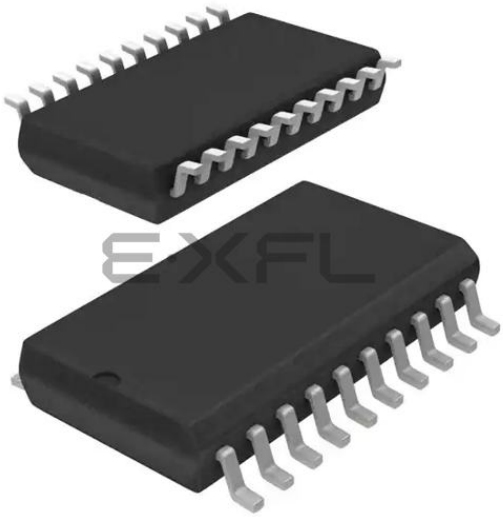


Welcome to E-XFL.COM

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	12
Program Memory Size	2KB (2K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	64 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 6V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SO
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st62t10cm6

3.3 OPTION BYTES

Each device is available for production in user programmable versions (OTP) as well as in factory coded versions (ROM). OTP devices are shipped to customers with a default content (00h), while ROM factory coded parts contain the code supplied by the customer. This implies that OTP devices have to be configured by the customer using the Option Bytes while the ROM devices are factory-configured.

The two option bytes allow the hardware configuration of the microcontroller to be selected.
The option bytes have no address in the memory map and can be accessed only in programming mode (for example using a standard ST6 programming tool).
In masked ROM devices, the option bytes are fixed in hardware by the ROM code (see Section 11.6.2 "ROM VERSION" on page 98). It is therefore impossible to read the option bytes.

The option bytes can be only programmed once. It is not possible to change the selected options after they have been programmed.

In order to reach the power consumption value indicated in Section 10.4, the option byte must be programmed to its default value. Otherwise, an over-consumption will occur.

MSB OPTION BYTE

Bits 15:10 = **Reserved**, must be always cleared.

Bit 9 = **EXTCNTL** *External STOP MODE control.*
0: EXTCNTL mode not available. STOP mode is not available with the watchdog active.
1: EXTCNTL mode available. STOP mode is available with the watchdog active by setting NMI pin to one.

Bit 8 = **LVD** *Low Voltage Detector* on/off.
This option bit enable or disable the Low Voltage Detector (LVD) feature.
0: Low Voltage Detector disabled
1: Low Voltage Detector enabled

LSB OPTION BYTE

Bit 7 = **PROTECT** *Readout Protection.*
This option bit enables or disables external access to the internal program memory.
0: Program memory not read-out protected
1: Program memory read-out protected

Bit 6 = **OSC** *Oscillator selection*.
This option bit selects the main oscillator type.
0: Quartz crystal, ceramic resonator or external clock
1: RC network

Bit 5 = **Reserved**, must be always cleared.

Bit 4 = **Reserved**, must be always set.

Bit 3 = **NMI PULL** *NMI Pull-Up* on/off.
This option bit enables or disables the internal pull-up on the NMI pin.
0: Pull-up disabled
1: Pull-up enabled

Bit 2 = **TIM PULL** *TIMER Pull-Up* on/off.
This option bit enables or disables the internal pull-up on the TIMER pin.
0: Pull-up disabled
1: Pull-up enabled

Bit 1 = **WDACT** *Hardware or software watchdog.*
This option bit selects the watchdog type.
0: Software (watchdog to be enabled by software)
1: Hardware (watchdog always enabled)

Bit 0 = **OSGEN** *Oscillator Safeguard* on/off.
This option bit enables or disables the oscillator Safeguard (OSG) feature.
0: Oscillator Safeguard disabled
1: Oscillator Safeguard enabled

	MSB OPTION BYTE								LSB OPTION BYTE							
	15							8	7							0
	Reserved						EXT CTL	LVD	PRO-TECT	OSC	Res.	Res.	NMI PULL	TIM PULL	WD ACT	OSG EN
Default Value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

CLOCK SYSTEM (Cont'd)

5.1.1 Main Oscillator

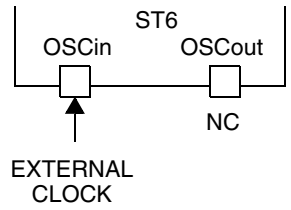
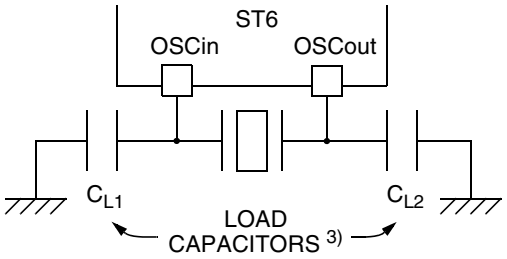
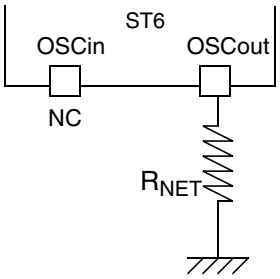
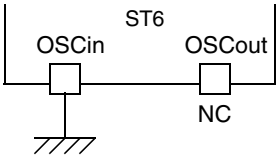
The oscillator configuration is specified by selecting the appropriate option in the option bytes (refer to the Option Bytes section of this document). When the CRYSTAL/RESONATOR option is selected, it must be used with a quartz crystal, a ceramic resonator or an external signal provided on the OSCin pin. When the RC NETWORK option is selected, the system clock is generated by an external resistor (the capacitor is implemented internally).

The main oscillator can be turned off (when the OSG ENABLED option is selected) by setting the OSCOFF bit of the ADC Control Register (not available on some devices). This will automatically start the Low Frequency Auxiliary Oscillator (LFAO).

The main oscillator can be turned off by resetting the OSCOFF bit of the A/D Converter Control Register or by resetting the MCU. When the main oscillator starts there is a delay made up of the oscillator start-up delay period plus the duration of the software instruction at a clock frequency f_{LFAO}.

Caution: It should be noted that when the RC network option is selected, the accuracy of the frequency is about 20% so it may not be suitable for some applications (For more details, please refer to the Electrical Characteristics Section).

Table 6. Oscillator Configurations

	Hardware Configuration
Crystal/Resonator Option 1)	**External Clock** 
Crystal/Resonator Option 1)	**Crystal/Resonator Clock 2)** 
RC Network Option 1)	**RC Network** 
OSG Enabled Option 1)	**LFAO** 

Notes:
1. To select the options shown in column 1 of the above table, refer to the Option Byte section.

2.This schematic are given for guidance only and are subject to the schematics given by the crystal or ceramic resonator manufacturer.

3. For more details, please refer to the Electrical Characteristics Section.

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:

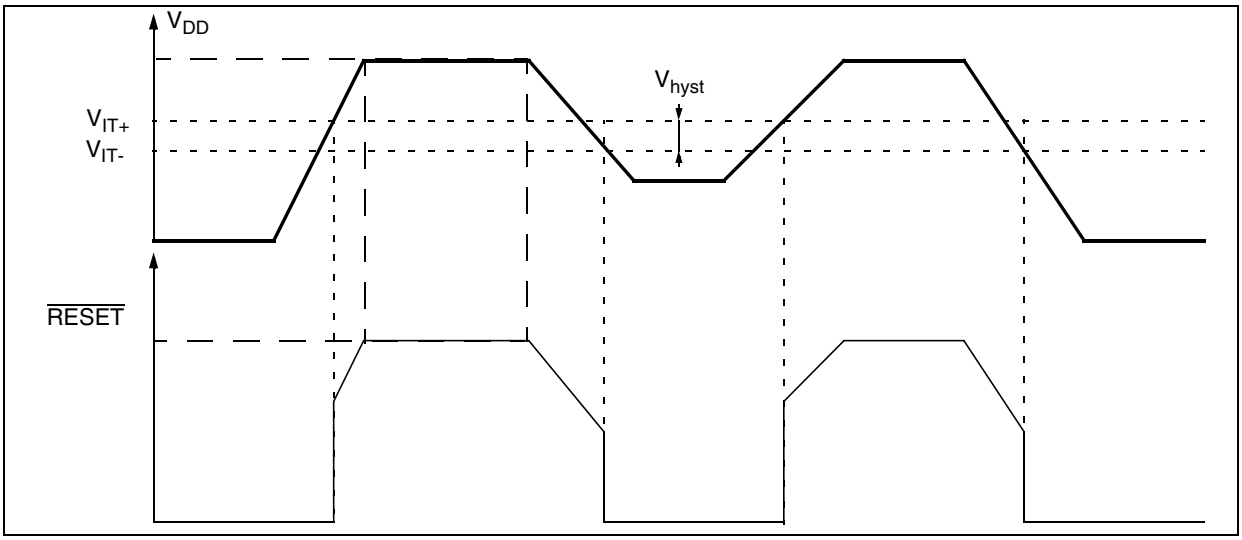
– Over the input threshold voltage, it is running under full software control

– 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



5.10 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.

5.10.1 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

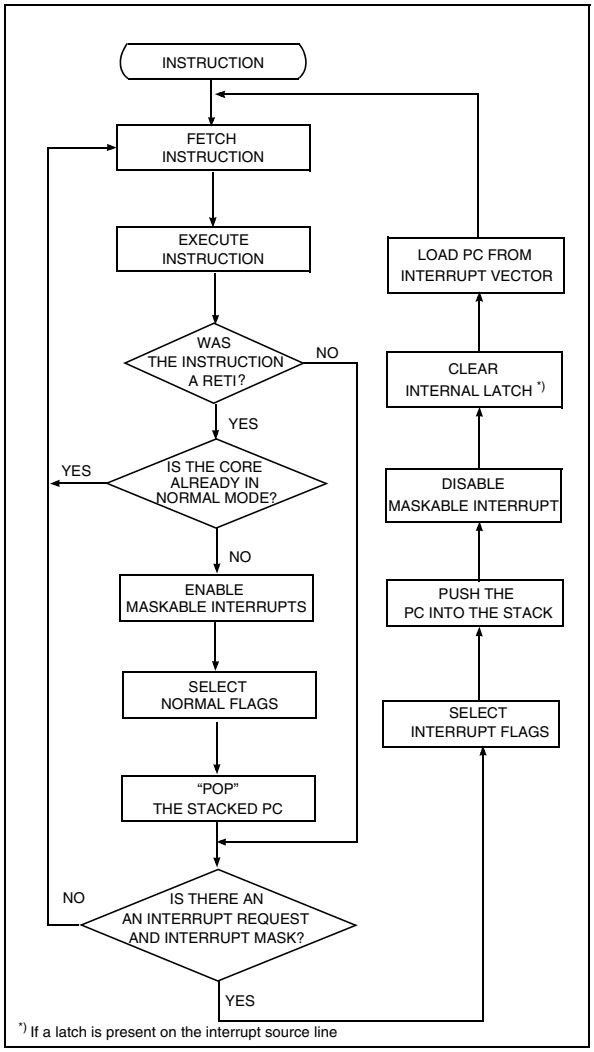


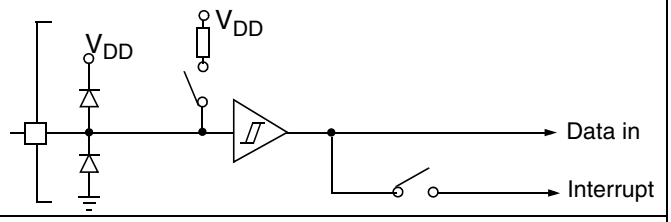
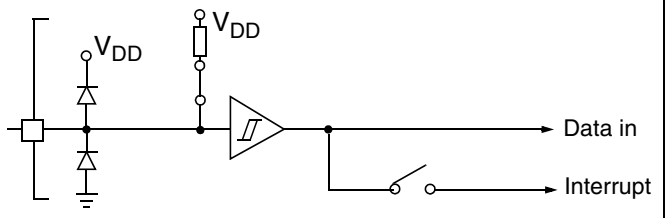
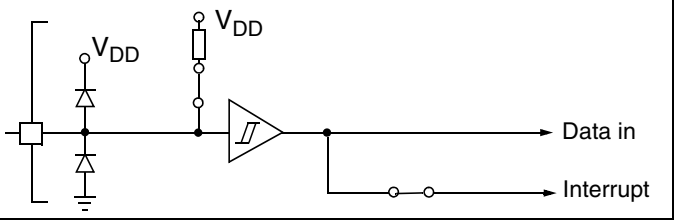
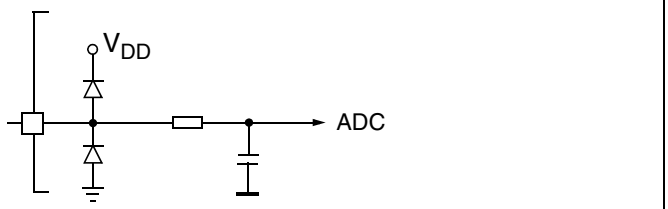
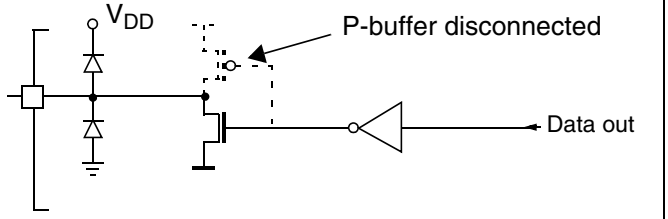
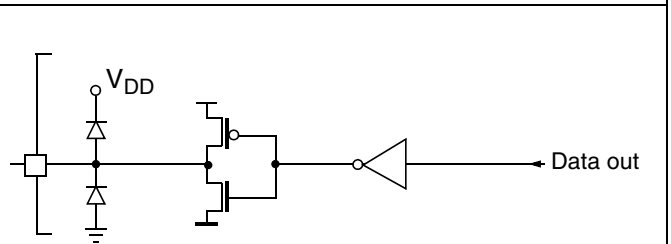
Table 7. Interrupt Response Time

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.

I/O PORTS (Cont'd)

Table 10. I/O Port Option Selections

	MODE				AVAILABLE ON(1)	SCHEMATIC
Digital Input	Input	DDRx 0	ORx 0	DRx 1	PA0-PA3 PB0-PB7	
	Reset state Input with pull up	DDRx 0	ORx 0	DRx 0	PA0-PA3 PB0-PB7	
	Input with pull up with interrupt	DDRx 0	ORx 1	DRx 0	PA0-PA3 PB0-PB7	
Analog Input	Analog Input	DDRx 0	ORx 1	DRx 1	PB0-PB3 (ST6210C/20C only) PB4-PB7 (All devices, except ST6208C)	
Digital output	Open drain output (5mA)				PB0-PB7	
	Open drain output (20 mA)	DDRx 1	ORx 0	DRx 0/1	PA0-PA3	
	Push-pull output (5mA)				PB0-PB7	
	Push-pull output (20 mA)	DDRx 1	ORx 1	DRx 0/1	PA0-PA3	

Note 1. Provided the correct configuration has been selected (see Table 9).

8 ON-CHIP PERIPHERALS

8.1 WATCHDOG TIMER (WDG)

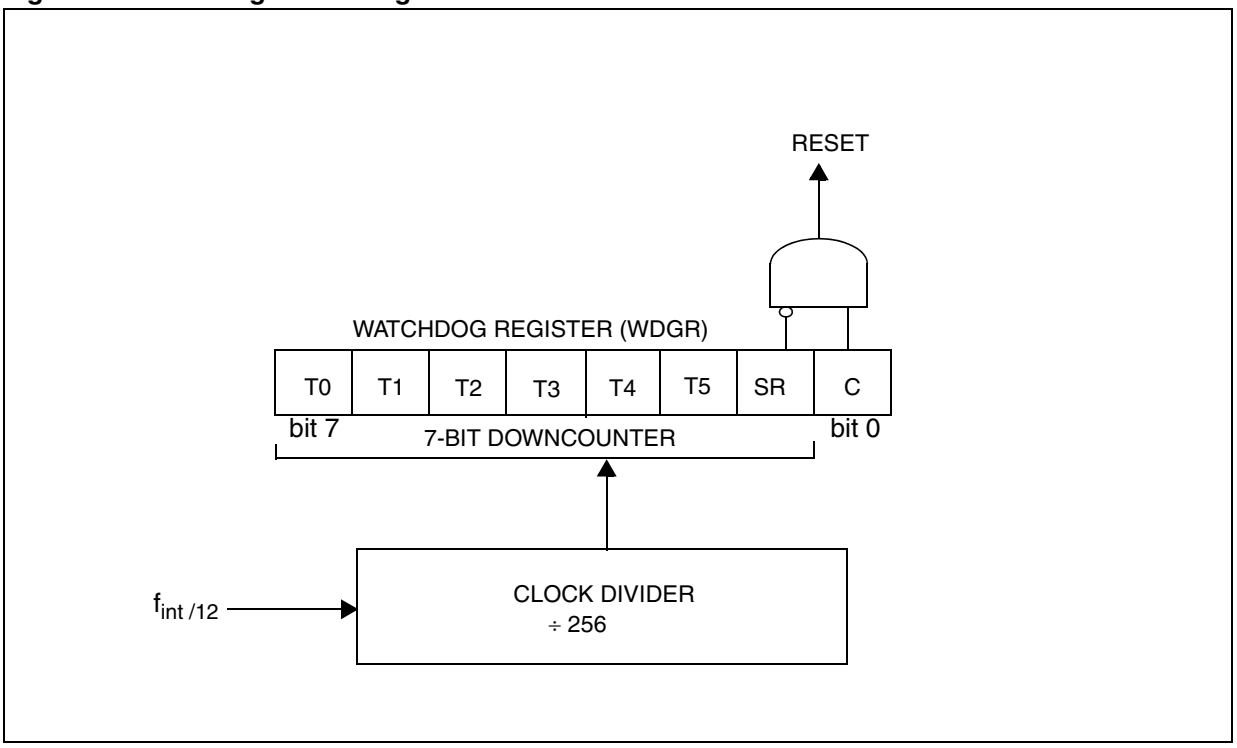
8.1.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the SR bit becomes cleared.

8.1.2 Main Features

- Programmable timer (64 steps of 3072 clock cycles)
- Software reset
- Reset (if watchdog activated) when the SR bit reaches zero
- Hardware or software watchdog activation selectable by option bit (Refer to the option bytes section)

Figure 25. Watchdog Block Diagram



WATCHDOG TIMER (Cont'd)

8.1.7 Register Description

WATCHDOG REGISTER (WDGR)

Address: 0D8h - Read/Write

Reset Value: 1111 1110 (FEh)

7							0
T0	T1	T2	T3	T4	T5	SR	C

Bits 7:2 = **T[5:0]** *Downcounter bits*
Caution: These bits are reversed and shifted with respect to the physical counter: bit-7 (T0) is the LSB of the Watchdog downcounter and bit-2 (T5) is the MSB.

Bit 1 = **SR**: *Software Reset bit*
Software can generate a reset by clearing this bit while the C bit is set. When C = 0 (Watchdog deactivated) the SR bit is the MSB of the 7-bit timer.
0: Generate (write)

1: No software reset generated, MSB of 7-bit timer

Bit 0 = **C** *Watchdog Control bit*.
If the hardware option is selected (WDACT bit in Option byte), this bit is forced high and cannot be changed by the user (the Watchdog is always active). When the software option is selected (WDACT bit in Option byte), the Watchdog function is activated by setting the C bit, and cannot then be deactivated (except by resetting the MCU).

When C is kept cleared the counter can be used as a 7-bit timer.
0: Watchdog deactivated
1: Watchdog activated

8-BIT TIMER (Cont'd)

8.2.4.2 Event Counter Mode

(TOUT = "0", DOUT = "0")

In this mode, the TIMER pin is the input clock of the Timer prescaler which is decremented on every rising edge of the input clock (allowing event count). See Figure 30 and Figure 31.

This mode is selected by clearing the TOUT bit in the TSCR register (i.e. as input) and clearing the DOUT bit.

Note: In this mode, if the TIMER pin is multiplexed, the corresponding port control bits have to be set in input with pull-up configuration.

Figure 30. f_{TIMER} Clock in Event Counter Mode

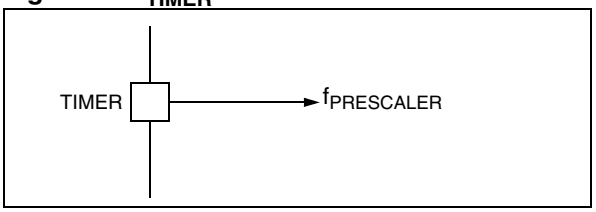
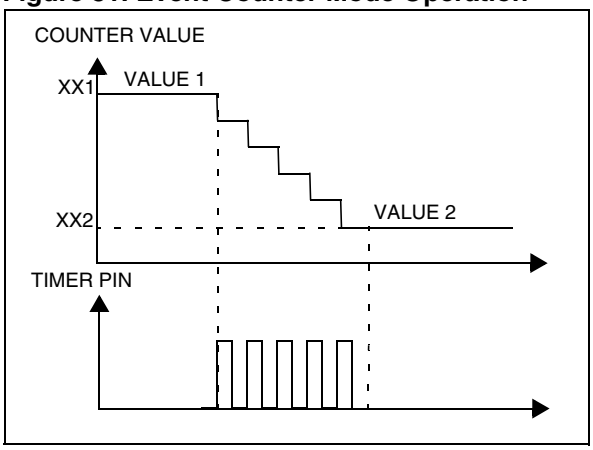


Figure 31. Event Counter Mode Operation



8.2.4.3 Output Mode

(TOUT = "1", DOUT = "data out")

In Output mode, the TIMER pin is connected to the DOUT latch, hence the Timer prescaler is clocked by the prescaler clock input (f_{INT}/12). See Figure 32.

The user can select the prescaler division ratio using the PS[2:0] bits in the TSCR register. When TCR decrements to zero, it sets the TMZ bit in the TSCR. The TMZ bit can be tested under program control to perform a timer function whenever it goes high and has to be cleared by the user. The low-to-high TMZ bit transition is used to latch the DOUT bit in the TSCR and, if the TOUT bit is set, DOUT is transferred to the TIMER pin. This operating mode allows external signal generation on the TIMER pin. See Figure 33.

This mode is selected by setting the TOUT bit in the TSCR register (i.e. as output) and setting the DOUT bit to output a high level or clearing the DOUT bit to output a low level.

Note: As soon as the TOUT bit is set, The timer pin is configured as output push-pull regardless of the corresponding I/O port control registers setting (if the TIMER pin is multiplexed).

Figure 32. Output Mode Control

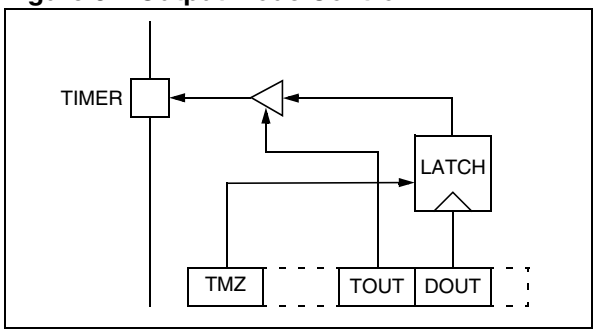
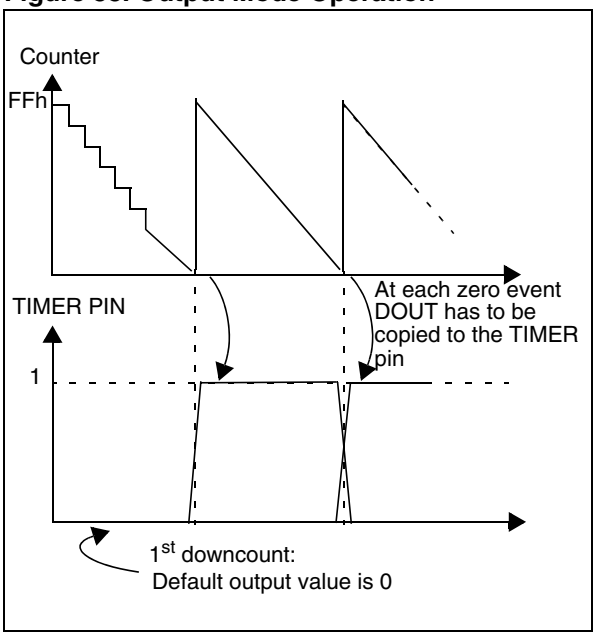


Figure 33. Output Mode Operation



8-BIT TIMER (Cont'd)

8.2.5 Low Power Modes

Mode	Description
WAIT	No effect on timer. Timer interrupt events cause the device to exit from WAIT mode.
STOP	Timer registers are frozen except in Event Counter mode (with external clock on TIMER pin).

8.2.6 Interrupts

Interrupt Event	Event Flag	Enable Bit	Exit from Wait	Exit from Stop
Timer Zero Event	TMZ	ETI	Yes	Yes

A/D CONVERTER (Cont'd)

8.3.4 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.

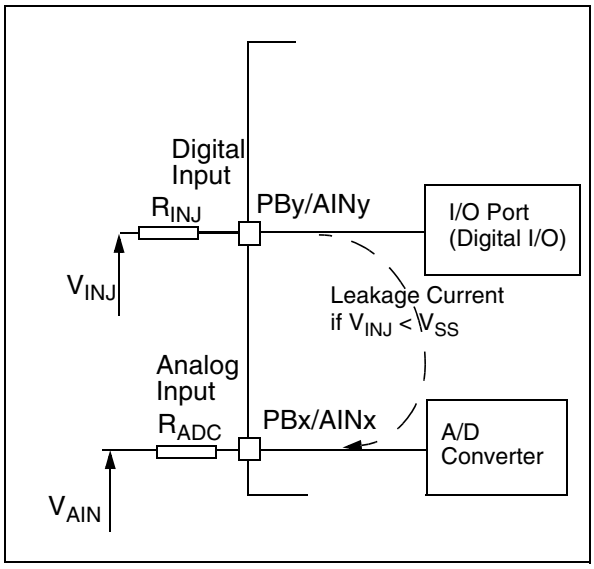
6. In order to optimize the conversion resolution, the user can configure the microcontroller in WAIT mode, because this mode minimises noise disturbances and power supply variations due to output switching. Nevertheless, the WAIT instruction should be executed as soon as possible after the beginning 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



9 INSTRUCTION SET

9.1 ST6 ARCHITECTURE

The ST6 architecture has been designed for maximum efficiency while keeping byte usage to a minimum; in short, to provide byte-efficient programming. The ST6 core has the ability to set or clear any register or RAM location bit in Data space using a single instruction. Furthermore, programs can branch to a selected address depending on the status of any bit in Data space.

9.2 ADDRESSING MODES

The ST6 has nine addressing modes, which are described in the following paragraphs. Three different address spaces are available: Program space, Data space, and Stack space. Program space contains the instructions which are to be executed, plus the data for immediate mode instructions. Data space contains the Accumulator, the X, Y, V and W registers, peripheral and Input/Output registers, the RAM locations and Data ROM locations (for storage of tables and constants). Stack space contains six 12-bit RAM cells used to stack the return addresses for subroutines and interrupts.

Immediate. In immediate addressing mode, the operand of the instruction follows the opcode location. As the operand is a ROM byte, the immediate addressing mode is used to access constants which do not change during program execution (e.g., a constant used to initialize a loop counter).

Direct. In direct addressing mode, the address of the byte which is processed by the instruction is stored in the location which follows the opcode. Direct addressing allows the user to directly address the 256 bytes in Data Space memory with a single two-byte instruction.

Short Direct. The core can address the four RAM registers X, Y, V, W (locations 80h, 81h, 82h, 83h) in short-direct addressing mode. In this case, the instruction is only one byte and the selection of the location to be processed is contained in the opcode. Short direct addressing is a subset of direct addressing mode. (Note that 80h and 81h are also indirect registers).

Extended. In extended addressing mode, the 12-bit address needed to define the instruction is obtained by concatenating the four least significant bits of the opcode with the byte following the opcode. The instructions (JP, CALL) which use extended addressing mode are able to branch to any address in the 4 Kbyte Program space.

Extended addressing mode instructions are two bytes long.

Program Counter Relative. Relative addressing mode is only used in conditional branch instructions. The instruction is used to perform a test and, if the condition is true, a branch with a span of -15 to +16 locations next to the address of the relative instruction. If the condition is not true, the instruction which follows the relative instruction is executed. Relative addressing mode instructions are one byte long. The opcode is obtained by adding the three most significant bits which characterize the test condition, one bit which determines whether it is a forward branch (when it is 0) or backward branch (when it is 1) and the four least significant bits which give the span of the branch (0h to Fh) which must be added or subtracted from the address of the relative instruction to obtain the branch destination address.

Bit Direct. In bit direct addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode points to the address of the byte in which the specified bit must be set or cleared. Thus, any bit in the 256 locations of Data space memory can be set or cleared.

Bit Test & Branch. Bit test and branch addressing mode is a combination of direct addressing and relative addressing. Bit test and branch instructions are three bytes long. The bit identification and the test condition are included in the opcode byte. The address of the byte to be tested is given in the next byte. The third byte is the jump displacement, which is in the range of -127 to +128. This displacement can be determined using a label, which is converted by the assembler.

Indirect. In indirect addressing mode, the byte processed by the register-indirect instruction is at the address pointed to by the content of one of the indirect registers, X or Y (80h, 81h). The indirect register is selected by bit 4 of the opcode. Register indirect instructions are one byte long.

Inherent. In inherent addressing mode, all the information necessary for executing the instruction is contained in the opcode. These instructions are one byte long.

INSTRUCTION SET (Cont'd)

Arithmetic and Logic. These instructions are used to perform arithmetic calculations and logic operations. In AND, ADD, CP, SUB instructions one operand is always the accumulator while, depending on the addressing mode, the other can be either a data space memory location or an immediate value. In CLR, DEC, INC instructions the operand can be any of the 256 data space addresses. In COM, RLC, SLA the operand is always the accumulator.

Table 18. Arithmetic & Logic Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
ADD A, (X)	Indirect	1	4	Δ	Δ
ADD A, (Y)	Indirect	1	4	Δ	Δ
ADD A, rr	Direct	2	4	Δ	Δ
ADDI A, #N	Immediate	2	4	Δ	Δ
AND A, (X)	Indirect	1	4	Δ	Δ
AND A, (Y)	Indirect	1	4	Δ	Δ
AND A, rr	Direct	2	4	Δ	Δ
ANDI A, #N	Immediate	2	4	Δ	Δ
CLR A	Short Direct	2	4	Δ	Δ
CLR r	Direct	3	4	*	*
COM A	Inherent	1	4	Δ	Δ
CP A, (X)	Indirect	1	4	Δ	Δ
CP A, (Y)	Indirect	1	4	Δ	Δ
CP A, rr	Direct	2	4	Δ	Δ
CPI A, #N	Immediate	2	4	Δ	Δ
DEC X	Short Direct	1	4	Δ	*
DEC Y	Short Direct	1	4	Δ	*
DEC V	Short Direct	1	4	Δ	*
DEC W	Short Direct	1	4	Δ	*
DEC A	Direct	2	4	Δ	*
DEC rr	Direct	2	4	Δ	*
DEC (X)	Indirect	1	4	Δ	*
DEC (Y)	Indirect	1	4	Δ	*
INC X	Short Direct	1	4	Δ	*
INC Y	Short Direct	1	4	Δ	*
INC V	Short Direct	1	4	Δ	*
INC W	Short Direct	1	4	Δ	*
INC A	Direct	2	4	Δ	*
INC rr	Direct	2	4	Δ	*
INC (X)	Indirect	1	4	Δ	*
INC (Y)	Indirect	1	4	Δ	*
RLC A	Inherent	1	4	Δ	Δ
SLA A	Inherent	2	4	Δ	Δ
SUB A, (X)	Indirect	1	4	Δ	Δ
SUB A, (Y)	Indirect	1	4	Δ	Δ
SUB A, rr	Direct	2	4	Δ	Δ
SUBI A, #N	Immediate	2	4	Δ	Δ

Notes:
X,Y Index Registers
V, W Short Direct Registers
Δ Affected
Immediate data (stored in ROM memory)
* Not Affected
rr Data space register

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

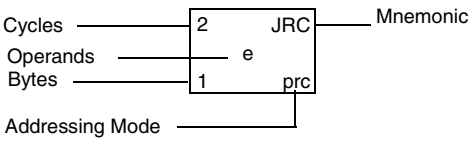
LOW / HI	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	2 NOP #	2 JRC e 1 prc	4 LD a,(x) 1 ind	0 0000
1 0001	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRS b0,rr,ee 3 bt	2 JRZ e 1 pcr	4 INC 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



10 ELECTRICAL CHARACTERISTICS

10.1 PARAMETER CONDITIONS

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

10.1.1 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).

10.1.2 Typical Values

Unless otherwise specified, typical data are based on T_A=25°C, V_{DD}=5V (for the 4.5V≤V_{DD}≤6.0V 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.

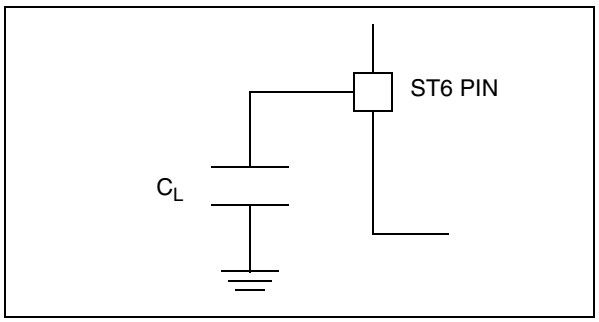
10.1.3 Typical Curves

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

10.1.4 Loading Capacitor

The loading conditions used for pin parameter measurement is shown in Figure 36.

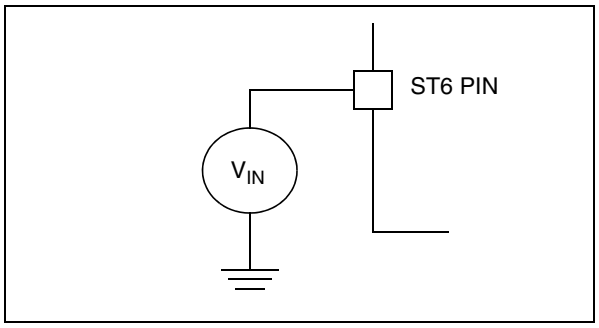
Figure 36. Pin Loading Conditions



10.1.5 Pin Input Voltage

The input voltage measurement on a pin of the device is described in Figure 37.

Figure 37. Pin Input Voltage



CLOCK AND TIMING CHARACTERISTICS (Cont'd)

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

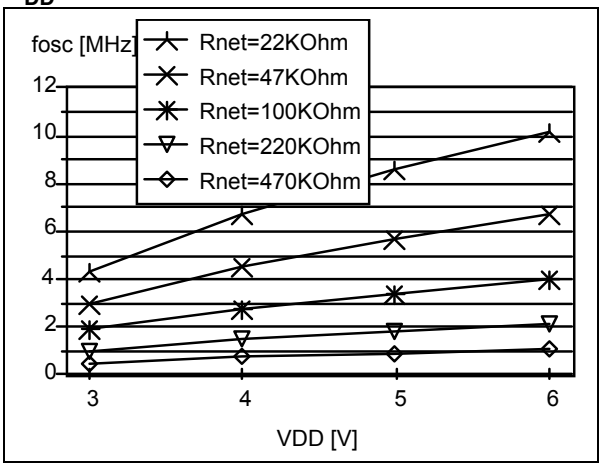
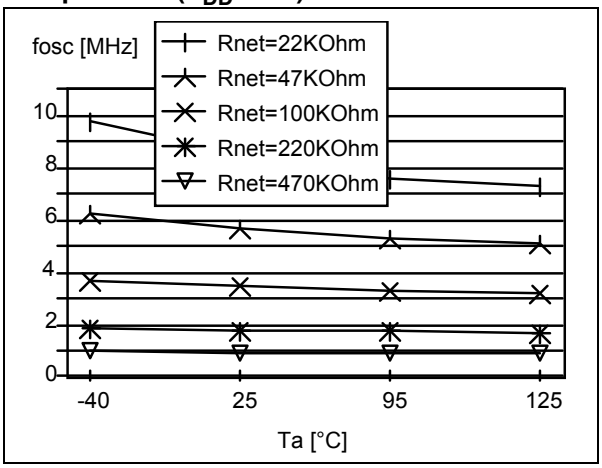


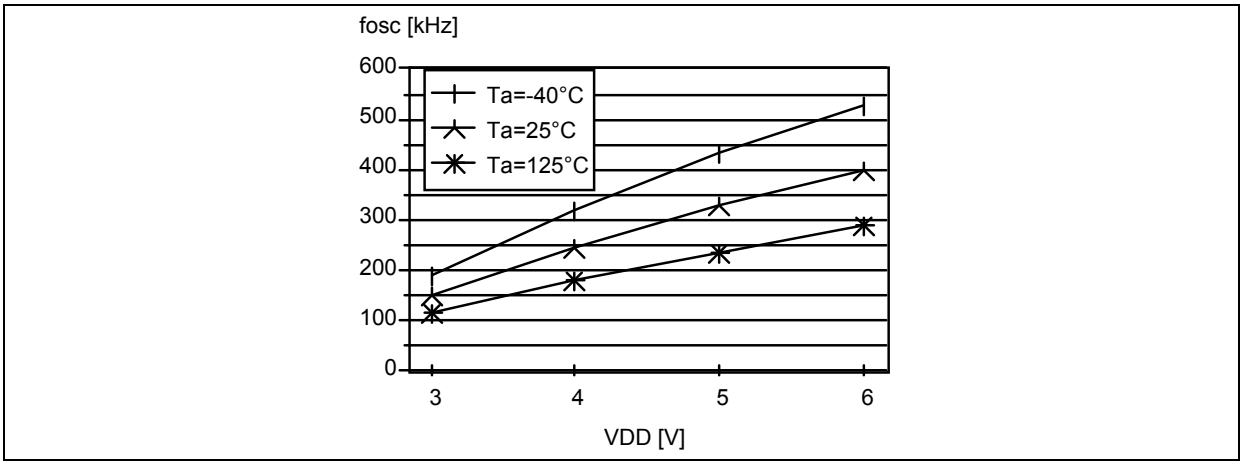
Figure 53. Typical RC Oscillator frequency vs. Temperature (V_{DD} = 5V)



10.5.5 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			

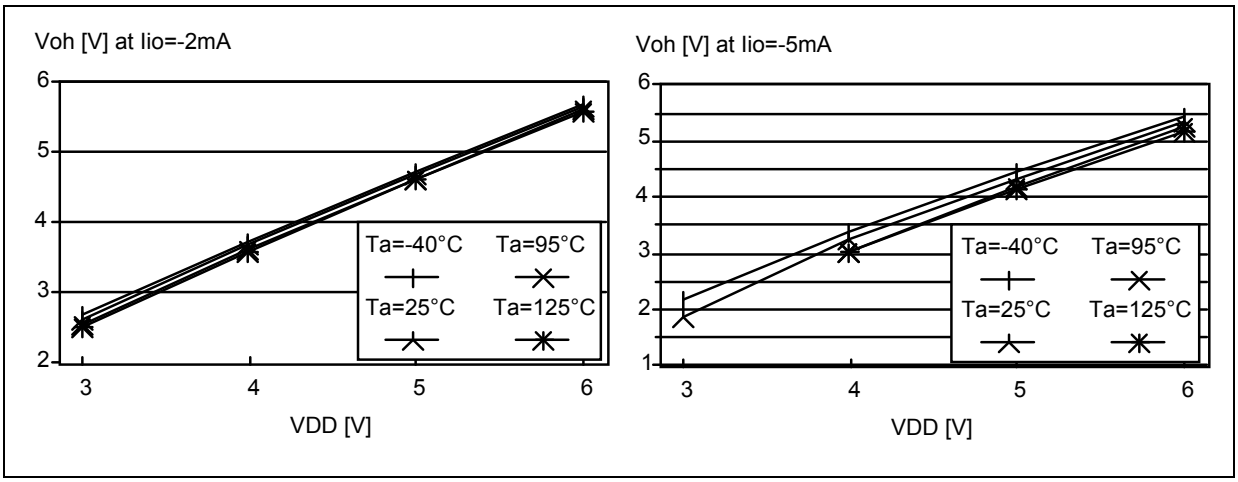
Figure 54. Typical LFAO Frequencies



Note:
1. Data based on characterization results.

I/O PORT PIN CHARACTERISTICS (Cont'd)

Figure 68. Typical V_{OH} vs V_{DD}



10.11 8-BIT ADC CHARACTERISTICS

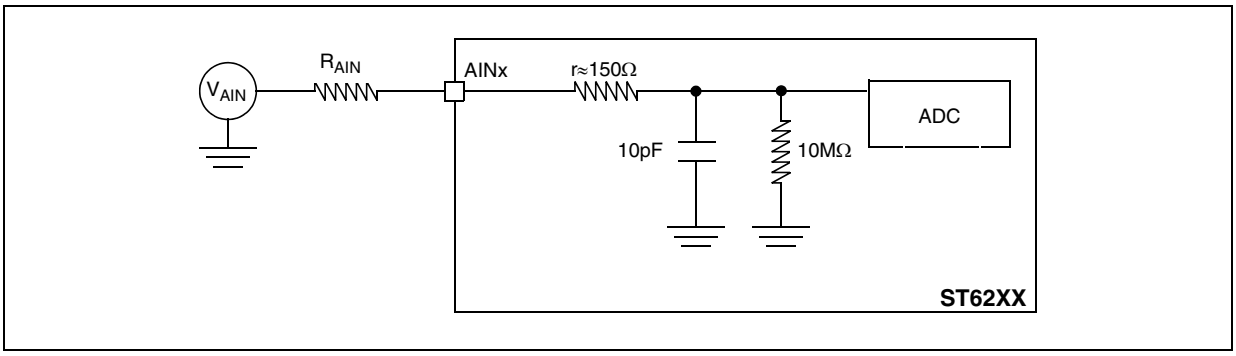
Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
f_{OSC}	Clock frequency		1.2		f_{OSC}	MHz
V_{AIN}	Conversion range voltage [2]		V_{SS}		V_{DD}	V
R_{AIN}	External input resistor				10 [3]	kΩ
t_{ADC}	Total convertion time	f_{OSC}=8MHz f_{OSC}=4MHz		70 140		µs
t_{STAB}	Stabilization time [4]			2	4	t_{CPU}
		f_{OSC}=8MHz		3.25	6.5	µs
AD_I	Analog input current during conversion				1.0	µA
AC_{IN}	Analog input capacitance			2	5	pF

Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}=5V.
2. The ADC refers to V_{DD} and V_{SS}.
3. Any added external serial resistor will downgrade the ADC accuracy (especially for resistance greater than 10kΩ). Data based on characterization results, not tested in production.
4. As a stabilization time for the AD converter is required, the first conversion after the enable can be wrong.

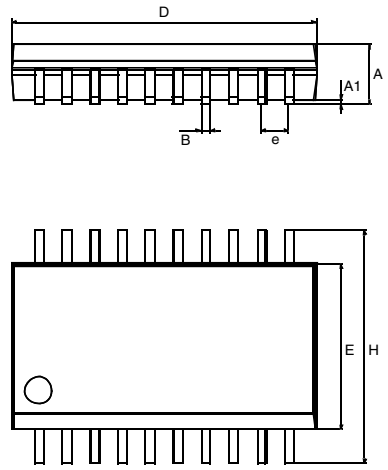
Figure 72. Typical Application with ADC



Note: ADC not present on some devices. See device summary on page 1.

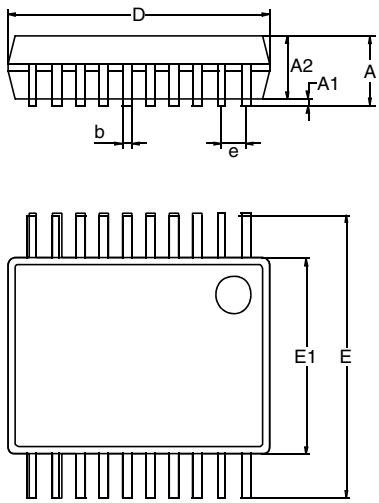
PACKAGE MECHANICAL DATA (Cont'd)

Figure 76. 20-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	12.60		13.00	0.496		0.512
E	7.40		7.60	0.291		0.299
e		1.27			0.050	
H	10.00		10.65	0.394		0.419
h	0.25		0.75	0.010		0.030
α	0°		8°	0°		8°
L	0.40		1.27	0.016		0.050
	Number of Pins					
N	20					

Figure 77. 20-Pin Plastic Shrink Small Outline Package



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			2.00			0.079
A1	0.05			0.002		
A2	1.65	1.75	1.85	0.065	0.069	0.073
b	0.22		0.38	0.009		0.015
c	0.09		0.25	0.004		0.010
D	6.90	7.20	7.50	0.272	0.283	0.295
E	7.40	7.80	8.20	0.291	0.307	0.323
E1	5.00	5.30	5.60	0.197	0.209	0.220
e		0.65			0.026	
θ	0°	4°	8°	0°	4°	8°
L	0.55	0.75	0.95	0.022	0.030	0.037
	Number of Pins					
N	20					

12 DEVELOPMENT TOOLS

STMicroelectronics offers a range of hardware and software development tools for the ST6 microcontroller family. Full details of tools available for the ST6 from third party manufacturers can be obtain from the STMicroelectronics Internet site: ➟ http://www.st.com.

Table 26. Dedicated Third Parties Development Tools

Third Party [1)]	Designation	ST Sales Type	Web site address
ACTUM	ST-REALIZER II: Graphical Schematic based Development available from STMicroelectronics.	STREALIZER-II	http://www.actum.com/
CEIBO	Low cost emulator available from CEIBO.		http://www.ceibo.com/
RAISONANCE	This tool includes in the same environment: an assembler, linker, C compiler, debugger and simulator. The assembler package (plus limited C compiler) is free and can be downloaded from raisonance web site. The full version is available both from STMicroelectronics and Raisonance.	ST6RAIS-SWC/PC	http://www.raisonance.com/
SOFTEC	High end emulator available from SOFTEC.		http://www.softecmicro.com/
	Gang programmer available from SOFTEC.		
ADVANCED EQUIPMENT	Single and gang programmers		http://www.aec.com.tw/
ADVANCED TRANSDATA			http://www.adv-transdata.com/
BP MICROSYSTEMS			http://www.bpmicro.com/
DATA I/O			http://www.data-io.com/
DATAMAN			http://www.dataman.com/
EE TOOLS			http://www.eetools.com/
ELNEC			http://www.elnec.com/
HI-LO SYSTEMS			http://www.hilosystems.com.tw/
ICE TECHNOLOGY			http://www.icetech.com/
LEAP			http://www.leap.com.tw/
LLOYD RESEARCH			http://www.lloyd-research.com/
LOGICAL DEVICES			http://www.chipprogrammers.com/
MQP ELECTRONICS			http://www.mqp.com/
NEEDHAMS ELECTRONICS			http://www.needhams.com/
STAG PROGRAMMERS			http://www.stag.co.uk/
SYSTEM GENERAL CORP			http://www.sg.com.tw
TRIBAL MICROSYSTEMS			http://www.tribalmicro.com/
XELTEK			http://www.xeltek.com/

Note 1: For latest information on third party tools, please visit our Internet site: ➟ http://www.st.com.

DEVELOPMENT TOOLS (Cont'd)

STMicroelectronics Tools

Four types of development tool are offered by ST, all of them connect to a PC via a parallel or serial port: see Table 27 and Table 28 for more details.

Table 27. STMicroelectronics Tool Features

	Emulation Type	Programming Capability	Software Included
ST6 Starter Kit	Device simulation (limited emulation as interrupts are not supported)	Yes (DIP packages only)	MCU CD ROM with: – Rkit-ST6 from Raisonance – ST6 Assembly toolchain – WGDB6 powerful Source Level Debugger for Win 3.1, Win 95 and NT – Various software demo versions. – Windows Programming Tools for Win 3.1, Win 95 and NT
ST6 HDS2 Emulator	In-circuit powerful emulation features including trace/ logic analyzer	No	
ST6 EPROM Programmer Board	No	Yes	

Table 28. Dedicated STMicroelectronics Development Tools

Supported Products	ST6 Starter Kit	ST6 HDS2 Emulator	ST6 Programming Board
ST6208C, ST6209C, ST6210C and ST6220C	ST622XC-KIT	Complete: ST62GP-EMU2 Dedication board: ST62GP-DBE	ST62E2XC-EPB