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	SPI
Peripherals	LED, LVD, POR, WDT
Number of I/O	13
Program Memory Size	3.8KB (3.8K x 8)
Program Memory Type	OTP
EEPROM Size	128 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 6V
Data Converters	A/D 7x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st62t60cb6

MEMORY MAP (Cont'd)

1.3.3 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.

1.3.3.1 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.

1.3.3.2 Data RAM/EEPROM

In ST62T53C, T60C, T63C and ST62E60C devices, the data space includes 60 bytes of RAM, 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 (DRW register).

Additional RAM and EEPROM pages can also be addressed using banks of 64 bytes located between addresses 00h and 3Fh.

1.3.4 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.

Table 1. Additional RAM/EEPROM Banks

Device	RAM	EEPROM
ST62T53C	1 x 64 bytes	-
ST62T60C/E60C	1 x 64 bytes	2 x 64 bytes
ST62T63C	1 x 64 bytes	1 x 64 bytes

Table 2ST62T53C, T60C, T63C and ST62E60C Data Memory Space

RAM and EEPROM	000h 03Fh
DATA ROM WINDOW AREA	040h 07Fh
X REGISTER	080h
Y REGISTER	081h
V REGISTER	082h
W REGISTER	083h
DATA RAM 60 BYTES	084h 0BFh
PORT A DATA REGISTER	0C0h
PORT B DATA REGISTER	0C1h
PORT C DATA REGISTER	0C2h
RESERVED	0C3h
PORT A DIRECTION REGISTER	0C4h
PORT B DIRECTION REGISTER	0C5h
PORT C DIRECTION REGISTER	0C6h
RESERVED	0C7h
INTERRUPT OPTION REGISTER	0C8h*
DATA ROM WINDOW REGISTER	0C9h*
RESERVED	0CAh 0CBh
PORT A OPTION REGISTER	0CCh
PORT B OPTION REGISTER	0CDh
PORT C OPTION REGISTER	0CEh
RESERVED	0CFh
A/D DATA REGISTER	0D0h
A/D CONTROL REGISTER	0D1h
TIMER PRESCALER REGISTER	0D2h
TIMER COUNTER REGISTER	0D3h
TIMER STATUS CONTROL REGISTER	0D4h
AR TIMER MODE CONTROL REGISTER	0D5h
AR TIMER STATUS/CONTROL REGISTER1	0D6h
AR TIMER STATUS/CONTROL REGISTER2	0D7h
WATCHDOG REGISTER	0D8h
AR TIMER RELOAD/CAPTURE REGISTER	0D9h
AR TIMER COMPARE REGISTER	0DAh
AR TIMER LOAD REGISTER	0DBh
OSCILLATOR CONTROL REGISTER	0DCh*
MISCELLANEOUS	0DDh
RESERVED	0DEh 0DFh
SPI DATA REGISTER	0E0h
SPI DIVIDER REGISTER	0E1h
SPI MODE REGISTER	0E2h
RESERVED	0E3h 0E7h
DATA RAM/EEPROM REGISTER	0E8h*
RESERVED	0E9h
EEPROM CONTROL REGISTER (except ST62T53C)	0EAh
RESERVED	0EBh 0FEh
ACCUMULATOR	0FFh

* WRITE ONLY REGISTER

MEMORY MAP (Cont'd)

Additional Notes on Parallel Mode:

If the user wishes to perform parallel programming, the first step should be to set the E2PAR2 bit. From this time on, the EEPROM will be addressed in write mode, the ROW address and the data will be latched and it will be possible to change them only at the end of the programming cycle or by resetting E2PAR2 without programming the EEPROM. After the ROW address is latched, the MCU can only "see" the selected EEPROM row and any attempt to write or read other rows will produce errors.

The EEPROM should not be read while E2PAR2 is set.

As soon as the E2PAR2 bit is set, the 8 volatile ROW latches are cleared. From this moment on, the user can load data in all or in part of the ROW. Setting E2PAR1 will modify the EEPROM registers corresponding to the ROW latches accessed after E2PAR2. For example, if the software sets E2PAR2 and accesses the EEPROM by writing to addresses 18h, 1Ah and 1Bh, and then sets E2PAR1, these three registers will be modified simultaneously; the remaining bytes in the row will be unaffected.

Note that E2PAR2 is internally reset at the end of the programming cycle. This implies that the user must set the E2PAR2 bit between two parallel programming cycles. Note that if the user tries to set E2PAR1 while E2PAR2 is not set, there will be no programming cycle and the E2PAR1 bit will be unaffected. Consequently, the E2PAR1 bit cannot be set if E2ENA is low. The E2PAR1 bit can be set by the user, only if the E2ENA and E2PAR2 bits are also set.

Notes: The EEPROM page shall not be changed through the DRBR register when the E2PAR2 bit is set.

EEPROM Control Register (EECTL)

Address: EAh — Read/Write

Reset status: 00h

7							0
D7	E2OFF	D5	D4	E2PAR1	E2PAR2	E2BUSY	E2ENA

Bit 7 = **D7**: *Unused.*

Bit 6 = **E2OFF**: *Stand-by Enable Bit.* WRITE ONLY. If this bit is set the EEPROM is disabled (any access will be meaningless) and the power consumption of the EEPROM is reduced to its lowest value.

Bit 5-4 = **D5-D4**: *Reserved.* MUST be kept reset.

Bit 3 = **E2PAR1**: *Parallel Start Bit.* WRITE ONLY. Once in Parallel Mode, as soon as the user software sets the E2PAR1 bit, parallel writing of the 8 adjacent registers will start. This bit is internally reset at the end of the programming procedure. Note that less than 8 bytes can be written if required, the undefined bytes being unaffected by the parallel programming cycle; this is explained in greater detail in the Additional Notes on Parallel Mode overleaf.

Bit 2 = **E2PAR2**: *Parallel Mode En. Bit.* WRITE ONLY. This bit must be set by the user program in order to perform parallel programming. If E2PAR2 is set and the parallel start bit (E2PAR1) is reset, up to 8 adjacent bytes can be written simultaneously. These 8 adjacent bytes are considered as a row, whose address lines A7, A6, A5, A4, A3 are fixed while A2, A1 and A0 are the changing bits, as illustrated in Table 4. E2PAR2 is automatically reset at the end of any parallel programming procedure. It can be reset by the user software before starting the programming procedure, thus leaving the EEPROM registers unchanged.

Bit 1 = **E2BUSY**: *EEPROM Busy Bit.* READ ONLY. This bit is automatically set by the EEPROM control logic when the EEPROM is in programming mode. The user program should test it before any EEPROM read or write operation; any attempt to access the EEPROM while the busy bit is set will be aborted and the writing procedure in progress will be completed.

Bit 0 = **E2ENA**: *EEPROM Enable Bit.* WRITE ONLY. This bit enables programming of the EEPROM cells. It must be set before any write to the EEPROM register. Any attempt to write to the EEPROM when E2ENA is low is meaningless and will not trigger a write cycle.

The EEPROM is disabled as soon as a STOP instruction is executed in order to achieve the lowest power-consumption.

PROGRAMMING MODES (Cont'd)

1.4.2 EPROM Erasing

The EPROM of the windowed package of the MCUs may be erased by exposure to Ultra Violet light. The erasure characteristic of the MCUs is such that erasure begins when the memory is exposed to light with a wave lengths shorter than approximately 4000Å. It should be noted that sunlights and some types of fluorescent lamps have wavelengths in the range 3000-4000Å.

It is thus recommended that the window of the MCUs packages be covered by an opaque label to prevent unintentional erasure problems when testing the application in such an environment.

The recommended erasure procedure of the MCUs EPROM is the exposure to short wave ultraviolet light which have a wave-length 2537A. The integrated dose (i.e. U.V. intensity x exposure time) for erasure should be a minimum of 15W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with 12000µW/cm^2 power rating. The ST62E60C should be placed within 2.5cm (1Inch) of the lamp tubes during erasure.

3 CLOCKS, RESET, INTERRUPTS AND POWER SAVING MODES

3.1 CLOCK SYSTEM

The MCU features a Main Oscillator which can be driven by an external clock, or used in conjunction with an AT-cut parallel resonant crystal or a suitable ceramic resonator, or with an external resistor (R_{NET}). In addition, a Low Frequency Auxiliary Oscillator (LFAO) can be switched in for security reasons, to reduce power consumption, or to offer the benefits of a back-up clock system.

The Oscillator Safeguard (OSG) option filters spikes from the oscillator lines, provides access to the LFAO to provide a backup oscillator in the event of main oscillator failure and also automatically limits the internal clock frequency (f_{INT}) as a function of V_{DD}, in order to guarantee correct operation. These functions are illustrated in Figure 10, Figure 11, Figure 12 and Figure 13.

A programmable divider on F_{INT} is also provided in order to adjust the internal clock of the MCU to the best power consumption and performance tradeoff.

Figure 9 illustrates various possible oscillator configurations using an external crystal or ceramic resonator, an external clock input, an external resistor (R_{NET}), or the lowest cost solution using only the LFAO. C_{L1} an C_{L2} should have a capacitance in the range 12 to 22 pF for an oscillator frequency in the 4-8 MHz range.

The internal MCU clock frequency (f_{INT}) is divided by 12 to drive the Timer, the A/D converter and the Watchdog timer, and by 13 to drive the CPU core, as may be seen in Figure 12.

With an 8 MHz oscillator frequency, the fastest machine cycle is therefore 1.625µs.

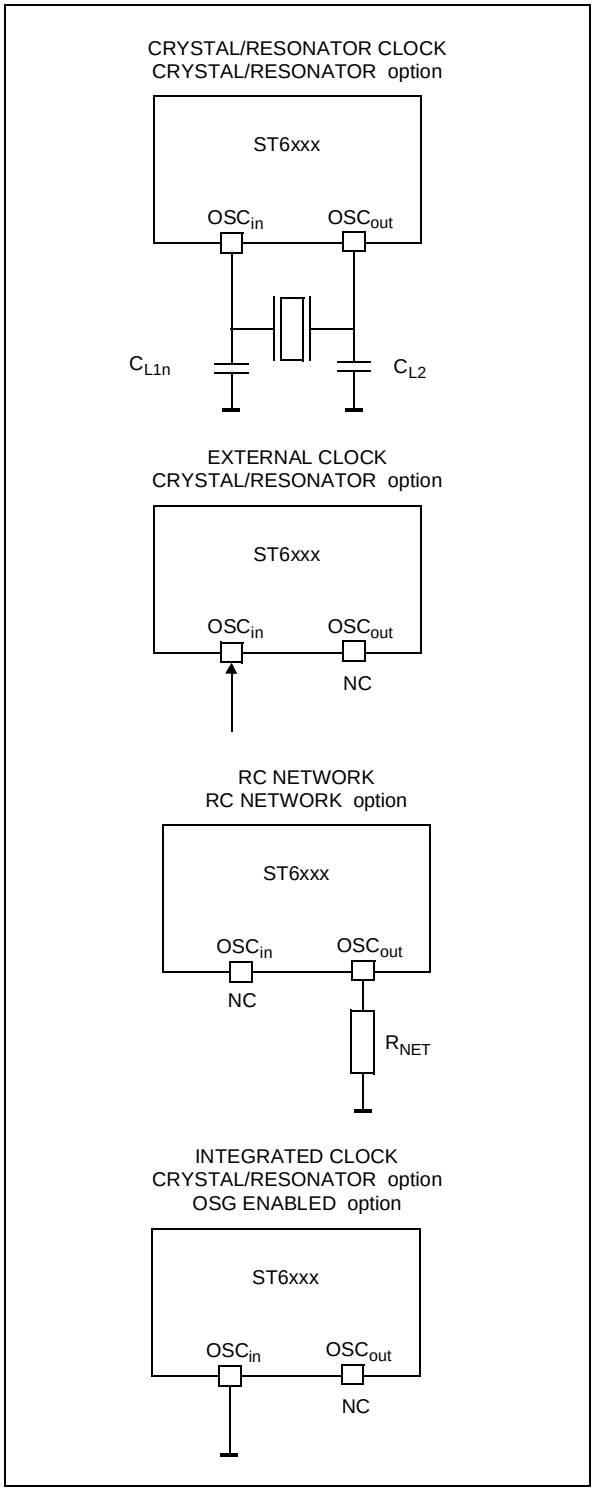
A machine cycle is the smallest unit of time needed to execute any operation (for instance, to increment the Program Counter). An instruction may require two, four, or five machine cycles for execution.

3.1.1 Main Oscillator

The oscillator configuration may be specified by selecting the appropriate option. 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 main oscillator can be turned off (when the OSG ENABLED option is selected) by setting the OSCOFF bit of the ADC Control Register. The Low Frequency Auxiliary Oscillator is automatically started.

Figure 9. Oscillator Configurations



CLOCK SYSTEM (Cont'd)

Figure 10. OSG Filtering Principle

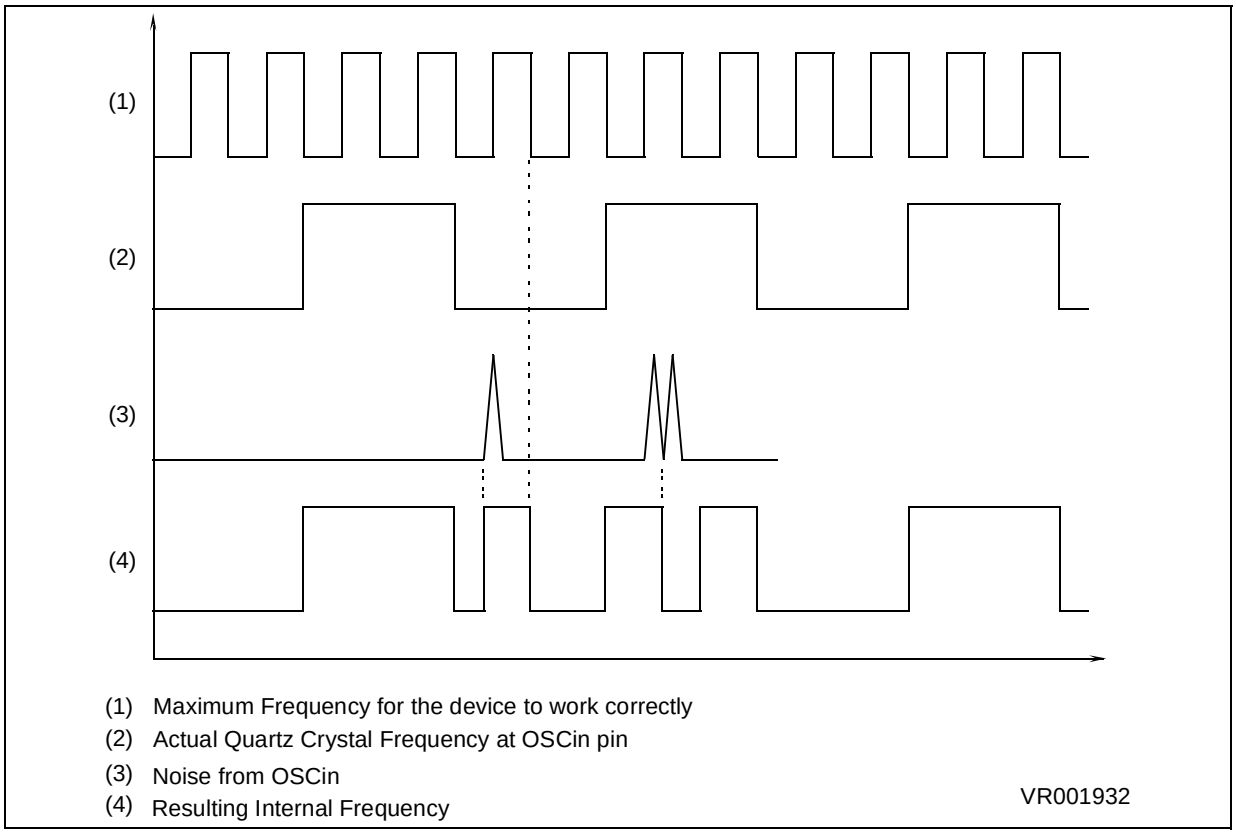
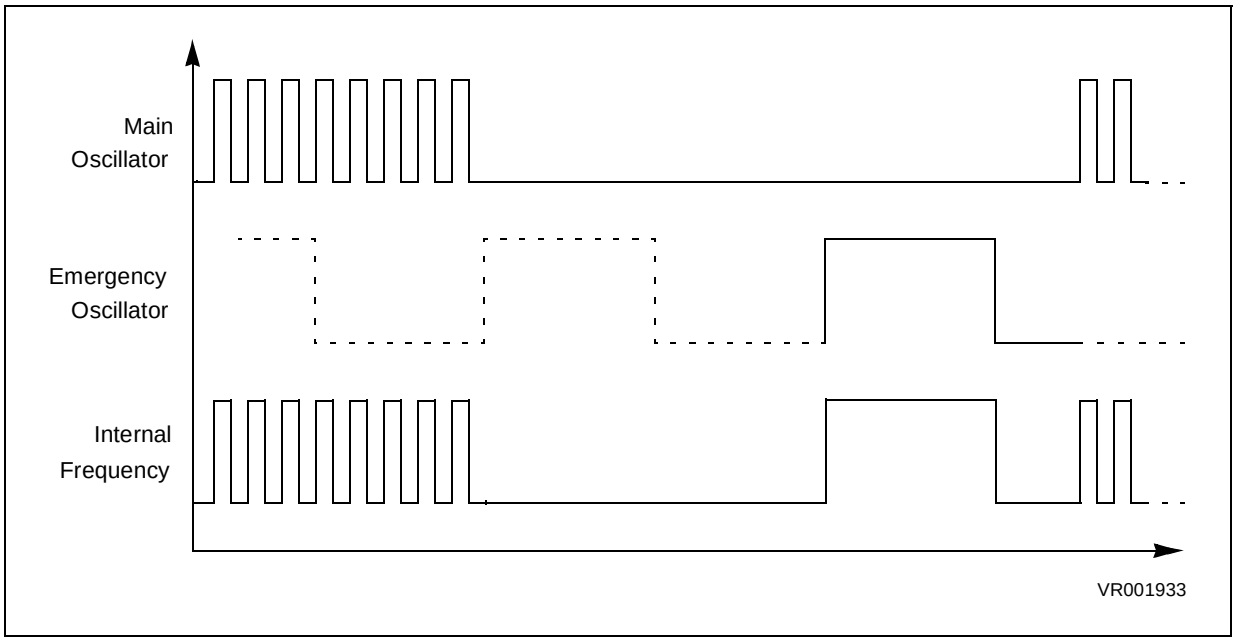


Figure 11. OSG Emergency Oscillator Principle



RESETS (Cont'd)

3.2.6 MCU Initialization Sequence

When a reset occurs the stack is reset, the PC is loaded with the address of the Reset Vector (located in program ROM starting at address 0FFEh). A jump to the beginning of the user program must be coded at this address. Following a Reset, the Interrupt flag is automatically set, so that the CPU is in Non Maskable Interrupt mode; this prevents the initialisation routine from being interrupted. The initialisation routine should therefore be terminated by a RETI instruction, in order to revert to normal mode and enable interrupts. If no pending interrupt is present at the end of the initialisation routine, the MCU will continue by processing the instruction immediately following the RETI instruction. If, however, a pending interrupt is present, it will be serviced.

Figure 16. Reset and Interrupt Processing

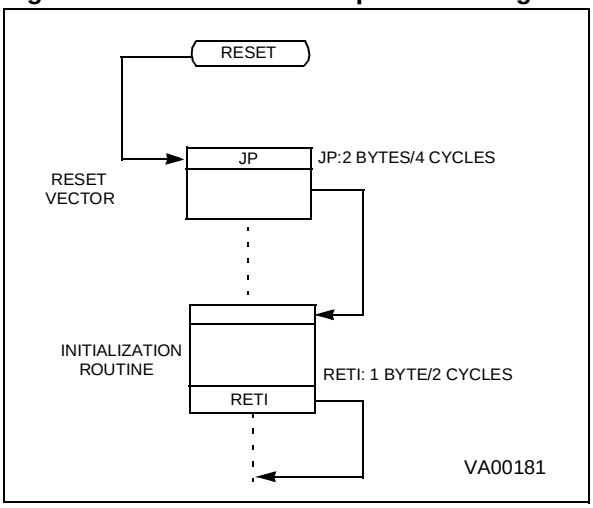
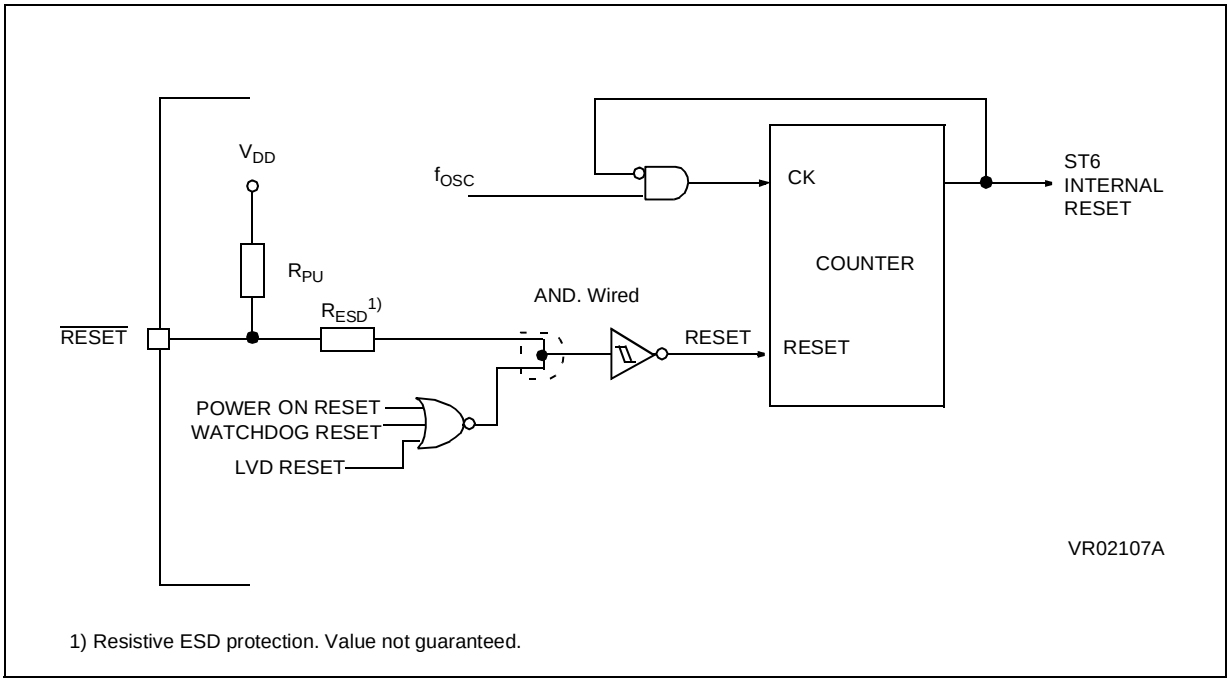


Figure 17. Reset Block Diagram



1) Resistive ESD protection. Value not guaranteed.

DIGITAL WATCHDOG (Cont'd)

3.3.1 Digital Watchdog Register (DWDR)

Address: 0D8h — Read/Write

Reset status: 1111 1110b

7							0
T0	T1	T2	T3	T4	T5	SR	C

Bit 0 = **C**: *Watchdog Control bit*

If the hardware option is selected, this bit is forced high and the user cannot change it (the Watchdog is always active). When the software option is selected, the Watchdog function is activated by setting bit C to 1, and cannot then be disabled (save by resetting the MCU).

When C is kept low the counter can be used as a 7-bit timer.

This bit is cleared to "0" on Reset.

Bit 1 = **SR**: *Software Reset bit*

This bit triggers a Reset when cleared.

When C = "0" (Watchdog disabled) it is the MSB of the 7-bit timer.

This bit is set to "1" on Reset.

Bits 2-7 = **T5-T0**: *Downcounter bits*

It should be noted that the register 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.

These bits are set to "1" on Reset.

3.3.2 Application Notes

The Watchdog plays an important supporting role in the high noise immunity of ST62xx devices, and should be used wherever possible. Watchdog related options should be selected on the basis of a trade-off between application security and STOP mode availability.

When STOP mode is not required, hardware activation without EXTERNAL STOP MODE CONTROL should be preferred, as it provides maximum security, especially during power-on.

When STOP mode is required, hardware activation and EXTERNAL STOP MODE CONTROL should be chosen. NMI should be high by default, to allow STOP mode to be entered when the MCU is idle.

The NMI pin can be connected to an I/O line (see Figure 19) to allow its state to be controlled by software. The I/O line can then be used to keep NMI low while Watchdog protection is required, or to avoid noise or key bounce. When no more processing is required, the I/O line is released and the device placed in STOP mode for lowest power consumption.

When software activation is selected and the Watchdog is not activated, the downcounter may be used as a simple 7-bit timer (remember that the bits are in reverse order).

The software activation option should be chosen only when the Watchdog counter is to be used as a timer. To ensure the Watchdog has not been unexpectedly activated, the following instructions should be executed within the first 27 instructions:

```
jrr 0, WD, #+3
ldi WD, 0FDH
```

3.4 INTERRUPTS

The CPU can manage four Maskable Interrupt sources, in addition to a Non Maskable Interrupt source (top priority interrupt). Each source is associated with a specific Interrupt Vector which contains a Jump instruction to the associated interrupt service routine. These vectors are located in Program space (see Table 7).

When an interrupt source generates an interrupt request, and interrupt processing is enabled, the PC register is loaded with the address of the interrupt vector (i.e. of the Jump instruction), which then causes a Jump to the relevant interrupt service routine, thus servicing the interrupt.

Interrupt sources are linked to events either on external pins, or on chip peripherals. Several events can be ORed on the same interrupt source, and relevant flags are available to determine which event triggered the interrupt.

The Non Maskable Interrupt request has the highest priority and can interrupt any interrupt routine at any time; the other four interrupts cannot interrupt each other. If more than one interrupt request is pending, these are processed by the processor core according to their priority level: source #1 has the higher priority while source #4 the lower. The priority of each interrupt source is fixed.

Table 7. Interrupt Vector Map

Interrupt Source	Priority	Vector Address
Interrupt source #0	1	(FFCh-FFDh)
Interrupt source #1	2	(FF6h-FF7h)
Interrupt source #2	3	(FF4h-FF5h)
Interrupt source #3	4	(FF2h-FF3h)
Interrupt source #4	5	(FF0h-FF1h)

3.4.1 Interrupt request

All interrupt sources but the Non Maskable Interrupt source can be disabled by setting accordingly the GEN bit of the Interrupt Option Register (IOR). This GEN bit also defines if an interrupt source, including the Non Maskable Interrupt source, can restart the MCU from STOP/WAIT modes.

Interrupt request from the Non Maskable Interrupt source #0 is latched by a flip flop which is automatically reset by the core at the beginning of the non-maskable interrupt service routine.

Interrupt request from source #1 can be configured either as edge or level sensitive by setting accordingly the LES bit of the Interrupt Option Register (IOR).

Interrupt request from source #2 are always edge sensitive. The edge polarity can be configured by setting accordingly the ESB bit of the Interrupt Option Register (IOR).

Interrupt request from sources #3 & #4 are level sensitive.

In edge sensitive mode, a latch is set when a edge occurs on the interrupt source line and is cleared when the associated interrupt routine is started. So, the occurrence of an interrupt can be stored, until completion of the running interrupt routine before being processed. If several interrupt requests occurs before completion of the running interrupt routine, only the first request is stored.

Storage of interrupt requests is not available in level sensitive mode. To be taken into account, the low level must be present on the interrupt pin when the MCU samples the line after instruction execution.

At the end of every instruction, the MCU tests the interrupt lines: if there is an interrupt request the next instruction is not executed and the appropriate interrupt service routine is executed instead.

Table 8. Interrupt Option Register Description

GEN	SET	Enable all interrupts
	CLEARED	Disable all interrupts
ESB	SET	Rising edge mode on interrupt source #2
	CLEARED	Falling edge mode on interrupt source #2
LES	SET	Level-sensitive mode on interrupt source #1
	CLEARED	Falling edge mode on interrupt source #1
OTHERS	NOT USED	

INTERRUPTS (Cont'd)

3.4.2 Interrupt Procedure

The interrupt procedure is very similar to a call procedure, indeed 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. There are separate sets of processor flags for normal, interrupt and non-maskable interrupt modes, which are automatically switched and so do not need to be saved.

The following list summarizes the interrupt procedure:

MCU

– The interrupt is detected.
– The C and Z flags are replaced by the interrupt flags (or by the NMI flags).
– The PC contents are stored in the first level of the stack.
– The normal interrupt lines are inhibited (NMI still active).
– The first internal latch is cleared.
– The associated interrupt vector is loaded in the PC.

WARNING: In some circumstances, when a maskable interrupt occurs while the ST6 core is in NORMAL mode and especially during the execution of an "ldi IOR, 00h" instruction (disabling all maskable interrupts): if the interrupt arrives 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.

User

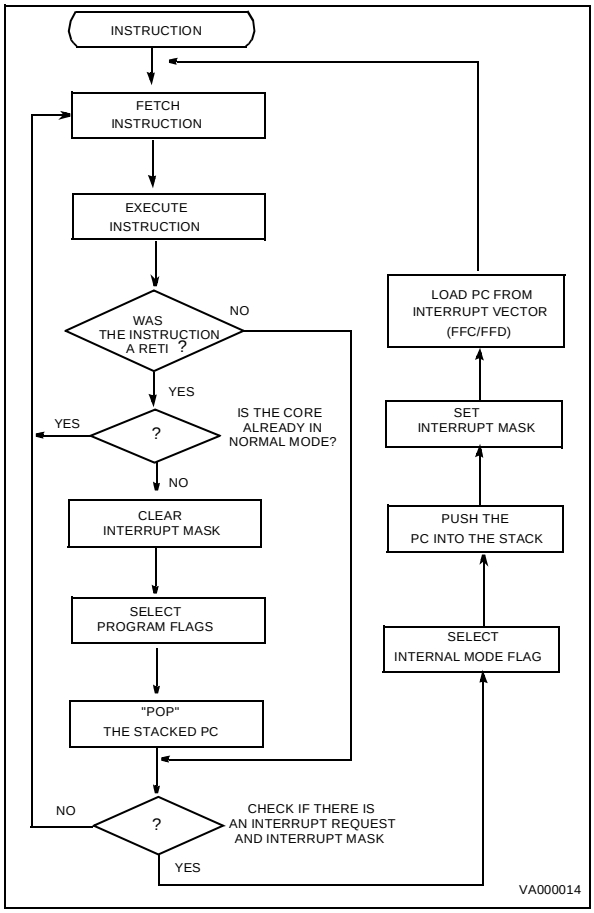
– User selected registers are saved within the interrupt service routine (normally on a software stack).
– The source of the interrupt is found by polling the interrupt flags (if more than one source is associated with the same vector).
– The interrupt is serviced.
– Return from interrupt (RETI)

MCU

– Automatically the MCU switches back to the normal flag set (or the interrupt flag set) and pops the previous PC value from the stack.

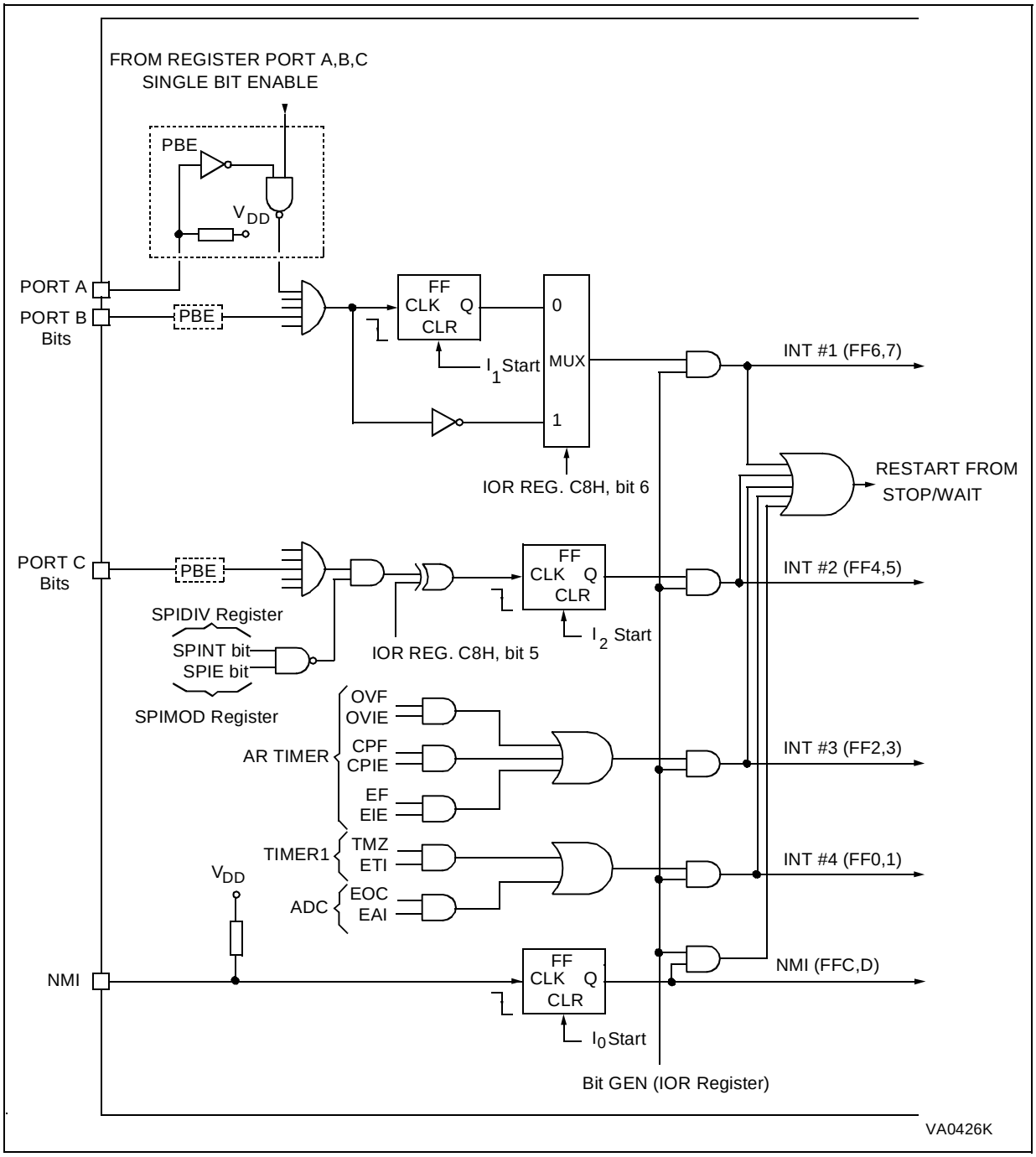
The interrupt routine usually begins by the identifying the device which generated the interrupt request (by polling). The user should save the registers which are used within the interrupt routine in a software stack. After the RETI instruction is executed, the MCU returns to the main routine.

Figure 21. Interrupt Processing Flow Chart



INTERRUPTS (Cont'd)

Figure 22. Interrupt Block Diagram



3.5 POWER SAVING MODES

The WAIT and STOP modes have been implemented in the ST62xx family of MCUs in order to reduce the product's electrical consumption during idle periods. These two power saving modes are described in the following paragraphs.

3.5.1 WAIT Mode

The MCU goes into WAIT mode as soon as the WAIT instruction is executed. The microcontroller can be considered as being in a "software frozen" state where the core stops processing the program instructions, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage. In this mode the peripherals 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 capability of monitoring external events. The active oscillator is not stopped in order to provide a clock signal to the peripherals. Timer counting may be enabled as well as the Timer interrupt, before entering the WAIT mode: this allows the WAIT mode to be exited when a Timer interrupt occurs. The same applies to other peripherals which use the clock signal.

If the WAIT mode is exited due to a Reset (either by activating the external pin or generated by the Watchdog), the MCU enters a normal reset procedure. If an interrupt is generated during WAIT mode, the MCU's behaviour depends on the state of the processor core prior to the WAIT instruction, but also on the kind of interrupt request which is generated. This is described in the following paragraphs. The processor core does not generate a delay following the occurrence of the interrupt, because the oscillator clock is still available and no stabilisation period is necessary.

3.5.2 STOP Mode

If the Watchdog is disabled, STOP mode is available. When in STOP mode, the MCU is placed in the lowest power consumption mode. In this operating mode, the microcontroller can be considered as being "frozen", no instruction is executed, the oscillator is stopped, the RAM contents and peripheral registers are preserved as long as the power supply voltage is higher than the RAM retention voltage, and the ST62xx core waits for the occurrence of an external interrupt request or a Reset to exit the STOP state.

If the STOP state is exited due to a Reset (by activating the external pin) the MCU will enter a normal reset procedure. Behaviour in response to interrupts depends on the state of the processor core prior to issuing the STOP instruction, and also on the kind of interrupt request that is generated.

This case will be described in the following paragraphs. The processor core generates a delay after occurrence of the interrupt request, in order to wait for complete stabilisation of the oscillator, before executing the first instruction.

4.2 TIMER

The MCU features an on-chip Timer peripheral, consisting of an 8-bit counter with a 7-bit programmable prescaler, giving a maximum count of 2^{15}.

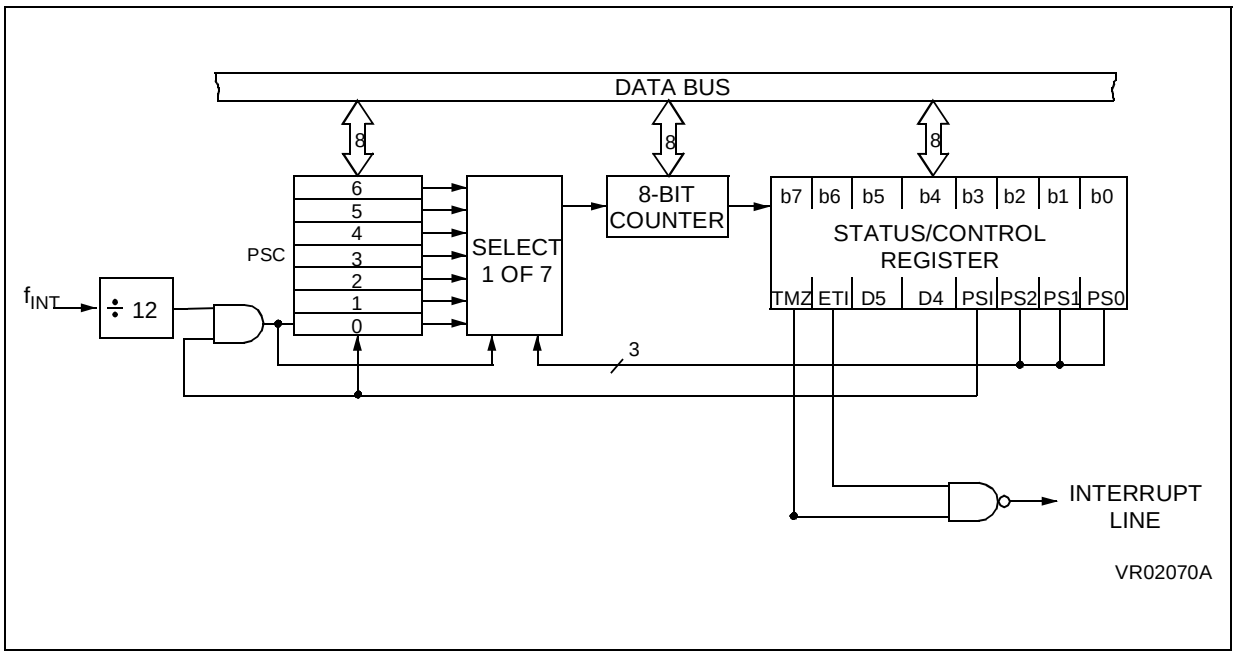
Figure 26 shows the Timer Block Diagram. The content of the 8-bit counter can be read/written in the Timer/Counter register, TCR, which can be addressed in Data space as a RAM location at address 0D3h. The state of the 7-bit prescaler can be read in the PSC register at address 0D2h. The control logic device is managed in the TSCR register as described in the following paragraphs.

The 8-bit counter is decrement by the output (rising edge) coming from the 7-bit prescaler and can be loaded and read under program control. When it decrements to zero then the TMZ (Timer Zero)bit in the TSCR is set. If the ETI (Enable Timer Interrupt) bit in the TSCR is also set, an interrupt request is generated. The Timer interrupt can be used to exit the MCU from WAIT mode.

The prescaler input is the internal frequency (f_{INT}) divided by 12. The prescaler decrements on the rising edge. Depending on the division factor programmed by PS2, PS1 and PS0 bits in the TSCR (see Figure 12), the clock input of the timer/counter register is multiplexed to different sources. For division factor 1, the clock input of the prescaler is also that of timer/counter; for factor 2, bit 0 of the prescaler register is connected to the clock input of TCR. This bit changes its state at half the frequency of the prescaler input clock. For factor 4, bit 1 of the PSC is connected to the clock input of TCR, and so forth. The prescaler initialize bit, PSI, in the TSCR register must be set to allow the prescaler (and hence the counter) to start. If it is cleared, all the prescaler bits are set and the counter is inhibited from counting. The prescaler can be loaded with any value between 0 and 7Fh, if bit PSI is set. The prescaler tap is selected by means of the PS2/PS1/PS0 bits in the control register.

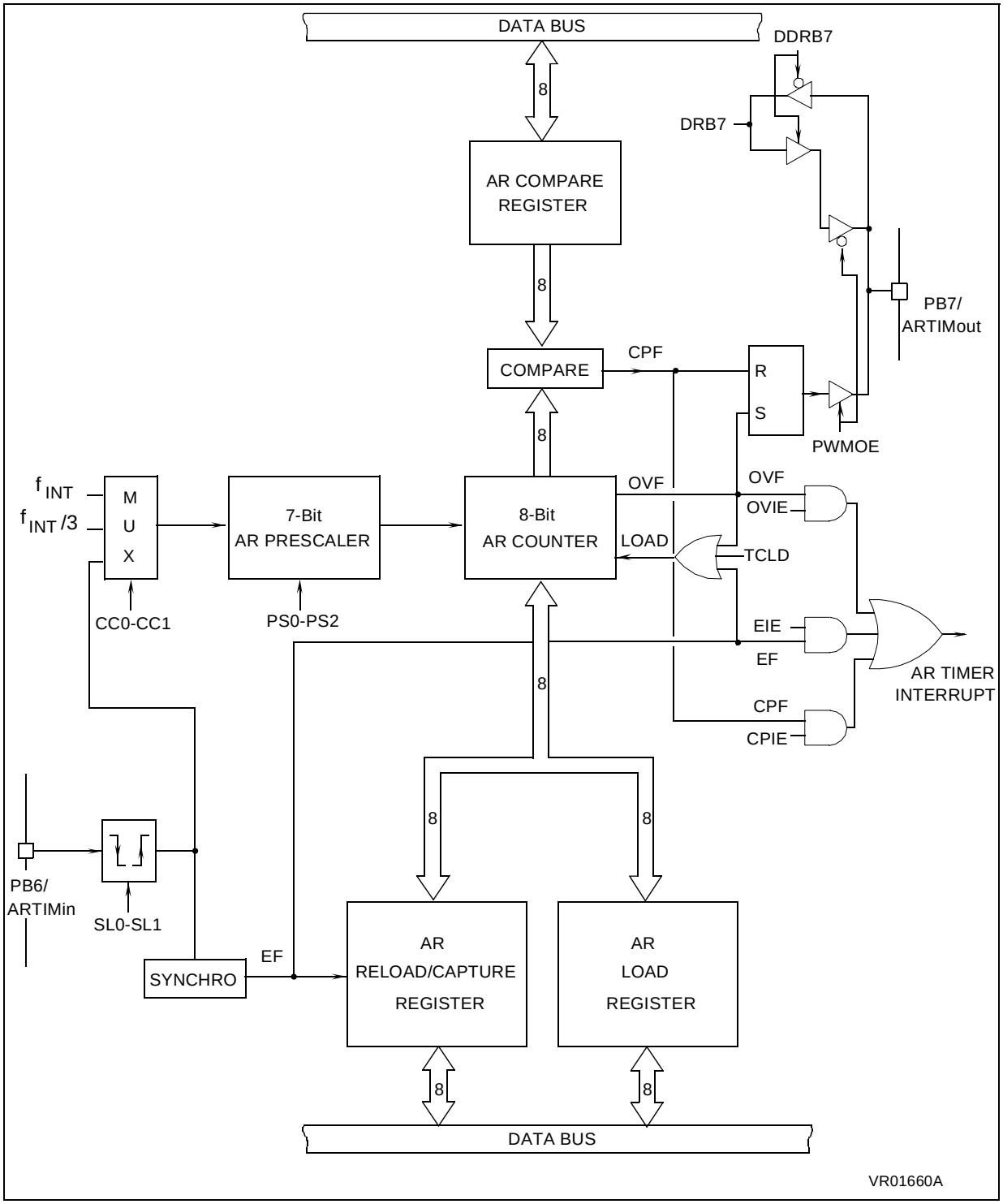
Figure 27 illustrates the Timer's working principle.

Figure 26. Timer Block Diagram



AUTO-RELOAD TIMER (Cont'd)

Figure 28. AR Timer Block Diagram



AUTO-RELOAD TIMER (Cont'd)

It should be noted that the reload values will also affect the value and the resolution of the duty cycle of PWM output signal. To obtain a signal on ARTIMout, the contents of the ARCP register must be greater than the contents of the ARRC register.

The maximum available resolution for the ARTIMout duty cycle is:

Resolution = 1/[255-(ARRC)]

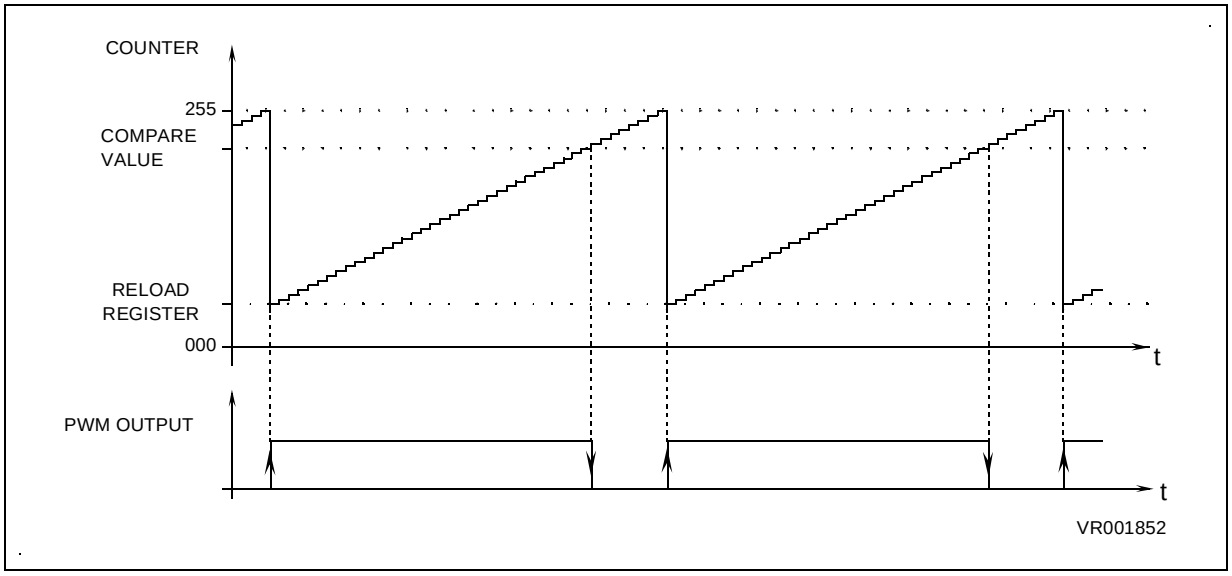
Where ARRC is the content of the Reload/Capture register. The compare value loaded in the Compare Register, ARCP, must be in the range from (ARRC) to 255.

The ARTC counter is initialized by writing to the ARRC register and by then setting the TCLD (Timer Load) and the TEN (Timer Clock Enable) bits in the Mode Control register, ARMC.

Enabling and selection of the clock source is controlled by the CC0, CC1, SL0 and SL1 bits in the Status Control Register, ARSC1. The prescaler division ratio is selected by the PS0, PS1 and PS2 bits in the ARSC1 register.

In Auto-reload Mode, any of the three available clock sources can be selected: Internal Clock, Internal Clock divided by 3 or the clock signal present on the ARTIMin pin.

Figure 29. Auto-reload Timer PWM Function



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

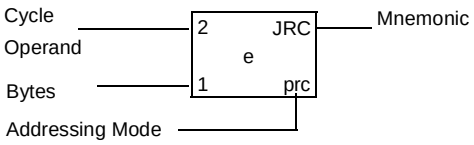
LOW / HI	**0** **0000**	**1** **0001**	**2** **0010**	**3** **0011**	**4** **0100**	**5** **0101**	**6** **0110**	**7** **0111**	LOW / HI
0 **0000**	2 JRNZ e 1 pcr	4 CALL abc 2 ext	2 JRNC e 1 pcr	5 JRR b0,rr,ee 3 bt	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD a,(x) 1 ind	**0** **0000**
1 **0001**	2 JRNZ e 1 pcr	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 RNZ 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	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



Opcode Map Summary (Continued)

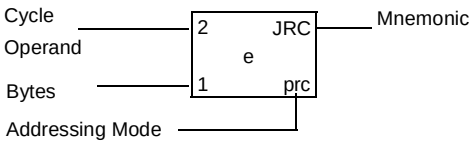
LOW / HI	**8 1000**	**9 1001**	**A 1010**	**B 1011**	**C 1100**	**D 1101**	**E 1110**	**F 1111**	LOW / HI
0 0000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 prc	4 LD a,(y) 1 ind	**0 0000**
1 0001	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 JRZ e 1 pcr	4 DEC x 1 sd	2 JRC e 1 prc	4 LD a,rr 2 dir	**1 0001**
2 0010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b4,rr 2 b.d	2 JRZ e 1 pcr	4 COM a	2 JRC e 1 prc	4 CP a,(y) 1 ind	**2 0010**
3 0011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b4,rr 2 b.d	2 JRZ e 1 pcr	4 LD x,a 1 sd	2 JRC e 1 prc	4 CP a,rr 2 dir	**3 0011**
4 0100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b2,rr 2 b.d	2 JRZ e 1 pcr	2 RETI 1 inh	2 JRC e 1 prc	4 ADD a,(y) 1 ind	**4 0100**
5 0101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b2,rr 2 b.d	2 JRZ e 1 pcr	4 DEC y 1 sd	2 JRC e 1 prc	4 ADD a,rr 2 dir	**5 0101**
6 0110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b6,rr 2 b.d	2 JRZ e 1 pcr	2 STOP 1 inh	2 JRC e 1 prc	4 INC (y) 1 ind	**6 0110**
7 0111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b6,rr 2 b.d	2 JRZ e 1 pcr	4 LD y,a 1 sd	2 JRC e 1 prc	4 INC rr 2 dir	**7 0111**
8 1000	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b1,rr 2 b.d	2 JRZ e 1 pcr	#	2 JRC e 1 prc	4 LD (y),a 1 ind	**8 1000**
9 1001	2 RNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b1,rr 2 b.d	2 JRZ e 1 pcr	4 DEC v 1 sd	2 JRC e 1 prc	4 LD rr,a 2 dir	**9 1001**
A 1010	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b5,rr 2 b.d	2 JRZ e 1 pcr	4 RCL a 1 inh	2 JRC e 1 prc	4 AND a,(y) 1 ind	**A 1010**
B 1011	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b5,rr 2 b.d	2 JRZ e 1 pcr	4 LD v,a 1 sd	2 JRC e 1 prc	4 AND a,rr 2 dir	**B 1011**
C 1100	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b3,rr 2 b.d	2 JRZ e 1 pcr	2 RET 1 inh	2 JRC e 1 prc	4 SUB a,(y) 1 ind	**C 1100**
D 1101	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b3,rr 2 b.d	2 JRZ e 1 pcr	4 DEC w 1 sd	2 JRC e 1 prc	4 SUB a,rr 2 dir	**D 1101**
E 1110	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b7,rr 2 b.d	2 JRZ e 1 pcr	2 WAIT 1 inh	2 JRC e 1 prc	4 DEC (y) 1 ind	**E 1110**
F 1111	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b7,rr 2 b.d	2 JRZ e 1 pcr	4 LD w,a 1 sd	2 JRC e 1 prc	4 DEC rr 2 dir	**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	1byte dataspace address
nn	1 byte immediate data
abc	12 bit address
ee	8 bit Displacement



6.2 RECOMMENDED OPERATING CONDITIONS

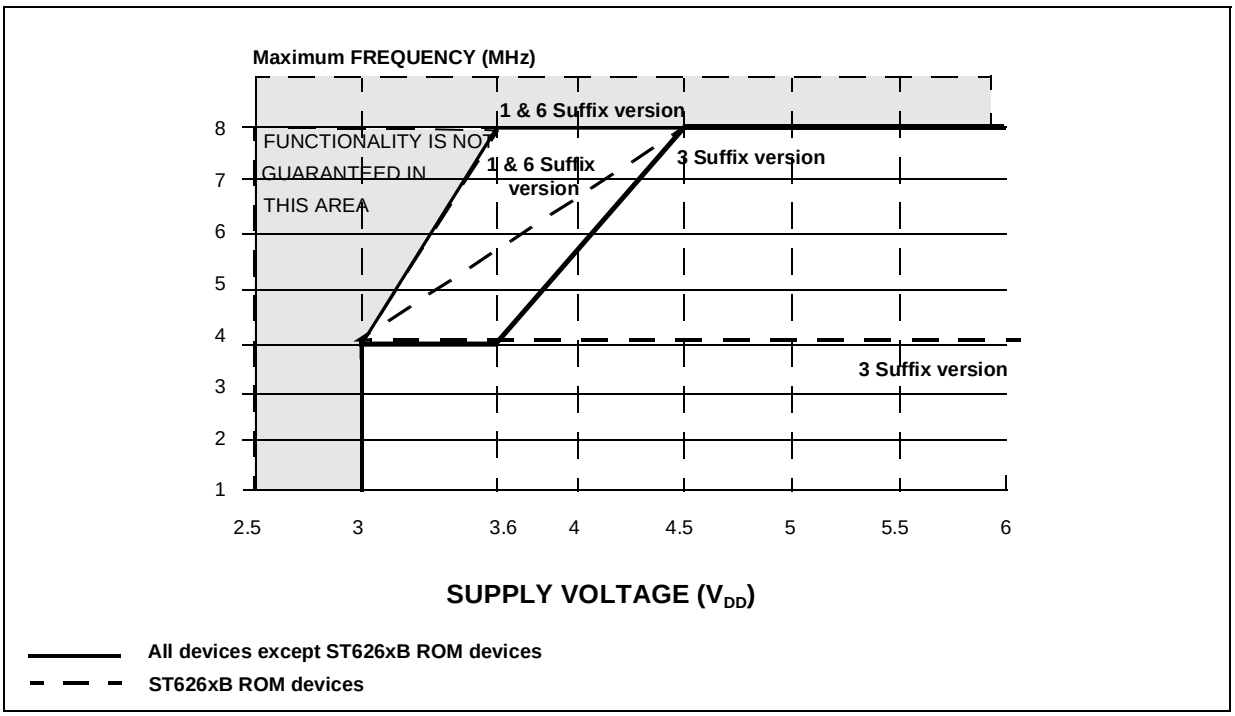
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
T_A	Operating Temperature	6 Suffix Version 1 Suffix Version 3 Suffix Version	-40 0 -40		85 70 125	°C
V_{DD}	Operating Supply Voltage (Except ST626xB ROM devices)	f_{OSC} = 4MHz, 1 & 6 Suffix f_{OSC} = 4MHz, 3 Suffix fosc= 8MHz , 1 & 6 Suffix fosc= 8MHz , 3 Suffix	3.0 3.0 3.6 4.5		6.0 6.0 6.0 6.0	V
	Operating Supply Voltage (ST626xB ROM devices)	f_{OSC} = 4MHz, 1 & 6 Suffix f_{OSC} = 4MHz, 3 Suffix fosc= 8MHz , 1 & 6 Suffix fosc= 8MHz , 3 Suffix	3.0 3.0 4.0 4.5		6.0 6.0 6.0 6.0	V
f_{OSC}	Oscillator Frequency[2] (Except ST626xB ROM devices)	V_{DD} = 3.0V, 1 & 6 Suffix V_{DD} = 3.0V , 3 Suffix V_{DD} = 3.6V , 1 & 6 Suffix V_{DD} = 3.6V , 3 Suffix	0 0 0 0		4.0 4.0 8.0 4.0	MHz
	Oscillator Frequency[2] (ST626xB ROM devices)	V_{DD} = 3.0V, 1 & 6 Suffix V_{DD} = 3.0V , 3 Suffix V_{DD} = 4.0V , 1 & 6 Suffix V_{DD} = 4.0V , 3 Suffix	0 0 0 0		4.0 4.0 8.0 4.0	MHz
I_{INJ+}	Pin Injection Current (positive)	V_{DD} = 4.5 to 5.5V			+5	mA
I_{INJ-}	Pin Injection Current (negative)	V_{DD} = 4.5 to 5.5V			-5	mA

Notes:

1. Care must be taken in case of negative current injection, where adapted impedance must be respected on analog sources to not affect the A/D conversion. For a -1mA injection, a maximum 10 KΩ is recommended.

2.An oscillator frequency above 1MHz is recommended for reliable A/D results

Figure 36. Maximum Operating FREQUENCY (Fmax) Versus SUPPLY VOLTAGE (V_{DD})



The shaded area is outside the recommended operating range; device functionality is not guaranteed under these conditions.

6.5 A/D CONVERTER CHARACTERISTICS

(T_A = -40 to +125°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
Res	Resolution			8		Bit
A_{TOT}	Total Accuracy (1) (2)	f_{OSC} > 1.2MHz f_{OSC} > 32kHz			±2 ±4	LSB
t_C	Conversion Time	f_{OSC} = 8MHz (T_A < 85°C) f_{OSC} = 4 MHz		70 140		µs
ZIR	Zero Input Reading	Conversion result when $V_{IN} = V_{SS}$	00			Hex
FSR	Full Scale Reading	Conversion result when $V_{IN} = V_{DD}$			FF	Hex
AD_I	Analog Input Current During Conversion	V_{DD}= 4.5V			1.0	µA
AC_{IN}	Analog Input Capacitance			2	5	pF

Notes:
1. Noise at VDD, VSS <10mV
2. With oscillator frequencies less than 1MHz, the A/D Converter accuracy is decreased.

6.6 TIMER CHARACTERISTICS

(T_A = -40 to +125°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
f_{IN}	Input Frequency on TIMER Pin				$\frac{f_{INT}}{4}$	MHz
t_W	Pulse Width at TIMER Pin	V_{DD} = 3.0V V_{DD} >4.5V	1 125			µs ns

6.7 SPI CHARACTERISTICS

(T_A = -40 to +125°C unless otherwise specified)

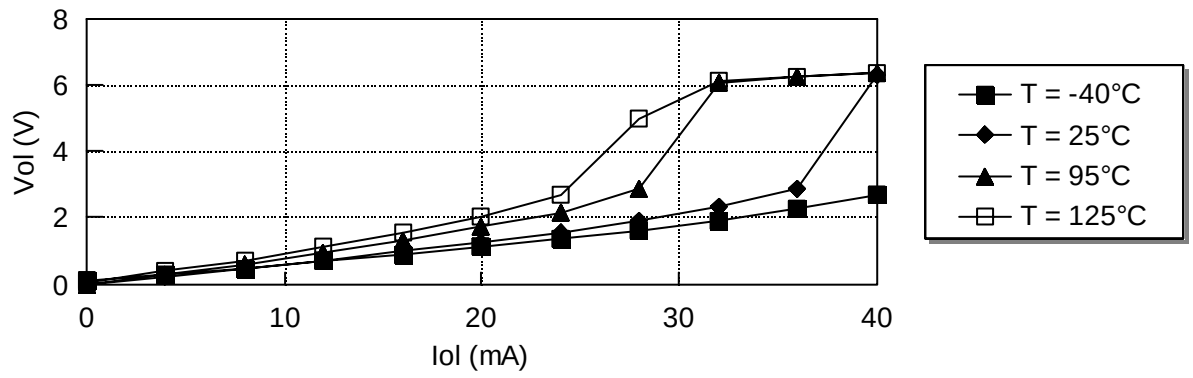
Symbol	Parameter	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
F_{CL}	Clock Frequency	Applied on Scl			500	kHz
t_{SU}	Set-up Time	Applied on Sin		250		ns
t_h	Hold Time	Applied onSin		50		ns

6.8 ARTIMER ELECTRICAL CHARACTERISTICS

(T_A = -40 to +125°C unless otherwise specified)

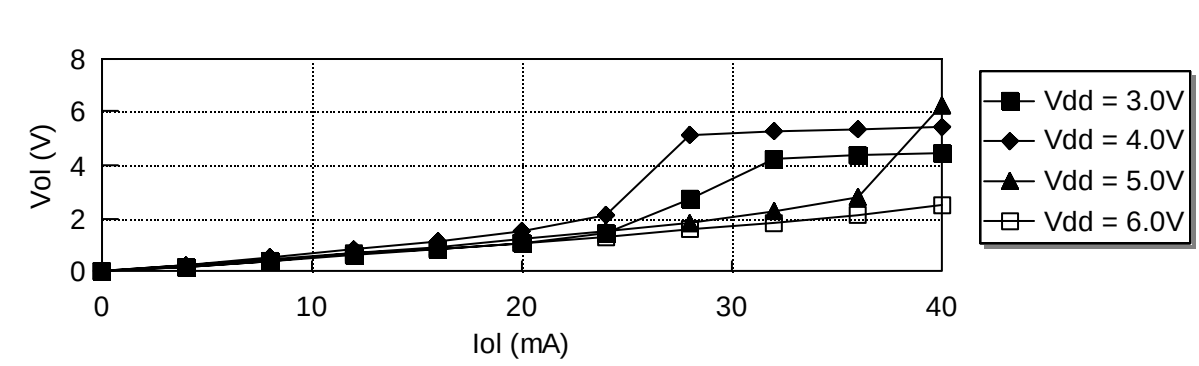
Symbol	Parameter	Test Conditions	Value			Unit
			Min	Typ	Max	
f_{IN}	Input Frequency on ARTIMin Pin	RUN and WAIT Modes				MHz
		STOP mode			2	

Figure 37. Vol versus Iol on all I/O port at Vdd=5V



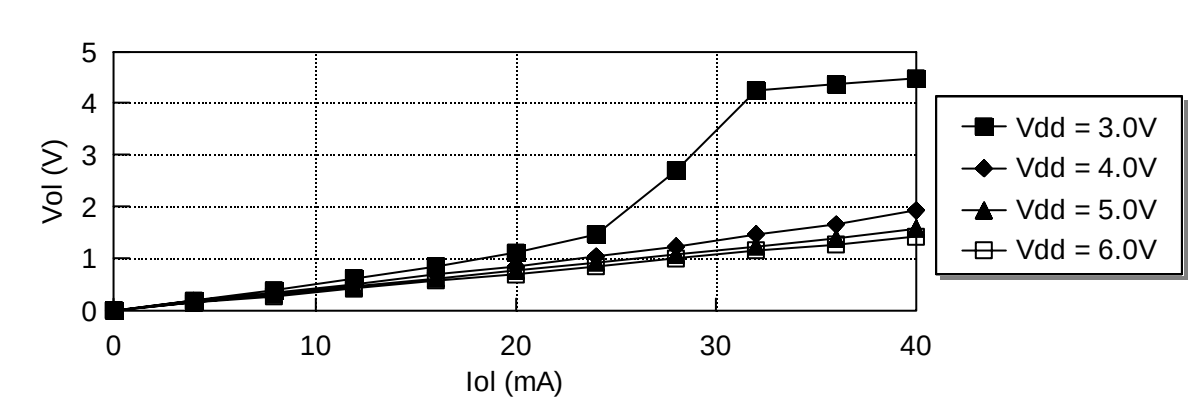
This curves represents typical variations and is given for guidance only

Figure 38. Vol versus Iol on all I/O port at T=25°C



This curves represents typical variations and is given for guidance only

Figure 39. Vol versus Iol for High sink (30mA) I/Oports at T=25°C



This curves represents typical variations and is given for guidance only

ST6253C/60B/63B/P53C/P60C/P63C MICROCONTROLLER OPTION LIST

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

STMicroelectronics references:

Device: [ ] ST6253C (2 KB) [ ] ST6260B (4 KB) [ ] ST6263B (2 KB)
[ ] ST62P53C (2 KB) [ ] ST62P60C (4 KB) [ ] ST62P63C (2 KB)

Package: [ ] Dual in Line Plastic
[ ] 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

Oscillator Selection: [ ] Quartz crystal / Ceramic resonator
[ ] RC network

Reset Delay: [ ] 32768 cycle delay [ ] 2048 cycle delay

Watchdog Selection: [ ] Software Activation [ ] Hardware Activation

PB1:PB0 Pull-Up at RESET*: [ ] Enabled [ ] Disabled

PB3:PB2 Pull-Up at RESET*: [ ] Enabled [ ] Disabled

External STOP Mode Control: [ ] Enabled [ ] Disabled

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

NMI pull-up*: [ ] Enabled [ ] Disabled

ADC Synchro*: [ ] Enabled [ ] Disabled

*except on ST6260B/63B

Comments:

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