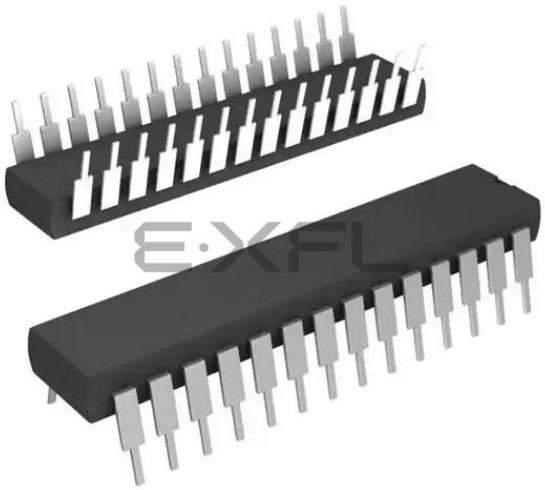


Welcome to E-XFL.COM

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	ST6
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LVD, POR, WDT
Number of I/O	20
Program Memory Size	4KB (4K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	64 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 6V
Data Converters	A/D 16x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.600", 15.24mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st62t25cb6

2 PIN DESCRIPTION

Figure 2. 28-Pin Package Pinout

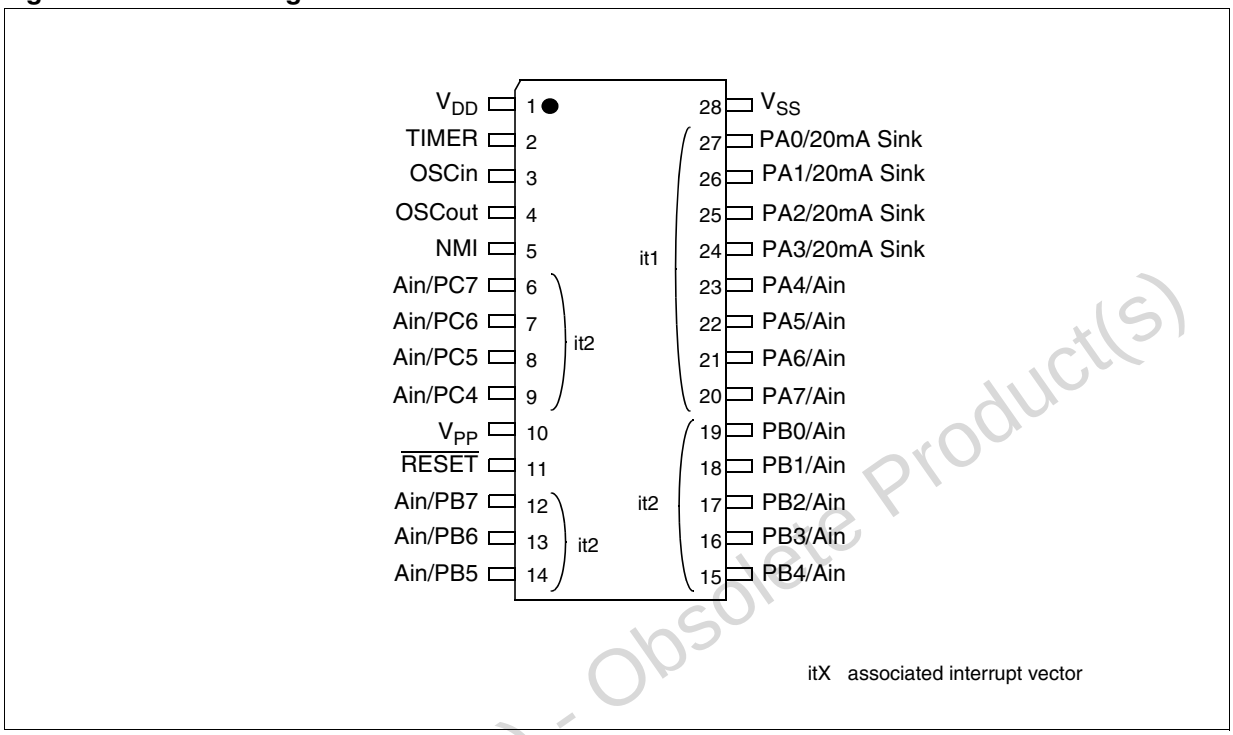


Table 1. Device Pin Description

Pin n°	Pin Name	Type	Main Function (after Reset)	Alternate Function
1	V_{DD}	S	Main power supply	
2	TIMER	I/O	Timer input or output	
3	OSCin	I	External clock input or resonator oscillator inverter input	
4	OSCout	O	Resonator oscillator inverter output or resistor input for RC oscillator	
5	NMI	I	Non maskable interrupt (falling edge sensitive)	
6	PC7/Ain	I/O	Pin C7 (IPU)	Analog input
7	PC6/Ain	I/O	Pin C6 (IPU)	Analog input
8	PC5/Ain	I/O	Pin C5 (IPU)	Analog input
9	PC4/Ain	I/O	Pin C4 (IPU)	Analog input
10	V_{PP}		Must be held at Vss for normal operation, if a 12.5V level is applied to the pin during the reset phase, the device enters EPROM programming mode.	
11	$\overline{\text{RESET}}$	I/O	Top priority non maskable interrupt (active low)	
12	PB7/Ain	I/O	Pin B7 (IPU)	Analog input
13	PB6/Ain	I/O	Pin B6 (IPU)	Analog input

MEMORY MAP (Cont'd)

3.1.2 Program Space

Program Space comprises the instructions to be executed, the data required for immediate addressing mode instructions, the reserved factory test area and the user vectors. Program Space is addressed via the 12-bit Program Counter register (PC register). Thus, the MCU is capable of addressing 4K bytes of memory directly.

3.1.3 Readout Protection

The Program Memory in OTP or EPROM devices can be protected against external readout of memory by setting the Readout Protection bit in the option byte (Section 3.3 on page 16).

In the EPROM parts, Readout Protection option can be desactivated only by U.V. erasure that also results in the whole EPROM context being erased.

Note: Once the Readout Protection is activated, it is no longer possible, even for STMicroelectronics, to gain access to the OTP contents. Returned parts can therefore not be accepted if the Readout Protection bit is set.

3.1.4 Data Space

Data Space accommodates all the data necessary for processing the user program. This space comprises the RAM resource, the processor core and peripheral registers, as well as read-only data such as constants and look-up tables in OTP/EPROM.

3.1.4.1 Data ROM

All read-only data is physically stored in program memory, which also accommodates the Program Space. The program memory consequently contains the program code to be executed, as well as the constants and look-up tables required by the application.

The Data Space locations in which the different constants and look-up tables are addressed by the processor core may be thought of as a 64-byte window through which it is possible to access the read-only data stored in OTP/EPROM.

3.1.4.2 Data RAM

The data space includes the user RAM area, the accumulator (A), the indirect registers (X), (Y), the short direct registers (V), (W), the I/O port registers, the peripheral data and control registers, the interrupt option register and the Data ROM Window register (DRWR register).

3.1.5 Stack Space

Stack space consists of six 12-bit registers which are used to stack subroutine and interrupt return addresses, as well as the current program counter contents.

5.2 LOW VOLTAGE DETECTOR (LVD)

The on-chip Low Voltage Detector is enabled by setting a bit in the option bytes (refer to the Option Bytes section of this document).

The LVD allows the device to be used without any external RESET circuitry. In this case, the $\overline{\text{RESET}}$ pin should be left unconnected.

If the LVD is not used, an external circuit is mandatory to ensure correct Power On Reset operation, see figure in the Reset section. For more details, please refer to the application note AN669.

The LVD generates a static Reset when the supply voltage is below a reference value. This means that it secures the power-up as well as the power-down keeping the ST6 in reset.

The V_{IT-} reference value for a voltage drop is lower than the V_{IT+} reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

The LVD Reset circuitry generates a reset when V_{DD} is below:

– V_{IT+} when V_{DD} is rising

– V_{IT-} when V_{DD} is falling

The LVD function is illustrated in Figure 12.

If the LVD is enabled, the MCU can be in only one of two states:

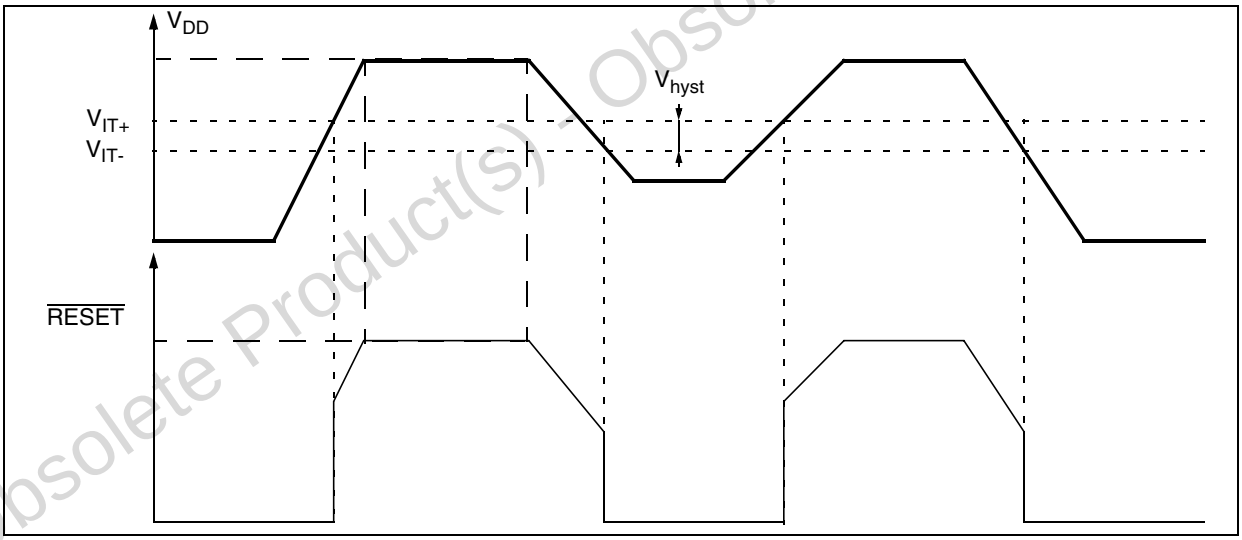
– Over the input threshold voltage, it is running under full software control

– Below the input threshold voltage, it is in static safe reset

In these conditions, secure operation is guaranteed without the need for external reset hardware.

During a Low Voltage Detector Reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

Figure 12. Low Voltage Detector Reset



5.3 RESET

5.3.1 Introduction

The MCU can be reset in three ways:

- A low pulse input on the $\overline{\text{RESET}}$ pin
- Internal Watchdog reset
- Internal Low Voltage Detector (LVD) reset

5.3.2 RESET Sequence

The basic RESET sequence consists of 3 main phases:

- Internal (watchdog or LVD) or external Reset event
- A delay of 2048 clock (f_{INT}) cycles
- RESET vector fetch

The reset delay allows the oscillator to stabilise and ensures that recovery has taken place from the Reset state.

The RESET vector fetch phase duration is 2 clock cycles.

When a reset occurs:

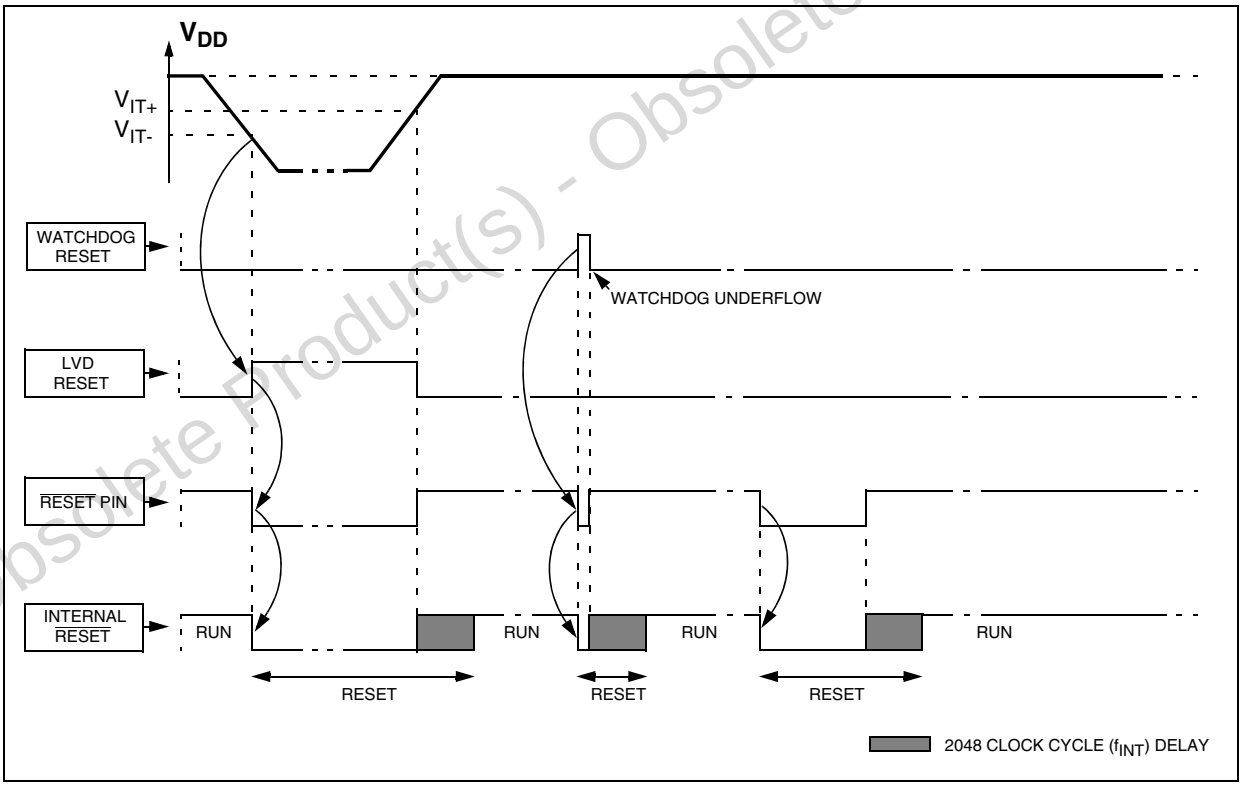
– The stack is cleared

– The PC is loaded with the address of the Reset vector. It is located in program ROM starting at address 0FFEh.

A jump to the beginning of the user program must be coded at this address.

– The interrupt flag is automatically set, so that the CPU is in Non Maskable Interrupt mode. This prevents the initialization routine from being interrupted. The initialization routine should therefore be terminated by a RETI instruction, in order to go back to normal mode.

Figure 13. RESET Sequence



RESET (Cont'd)

5.3.4 Watchdog Reset

The MCU provides a Watchdog timer function in order to be able to recover from software hang-ups. If the Watchdog register is not refreshed before an end-of-count condition is reached, a Watchdog reset is generated.

After a Watchdog reset, the MCU restarts in the same way as if a Reset was generated by the $\overline{\text{RESET}}$ pin.

Note: When a watchdog reset occurs, the $\overline{\text{RESET}}$ pin is tied low for very short time period, to flag the reset phase. This time is not long enough to reset external circuits.

For more details refer to the Watchdog Timer chapter.

5.3.5 LVD Reset

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

- Power-On RESET
- Voltage Drop RESET

During an LVD reset, the $\overline{\text{RESET}}$ pin is pulled low when $V_{DD}<V_{IT+}$ (rising edge) or $V_{DD}<V_{IT-}$ (falling edge).

For more details, refer to the LVD chapter.

Caution: Do not externally connect directly the $\overline{\text{RESET}}$ pin to V_{DD}, this may cause damage to the component in case of internal RESET (Watchdog or LVD).

Figure 15. Simple External Reset Circuitry

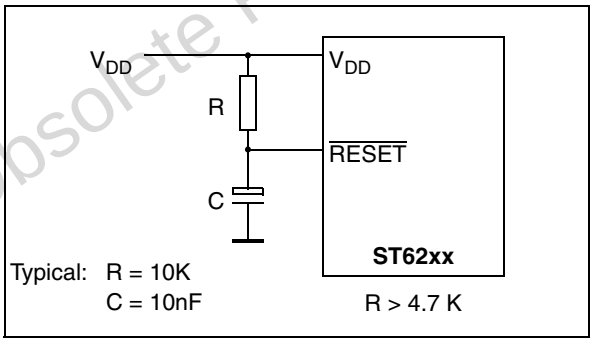
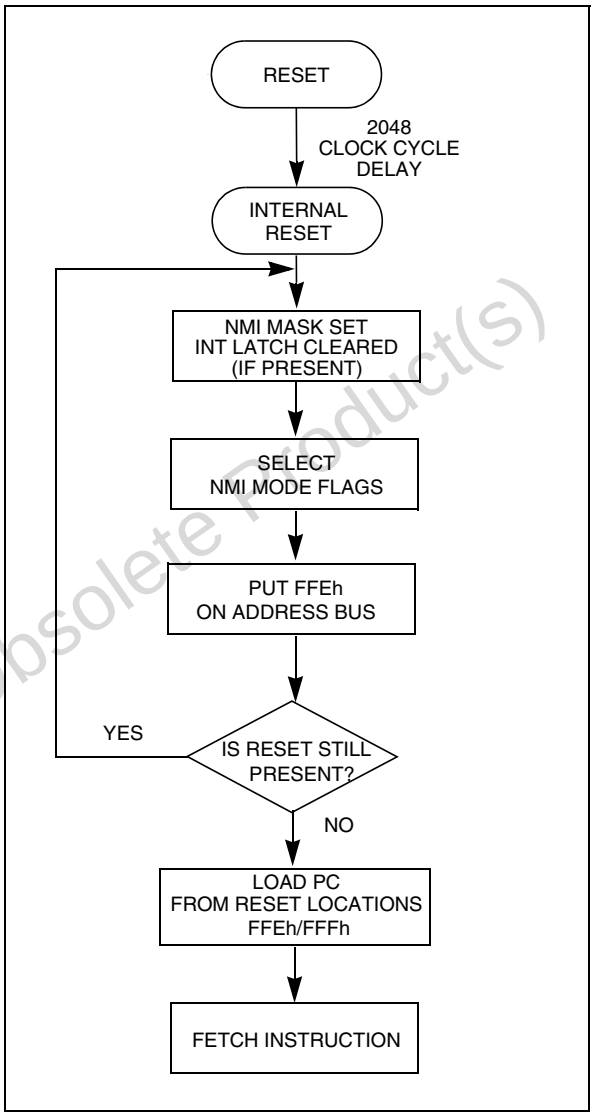


Figure 16. Reset Processing



6.6 INTERRUPT HANDLING PROCEDURE

The interrupt procedure is very similar to a call procedure, in fact the user can consider the interrupt as an asynchronous call procedure. As this is an asynchronous event, the user cannot know the context and the time at which it occurred. As a result, the user should save all Data space registers which may be used within the interrupt routines. The following list summarizes the interrupt procedure:

When an interrupt request occurs, the following actions are performed by the MCU automatically:

– The core switches from the normal flags to the interrupt flags (or the NMI flags).
– The PC contents are stored in the top level of the stack.
– The normal interrupt lines are inhibited (NMI still active).
– The internal latch (if any) is cleared.
– The associated interrupt vector is loaded in the PC.

When an interrupt request occurs, the following actions must be performed by the user software:

– User selected registers have to be saved within the interrupt service routine (normally on a software stack).
– The source of the interrupt must be determined by polling the interrupt flags (if more than one source is associated with the same vector).
– The RETI (RETurn from Interrupt) instruction must end the interrupt service routine.

After the RETI instruction is executed, the MCU returns to the main routine.

Caution: When a maskable interrupt occurs while the ST6 core is in NORMAL mode and during the execution of an “ldi IOR, 00h” instruction (disabling all maskable interrupts): if the interrupt request occurs during the first 3 cycles of the “ldi” instruction (which is a 4-cycle instruction) the core will switch to interrupt mode BUT the flags CN and ZN will NOT switch to the interrupt pair CI and ZI.

6.6.1 Interrupt Response Time

This is defined as the time between the moment when the Program Counter is loaded with the interrupt vector and when the program has jump to the interrupt subroutine and is ready to execute the code. It depends on when the interrupt occurs while the core is processing an instruction.

Figure 18. Interrupt Processing Flow Chart

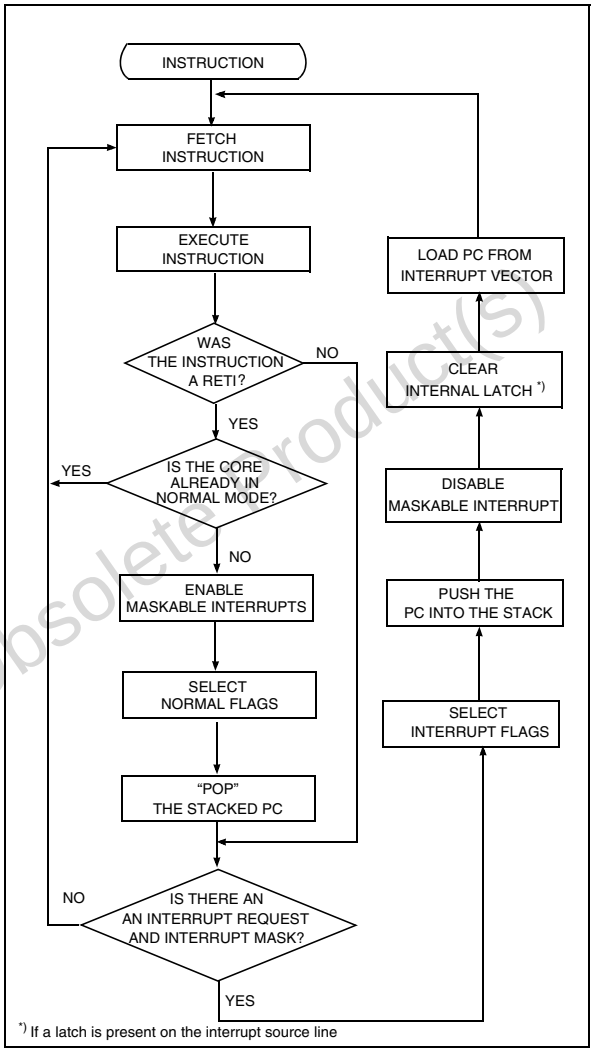


Table 6. Interrupt Response Time

Minimum	6 CPU cycles
Maximum	11 CPU cycles

One CPU cycle is 13 external clock cycles thus 11 CPU cycles = 11 x (13 /8M) = 17.875 µs with an 8 MHz external quartz.

6.7 REGISTER DESCRIPTION

INTERRUPT OPTION REGISTER (IOR)

Address: 0C8h — Write Only

Reset status: 00h

7							0
-	LES	ESB	GEN	-	-	-	-

Caution: This register is write-only and cannot be accessed by single-bit operations (SET, RES, DEC,...).

Bit 7 =Reserved, must be cleared.

Bit 6 = **LES** *Level/Edge Selection bit*.
0: Falling edge sensitive mode is selected for interrupt vector #1
1: Low level sensitive mode is selected for interrupt vector #1

Bit 5 = **ESB** *Edge Selection bit*.
0: Falling edge mode on interrupt vector #2
1: Rising edge mode on interrupt vector #2

Bit 4 = **GEN** *Global Enable Interrupt*.
0: Disable all maskable interrupts
1: Enable all maskable interrupts

Note: When the GEN bit is cleared, the NMI interrupt is active but cannot be used to exit from STOP or WAIT modes.

Bits 3:0 = Reserved, must be cleared.

Table 7. Interrupt Mapping

Vector number	Source Block	Description	Register Label	Flag	Exit from STOP	Vector Address	Priority Order
	RESET	Reset	N/A	N/A	yes	FFEh-FFFh	Highest Priority ↓ Lowest Priority
Vector #0	NMI	Non Maskable Interrupt	N/A	N/A	yes	FFCh-FFDh	
NOT USED						FFAh-FFBh	
						FF8h-FF9h	
Vector #1	Port A	Ext. Interrupt Port A	N/A	N/A	yes	FF6h-FF7h	
Vector #2	Port B, C	Ext. Interrupt Port B, C	N/A	N/A	yes	FF4h-FF5h	
Vector #3	TIMER	Timer underflow	TSCR	TMZ	yes	FF2h-FF3h	
Vector #4	ADC	End Of Conversion	ADCR	EOC	no	FF0h-FF1h	

7 POWER SAVING MODES

7.1 INTRODUCTION

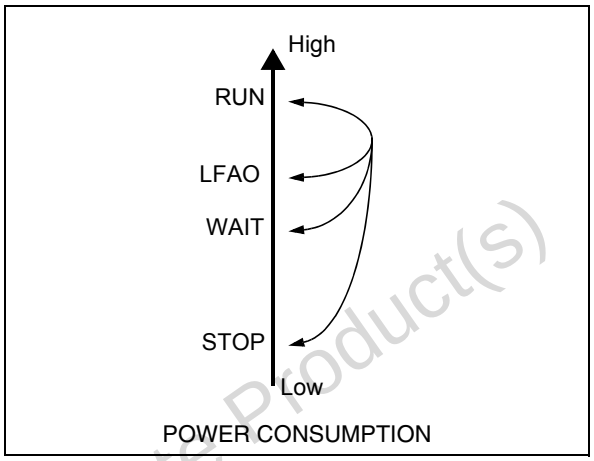
To give a large measure of flexibility to the application in terms of power consumption, two main power saving modes are implemented in the ST6 (see Figure 19).

In addition, the Low Frequency Auxiliary Oscillator (LFAO) can be used instead of the main oscillator to reduce power consumption in RUN and WAIT modes.

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.

From Run mode, the different power saving modes may be selected by calling the specific ST6 software instruction or for the LFAO by setting the relevant register bit. For more information on the LFAO, please refer to the Clock chapter.

Figure 19. Power Saving Mode Transitions



7.2 WAIT MODE

The MCU goes into WAIT mode as soon as the WAIT instruction is executed. This has the following effects:

– Program execution is stopped, the microcontroller software can be considered as being in a “frozen” state.

– RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage.

– The oscillator is kept running to provide a clock to the peripherals; they are still active.

WAIT mode can be used when the user wants to reduce the MCU power consumption during idle periods, while not losing track of time or the ability to monitor external events. WAIT mode places the MCU in a low power consumption mode by stopping the CPU. The active oscillator (main oscillator or LFAO) is kept running in order to provide a clock signal to the peripherals.

If the power consumption has to be further reduced, the Low Frequency Auxiliary Oscillator (LFAO) can be used in place of the main oscillator, if its operating frequency is lower. If required, the LFAO must be switched on before entering WAIT mode.

Exit from Wait mode

The MCU remains in WAIT mode until one of the following events occurs:

– RESET (Watchdog, LVD or $\overline{\text{RESET}}$ pin)

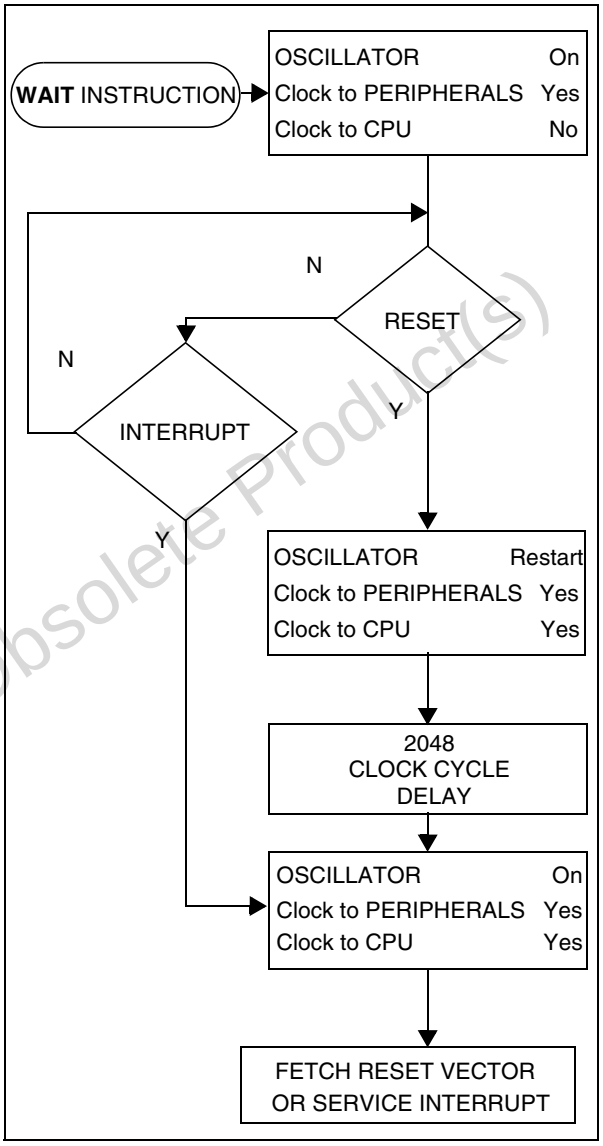
– A peripheral interrupt (timer, ADC,...),

– An external interrupt (I/O port, NMI)

The Program Counter then branches to the starting address of the interrupt or RESET service routine. Refer to Figure 20.

See also Section 7.4.1.

Figure 20. WAIT Mode Flowchart



8-BIT TIMER (Cont'd)

9.2.7 Register Description

PRESCALER COUNTER REGISTER (PSCR)
Address: 0D2h - Read/Write
Reset Value: 0111 1111 (7Fh)

7							0
PSCR 7	PSCR 6	PSCR 5	PSCR 4	PSCR 3	PSCR 2	PSCR 1	PSCR 0

Bit 7 = **PSCR7:** Not used, always read as "0".
Bits 6:0 = **PSCR[6:0]** *Prescaler LSB.*

TIMER COUNTER REGISTER (TCR)
Address: 0D3h - Read / Write
Reset Value: 1111 1111 (FFh)

7							0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

Bits 7:0 = **TCR[7:0]** *Timer counter bits.*

TIMER STATUS CONTROL REGISTER (TSCR)
Address: 0D4h - Read/Write
Reset Value: 0000 0000 (00h)

7							0
TMZ	ETI	TOUT	DOUT	PSI	PS2	PS1	PS0

Bit 7 = **TMZ** *Timer Zero bit.*
A low-to-high transition indicates that the timer count register has underflowed. It means that the TCR value has changed from 00h to FFh.
This bit must be cleared by user software.
0: Counter has not underflowed
1: Counter underflow occurred

Bit 6 = **ETI** *Enable Timer Interrupt.*
When set, enables the timer interrupt request. If ETI=0 the timer interrupt is disabled. If ETI=1 and TMZ=1 an interrupt request is generated.
0: Interrupt disabled (reset state)
1: Interrupt enabled

Bit 5 = **TOUT** Timer Output Control.
When low, this bit selects the input mode for the TIMER pin. When high the output mode is selected.
0: Input mode (reset state)
1: Output mode, the TIMER pin is configured as push-pull output

Bit 4 = **DOUT** *Data Output.*
Data sent to the timer output when TMZ is set high (output mode only). Input mode selection (input mode only).

Bit 3 = **PSI**: *Prescaler Initialize bit.*
Used to initialize the prescaler and inhibit its counting. When PSI="0" the prescaler is set to 7Fh and the counter is inhibited. When PSI="1" the prescaler is enabled to count downwards. As long as PSE="1" both counter and prescaler are not running
0: Counting disabled
1: Counting enabled

Bits 1:0 = **PS[2:0]** *Prescaler Mux. Select.*
These bits select the division ratio of the prescaler register.

Table 13. Prescaler Division Factors

PS2	PS1	PS0	Divided by
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Table 14. 8-Bit Timer Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0D2h	**PSCR**	PSCR7	PSCR6	PSCR5	PSCR4	PSCR3	PSCR2	PSCR1	PSCR0
	Reset Value	0	1	1	1	1	1	1	1
0D3h	**TCR**	TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0
	Reset Value	1	1	1	1	1	1	1	1
0D4h	**TSCR**	TMZ	ETI	TOUT	DOUT	PSI	PS2	PS1	PS0
	Reset Value	0	0	0	0	0	0	0	0

9.3 A/D CONVERTER (ADC)

9.3.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 8-bit, successive approximation converter. This peripheral has multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from different sources.

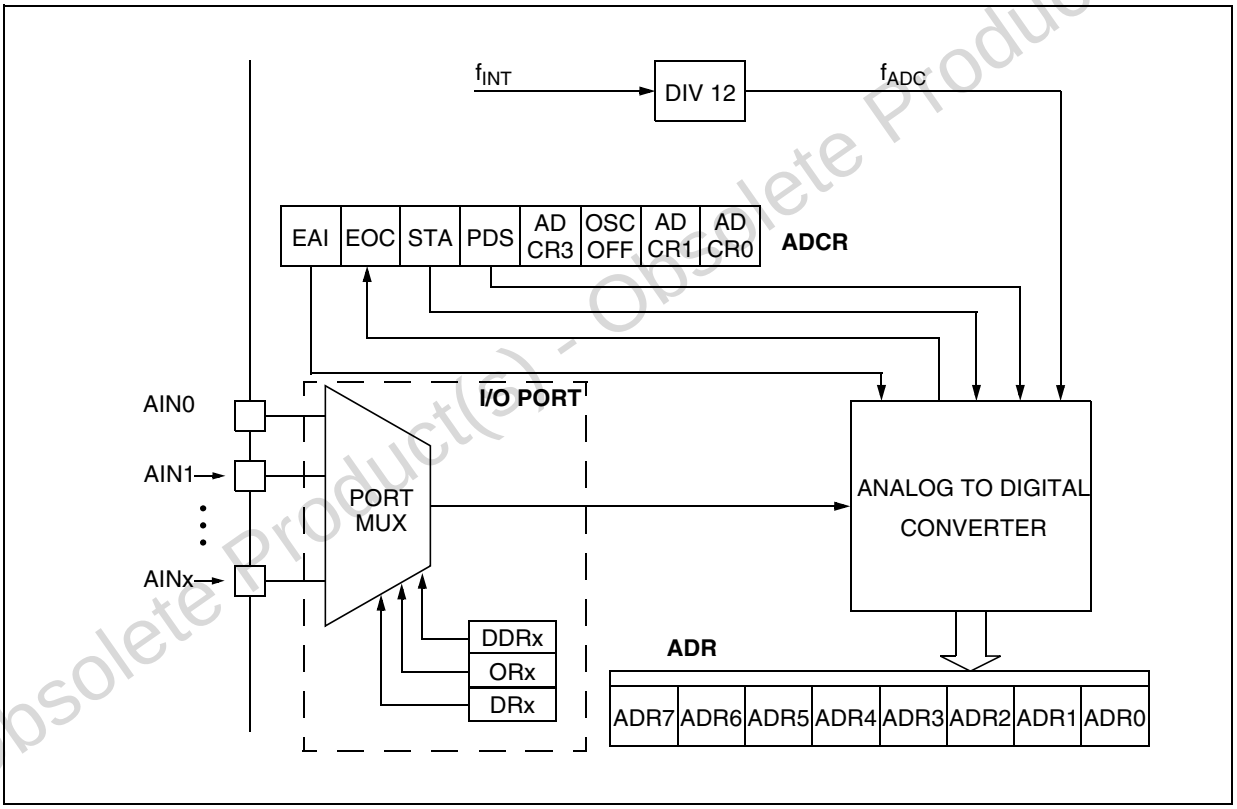
The result of the conversion is stored in a 8-bit Data Register. The A/D converter is controlled through a Control Register.

9.3.2 Main Features

- 8-bit conversion
- Multiplexed analog input channels
- Linear successive approximation
- Data register (DR) which contains the results
- End of Conversion flag
- On/Off bit (to reduce consumption)
- Typical conversion time 70 µs (with an 8 MHz crystal)

The block diagram is shown in Figure 34.

Figure 34. ADC Block Diagram



A/D CONVERTER (Cont'd)

9.3.4 Recommendations

The following six notes provide additional information on using the A/D converter.

1.The A/D converter does not feature a sample and hold circuit. The analog voltage to be measured should therefore be stable during the entire conversion cycle. Voltage variation should not exceed ±1/2 LSB for optimum conversion accuracy. A low pass filter may be used at the analog input pins to reduce input voltage variation during conversion.

2. When selected as an analog channel, the input pin is internally connected to a capacitor C_{ad} of typically 9pF. For maximum accuracy, this capacitor must be fully charged at the beginning of conversion. In the worst case, conversion starts one instruction (6.5 µs) after the channel has been selected. The impedance of the analog voltage source (ASI) in worst case conditions, is calculated using the following formula:

$$6.5\mu s = 9 \times C_{ad} \times ASI$$

(capacitor charged to over 99.9%), i.e. 30 kΩ including a 50% guardband.
The ASI can be higher if C_{ad} has been charged for a longer period by adding instructions before the start of conversion (adding more than 26 CPU cycles is pointless).

3. Since the ADC is on the same chip as the microprocessor, the user should not switch heavily loaded output signals during conversion, if high precision is required. Such switching will affect the supply voltages used as analog references.

4. Conversion accuracy depends on the quality of the power supplies (V_{DD} and V_{SS}). The user must take special care to ensure a well regulated reference voltage is present on the V_{DD} and V_{SS} pins (power supply voltage variations must be less than 0.1V/ms). This implies, in particular, that a suitable decoupling capacitor is used at the V_{DD} pin.
The converter resolution is given by:

$$\frac{V_{DD} - V_{SS}}{256}$$

The Input voltage (Ain) which is to be converted must be constant for 1µs before conversion and remain constant during conversion.

5. Conversion resolution can be improved if the power supply voltage (V_{DD}) to the microcontroller is lowered.

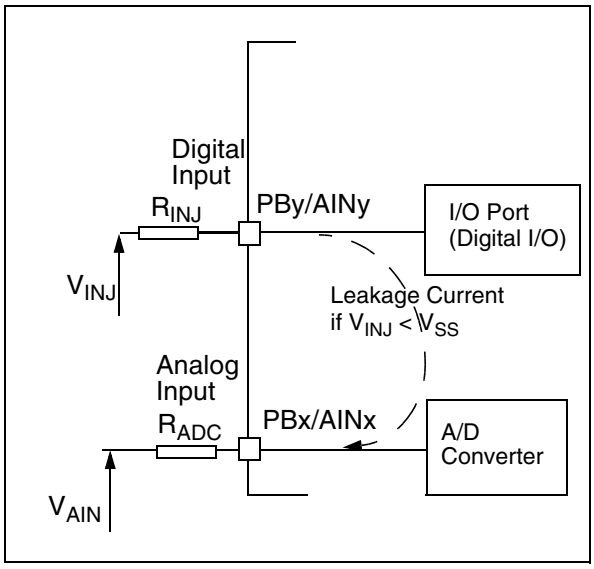
6. In order to optimize the conversion resolution, the user can configure the microcontroller in WAIT mode, because this mode minimises noise disturbances and power supply variations due to output switching. Nevertheless, the WAIT instruction should be executed as soon as possible after the beginning of the conversion, because execution of the WAIT instruction may cause a small variation of the V_{DD} voltage. The negative effect of this variation is minimized at the beginning of the conversion when the converter is less sensitive, rather than at the end of conversion, when the least significant bits are determined.
The best configuration, from an accuracy standpoint, is WAIT mode with the Timer stopped. In this case only the ADC peripheral and the oscillator are then still working. The MCU must be woken up from WAIT mode by the ADC interrupt at the end of the conversion. The microcontroller can also be woken up by the Timer interrupt, but this means the Timer must be running and the resulting noise could affect conversion accuracy.

Caution: When an I/O pin is used as an analog input, A/D conversion accuracy will be impaired if negative current injections ($V_{INJ} < V_{SS}$) occur from adjacent I/O pins with analog input capability. Refer to Figure 35. To avoid this:

– Use another I/O port located further away from the analog pin, preferably not multiplexed on the A/D converter

– Increase the input resistance $R_{IN\,J}$ (to reduce the current injections) and reduce R_{ADC} (to preserve conversion accuracy).

Figure 35. Leakage from Digital Inputs



Opcode Map Summary. The following table contains an opcode map for the instructions used by the ST6

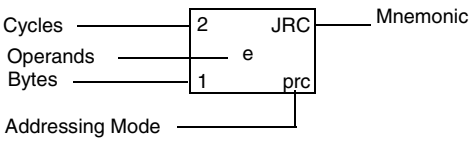
LOW / HI	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	LOW / HI
0 **0000**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e NOP 1 pcr	#	2 JRC e 1 prc	4 LD a,(x) 1 ind	**0** **0000**
1 **0001**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC x 1 sd	2 JRC e 1 prc	4 LDI a,nn 2 imm	**1** **0001**
2 **0010**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b4,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 CP a,(x) 1 ind	**2** **0010**
3 **0011**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b4,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,x 1 sd	2 JRC e 1 prc	4 CPI a,nn 2 imm	**3** **0011**
4 **0100**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b2,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 ADD a,(x) 1 ind	**4** **0100**
5 **0101**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b2,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC y 1 sd	2 JRC e 1 prc	4 ADDI a,nn 2 imm	**5** **0101**
6 **0110**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b6,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 INC (x) 1 ind	**6** **0110**
7 **0111**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b6,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,y 1 sd	2 JRC e 1 prc	#	**7** **0111**
8 **1000**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b1,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (x),a 1 ind	**8** **1000**
9 **1001**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b1,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC v 1 sd	2 JRC e 1 prc	#	**9** **1001**
A **1010**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b5,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 AND a,(x) 1 ind	**A** **1010**
B **1011**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b5,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,v 1 sd	2 JRC e 1 prc	4 ANDI a,nn 2 imm	**B** **1011**
C **1100**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b3,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 SUB a,(x) 1 ind	**C** **1100**
D **1101**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b3,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC w 1 sd	2 JRC e 1 prc	4 SUBI a,nn 2 imm	**D** **1101**
E **1110**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b7,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 DEC (x) 1 ind	**E** **1110**
F **1111**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b7,rr,ee 3 bt	2 JRZ e 1 pcr	4 LD a,w 1 sd	2 JRC e 1 prc	#	**F** **1111**

Abbreviations for Addressing Modes:

dir	Direct
sd	Short Direct
imm	Immediate
inh	Inherent
ext	Extended
b.d	Bit Direct
bt	Bit Test
pcr	Program Counter Relative
ind	Indirect

Legend:

#	Indicates Illegal Instructions
e	5-bit Displacement
b	3-bit Address
rr	1-byte Data space address
nn	1-byte immediate data
abc	12-bit address
ee	8-bit displacement



SUPPLY CURRENT CHARACTERISTICS (Cont'd)

11.4.2 WAIT Modes

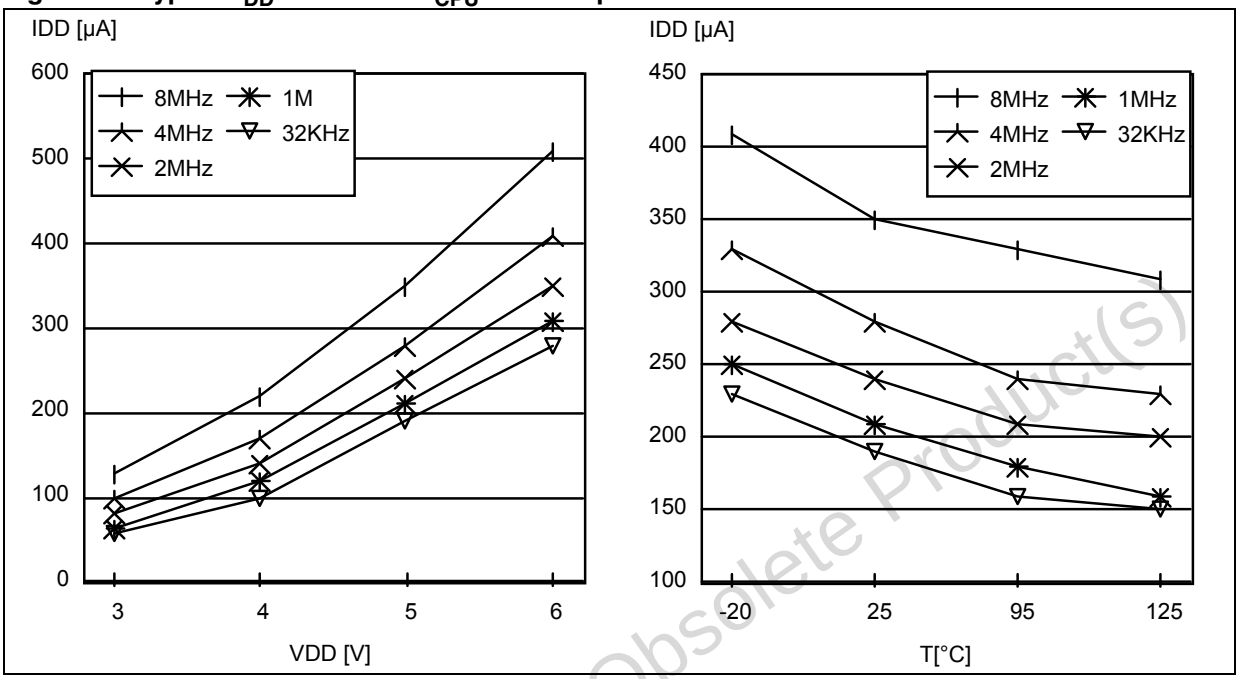
Symbol	Parameter	Conditions			Typ [1]	Max [2]	Unit
I_{DD}	Supply current in WAIT mode [3] Option bytes not programmed (see Figure 44)	$4.5V \le V_{DD} \le 6.0V$	OTP devices	f_{OSC}=32kHz f_{OSC}=1MHz f_{OSC}=2MHz f_{OSC}=4MHz f_{OSC}=8MHz	330 350 370 410 480	550 600 650 700 800	µA
	Supply current in WAIT mode [3] Option bytes programmed to 00H (see Figure 45)			f_{OSC}=32kHz f_{OSC}=1MHz f_{OSC}=2MHz f_{OSC}=4MHz f_{OSC}=8MHz	18 26 41 57 70	60 80 120 180 200	
	Supply current in WAIT mode [3] (see Figure 46)		ROM devices	f_{OSC}=32kHz f_{OSC}=1MHz f_{OSC}=2MHz f_{OSC}=4MHz f_{OSC}=8MHz	190 210 240 280 350	300 350 400 500 600	
	Supply current in WAIT mode [3] Option bytes not programmed (see Figure 44)	$3V \le V_{DD} \le 3.6V$	OTP devices	f_{OSC}=32kHz f_{OSC}=1MHz f_{OSC}=2MHz f_{OSC}=4MHz f_{OSC}=8MHz	80 90 100 120 150	120 140 150 200 250	
	Supply current in WAIT mode [3] Option bytes programmed to 00H (see Figure 45)			f_{OSC}=32kHz f_{OSC}=1MHz f_{OSC}=2MHz f_{OSC}=4MHz f_{OSC}=8MHz	5 8 16 18 20	30 40 50 60 100	
	Supply current in WAIT mode [3] Option bytes not programmed (see Figure 46)		ROM devices	f_{OSC}=32kHz f_{OSC}=1MHz f_{OSC}=2MHz f_{OSC}=4MHz f_{OSC}=8MHz	60 65 80 100 130	100 110 120 150 210	

Notes:

1. Typical data are based on T_A=25°C, V_{DD}=5V ($4.5V \le V_{DD} \le 6.0V$ range) and V_{DD}=3.3V ($3V \le V_{DD} \le 3.6V$ range).
2. Data based on characterization results, tested in production at V_{DD} max. and f_{OSC} max.
3. All I/O pins in input with pull-up mode (no load), all peripherals in reset state; clock input (OSC_{IN}) driven by external square wave, OSG and LVD disabled.

SUPPLY CURRENT CHARACTERISTICS (Cont'd)

Figure 46. Typical I_{DD} in WAIT vs f_{CPU} and Temperature for ROM devices



SUPPLY CURRENT CHARACTERISTICS (Cont'd)

11.4.3 STOP Mode

Symbol	Parameter	Conditions	Typ [1]	Max	Unit
I_{DD}	Supply current in STOP mode [2] (see Figure 47 & Figure 48)	OTP devices	0.3	10 [3] 20 [4]	µA
		ROM devices	0.1	2 [3] 20 [4]	

Notes:

1. Typical data are based on V_{DD}=5.0V at T_A=25°C.
2. All I/O pins in input with pull-up mode (no load), all peripherals in reset state, OSG and LVD disabled, option bytes programmed to 00H. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
3. Maximum STOP consumption for -40°C<Ta<90°C
4. Maximum STOP consumption for -40°C<Ta<125°C

Figure 47. Typical I_{DD} in STOP vs Temperature for OTP devices

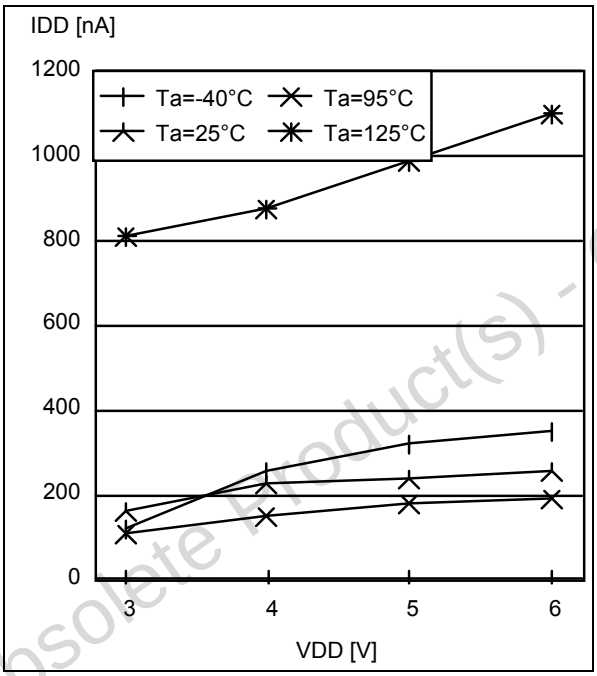
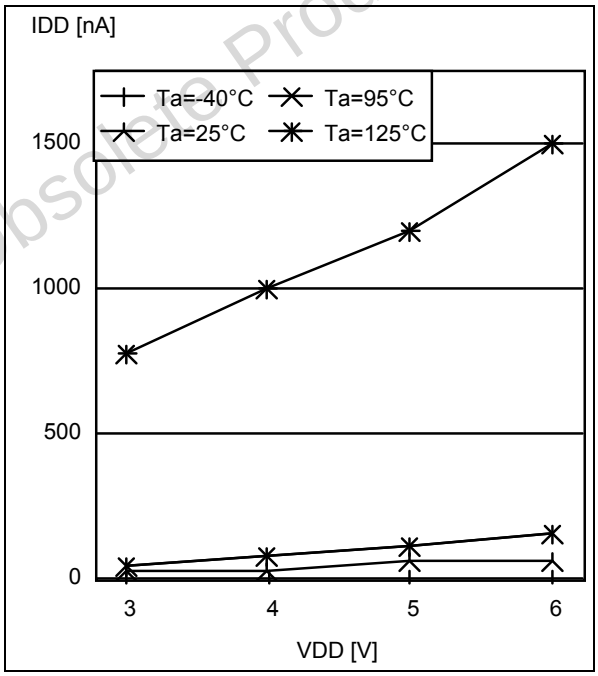


Figure 48. Typical I_{DD} in STOP vs Temperature for ROM devices



CLOCK AND TIMING CHARACTERISTICS (Cont'd)

Figure 52. Typical RC Oscillator frequency vs. V_{DD}

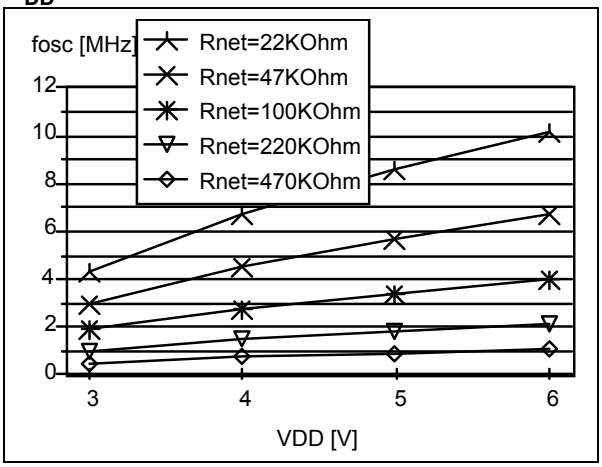
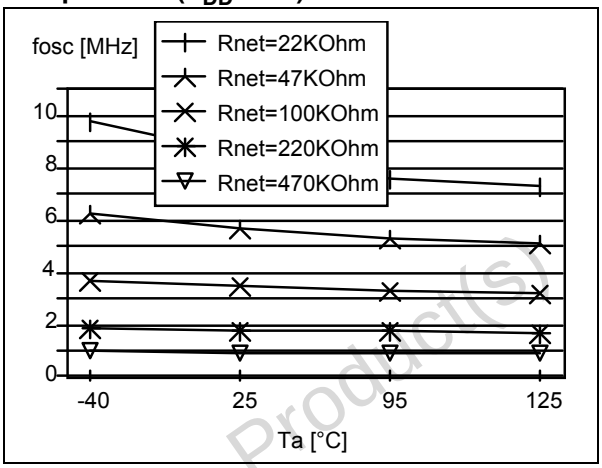


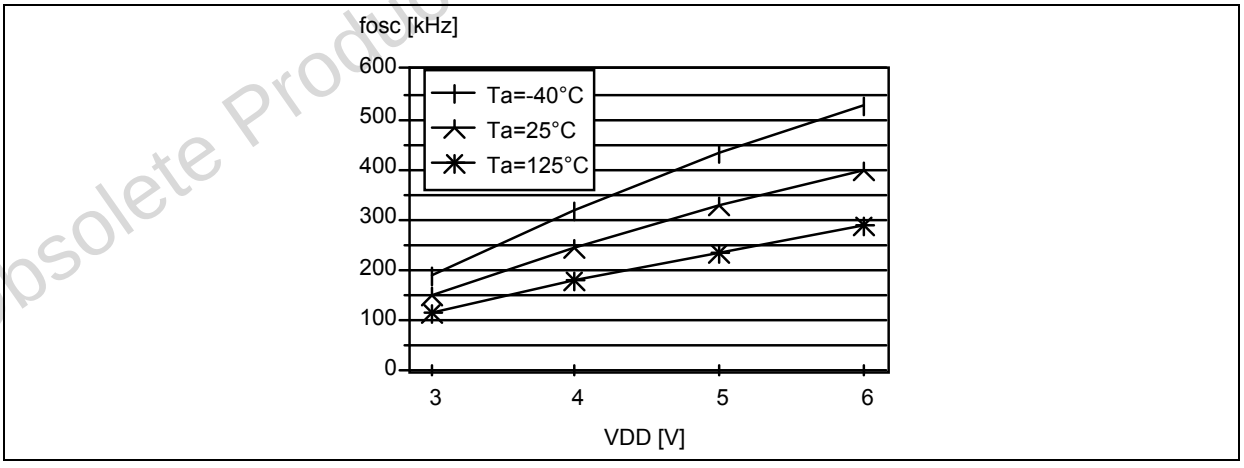
Figure 53. Typical RC Oscillator frequency vs. Temperature (V_{DD} = 5V)



11.5.5 Oscillator Safeguard (OSG) and Low Frequency Auxiliary Oscillator (LFAO)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{LFAO}	Low Frequency Auxiliary Oscillator Frequency [1]	T_A=25° C, V_{DD}=5.0 V	200	350	800	kHz
		T_A=25° C, V_{DD}=3.3 V	86	150	340	
f_{OSG}	Internal Frequency with OSG enabled	T_A=25° C, V_{DD}=4.5 V	4			MHz
		T_A=25° C, V_{DD}=3.3 V	2			

Figure 54. Typical LFAO Frequencies



Note:
1. Data based on characterization results.

EMC CHARACTERISTICS (Cont'd)

11.7.2.2 Static and Dynamic Latch-Up

- **LU**: 3 complementary static tests are required on 10 parts to assess the latch-up performance. A supply overvoltage (applied to each power supply pin), a current injection (applied to each input, output and configurable I/O pin) and a power supply switch sequence are performed on each sample. This test conforms to the EIA/JESD 78 IC latch-up standard. For more details, refer to the AN1181 application note.
- **DLU**: Electro-Static Discharges (one positive then one negative test) are applied to each pin of 3 samples when the micro is running to assess the latch-up performance in dynamic mode. Power supplies are set to the typical values, the oscillator is connected as near as possible to the pins of the micro and the component is put in reset mode. This test conforms to the IEC1000-4-2 and SAEJ1752/3 standards and is described in Figure 58. For more details, refer to the AN1181 application note.

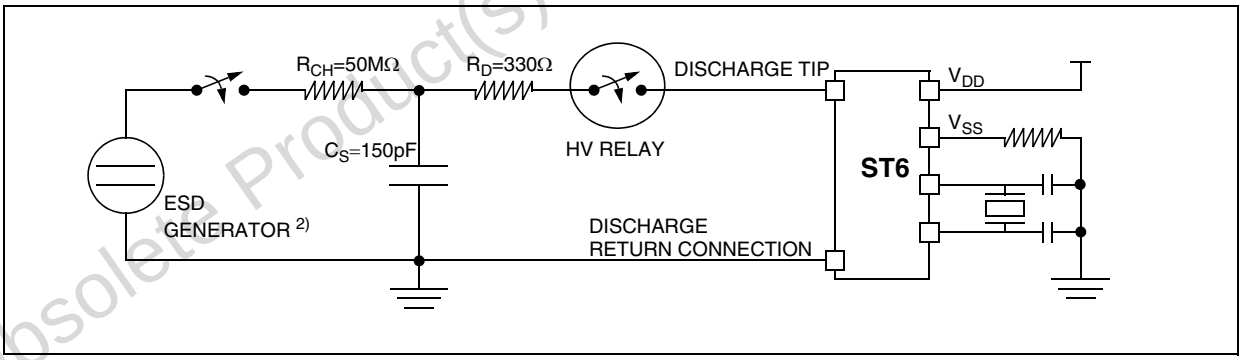
Electrical Sensitivities

Symbol	Parameter	Conditions	Class [1]
LU	Static latch-up class	T_A=+25°C T_A=+85°C	A A
DLU	Dynamic latch-up class	V_{DD}=5V, f_{OSC}=4MHz, T_A=+25°C	A

Notes:

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).
2. Schaffner NSG435 with a pointed test finger.

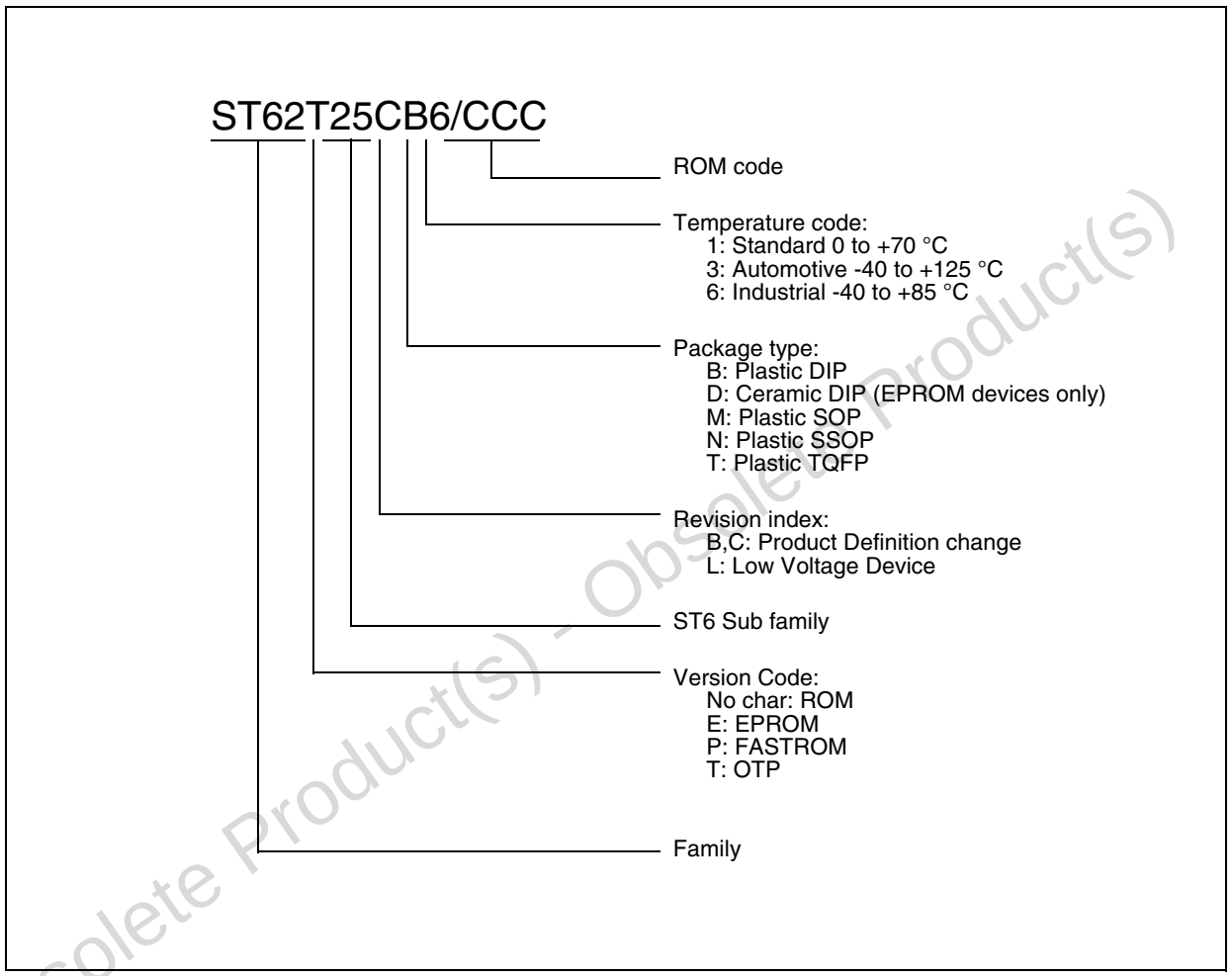
Figure 58. Simplified Diagram of the ESD Generator for DLU



12.5 ORDERING INFORMATION

The following section deals with the procedure for transfer of customer codes to STMicroelectronics and also details the ST6 factory coded device type.

Figure 78. ST6 Factory Coded Device Types



ST6215C/25C/P15C/P25C MICROCONTROLLER OPTION LIST

Customer: . . . . . .
Address: . . . . . .
. . . . . .
Contact: . . . . . .
Phone: . . . . . .
Reference: . . . . . .

STMicroelectronics references:

Device: [ ] ST6215C (2 KB) [ ] ST6225C (4 KB)
[ ] ST62P15C (2 KB) [ ] ST62P25C (4 KB)

Package: [ ] Dual in Line Plastic
[ ] Small Outline Plastic with conditioning
[ ] Shrink Small Outline Plastic with conditioning

Conditioning option: [ ] Standard (Tube) [ ] Tape & Reel

Temperature Range: [ ] 0°C to + 70°C [ ] - 40°C to + 85°C
[ ] - 40°C to + 125°C

Marking: [ ] Standard marking
[ ] Special marking (ROM only):
PDIP28 (10 char. max): _ _ _ _ _ _ _ _ _ _
PSO28 (8 char. max): _ _ _ _ _ _ _ _
SSOP28 (11 char. max): _ _ _ _ _ _ _ _ _ _ _

Authorized characters are letters, digits, '.', '-', '/' and spaces only.

Oscillator Safeguard: [ ] Enabled [ ] Disabled
Watchdog Selection: [ ] Software Activation [ ] Hardware Activation
Timer pull-up: [ ] Enabled [ ] Disabled
NMI pull-up: [ ] Enabled [ ] Disabled
Oscillator Selection: [ ] Quartz crystal / Ceramic resonator
[ ] RC network

Readout Protection: FASTROM:
[ ] Enabled [ ] Disabled
ROM:
[ ] Enabled:
[ ] Fuse is blown by STMicroelectronics
[ ] Fuse can be blown by the customer
[ ] Disabled

Low Voltage Detector: [ ] Enabled [ ] Disabled
External STOP Mode Control: [ ] Enabled [ ] Disabled
Identifier (FASTROM only): [ ] Enabled [ ] Disabled

Comments:

Oscillator Frequency in the application: . . . . . .
Supply Operating Range in the application: . . . . . .
Notes: . . . . . .
Date: . . . . . .
Signature: . . . . . .