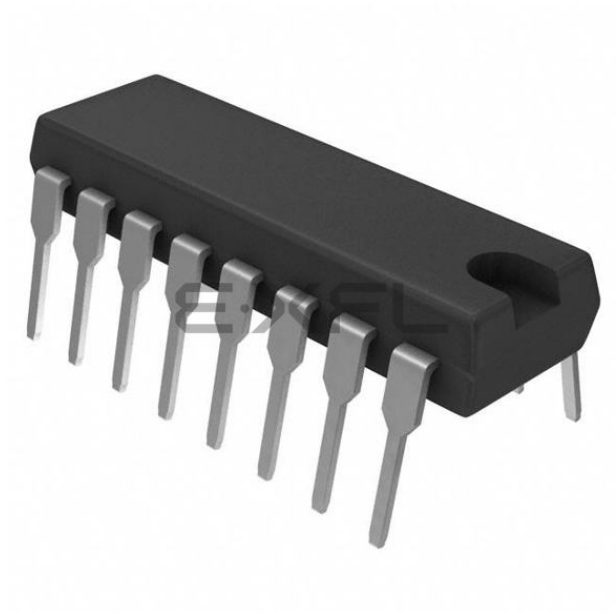


Welcome to E-XFL.COM

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	9
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 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	16-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st62t01cb3

1 INTRODUCTION

The ST6200C, 01C and 03C devices are low cost members of the ST62xx 8-bit HCMOS family of microcontrollers, which is targeted at low to medium complexity applications. All ST62xx devices are based on a building block approach: a common core is surrounded by a number of on-chip peripherals.

The ST62E01C is the erasable EPROM version of the ST62T00C, T01 and T03C devices, which may be used during the development phase for the ST62T00C, T01 and T03C target devices, as well as the respective ST6200C, 01C and 03C ROM devices.

OTP and EPROM devices are functionally identical. OTP devices offer all the advantages of user programmability at low cost, which make them the ideal choice in a wide range of applications where frequent code changes, multiple code versions or last minute programmability are required.

The ROM based versions offer the same functionality, selecting the options defined in the programmable option bytes of the OTP/EPROM versions in the ROM option list (See Section 11.6 on page 92).

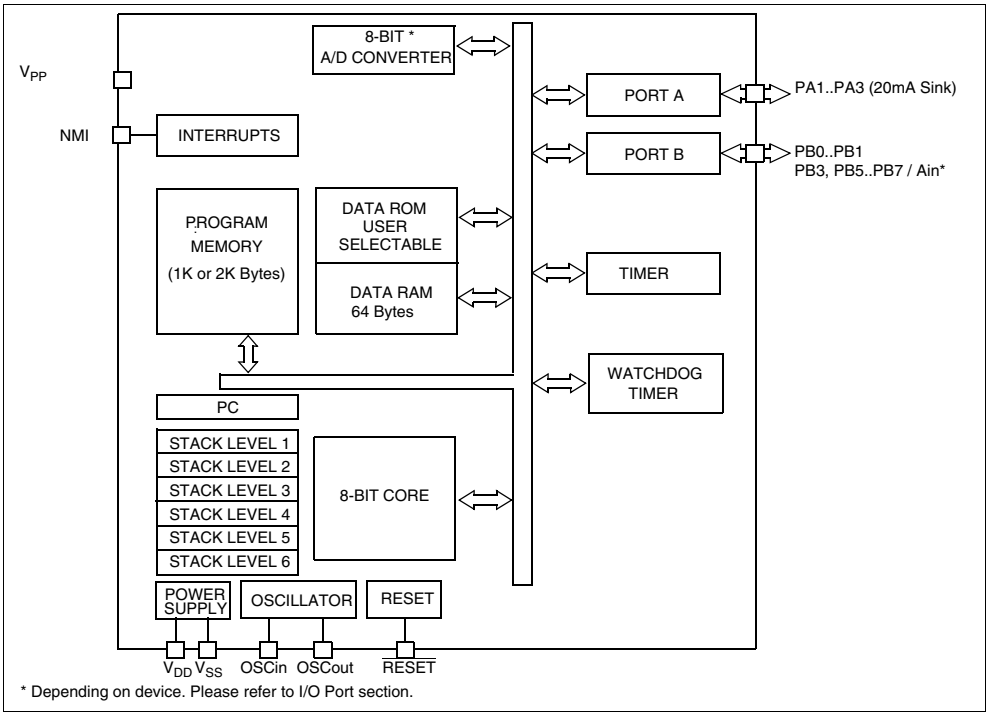
The ST62P00C, P01C and P03C are the **F**actory **A**dvanced **S**ervice **T**echnique ROM (FASTROM) versions of ST62T00C, T01 and T03C OTP devices.

They offer the same functionality as OTP devices, but they do not have to be programmed by the customer (See Section 11 on page 86).

These compact low-cost devices feature a Timer comprising an 8-bit counter with a 7-bit programmable prescaler, an 8-bit A/D Converter with 4 analog inputs (depending on device, see device summary on page 1) and a Digital Watchdog timer, making them well suited for a wide range of automotive, appliance and industrial applications.

For easy reference, all parametric data are located in Section 10 on page 58.

Figure 1. Block Diagram



* Depending on device. Please refer to I/O Port section.

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 in OTP, EPROM or ROM devices can be protected against external readout of memory by setting the Readout Protection bit in the option byte (Section 3.3 on page 15).

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 or ROM 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.

5.5 INTERRUPT RULES AND PRIORITY MANAGEMENT

- A Reset can interrupt the NMI and peripheral interrupt routines
- The Non Maskable Interrupt request has the highest priority and can interrupt any peripheral interrupt routine at any time but cannot interrupt another NMI interrupt.
- No peripheral interrupt can interrupt another. If more than one interrupt request is pending, these are processed by the processor core according to their priority level: vector #1 has the highest priority while vector #4 the lowest. The priority of each interrupt source is fixed by hardware (see Interrupt Mapping table).

5.6 INTERRUPTS AND LOW POWER MODES

All interrupts cause the processor to exit from WAIT mode. Only the external and some specific interrupts from the on-chip peripherals cause the processor to exit from STOP mode (refer to the “Exit from STOP“ column in the Interrupt Mapping Table).

5.7 NON MASKABLE INTERRUPT

This interrupt is triggered when a falling edge occurs on the NMI pin regardless of the state of the GEN bit in the IOR register. An interrupt request on NMI vector #0 is latched by a flip flop which is automatically reset by the core at the beginning of the NMI service routine.

5.8 PERIPHERAL INTERRUPTS

Different peripheral interrupt flags in the peripheral control registers are able to cause an interrupt when they are active if both:

– The GEN bit of the IOR register is set

– The corresponding enable bit is set in the peripheral control register.

Peripheral interrupts are linked to vectors #3 and #4. Interrupt requests are flagged by a bit in their corresponding control register. This means that a request cannot be lost, because the flag bit must be cleared by user software.

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

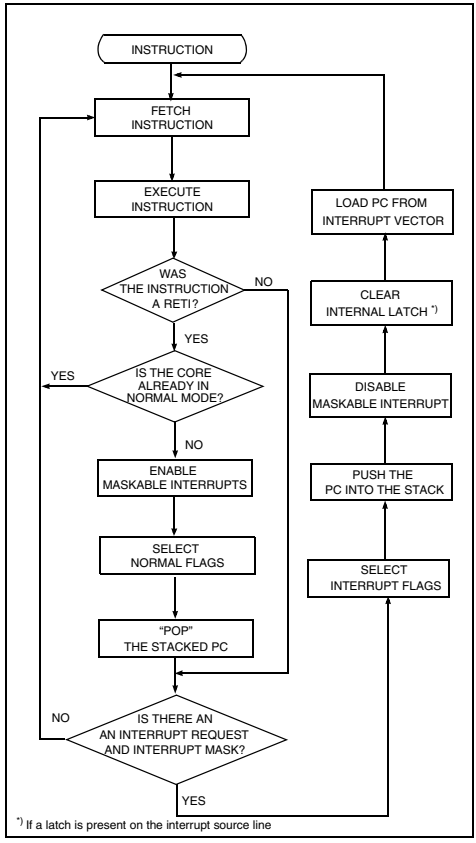


Table 6. Interrupt Response Time

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.

5.11 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
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	Ext. Interrupt Port B	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	Lowest Priority

* Depending on device. See device summary on page 1.

6.2 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)

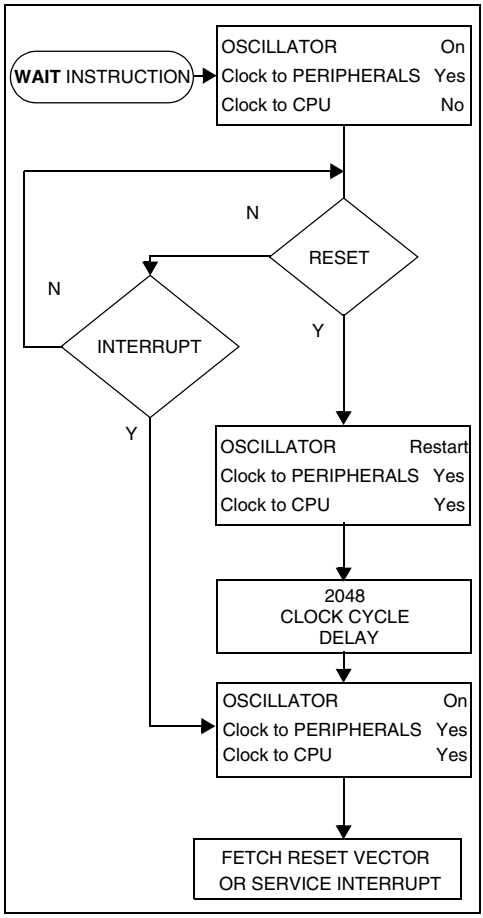
– A peripheral interrupt (timer, ADC,...),

– 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.

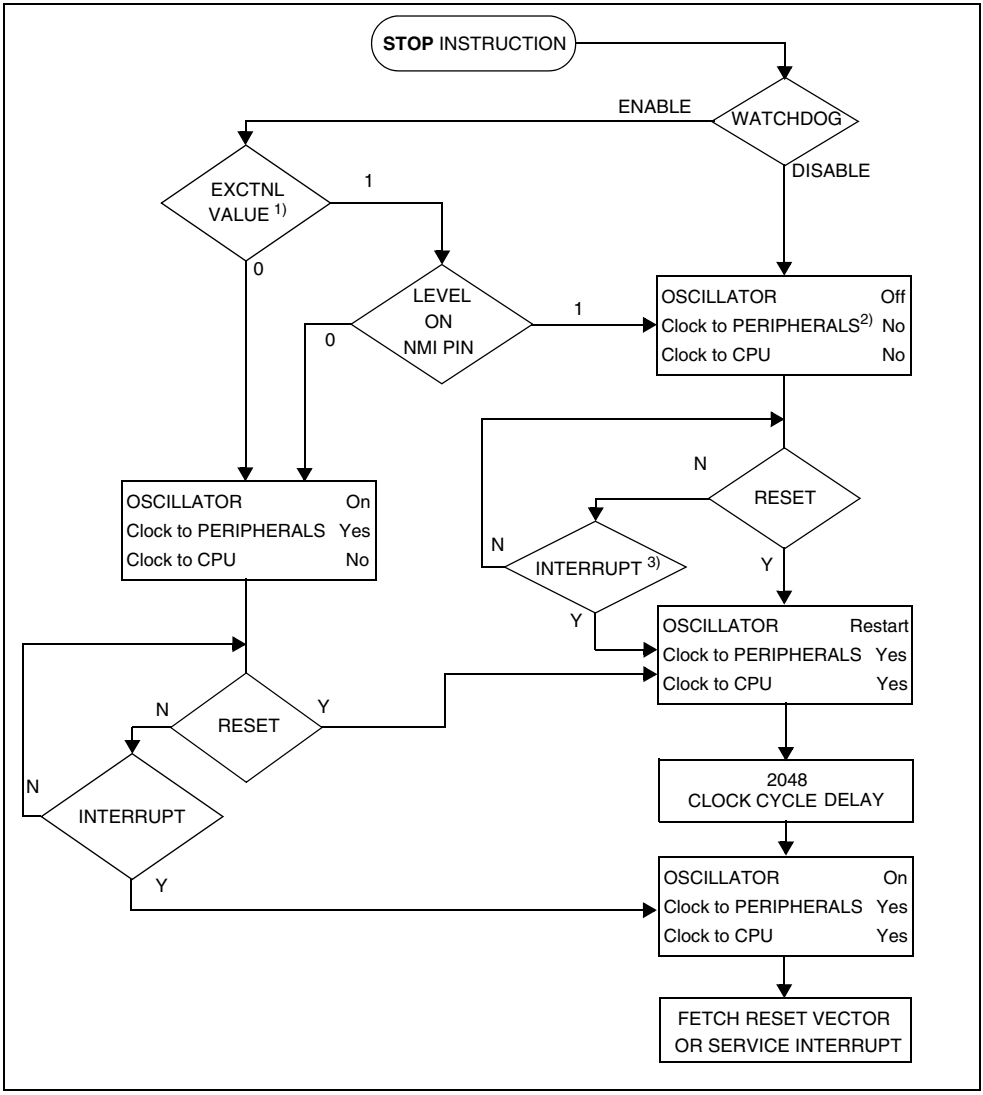
See also Section 6.4.1.

Figure 20. WAIT Mode Flowchart



STOP MODE (Cont'd)

Figure 22. STOP Mode Flowchart



Notes:

1. EXCTNL is an option bit. See option byte section for more details.

2. Peripheral clocked with an external clock source can still be active.

3. Only some specific interrupts can exit the MCU from STOP mode (such as external interrupt). Refer to the Interrupt Mapping table for more details.

I/O PORTS (Cont'd)

7.2.5 Instructions NOT to be used to access Port Data registers (SET, RES, INC and DEC)

DO NOT USE READ-MODIFY-WRITE INSTRUCTIONS (SET, RES, INC and DEC) ON PORT DATA REGISTERS IF ANY PIN OF THE PORT IS CONFIGURED IN INPUT MODE.

These instructions make an implicit read and write back of the entire register. In port input mode, however, the data register reads from the input pins directly, and not from the data register latches. Since data register information in input mode is used to set the characteristics of the input pin (interrupt, pull-up, analog input), these may be unintentionally reprogrammed depending on the state of the input pins.

As a general rule, it is better to only use single bit instructions on data registers when the whole (8-bit) port is in output mode. In the case of inputs or of mixed inputs and outputs, it is advisable to keep a copy of the data register in RAM. Single bit instructions may then be used on the RAM copy, after which the whole copy register can be written to the port data register:

```
SET bit, datacopy
LD a, datacopy
LD DRA, a
```

7.2.6 Recommendations

1. Safe I/O State Switching Sequence

Switching the I/O ports from one state to another should be done in a sequence which ensures that no unwanted side effects can occur. The recommended safe transitions are illustrated in Figure 24 The Interrupt Pull-up to Input Analog transition (and vice-vesra) is potentially risky and should be avoided when changing the I/O operating mode.

2. Handling Unused Port Bits

On ports that have less than 8 external pins connected:

– Leave the unbonded pins in reset state and do not change their configuration.

– Do not use instructions that act on a whole port register (INC, DEC, or read operations). Unavailable bits must be masked by software (AND instruction). Thus, when a read operation performed on an incomplete port is followed by a comparison, use a mask.

3. High Impedance Input

On any CMOS device, it is not recommended to connect high impedance on input pins. The choice of these impedance has to be done with respect to the maximum leakage current defined in the datasheet. The risk is to be close or out of specification on the input levels applied to the device.

7.3 LOW POWER MODES

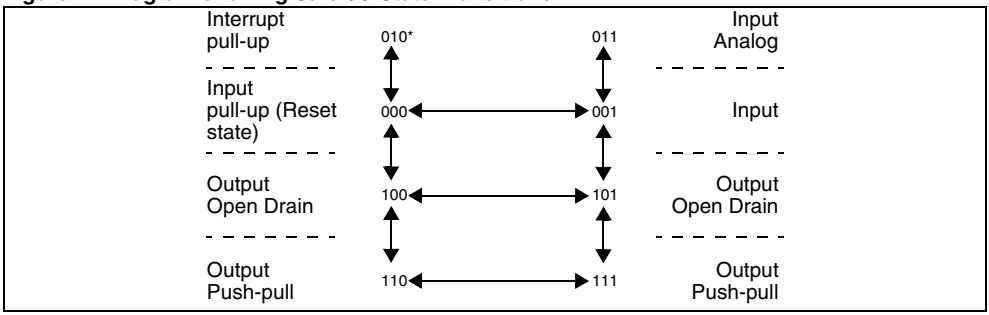
The WAIT and STOP instructions allow the ST62xx to be used in situations where low power consumption is needed. The lowest power consumption is achieved by configuring I/Os in output push-pull low mode.

Mode	Description
WAIT	No effect on I/O ports. External interrupts cause the device to exit from WAIT mode.
STOP	No effect on I/O ports. External interrupts cause the device to exit from STOP mode.

7.4 INTERRUPTS

The external interrupt event generates an interrupt if the corresponding configuration is selected with DDR, DR and OR registers (see Table 8) and the GEN-bit in the IOR register is set.

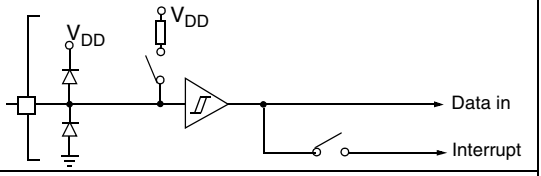
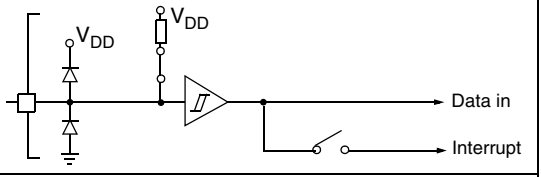
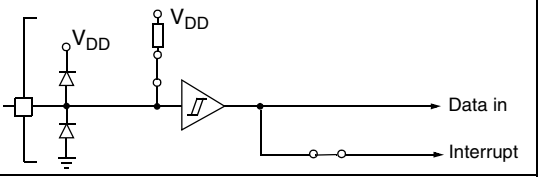
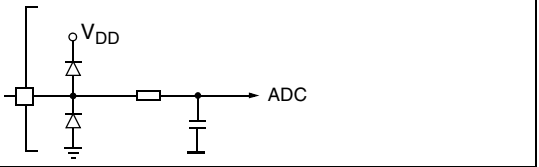
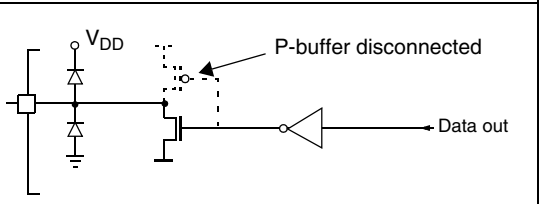
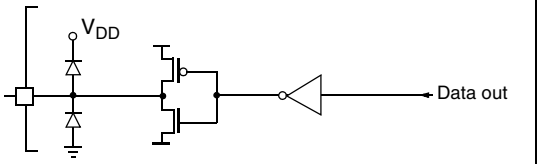
Figure 24. Diagram showing Safe I/O State Transitions



Note *. xxx = DDR, OR, DR Bits respectively

I/O PORTS (Cont'd)

Table 9. I/O Port Option Selections

	MODE	DDRx	ORx	DRx	AVAILABLE ON(1)	SCHEMATIC
Digital Input	Input	0	0	1	PA1-PA3 PB0, PB1, PB3, PB5-PB7	
	Reset state Input with pull up	0	0	0	PA1-PA3 PB0, PB1, PB3, PB5-PB7	
	Input with pull up with interrupt	0	1	0	PA1-PA3 PB0, PB1, PB3, PB5-PB7	
Analog Input	Analog Input	0	1	1	PB3, PB5-PB7 (Except on ST6203C)	
Digital output	Open drain output (5mA)	1	0	0/1	PB0, PB1, PB3, PB5-PB7	
	Open drain output (20 mA)	1	0	0/1	PA1-PA3	
	Push-pull output (5mA)	1	1	0/1	PB0, PB1, PB3, PB5-PB7	
	Push-pull output (20 mA)	1	1	0/1	PA1-PA3	

Note 1. Provided the correct configuration has been selected (see Table 8).

WATCHDOG TIMER (Cont'd)

8.1.3 Functional Description

The watchdog activation is selected through an option in the option bytes:

– **HARDWARE** Watchdog option

After reset, the watchdog is permanently active, the C bit in the WDGR is forced high and the user can not change it. However, this bit can be read equally as 0 or 1.

– **SOFTWARE** Watchdog option

After reset, the watchdog is deactivated. The function is activated by setting C bit in the WDGR register. Once activated, it cannot be deactivated. The counter value stored in the WDGR register (bits SR:T0), is decremented every 3072 clock cycles. The length of the timeout period can be programmed by the user in 64 steps of 3072 clock cycles.

If the watchdog is activated (by setting the C bit) and when the SR bit is cleared, the watchdog initiates a reset cycle pulling the reset pin low for typically 500ns.

The application program must write in the WDGR register at regular intervals during normal operation to prevent an MCU reset. The value to be stored in the WDGR register must be between FEh and 02h (see Table 11). To run the watchdog function the following conditions must be true:

– The C bit is set (watchdog activated)

– The SR bit is set to prevent generating an immediate reset

– The T[5:0] bits contain the number of decrements which represent the time delay before the watchdog produces a reset.

Table 11. Watchdog Timing (f_{OSC} = 8 MHz)

	WDGR Register initial value	WDG timeout period (ms)
Max.	FEh	24.576
Min.	02h	0.384

8.1.3.1 Software Reset

The SR bit can be used to generate a software reset by clearing the SR bit while the C bit is set.

8.1.4 Recommendations

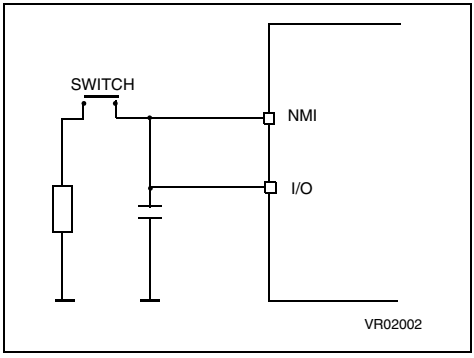
1. The Watchdog plays an important supporting role in the high noise immunity of ST62xx devices, and should be used wherever possible. Watchdog related options should be selected on the basis of a trade-off between application security and STOP mode availability (refer to the description of the WDACT and EXTCNTL bits on the Option Bytes).

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 26) 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.

Figure 26. A typical circuit making use of the EXERNAL STOP MODE CONTROL feature



2. When software activation is selected (WDACT bit in Option byte) 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:

```
jrr 0, WDGR, #+3 ; If C=0,jump to next
ldi WDGR, 0FDH   ; SR=0 -> reset

next :
```

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)

Conditional Branch. Branch instructions perform a branch in the program when the selected condition is met.

Bit Manipulation Instructions. These instructions can handle any bit in Data space memory. One group either sets or clears. The other group (see Conditional Branch) performs the bit test branch operations.

Control Instructions. Control instructions control microcontroller operations during program execution.

Jump and Call. These two instructions are used to perform long (12-bit) jumps or subroutine calls to any location in the whole program space.

Table 17. Conditional Branch Instructions

Instruction	Branch If	Bytes	Cycles	Flags	
				Z	C
JRC e	C = 1	1	2	*	*
JRNC e	C = 0	1	2	*	*
JRZ e	Z = 1	1	2	*	*
JRNZ e	Z = 0	1	2	*	*
JRR b, rr, ee	Bit = 0	3	5	*	Δ
JRS b, rr, ee	Bit = 1	3	5	*	Δ

Notes:
b 3-bit address
e 5 bit signed displacement in the range -15 to +16
ee 8 bit signed displacement in the range -126 to +129
rr Data space register
Δ Affected. The tested bit is shifted into carry.
* Not Affected

Table 18. Bit Manipulation Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
SET b,rr	Bit Direct	2	4	*	*
RES b,rr	Bit Direct	2	4	*	*

Notes:
b 3-bit address
rr Data space register
* Not Affected
Bit Manipulation Instructions should not be used on Port Data Registers and any registers with read only and/or write only bits (see I/O port chapter)

Table 19. Control Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
NOP	Inherent	1	2	*	*
RET	Inherent	1	2	*	*
RETI	Inherent	1	2	Δ	Δ
STOP (1)	Inherent	1	2	*	*
WAIT	Inherent	1	2	*	*

Notes:
1. This instruction is deactivated and a WAIT is automatically executed instead of a STOP if the watchdog function is selected.
Δ Affected
*Not Affected

Table 20. Jump & Call Instructions

Instruction	Addressing Mode	Bytes	Cycles	Flags	
				Z	C
CALL abc	Extended	2	4	*	*
JP abc	Extended	2	4	*	*

Notes:
abc 12-bit address
* Not Affected

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.

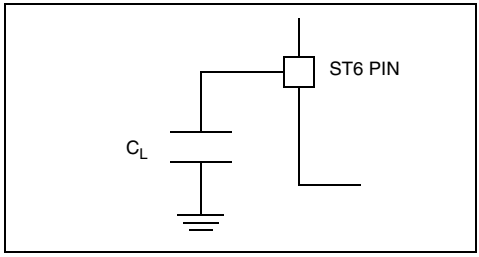
10.1.3 Typical Curves

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 30.

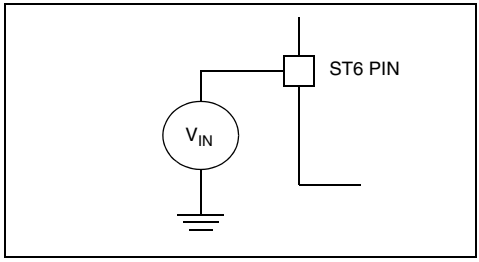
Figure 30. Pin Loading Conditions



10.1.5 Pin Input Voltage

The input voltage measurement on a pin of the device is described in Figure 31.

Figure 31. Pin Input Voltage



SUPPLY CURRENT CHARACTERISTICS (Cont'd)

10.4.2 WAIT Modes

Symbol	Parameter	Conditions			Typ [1]	Max [2]	Unit
I_{DD}	Supply current in WAIT mode [3] Option bytes not programmed (see Figure 38)	4.5V≤V_{DD}≤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 39)			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 40)		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 38)	3V≤V_{DD}≤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 39)			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 40)		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≤V_{DD}≤6.0V range) and V_{DD}=3.3V (3V≤V_{DD}≤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.

EMC CHARACTERISTICS (Cont'd)

10.7.2.2 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 52. For more details, refer to the AN1181 application note.

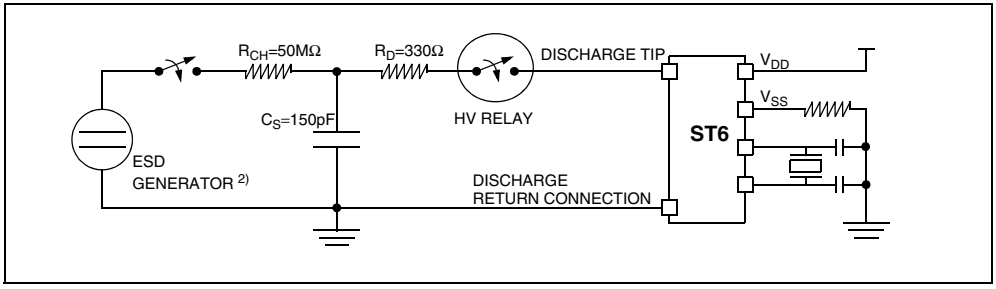
Electrical Sensitivities

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 52. Simplified Diagram of the ESD Generator for DLU



I/O PORT PIN CHARACTERISTICS (Cont'd)

Figure 59. Typical V_{OH} at V_{DD} = 5V

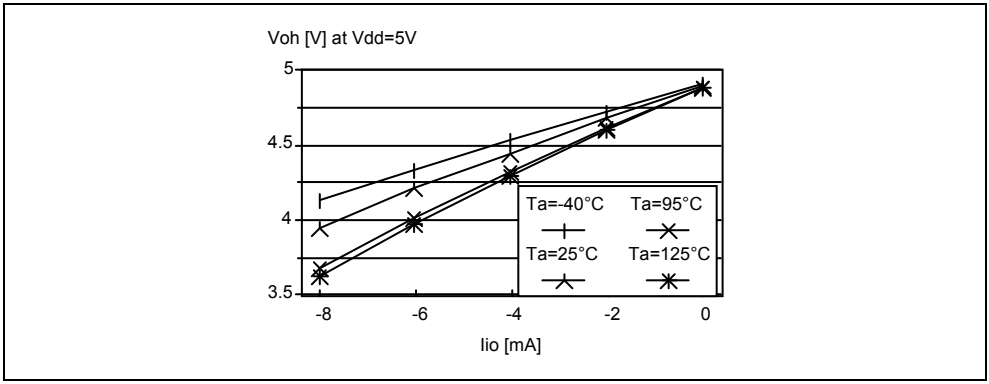


Figure 60. Typical V_{OL} vs V_{DD} (standard I/Os)

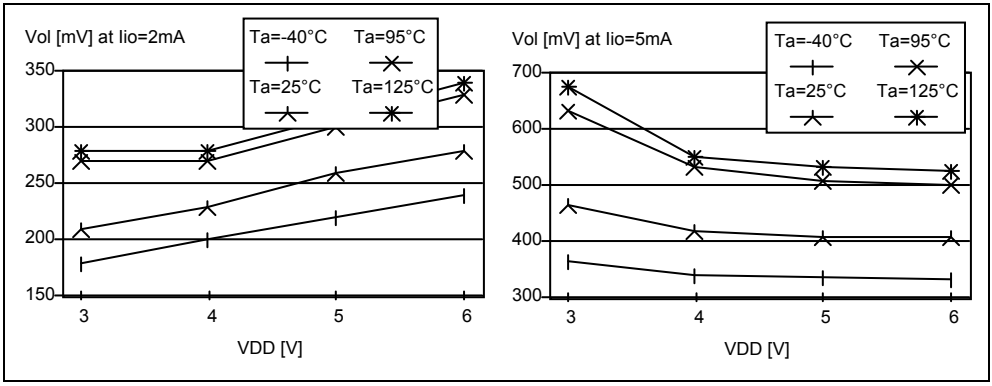
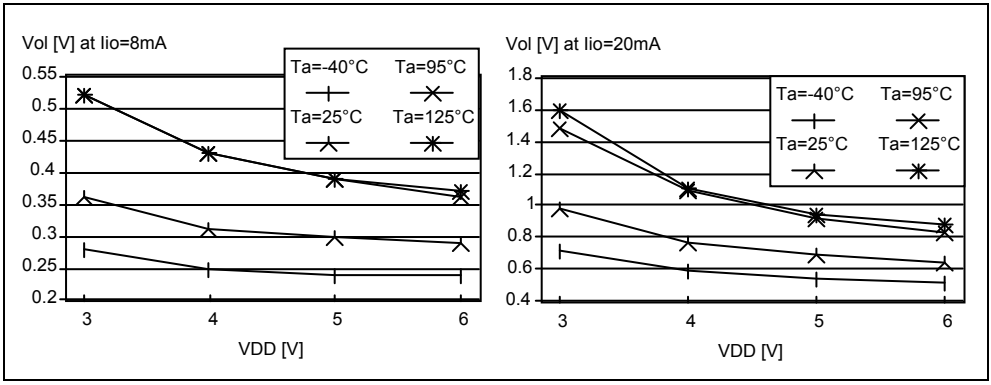


Figure 61. Typical V_{OL} vs V_{DD} (high-sink I/Os)

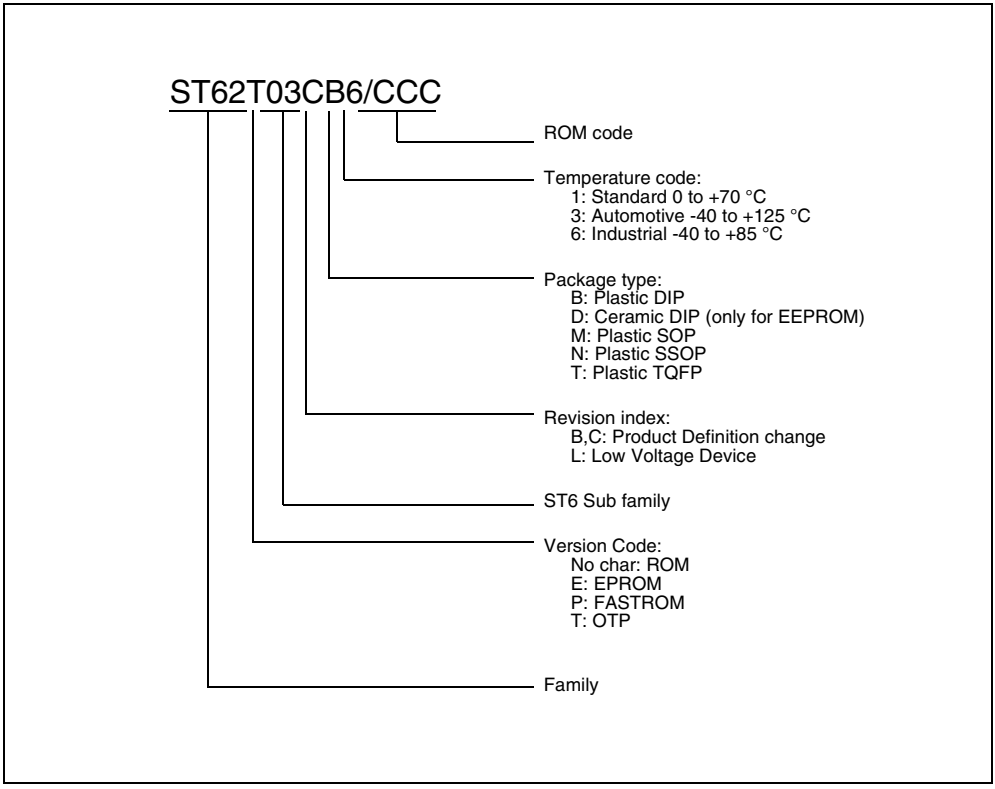


11.5 ORDERING INFORMATION

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

For a list of available options (e.g. memory size, package) and orderable part numbers or for further information on any aspect of this device, please go to www.st.com or contact the ST Sales Office nearest to you.

Figure 72. ST6 Factory Coded Device Types



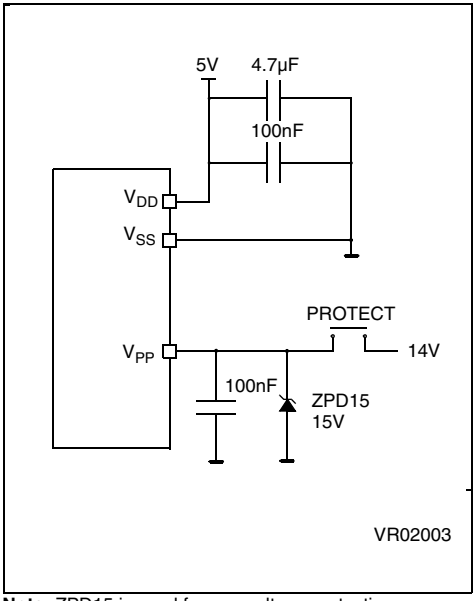
TRANSFER OF CUSTOMER CODE (Cont'd)

11.6.2 ROM VERSION

The ST6200C, 01C and 03C are mask programmed ROM version of ST62T00C, T01 and T03C OTP devices.

They offer the same functionality as OTP devices, selecting as ROM options the options defined in the programmable option byte of the OTP version.

Figure 73. Programming Circuit

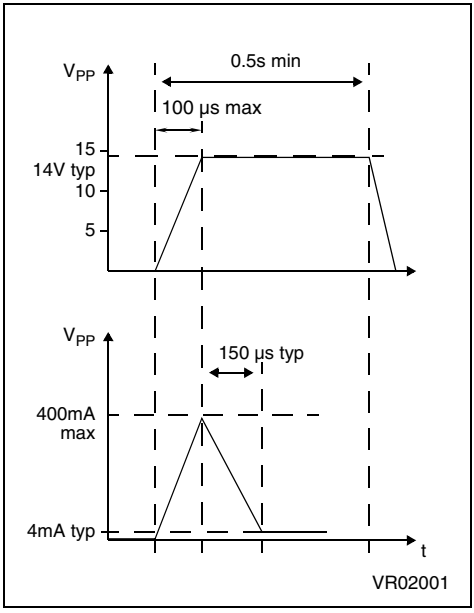


Note: ZPD15 is used for overvoltage protection

ROM Readout Protection. If the ROM READOUT PROTECTION option is selected, a protection fuse can be blown to prevent any access to the program memory content.

In case the user wants to blow this fuse, high voltage must be applied on the V_{PP} pin.

Figure 74. Programming wave form



13 ST6 APPLICATION NOTES

IDENTIFICATION	DESCRIPTION
MOTOR CONTROL	
AN392	MICROCONTROLLER AND TRIACS ON THE 110/240V MAINS
AN414	CONTROLLING A BRUSH DC MOTOR WITH AN ST6265 MCU
AN416	SENSORLESS MOTOR DRIVE WITH THE ST62 MCU + TRIAC
AN422	IMPROVES UNIVERSAL MOTOR DRIVE
AN863	IMPROVED SENSORLESS CONTROL WITH THE ST62 MCU FOR UNIVERSAL MOTOR
BATTERY MANAGEMENT	
AN417	FROM NICD TO NIMH FAST BATTERY CHARGING
AN433	ULTRA FAST BATTERY CHARGER USING ST6210 MICROCONTROLLER
AN859	AN INTELLIGENT ONE HOUR MULTICHARGER FOR Li-Ion, NiMH and NiCd BATTERIES
HOME APPLIANCE	
AN674	MICROCONTROLLERS IN HOME APPLIANCES: A SOFT REVOLUTION
AN885	ST62 MICROCONTROLLERS DRIVE HOME APPLIANCE MOTOR TECHNOLOGY
GRAPHICAL DESIGN	
AN676	BATTERY CHARGER USING THE ST6-REALIZER
AN677	PAINLESS MICROCONTROLLER CODE BY GRAPHICAL APPLICATION DESCRIPTION
AN839	ANALOG MULTIPLE KEY DECODING USING THE ST6-REALIZER
AN840	CODED LOCK USING THE ST6-REALIZER
AN841	A CLOCK DESIGN USING THE ST6-REALIZER
AN842	7 SEGMENT DISPLAY DRIVE USING THE ST6-REALIZER
COST REDUCTION	
AN431	USING ST6 ANALOG INPUTS FOR MULTIPLE KEY DECODING
AN594	DIRECT SOFTWARE LCD DRIVE WITH ST621X AND ST626X
AN672	OPTIMIZING THE ST6 A/D CONVERTER ACCURACY
AN673	REDUCING CURRENT CONSUMPTION AT 32KHZ WITH ST62
DESIGN IMPROVEMENTS	
AN420	EXPANDING A/D RESOLUTION OF THE ST6 A/D CONVERTER
AN432	USING ST62XX I/O PORTS SAFELY
AN434	MOVEMENT DETECTOR CONCEPTS FOR NOISY ENVIRONMENTS
AN435	DESIGNING WITH MICROCONTROLLERS IN NOISY ENVIRONMENTS
AN669	SIMPLE RESET CIRCUITS FOR THE ST6
AN670	OSCILLATOR SELECTION FOR ST62
AN671	PREVENTION OF DATA CORRUPTION IN ST6 ON-CHIP EEPROM
AN911	ST6 MICRO IS EMC CHAMPION
AN975	UPGRADING FROM ST625X/6XB TO ST625X/6XC
AN1015	SOFTWARE TECHNIQUES FOR IMPROVING ST6 EMC PERFORMANCE
PERIPHERAL OPERATIONS	
AN590	PWM GENERATION WITH ST62 AUTO-RELOAD TIMER
AN591	INPUT CAPTURE WITH ST62 AUTO-RELOAD TIMER
AN592	PLL GENERATION USING THE ST62 AUTO-RELOAD TIMER
AN593	ST62 IN-CIRCUIT PROGRAMMING
AN678	LCD DRIVING WITH ST6240

14 SUMMARY OF CHANGES

Description of the changes between the current release of the specification and the previous one.

Revision	Main Changes	Date
3.3	Removed references to 32768 clock cycle delay in Section 5 and in Section 6 Changed note 2 in Section 10.6.2 on page 72: added text on data retention and programmability.	October 2003
4	Updated device summary on page 1 Replaced soldering information by ECOPACK® information in Section 11.3 on page 89 Updated disclaimer on last page	January 2009
5	Added note 5 to Section 10.2.2 on page 59	October 2009

15 TO GET MORE INFORMATION

To get the latest information on this product please use the STMicroelectronics web server.

➟ http://www.st.com/