | | 50 | | µA |
| | Internal Voltage Reference Consumption during power-down | | | 200 | | pA |
| $t_{startup}$ | Startup duration | | | 1[2] | | µs |

## 13.14 CURRENT BIAS CHARACTERISTICS (for Comparator and Internal Voltage Reference)

| Symbol | Parameter | Conditions | Min | Typ [1] | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{DDA}$ | Supply range | | 4.5 | 5 | 5.5 | V |
| Temp | Temperature range | | -40 | 27 | 125 | °C |
| IDD (Bias) | Bias Consumption in run mode | | | 50 | | µA |
| | Bias Consumption during power-down | | | 36 | | pA |
| $t_{startup}$ | Startup time | | | 1[2] | | µs |

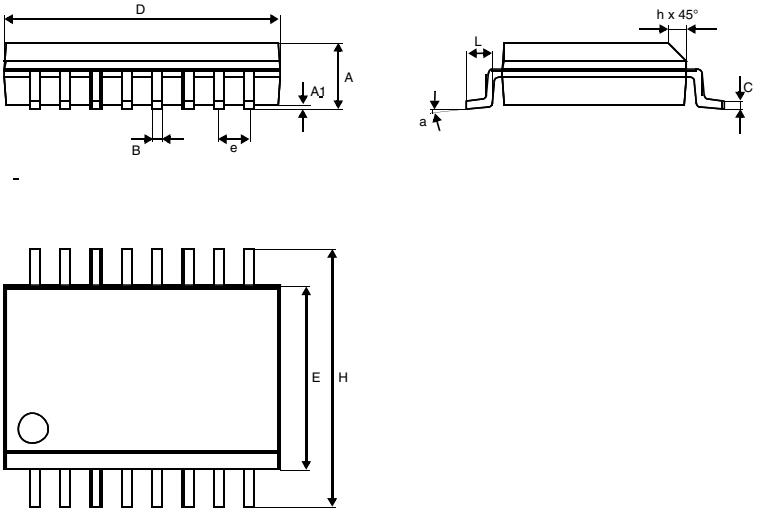
**Notes:**

**1.** Unless otherwise specified, typical data are based on $T_A$=25°C and $V_{DD}$-$V_{SS}$=5V. They are given only as design guidelines and are not tested.

**2.** Since startup time for internal voltage reference and bias is 1 µs, comparator correct output should not be expected before 1 µs during startup.

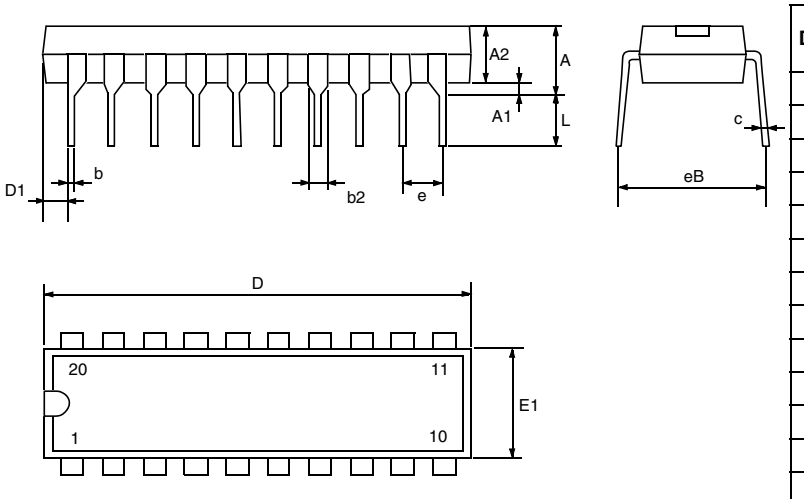
## PACKAGE CHARACTERISTICS (Cont'd)

### Figure 114. 16-Pin Plastic Small Outline Package, 300-mil Width




| Dim. | mm | | | inches | | |
|---|---|---|---|---|---|---|
| | Min | Typ | Max | Min | Typ | Max |
| A | 2.35 | | 2.65 | 0.093 | | 0.104 |
| A1 | 0.10 | | 0.30 | 0.004 | | 0.012 |
| B | 0.33 | | 0.51 | 0.013 | | 0.020 |
| C | 0.23 | | 0.32 | 0.009 | | 0.013 |
| D | 10.10 | | 10.50 | 0.398 | | 0.413 |
| E | 7.40 | | 7.60 | 0.291 | | 0.299 |
| H | 10.00 | | 10.65 | 0.394 | | 0.419 |
| e | | 1.27 | | | 0.050 | |
| h | 0.25 | | 0.75 | 0.010 | | 0.030 |
| $\alpha$ | 0° | | 8° | 0° | | 8° |
| L | 0.40 | | 1.27 | 0.016 | | 0.050 |
| | Number of Pins | | | | | |
| N | 16 | | | | | |

### Figure 115. 20-Pin Plastic Dual In-Line Package, 300-mil Width




| Dim. | mm | | | inches) | | |
|---|---|---|---|---|---|---|
| | Min | Typ | Max | Min | Typ | M |
| A | | | 5.33 | | | 0.2 |
| A1 | 0.38 | | | 0.0150 | | |
| A2 | 2.92 | 3.30 | 4.95 | 0.1150 | 0.1299 | 0.1 |
| b | 0.36 | 0.46 | 0.56 | 0.0142 | 0.0181 | 0.0 |
| b2 | 1.14 | 1.52 | 1.78 | 0.0449 | 0.0598 | 0.0 |
| c | 0.20 | 0.25 | 0.36 | 0.0079 | 0.0098 | 0.0 |
| D | 24.89 | 26.16 | 26.92 | 0.9799 | 1.0299 | 1.0 |
| D1 | 0.13 | | | 0.0051 | | |
| e | | 2.54 | | | 0.1000 | |
| eB | | | 10.92 | | | 0.4 |
| E1 | 6.10 | 6.35 | 7.11 | 0.2402 | 0.2500 | 0.2 |
| L | 2.92 | 3.30 | 3.81 | 0.1150 | 0.1299 | 0.1 |
| | Number of Pins | | | | | |
| N | 20 | | | | | |

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

**Please Read Carefully:**

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

**UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.**

**UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.**

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.



STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

**www.st.com**