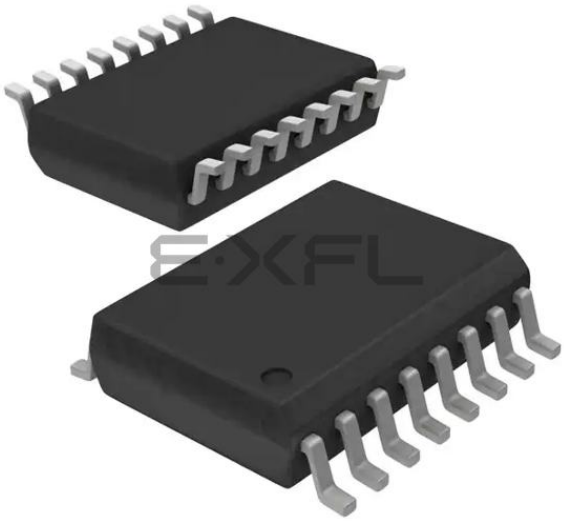


Welcome to E-XFL.COM

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	1KB (1K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	64 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st62t03cm6

Table of Contents

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 CLOCKS, SUPPLY AND RESET

5.1 CLOCK SYSTEM

The main oscillator of the MCU can be driven by any of these clock sources:

– external clock signal

– external AT-cut parallel-resonant crystal

– external ceramic resonator

– external RC network (R_{NET}).

In addition, an on-chip Low Frequency Auxiliary Oscillator (LFAO) is available as a back-up clock system or to reduce power consumption.

An optional Oscillator Safeguard (OSG) filters spikes from the oscillator lines, and switches to the LFAO backup oscillator in the event of main oscillator failure. It 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, and Figure 11.

Table 5 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.

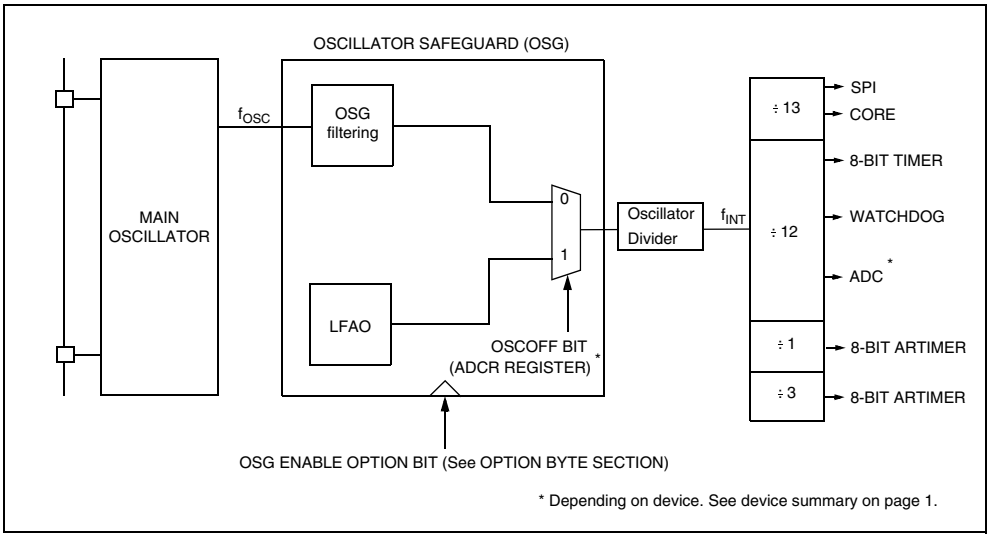
For more details on configuring the clock options, refer to the Option Bytes section of this document.

The internal MCU clock frequency (f_{INT}) is divided by 12 to drive the Timer, the Watchdog timer and the A/D converter, by 13 to drive the CPU core and the SPI and by 1 or 3 to drive the ARTIMER, as shown in Figure 9.

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

A CPU 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 CPU cycles for execution.

Figure 9. Clock Circuit Block Diagram



CLOCK SYSTEM (Cont'd)

5.1.1 Main Oscillator

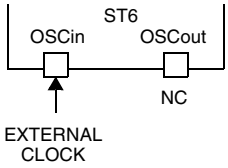
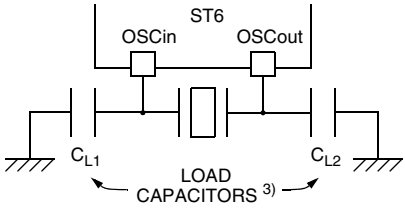
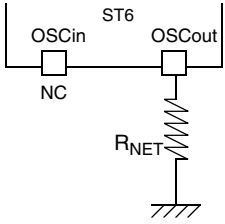
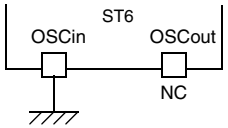
The oscillator configuration is specified by selecting the appropriate option in the option bytes (refer to the Option Bytes section of this document). When the CRYSTAL/RESONATOR option is selected, it must be used with a quartz crystal, a ceramic resonator or an external signal provided on the OSCin 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 5. 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.

RESET (Cont'd)

5.3.4 Watchdog Reset

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

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

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

For more details refer to the Watchdog Timer chapter.

5.3.5 LVD Reset

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

- Power-On RESET
- Voltage Drop RESET

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

For more details, refer to the LVD chapter.

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

Figure 15. Simple External Reset Circuitry

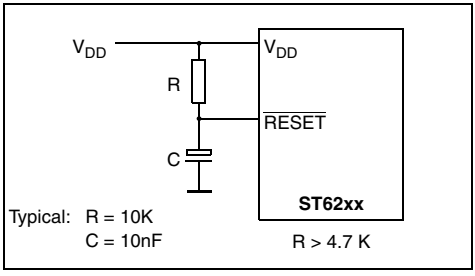
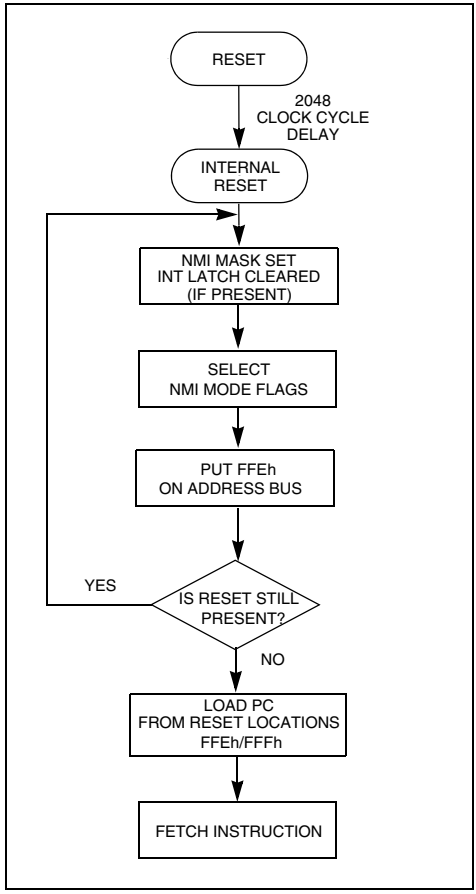


Figure 16. Reset Processing



5.9 EXTERNAL INTERRUPTS (I/O Ports)

External interrupt vectors can be loaded into the PC register if the corresponding external interrupt occurred and if the GEN bit is set. These interrupts allow the processor to exit from STOP mode.

The external interrupt polarity is selected through the IOR register.

External interrupts are linked to vectors #1 and # 2.

Interrupt requests on vector #1 can be configured either as edge or level-sensitive using the LES bit in the IOR Register.

Interrupt requests from vector #2 are always edge sensitive. The edge polarity can be configured using the ESB bit in the IOR Register.

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, an interrupt request can be stored until completion of the currently executing interrupt routine, before being processed. If several interrupt requests occurs before completion of the current interrupt routine, only the first request is stored.

Storing of interrupt requests is not possible 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.

5.9.1 Notes on using External Interrupts

ESB bit Spurious Interrupt on Vector #2

If a pin associated with interrupt vector #2 is configured as interrupt with pull-up, whenever vector #2 is configured to be rising edge sensitive (by setting the ESB bit in the IOR register), an interrupt is latched although a rising edge may not have occured on the associated pin.

This is due to the vector #2 circuitry.The workaround is to discard this first interrupt request in the routine (using a flag for example).

Masking of One Interrupt by Another on Vector #2.

When two or more port pins (associated with interrupt vector #2) are configured together as input with interrupt (falling edge sensitive), as long as one pin is stuck at '0', the other pin can never generate an interrupt even if an active edge occurs at this pin. The same thing occurs when one pin is stuck at '1' and interrupt vector #2 is configured as rising edge sensitive.

To avoid this the first pin must input a signal that goes back up to '1' right after the falling edge. Otherwise, in the interrupt routine for the first pin, deactivate the “input with interrupt” mode using the port control registers (DDR, OR, DR). An active edge on another pin can then be latched.

I/O port Configuration Spurious Interrupt on Vector #2

If a pin associated with interrupt vector #2 is in ‘input with pull-up’ state, a ‘0’ level is present on the pin and the ESB bit = 0, when the I/O pin is configured as interrupt with pull-up by writing to the DDRx, ORx and DRx register bits, an interrupt is latched although a falling edge may not have occurred on the associated pin.

In the opposite case, if the pin is in interrupt with pull-up state , a 0 level is present on the pin and the ESB bit =1, when the I/O port is configured as input with pull-up by writing to the DDRx, ORx and DRx bits, an interrupt is latched although a rising edge may not have occurred on the associated pin.

6.3 STOP MODE

STOP mode is the lowest power consumption mode of the MCU (see Figure 22).

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

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

– The contents of RAM and the peripheral registers are kept safely as long as the power supply voltage is higher than the RAM retention voltage.

– The oscillator is stopped, so peripherals cannot work except the those that can be driven by an external clock.

Exit from STOP Mode

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

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

– A peripheral interrupt (assuming this peripheral can be driven by an external clock)

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

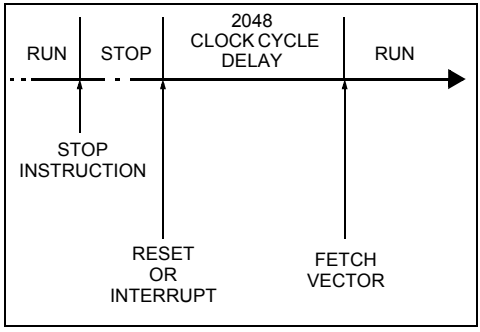
In all cases a delay of 2048 clock cycles (f_{INT}) is generated to make sure the oscillator has started properly.

The Program Counter then points to the starting address of the interrupt or RESET service routine (see Figure 21).

STOP Mode and Watchdog

When the Watchdog is active (hardware or software activation), the STOP instruction is disabled and a WAIT instruction will be executed in its place unless the EXCTNL option bit is set to 1 in the option bytes and a a high level is present on the NMI pin. In this case, the STOP instruction will be executed and the Watchdog will be frozen.

Figure 21. STOP Mode Timing Overview



7 I/O PORTS

7.1 INTRODUCTION

Each I/O port contains up to 8 pins. Each pin can be programmed independently as digital input (with or without pull-up and interrupt generation), digital output (open drain, push-pull) or analog input (when available).

The I/O pins can be used in either standard or alternate function mode.

Standard I/O mode is used for:

– Transfer of data through digital inputs and outputs (on specific pins):

– External interrupt generation

Alternate function mode is used for:

– Alternate signal input/output for the on-chip peripherals

The generic I/O block diagram is shown in Figure 23.

7.2 FUNCTIONAL DESCRIPTION

Each port is associated with 3 registers located in Data space:

– Data Register (DR)

– Data Direction Register (DDR)

– Option Register (OR)

Each I/O pin may be programmed using the corresponding register bits in the DDR, DR and OR registers: bit x corresponding to pin x of the port. Table 8 illustrates the various port configurations which can be selected by user software.

During MCU initialization, all I/O registers are cleared and the input mode with pull-up and no interrupt generation is selected for all the pins, thus avoiding pin conflicts.

7.2.1 Digital Input Modes

The input configuration is selected by clearing the corresponding DDR register bit.

In this case, reading the DR register returns the digital value applied to the external I/O pin.

Different input modes can be selected by software through the DR and OR registers, see Table 8.

External Interrupt Function

All input lines can be individually connected by software to the interrupt system by programming the OR and DR registers accordingly. The interrupt trigger modes (falling edge, rising edge and low level) can be configured by software for each port as described in the Interrupt section.

7.2.2 Analog Inputs

Some pins can be configured as analog inputs by programming the OR and DR registers accordingly, see Table 8. These analog inputs are connected to the on-chip 8-bit Analog to Digital Converter.

Caution: ONLY ONE pin should be programmed as an analog input at any time, since by selecting more than one input simultaneously their pins will be effectively shorted.

7.2.3 Output Modes

The output configuration is selected by setting the corresponding DDR register bit. In this case, writing to the DR register applies this digital value to the I/O pin through the latch. Then, reading the DR register returns the previously stored value.

Two different output modes can be selected by software through the OR register: push-pull and open-drain.

DR register value and output pin status:

DR	Push-pull	Open-drain
0	V_{SS}	V_{SS}
1	V_{DD}	Floating

Note: The open drain setting is not a true open drain. This means it has the same structure as the push-pull setting but the P-buffer is deactivated. To avoid damaging the device, please respect the V_{OUT} absolute maximum rating described in the Electrical Characteristics section.

7.2.4 Alternate Functions

When an on-chip peripheral is configured to use a pin, the alternate function (timer input/output...) is not systematically selected but has to be configured through the DDR, OR and DR registers. Refer to the chapter describing the peripheral 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

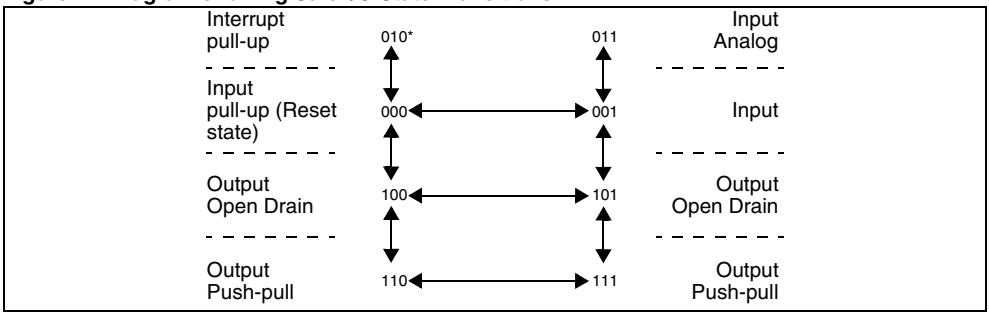
The WAIT and STOP instructions allow the ST62xx to be used in situations where low power consumption is needed. The lowest power consumption is achieved by configuring I/Os in output push-pull low mode.

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

WATCHDOG TIMER (Cont'd)

These instructions test the C bit and reset the MCU (i.e. disable the Watchdog) if the bit is set (i.e. if the Watchdog is active), thus disabling the Watchdog.

For more information on the use of the watchdog, please read application note AN1015.

Note: This note applies only when the watchdog is used as a standard timer. It is recommended to read the counter twice, as it may sometimes return an invalid value if the read is performed while the counter is decremented (counter bits in transient state). To validate the return value, both values read must be equal. The counter decrements every 384 µs at 8 MHz f_{OSC}.

8.1.5 Low Power Modes

Mode	Description
WAIT	No effect on Watchdog.
STOP	Behaviour depends on the EXTCNTL option in the Option bytes: **1. Watchdog disabled:** The MCU will enter Stop mode if a STOP instruction is executed. **2. Watchdog enabled and EXTCNTL option disabled:** If a STOP instruction is encountered, it is interpreted as a WAIT. **3. Watchdog and EXTCNTL option enabled:** If a STOP instruction is encountered when the NMI pin is low, it is interpreted as a WAIT. If, however, the STOP instruction is encountered when the NMI pin is high, the Watchdog counter is frozen and the CPU enters STOP mode. When the MCU exits STOP mode (i.e. when an interrupt is generated), the Watchdog resumes its activity.

8.1.6 Interrupts

None.

Opcode Map Summary (Continued)

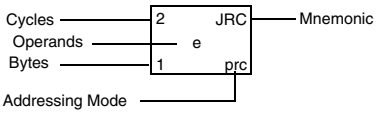
LOW / HI	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	LOW / HI
0 **0000**	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 RES b0,rr 2 b.d	2 JRZ e 1 pcr	4 LDI rr,nn 3 imm	2 JRC e 1 prc	4 LD a,(y) 1 ind	**0** **0000**
1 **0001**	2 JRNZ e 1 pcr	4 JP abc 2 ext	2 JRNC e 1 pcr	4 SET b0,rr 2 b.d	2 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	# 1 prc	2 JRC e 1 prc	4 LD (y),a 1 ind	**8** **1000**
9 **1001**	2 JRNZ 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

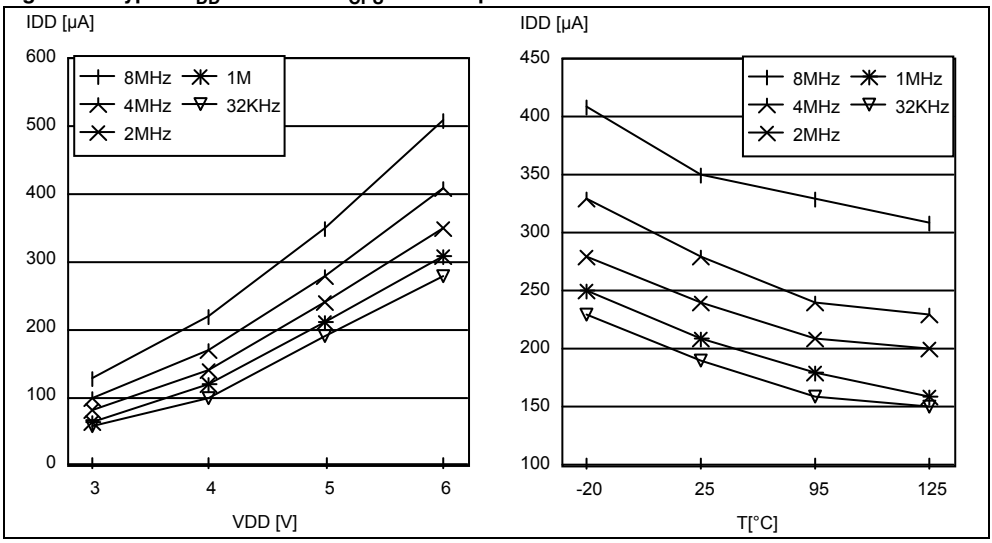
Legend:

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



SUPPLY CURRENT CHARACTERISTICS (Cont'd)

Figure 40. Typical I_{DD} in WAIT vs f_{CPU} and Temperature for ROM devices



CLOCK AND TIMING CHARACTERISTICS (Cont'd)

10.5.3 Crystal and Ceramic Resonator Oscillators

The ST6 internal clock can be supplied with several different Crystal/Ceramic resonator oscillators. Only parallel resonant crystals can be used. All the information given in this paragraph are based on characterization results with specified typical external components. Refer to the crystal/ceramic resonator manufacturer for more details (frequency, package, accuracy...).

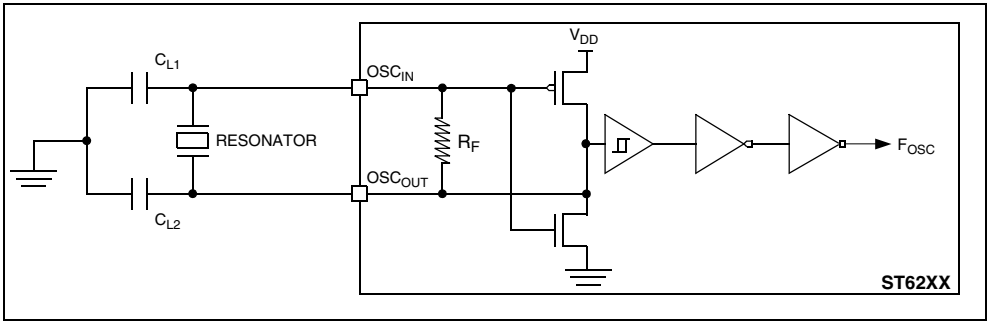
Symbol	Parameter	Conditions	Typ	Unit
R_F	Feedback resistor		3	MΩ
C_{L1} C_{L2}	Recommended load capacitances versus equivalent crystal or ceramic resonator frequency	f_{OSC}=32 kHz, f_{OSC}=1 MHz f_{OSC}=2 MHz f_{OSC}=4 MHz f_{OSC}=8 MHz	120 47 33 33 22	pF

Oscillator		Typical Crystal or Ceramic Resonators			C_{L1}	C_{L2}	$t_{SU(osc)}$
		Reference	Freq.	Characteristic [1]	[pF]	[pF]	[ms] [1]
Ceramic	MURATA	CSB455E	455KHz	Δf_{OSC}=[±0.5KHz$_{tolerance}$,±0.3%$_{\Delta Ta}$, ±0.5%$_{aging}$]	220	220	
		CSB1000J	1MHz	Δf_{OSC}=[±0.5KHz$_{tolerance}$,±0.3%$_{\Delta Ta}$, ±0.5%$_{aging}$]	100	100	
		CSTCC2.00MG0H6	2MHz	Δf_{OSC}=[±0.5%$_{tolerance}$,±0.5%$_{\Delta Ta}$, ±0.3%$_{aging}$]	47	47	
		CSTCC4.00MG0H6	4MHz	Δf_{OSC}=[±0.5%$_{tolerance}$,±0.3%$_{\Delta Ta}$, ±0.3%$_{aging}$]	47	47	
		CSTCC8.00MG	8MHz	Δf_{OSC}=[±0.5%$_{tolerance}$,±0.3%$_{\Delta Ta}$, ±0.3%$_{aging}$]	15	15	

Notes:

1. Resonator characteristics given by the crystal/ceramic resonator manufacturer.
2. $t_{SU(OSC)}$ is the typical oscillator start-up time measured between V_{DD}=2.8V and the fetch of the first instruction (with a quick V_{DD} ramp-up from 0 to 5V (<50µs).
3. The oscillator selection can be optimized in terms of supply current using an high quality resonator with small R_S value. Refer to crystal/ceramic resonator manufacturer for more details.

Figure 44. Typical Application with a Crystal or Ceramic Resonator



10.6 MEMORY CHARACTERISTICS

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

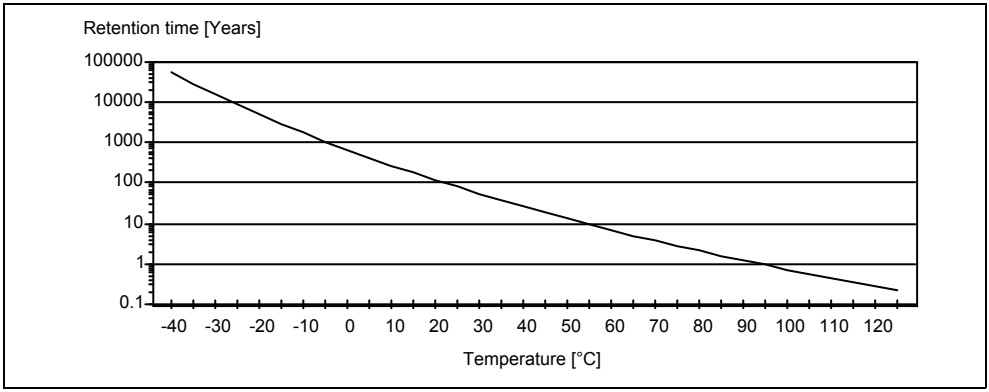
10.6.1 RAM and Hardware Registers

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention[1)]		0.7			V

10.6.2 EPROM Program Memory

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{ret}	Data retention [2)]	T_A=+55°C [3)]	10			years

Figure 49. EPROM Retention Time vs. Temperature



Notes:

1. Minimum V_{DD} supply voltage without losing data stored in RAM (in STOP mode or under RESET) or in hardware registers (only in STOP mode). Guaranteed by construction, not tested in production.
2. Data based on reliability test results and monitored in production. For OTP devices, data retention and programmability must be guaranteed by a screening procedure. Refer to Application Note AN886.
3. The data retention time increases when the T_A decreases, see Figure 49.

I/O PORT PIN CHARACTERISTICS (Cont'd)

10.8.2 Output Driving Current

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

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL} 1)	Output low level voltage for a standard I/O pin (see Figure 57 and Figure 60)	V_{DD}=5V	I_{IO}=+10µA, $T_A \leq 125°C$		0.1	V
			I_{IO}=+3mA, $T_A \leq 125°C$		0.8	
			I_{IO}=+5mA, $T_A \leq 85°C$		0.8	
			I_{IO}=+10mA, $T_A \leq 85°C$		1.2	
	Output low level voltage for a high sink I/O pin (see Figure 58 and Figure 61)		I_{IO}=+10µA, $T_A \leq 125°C$		0.1	
			I_{IO}=+7mA, $T_A \leq 125°C$		0.8	
			I_{IO}=+10mA, $T_A \leq 85°C$		0.8	
			I_{IO}=+15mA, $T_A \leq 125°C$		1.3	
			I_{IO}=+20mA, $T_A \leq 85°C$		1.3	
			I_{IO}=+30mA, $T_A \leq 85°C$		2	
V_{OH} 2)	Output high level voltage for an I/O pin (see Figure 59 and Figure 62)		I_{IO}=-10µA, $T_A \leq 125°C$	V_{DD}-0.1		
			I_{IO}=-3mA, $T_A \leq 125°C$	V_{DD}-1.5		
			I_{IO}=-5mA, $T_A \leq 85°C$	V_{DD}-1.5		

Notes:

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in Section 10.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
2. The I_{IO} current source must always respect the absolute maximum rating specified in Section 10.2.2 and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}. True open drain I/O pins does not have V_{OH}.

Figure 57. Typical V_{OL} at V_{DD} = 5V (standard)

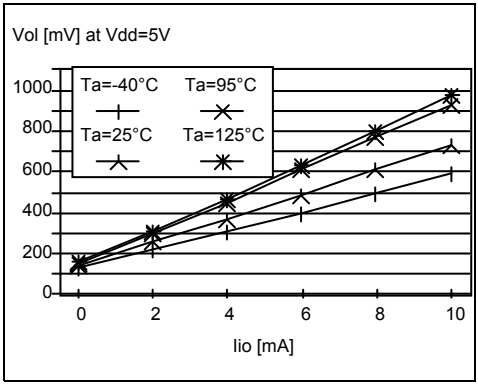
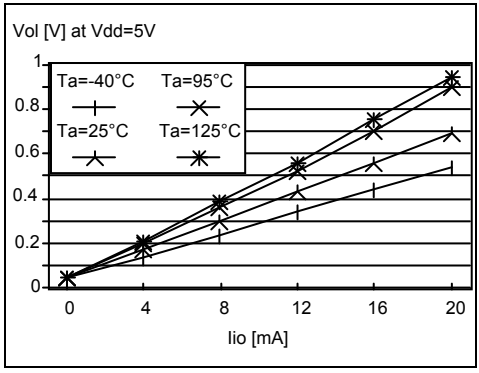


Figure 58. Typical V_{OL} at V_{DD} = 5V (high-sink)



10.9 CONTROL PIN CHARACTERISTICS

10.9.1 Asynchronous $\overline{\text{RESET}}$ Pin

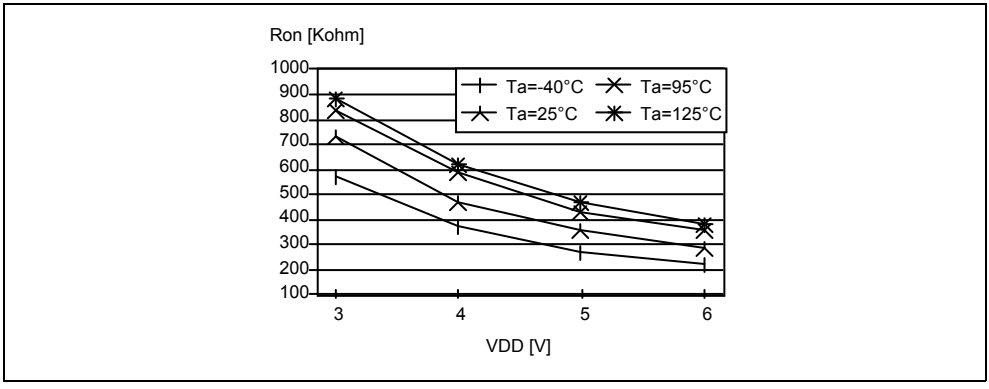
Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ [1]	Max	Unit
V_{IL}	Input low level voltage [2]					$0.3xV_{DD}$	V
V_{IH}	Input high level voltage [2]			$0.7xV_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis [3]			200	400		mV
R_{ON}	Weak pull-up equivalent resistor [4]	$V_{IN}=V_{SS}$	V_{DD}=5V	150	350	900	kΩ
			V_{DD}=3.3V	300	730	1900	
R_{ESD}	ESD resistor protection	$V_{IN}=V_{SS}$	V_{DD}=5V		2.8		kΩ
			V_{DD}=3.3V				
$t_{w(RSTL)out}$	Generated reset pulse duration	External pin or internal reset sources					t_{CPU} µs
$t_{h(RSTL)in}$	External reset pulse hold time [5]						µs
$t_{g(RSTL)in}$	Filtered glitch duration [6]						ns

Notes:

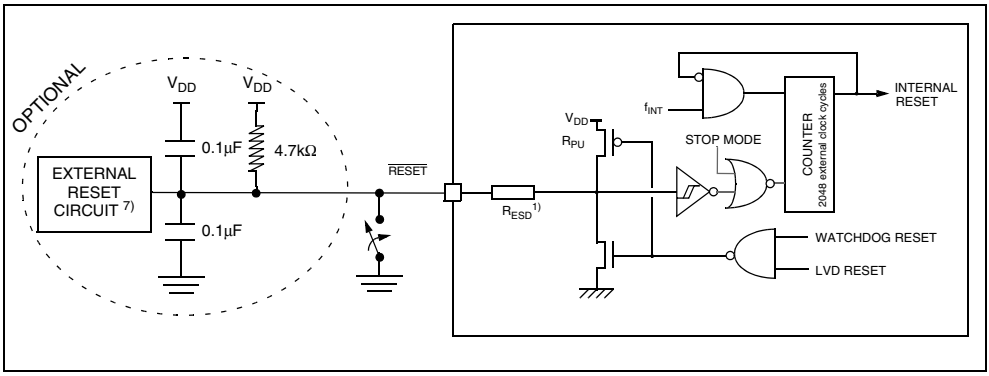
1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}=5V.
2. Data based on characterization results, not tested in production.
3. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.
4. The R_{ON} pull-up equivalent resistor is based on a resistive transistor. This data is based on characterization results, not tested in production.
5. All short pulse applied on $\overline{\text{RESET}}$ pin with a duration below $t_{h(RSTL)in}$ can be ignored.
6. The reset network protects the device against parasitic resets, especially in a noisy environment.
7. The output of the external reset circuit must have an open-drain output to drive the ST6 reset pad. Otherwise the device can be damaged when the ST6 generates an internal reset (LVD or watchdog).

Figure 63. Typical R_{ON} vs V_{DD} with $V_{IN}=V_{SS}$



CONTROL PIN CHARACTERISTICS (Cont'd)

Figure 64. Typical Application with $\overline{\text{RESET}}$ pin [8)]



10.9.2 NMI Pin

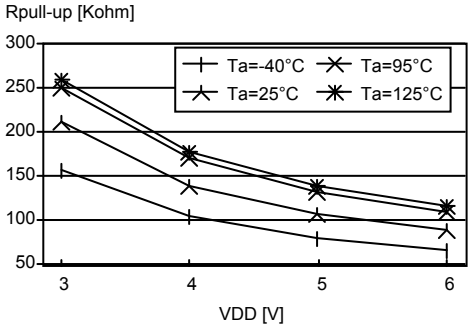
Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Symbol	Parameter	Conditions		Min	Typ [1)]	Max	Unit
V_{IL}	Input low level voltage [2)]					$0.3xV_{DD}$	V
V_{IH}	Input high level voltage [2)]			$0.7xV_{DD}$			
V_{hys}	Schmitt trigger voltage hysteresis [3)]			200	400		mV
$R_{pull-up}$	Weak pull-up equivalent resistor [4)]	$V_{IN}=V_{SS}$	V_{DD}=5V	40	100	350	kΩ
			V_{DD}=3.3V	80	200	700	

Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}=5V.
2. Data based on characterization results, not tested in production.
3. Hysteresis voltage between Schmitt trigger switching levels. Based on characterization results, not tested.
4. The $R_{pull-up}$ equivalent resistor is based on a resistive transistor. This data is based on characterization results, not tested in production.

Figure 65. Typical $R_{pull-up}$ vs. V_{DD} with $V_{IN}=V_{SS}$

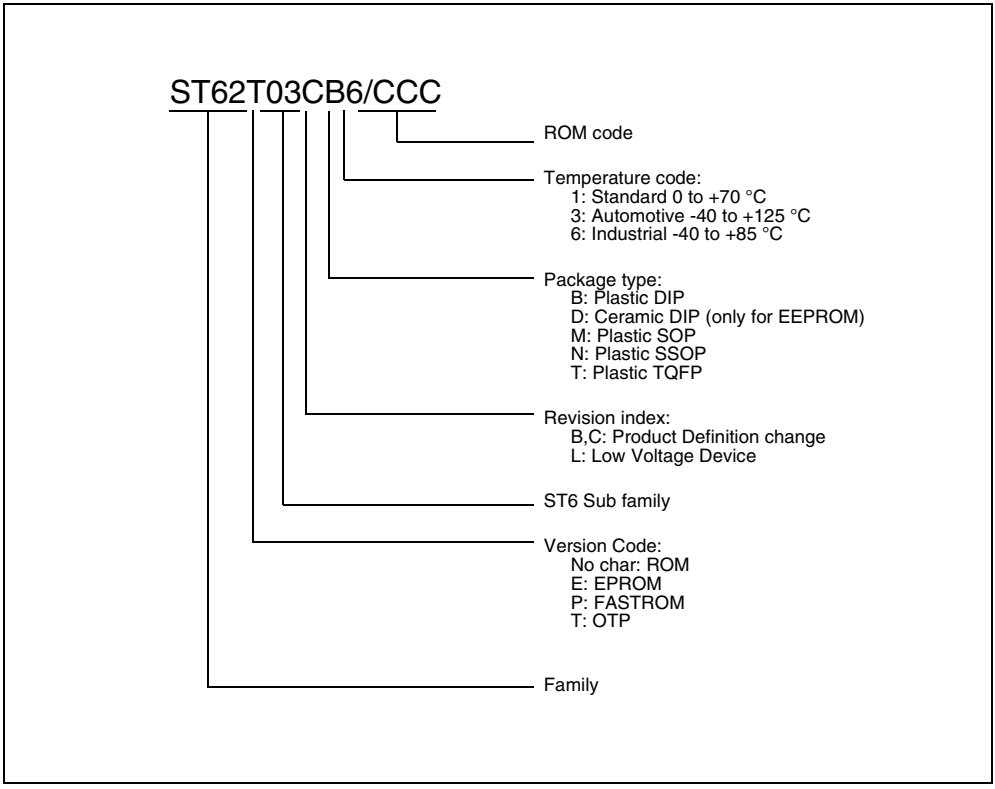


11.5 ORDERING INFORMATION

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



11.6 TRANSFER OF CUSTOMER CODE

Customer code is made up of the ROM contents and the list of the selected FASTROM options. The ROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected options are communicated to STMicroelectronics using the correctly filled OPTION LIST appended. See page 94.

The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.

Listing Generation and Verification. When STMicroelectronics receives the user's ROM contents, a computer listing is generated from it. This listing refers exactly to the ROM contents and options which will be used to produce the specified MCU. The listing is then returned to the customer who must thoroughly check, complete, sign and return it to STMicroelectronics. The signed listing forms a part of the contractual agreement for the production of the specific customer MCU.

11.6.1 FASTROM version

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. The customer code must be sent to STMicroelectronics in the same way as for ROM devices. The FASTROM option list has the same options as defined in the programmable option byte of the OTP version. It also offers an identifier option. If this option is enabled, each FASTROM device is programmed with a unique 5-byte number which is mapped at addresses 0F9Bh-0F9Fh. The user must therefore leave these bytes blanked.

The identification number is structured as follows:

0F9Bh	T0
0F9Ch	T1
0F9Dh	T2
0F9Eh	T3
0F9Fh	Test ID

with T0, T1, T2, T3 = time in seconds since 01/01/1970 and Test ID = Tester Identifier.

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