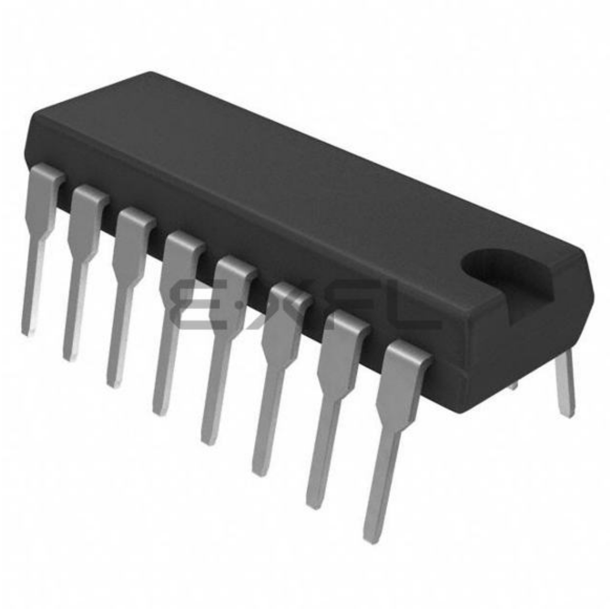


Welcome to E-XFL.COM

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	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	11
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 5x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°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/st7flit10by1b6

1 INTRODUCTION

The ST7LITE1xB is a member of the ST7 microcontroller family. All ST7 devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set.

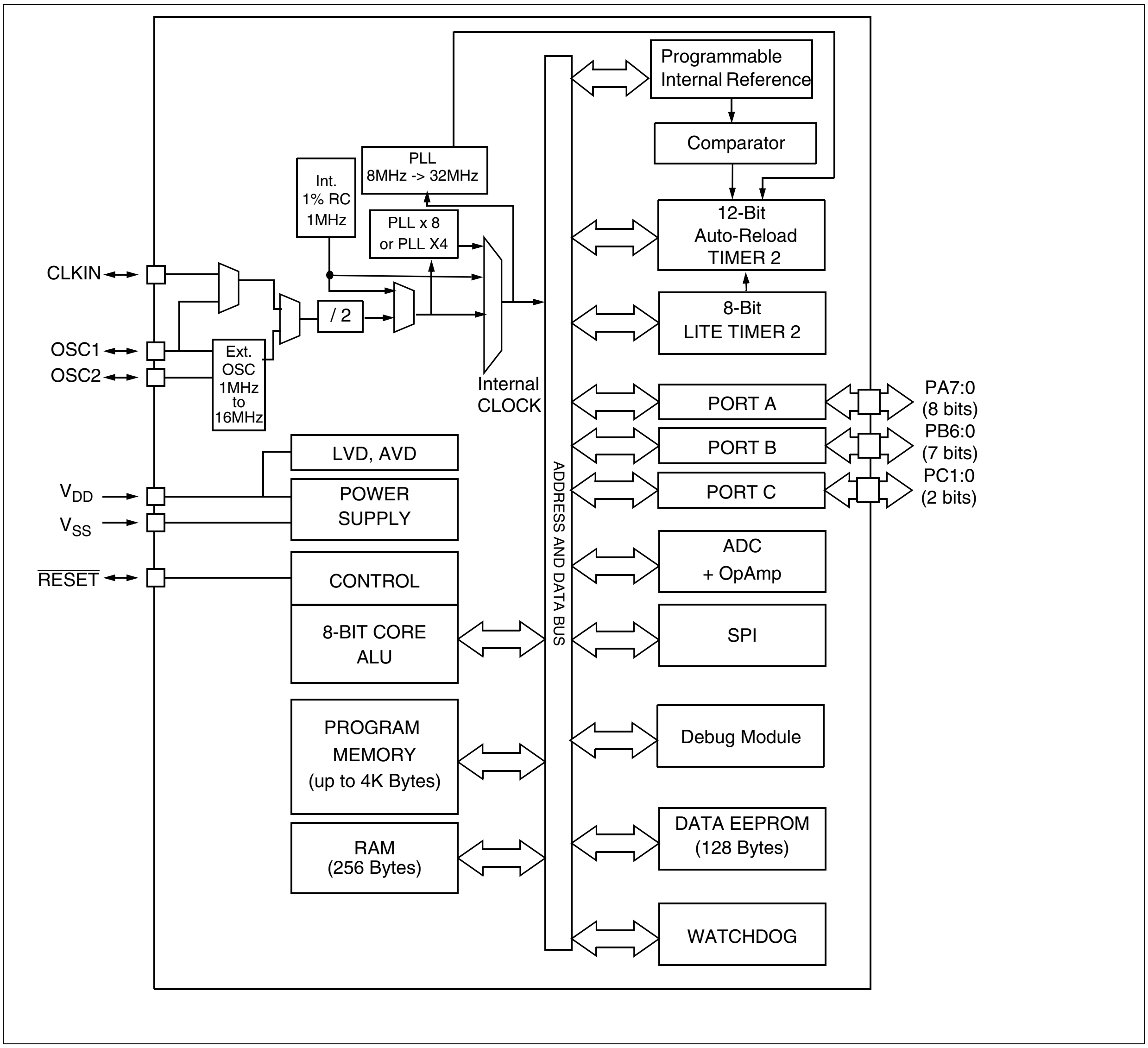
The ST7LITE1xB features FLASH memory with byte-by-byte In-Circuit Programming (ICP) and In-Application Programming (IAP) capability.

Under software control, the ST7LITE1xB device can be placed in WAIT, SLOW, or HALT mode, reducing power consumption when the application is in idle or standby state.

The enhanced instruction set and addressing modes of the ST7 offer both power and flexibility to software developers, enabling the design of highly efficient and compact application code. In addition to standard 8-bit data management, all ST7 microcontrollers feature true bit manipulation, 8x8 unsigned multiplication and indirect addressing modes.

For easy reference, all parametric data are located in section 13 on page 110. The ST7LITE1xB features an on-chip Debug Module (DM) to support In-Circuit Debugging (ICD). For a description of the DM registers, refer to the ST7 ICC Protocol Reference Manual.

Figure 1. General Block Diagram



2 PIN DESCRIPTION

Figure 2. 20-Pin SO and DIP Package Pinout

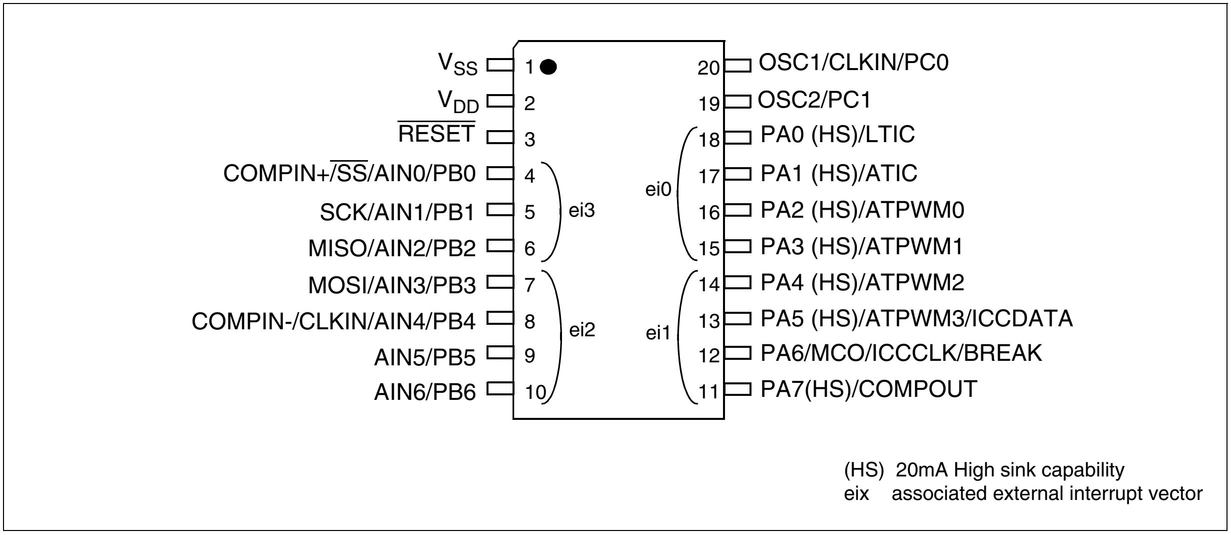
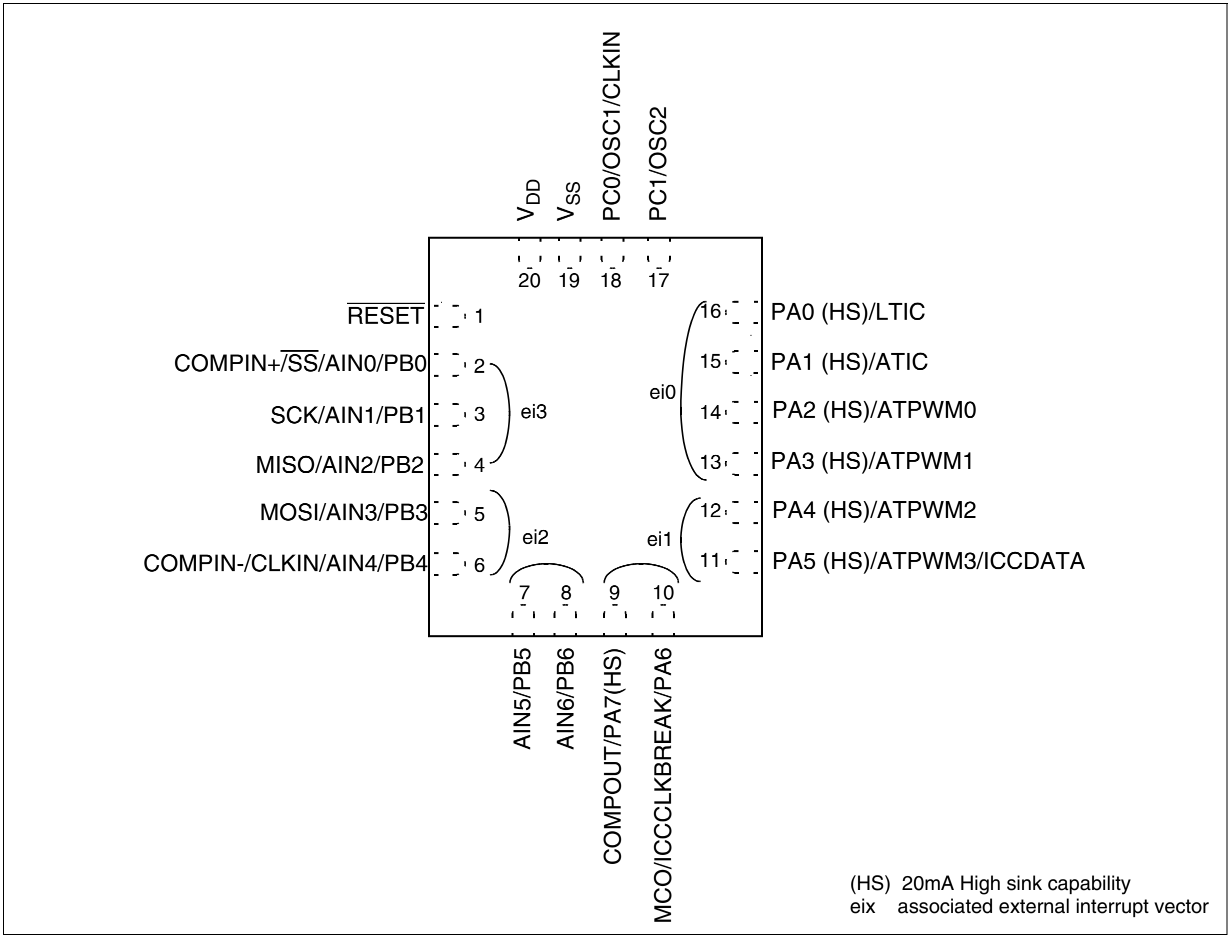


Figure 3. 20-Pin QFN Package Pinout



DATA EEPROM (Cont'd)

5.4 POWER SAVING MODES

Wait mode

The DATA EEPROM can enter WAIT mode on execution of the WFI instruction of the microcontroller or when the microcontroller enters Active-HALT mode.The DATA EEPROM will immediately enter this mode if there is no programming in progress, otherwise the DATA EEPROM will finish the cycle and then enter WAIT mode.

Active-Halt mode

Refer to Wait mode.

Halt mode

The DATA EEPROM immediately enters HALT mode if the microcontroller executes the HALT instruction. Therefore the EEPROM will stop the function in progress, and data may be corrupted.

5.5 ACCESS ERROR HANDLING

If a read access occurs while E2LAT=1, then the data bus will not be driven.

If a write access occurs while E2LAT=0, then the data on the bus will not be latched.

If a programming cycle is interrupted (by a RESET action), the integrity of the data in memory will not be guaranteed.

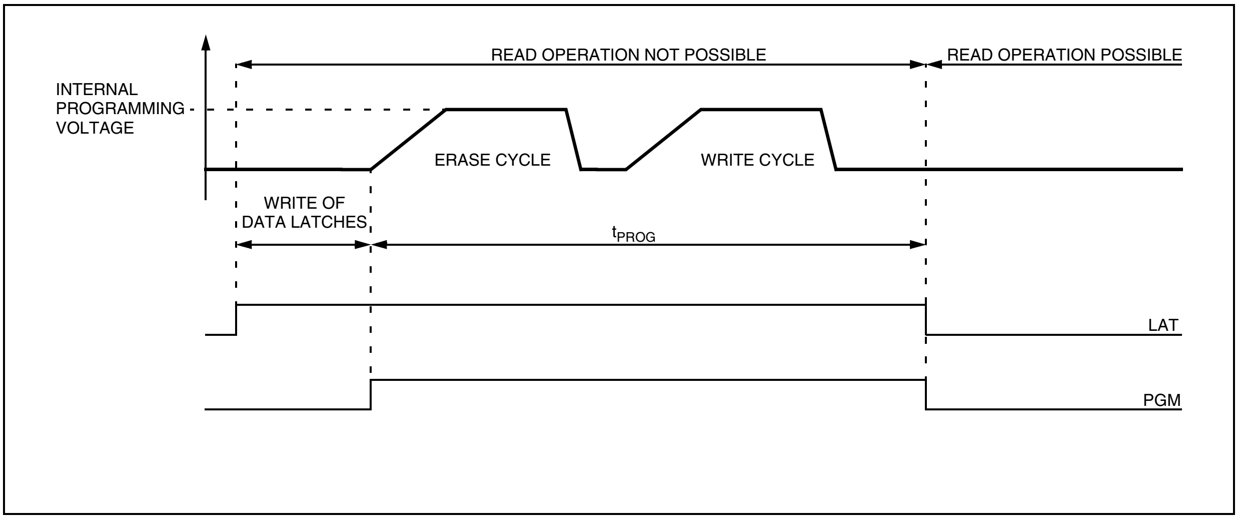
5.6 Data EEPROM Read-out Protection

The read-out protection is enabled through an option bit (see option byte section).

When this option is selected, the programs and data stored in the EEPROM memory are protected against read-out (including a re-write protection). In Flash devices, when this protection is removed by reprogramming the Option Byte, the entire Program memory and EEPROM is first automatically erased.

Note: Both Program Memory and data EEPROM are protected using the same option bit.

Figure 10. Data EEPROM Programming Cycle



7.3 REGISTER DESCRIPTION

MAIN CLOCK CONTROL/STATUS REGISTER (MCCSR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	MCO	SMS

Bits 7:2 = Reserved, must be kept cleared.

Bit 1 = **MCO** *Main Clock Out enable*
This bit is read/write by software and cleared by hardware after a reset. This bit allows to enable the MCO output clock.
0: MCO clock disabled, I/O port free for general purpose I/O.
1: MCO clock enabled.

Bit 0 = **SMS** *Slow Mode select*
This bit is read/write by software and cleared by hardware after a reset. This bit selects the input clock f_{OSC} or $f_{OSC}/32$.
0: Normal mode ($f_{CPU} = f_{OSC}$
1: Slow mode ($f_{CPU} = f_{OSC}/32$)

RC CONTROL REGISTER (RCCR)

Read / Write

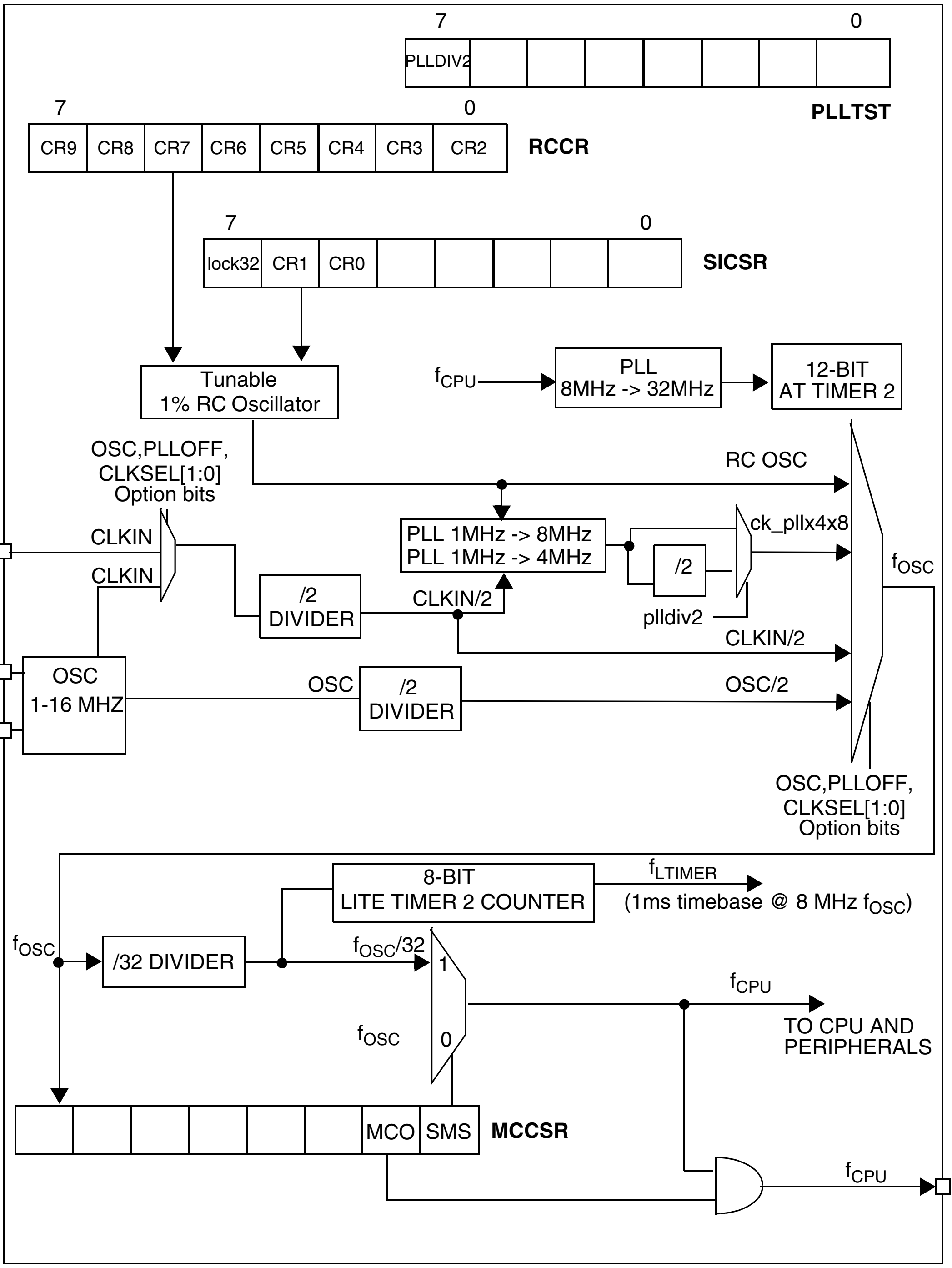
Reset Value: 1111 1111 (FFh)

7							0
CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2

Bits 7:0 = **CR[9:2]** *RC Oscillator Frequency Adjustment Bits*
These bits must be written immediately after reset to adjust the RC oscillator frequency and to obtain an accuracy of 1%. The application can store the correct value for each voltage range in EEPROM and write it to this register at start-up.
00h = maximum available frequency
FFh = lowest available frequency
These bits are used with the CR[1:0] bits in the SICSR register. Refer to section 7.6.4 on page 35.

Note: To tune the oscillator, write a series of different values in the register until the correct frequency is reached. The fastest method is to use a dichotomy starting with 80h.

Figure 14. Clock Management Block Diagram



Note: The PLL cannot be used with the external resonator oscillator

7.6 SYSTEM INTEGRITY MANAGEMENT (SI)

The System Integrity Management block contains the Low voltage Detector (LVD) and Auxiliary Voltage Detector (AVD) functions. It is managed by the SICSR register.

Note: A reset can also be triggered following the detection of an illegal opcode or prebyte code. Refer to section 12.2.1 on page 107 for further details.

7.6.1 Low Voltage Detector (LVD)

The Low Voltage Detector function (LVD) generates a static reset when the V_{DD} supply voltage is below a $V_{IT-(LVD)}$ reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

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

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

– $V_{IT+(LVD)}$when V_{DD} is rising

– $V_{IT-(LVD)}$ when V_{DD} is falling

The LVD function is illustrated in Figure 18.

The voltage threshold can be configured by option byte to be low, medium or high.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is above $V_{IT-(LVD)}$, the MCU can only be in two modes:

– under full software control

– in static safe reset

In these conditions, secure operation is always ensured for the application without the need for external reset hardware.

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

Notes:

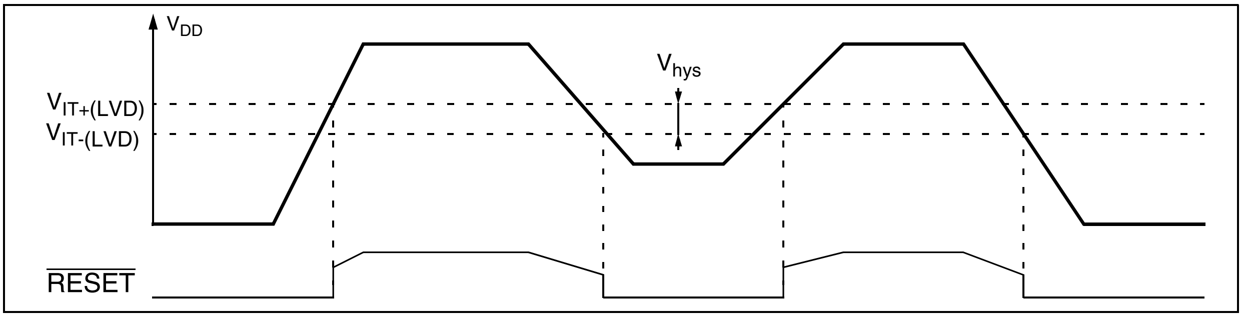
The LVD allows the device to be used without any external RESET circuitry.

The LVD is an optional function which can be selected by option byte.

Use of LVD with capacitive power supply: with this type of power supply, if power cuts occur in the application, it is recommended to pull V_{DD} down to 0V to ensure optimum restart conditions. Refer to circuit example in Figure 106 on page 136 and note 4.

It is recommended to make sure that the V_{DD} supply voltage rises monotonously when the device is exiting from Reset, to ensure the application functions properly.

Figure 18. Low Voltage Detector vs Reset



INTERRUPTS (Cont'd)

Bit 3:2 = **ei1[1:0]** *ei1 pin selection*
These bits are written by software. They select the Port A I/O pin used for the ei1 external interrupt according to the table below.

External Interrupt I/O pin selection

ei11	ei10	I/O Pin
0	0	PA4
0	1	PA5
1	0	PA6
1	1	PA7*

* Reset State

Bit 1:0 = **ei0[1:0]** *ei0 pin selection*
These bits are written by software. They select the Port A I/O pin used for the ei0 external interrupt according to the table below.

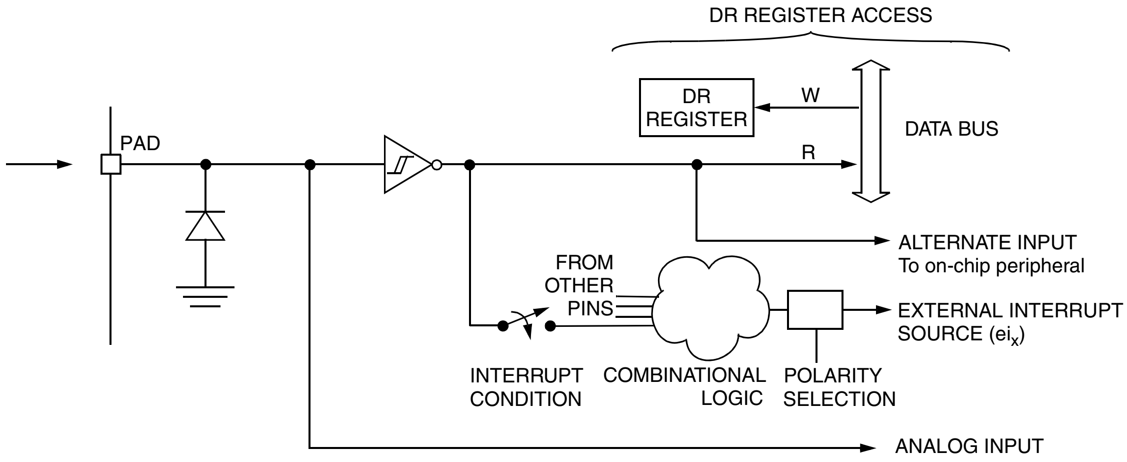
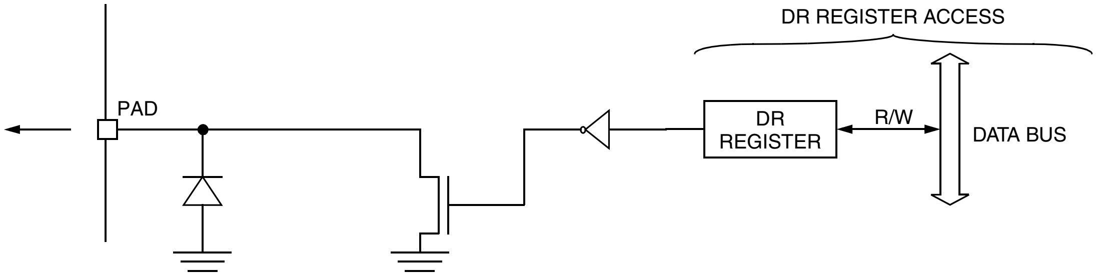
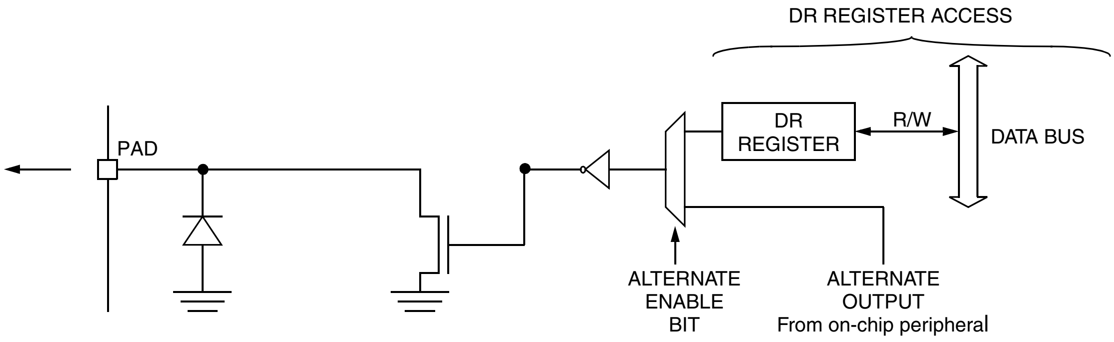
External Interrupt I/O pin selection

ei01	ei00	I/O Pin
0	0	PA0 *
0	1	PA1
1	0	PA2
1	1	PA3

* Reset State

I/O PORTS (Cont'd)

Table 9. I/O Configurations

	Hardware Configuration
INPUT 1)	 DR REGISTER ACCESS DR REGISTER W R DATA BUS PAD ALTERNATE INPUT To on-chip peripheral FROM OTHER PINS EXTERNAL INTERRUPT SOURCE (ei_x) INTERRUPT CONDITION COMBINATIONAL LOGIC POLARITY SELECTION ANALOG INPUT
OPEN-DRAIN OUTPUT 2)	
PUSH-PULL OUTPUT 2)	 DR REGISTER ACCESS DR REGISTER R/W DATA BUS PAD ALTERNATE ENABLE BIT ALTERNATE OUTPUT From on-chip peripheral

Notes:

1. When the I/O port is in input configuration and the associated alternate function is enabled as an output, reading the DR register will read the alternate function output status.
2. When the I/O port is in output configuration and the associated alternate function is enabled as an input, the alternate function reads the pin status given by the DR register content.

DUAL 12-BIT AUTORELOAD TIMER 4 (Cont'd)

11.2.3 Functional Description

11.2.3.1 PWM Mode

This mode allows up to four Pulse Width Modulated signals to be generated on the PWMx output pins.

PWM Frequency

The four PWM signals can have the same frequency (f_{PWM}) or can have two different frequencies. This is selected by the ENCNTR2 bit which enables single timer or dual timer mode (see Figure 1 and Figure 2).

The frequency is controlled by the counter period and the ATR register value. In dual timer mode, PWM2 and PWM3 can be generated with a different frequency controlled by CNTR2 and ATR2.

$$f_{PWM} = f_{COUNTER} / (4096 - ATR)$$

Following the above formula,

– If $f_{COUNTER}$ is 4 MHz, the maximum value of f_{PWM} is 2 MHz (ATR register value = 4094), the minimum value is 1 kHz (ATR register value = 0).

– If $f_{COUNTER}$ is 32 MHz, the maximum value of f_{PWM} is 8 MHz (ATR register value = 4092), the minimum value is 8 kHz (ATR register value = 0).

Notes:

1. The maximum value of ATR is 4094 because it must be lower than the DC4R value which must be 4095 in this case.

2. To update the DCRx registers at 32 MHz, the following precautions must be taken:

– if the PWM frequency is < 1 MHz and the TRANx bit is set asynchronously, it should be set twice after a write to the DCRx registers.

– if the PWM frequency is > 1 MHz, the TRANx bit should be set along with FORCEx bit with the same instruction (use a load instruction and not 2 bset instructions).

Duty Cycle

The duty cycle is selected by programming the DCRx registers. These are preload registers. The DCRx values are transferred in Active duty cycle registers after an overflow event if the corresponding transfer bit (TRANx bit) is set.

The TRAN1 bit controls the PWMx outputs driven by counter 1 and the TRAN2 bit controls the PWMx outputs driven by counter 2.

PWM generation and output compare are done by comparing these active DCRx values with the counter.

The maximum available resolution for the PWMx duty cycle is:

$$Resolution = 1 / (4096 - ATR)$$

where ATR is equal to 0. With this maximum resolution, 0% and 100% duty cycle can be obtained by changing the polarity.

At reset, the counter starts counting from 0.

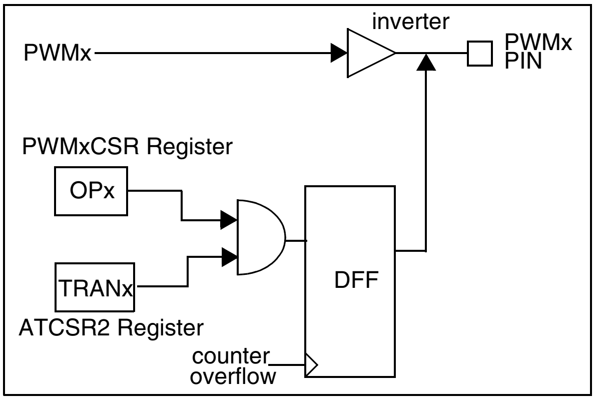
When a upcounter overflow occurs (OVF event), the preloaded Duty cycle values are transferred to the active Duty Cycle registers and the PWMx signals are set to a high level. When the upcounter matches the active DCRx value the PWMx signals are set to a low level. To obtain a signal on a PWMx pin, the contents of the corresponding active DCRx register must be greater than the contents of the ATR register.

The maximum value of ATR is 4094 because it must be lower than the DCR value which must be 4095 in this case.

Polarity Inversion

The polarity bits can be used to invert any of the four output signals. The inversion is synchronized with the counter overflow if the corresponding transfer bit in the ATCSR2 register is set (reset value). See Figure 3.

Figure 37. PWM Polarity Inversion



The Data Flip Flop (DFF) applies the polarity inversion when triggered by the counter overflow input.

Output Control

The PWMx output signals can be enabled or disabled using the OEx bits in the PWMCR register.

DUAL 12-BIT AUTORELOAD TIMER 4 (Cont'd)

11.2.3.2 Dead Time Generation

A dead time can be inserted between PWM0 and PWM1 using the DTGR register. This is required for half-bridge driving where PWM signals must not be overlapped. The non-overlapping PWM0/PWM1 signals are generated through a programmable dead time by setting the DTE bit.

Dead time value = DT[6:0] x Tcounter1

DTGR[7:0] is buffered inside so as to avoid deforming the current PWM cycle. The DTGR effect will take place only after an overflow.

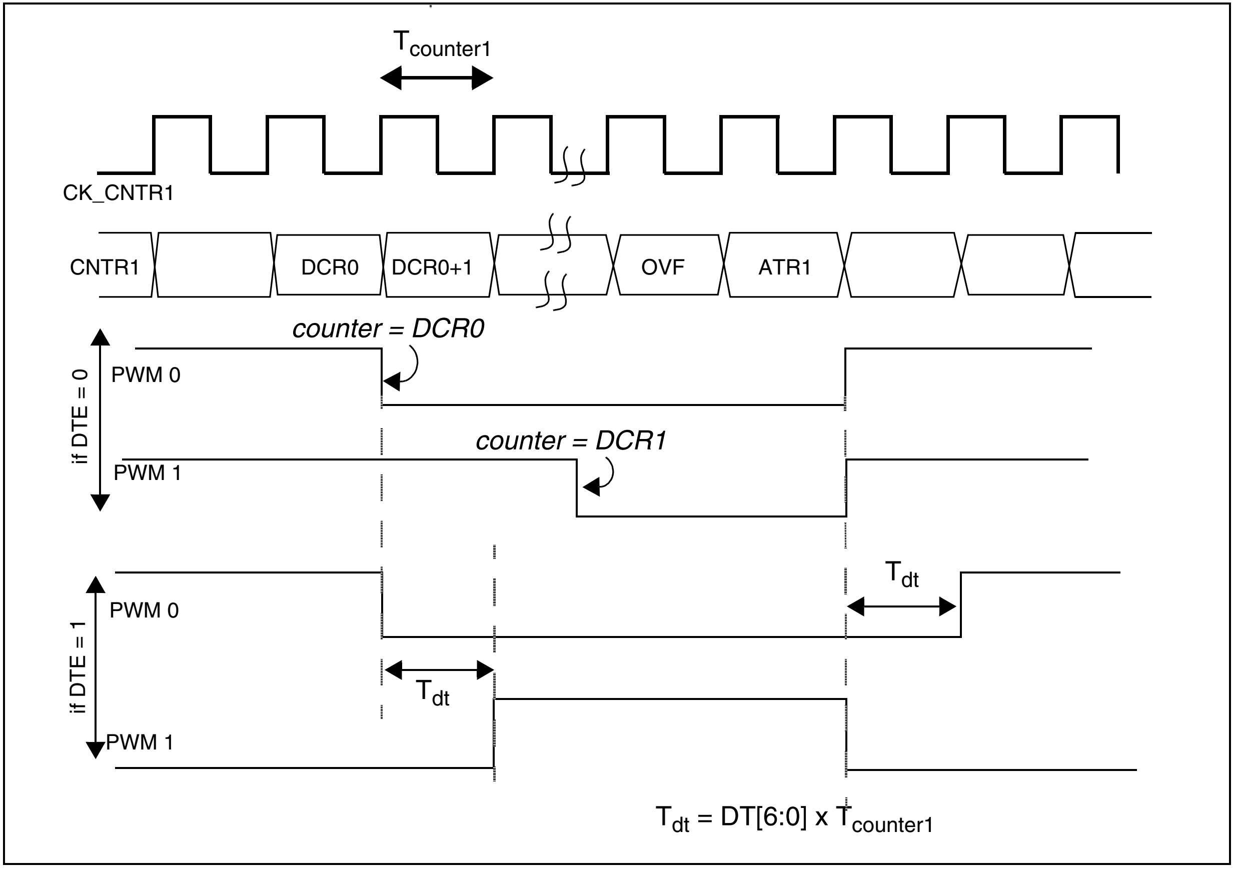
Notes:

1. Dead time is generated only when DTE=1 and DT[6:0] ≠ 0. If DTE is set and DT[6:0]=0, PWM output signals will be at their reset state.

2. Half Bridge driving is possible only if polarities of PWM0 and PWM1 are not inverted, i.e. if OP0 and OP1 are not set. If polarity is inverted, overlapping PWM0/PWM1 signals will be generated.

3. Dead Time generation does not work at 1 ms timebase.

Figure 40. Dead Time Generation



In the above example, when the DTE bit is set:

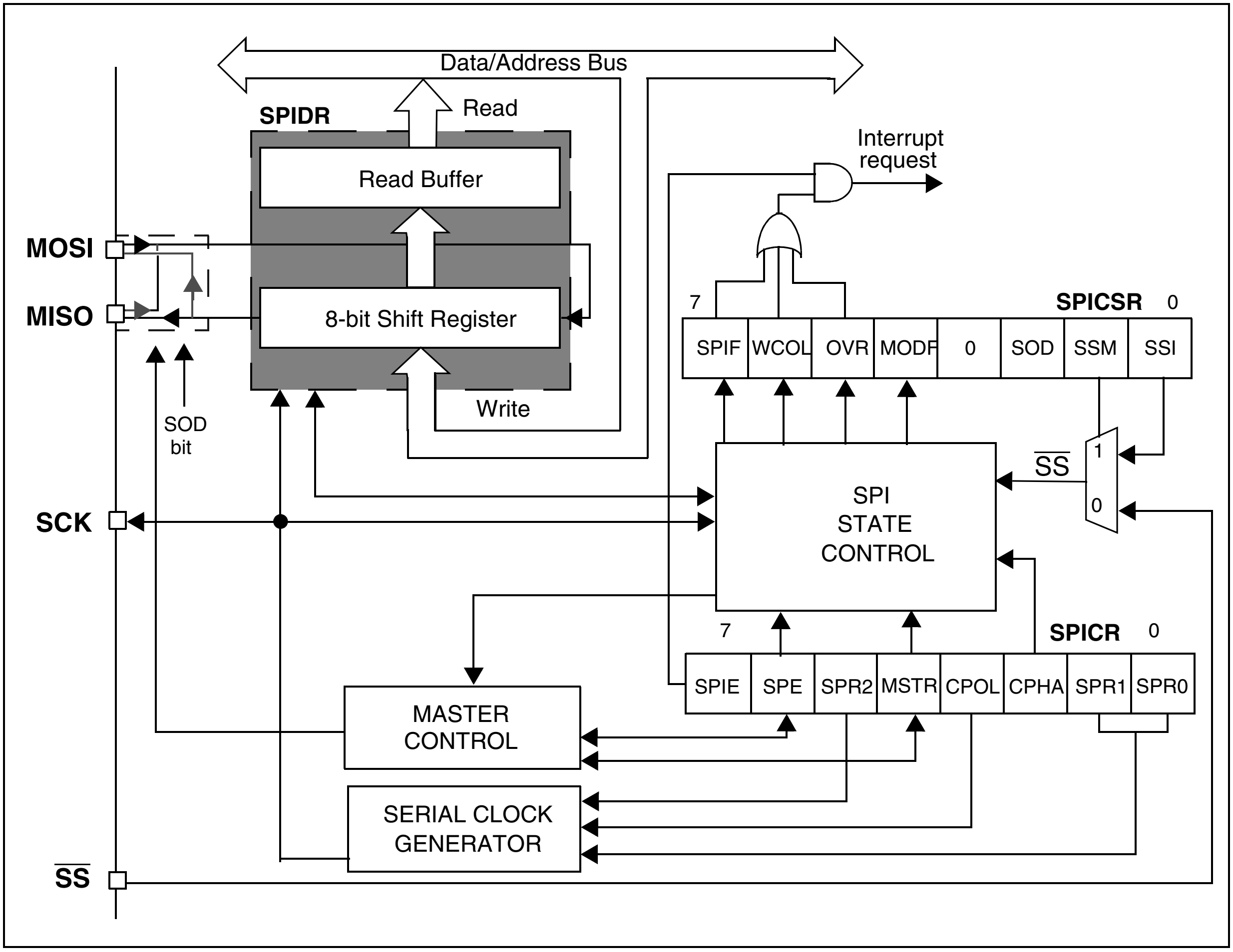
– PWM goes low at DCR0 match and goes high at ATR1+Tdt

– PWM1 goes high at DCR0+Tdt and goes low at ATR match.

With this programmable delay (Tdt), the PWM0 and PWM1 signals which are generated are not overlapped.

SERIAL PERIPHERAL INTERFACE (SPI) (cont'd)

Figure 53. Serial Peripheral Interface Block Diagram



SERIAL PERIPHERAL INTERFACE (cont'd)

11.4.3.3 Master Mode Operation

In master mode, the serial clock is output on the SCK pin. The clock frequency, polarity and phase are configured by software (refer to the description of the SPICSR register).

Note: The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL = 1 or pulling down SCK if CPOL = 0).

How to operate the SPI in master mode

To operate the SPI in master mode, perform the following steps in order:

1. Write to the SPICR register:
   - Select the clock frequency by configuring the SPR[2:0] bits.
   - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits. Figure 5 shows the four possible configurations. **Note:** The slave must have the same CPOL and CPHA settings as the master.
2. Write to the SPICSR register:
   - Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the SS pin high for the complete byte transmit sequence.
3. Write to the SPICR register:
   - Set the MSTR and SPE bits **Note:** MSTR and SPE bits remain set only if SS is high).

Important note: if the SPICSR register is not written first, the SPICR register setting (MSTR bit) may be not taken into account.

The transmit sequence begins when software writes a byte in the SPIDR register.

11.4.3.4 Master Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

- The SPIF bit is set by hardware.
- An interrupt request is generated if the SPIE bit is set and the interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SPICSR register while the SPIF bit is set
2. A read to the SPIDR register

Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

11.4.3.5 Slave Mode Operation

In slave mode, the serial clock is received on the SCK pin from the master device.

To operate the SPI in slave mode:

1. Write to the SPICSR register to perform the following actions:
   - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see Figure 5). **Note:** The slave must have the same CPOL and CPHA settings as the master.
   - Manage the SS pin as described in Section 0.1.3.2 and Figure 3. If CPHA = 1 SS must be held low continuously. If CPHA = 0 SS must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.
2. Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

11.4.3.6 Slave Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MISO pin most significant bit first.

The transmit sequence begins when the slave device receives the clock signal and the most significant bit of the data on its MOSI pin.

When data transfer is complete:

- The SPIF bit is set by hardware.
- An interrupt request is generated if SPIE bit is set and interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

1. An access to the SPICSR register while the SPIF bit is set
2. A write or a read to the SPIDR register

Notes: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an Overrun condition (see Section 0.1.5.2).

Figure 66. Typical accuracy with RCCR=RCCR0 vs V_{DD}= 4.5-5.5V and Temperature

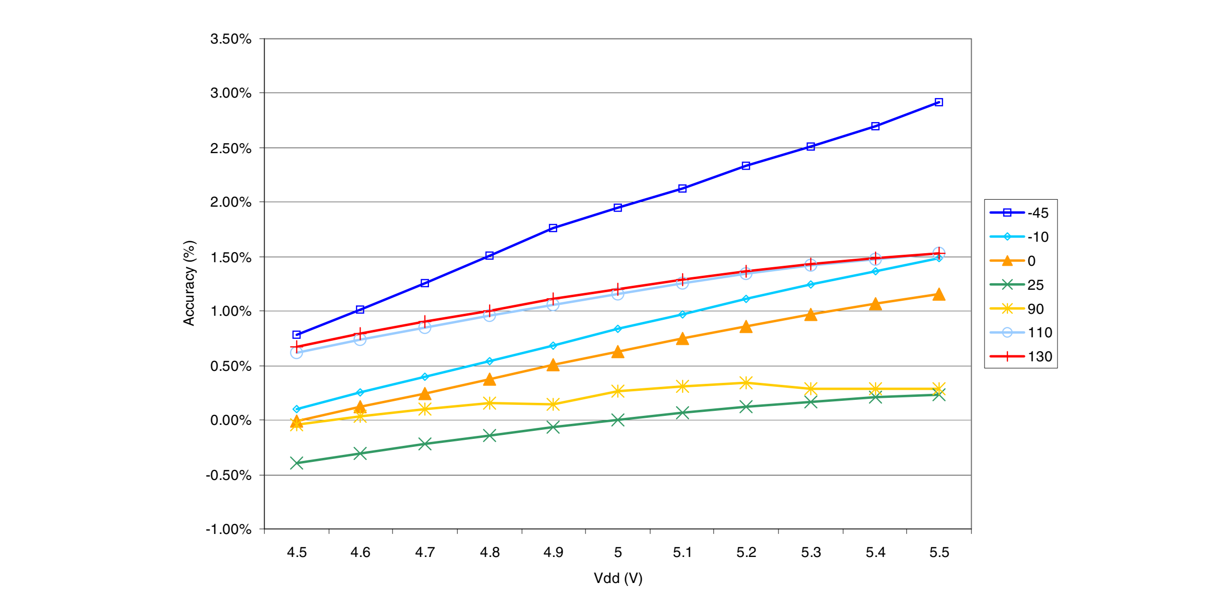
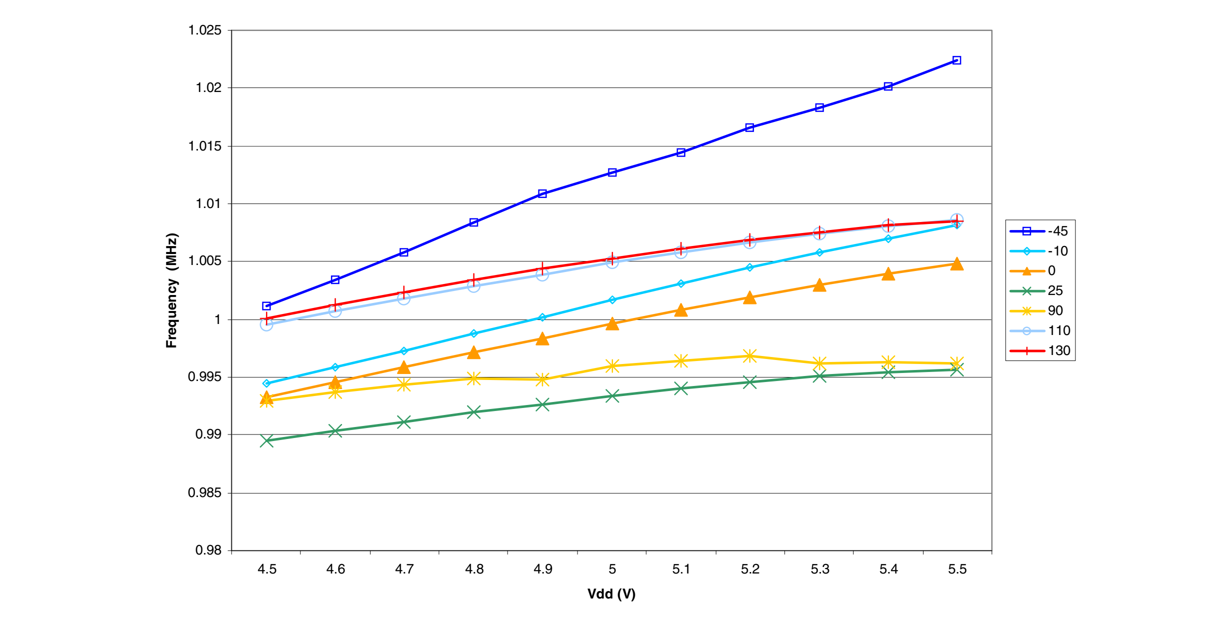


Figure 67. Typical RCCR0 vs V_{DD} and Temperature



13.4 SUPPLY CURRENT CHARACTERISTICS

The following current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To get the total device consumption, the two current values must be added (except for HALT mode for which the clock is stopped).

13.4.1 Supply Current

T_A = -40 to +85°C unless otherwise specified

Symbol	Parameter	Conditions		Typ	Max	Unit
I_{DD}	Supply current in RUN mode	V_{DD}=5.5V	f_{CPU}=8MHz [1]	7	9	mA
	Supply current in WAIT mode		f_{CPU}=8MHz [2]	3	3.6	
	Supply current in SLOW mode		f_{CPU}=250kHz [3]	0.7	0.9	
	Supply current in SLOW WAIT mode7		f_{CPU}=250kHz [4]	0.5	0.8	
	Supply current in HALT mode[5]		-40°C≤T_A≤+125°C	<1	6	µA
	Supply current in AWUFH mode [6)7]		-40°C≤T_A≤+125°C	20		

Notes:

1. CPU running with memory access, all I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.

2. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.

3. SLOW mode selected with f_{CPU} based on f_{OSC} divided by 32. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.

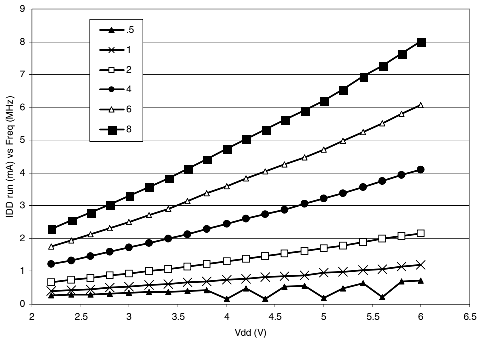
4. SLOW-WAIT mode selected with f_{CPU} based on f_{OSC} divided by 32. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.

5. All I/O pins in output mode with a static value at V_{SS} (no load), LVD disabled. Data based on characterization results, tested in production at V_{DD} max and f_{CPU} max.

6. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load). Data tested in production at V_{DD} max. and f_{CPU} max.

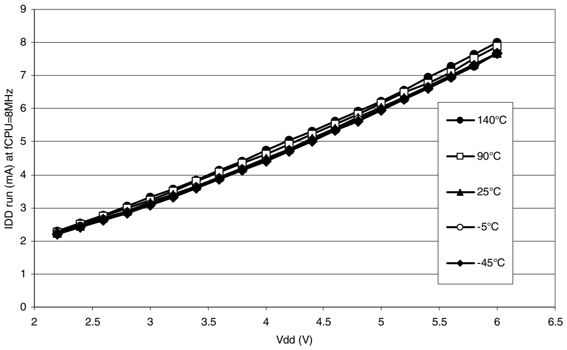
7. This consumption refers to the Halt period only and not the associated run period which is software dependent.

Figure 71. Typical I_{DD} in RUN vs. f_{CPU}



Note: Graph displays data beyond the normal operating range of 3V - 5.5V

Figure 72. Typical I_{DD} in RUN at f_{CPU} = 8MHz



Note: Graph displays data beyond the normal operating range of 3V - 5.5V

13.8 I/O PORT PIN CHARACTERISTICS

13.8.1 General Characteristics

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

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IL}	Input low level voltage			V_{SS} - 0.3		$0.3xV_{DD}$	V
V_{IH}	Input high level voltage			$0.7xV_{DD}$		V_{DD} + 0.3	
V_{hys}	Schmitt trigger voltage hysteresis [1]				400		mV
I_L	Input leakage current	$V_{SS} \leq V_{IN} \leq V_{DD}$				±1	µA
I_S	Static current consumption induced by each floating input pin[2]	Floating input mode			400		
R_{PU}	Weak pull-up equivalent resistor[3]	$V_{IN}=V_{SS}$	V_{DD}=5V	50	120	250	kΩ
			V_{DD}=3V		160		
C_{IO}	I/O pin capacitance				5		pF
$t_{f(IO)out}$	Output high to low level fall time [1]	C_L=50pF Between 10% and 90%			25		ns
$t_{r(IO)out}$	Output low to high level rise time [1]				25		
$t_{w(IT)in}$	External interrupt pulse time [4]			1			t_{CPU}

Notes:

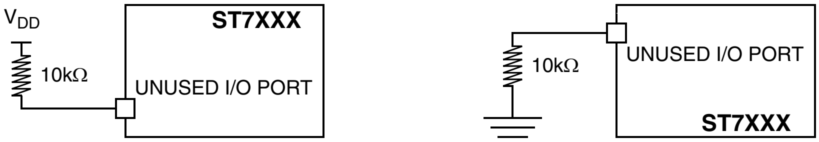
1. Data based on validation/design results.

2. Configuration not recommended, all unused pins must be kept at a fixed voltage: using the output mode of the I/O for example or an external pull-up or pull-down resistor (see Figure 80). Static peak current value taken at a fixed V_{IN} value, based on design simulation and technology characteristics, not tested in production. This value depends on V_{DD} and temperature values.

3. The R_{PU} pull-up equivalent resistor is based on a resistive transistor.

4. To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

Figure 80. Two typical Applications with unused I/O Pin



Caution: During normal operation the ICCCLK pin must be pulled-up, internally or externally (external pull-up of 10k mandatory in noisy environment). This is to avoid entering ICC mode unexpectedly during a reset.

Note: I/O can be left unconnected if it is configured as output (0 or 1) by the software. This has the advantage of greater EMC robustness and lower cost.

I/O PORT PIN CHARACTERISTICS (Cont'd)

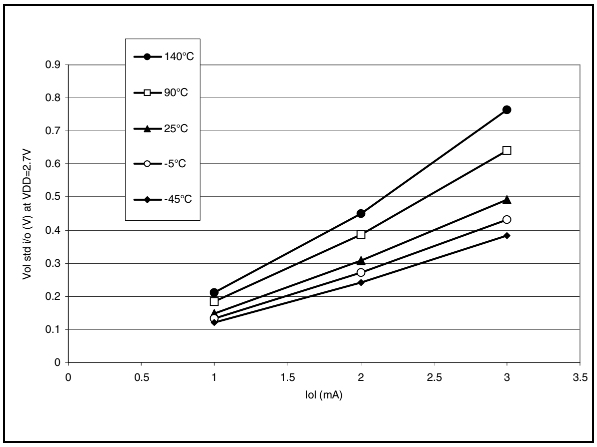


Figure 81. Typical V_{OL} at V_{DD}=2.7V (standard)

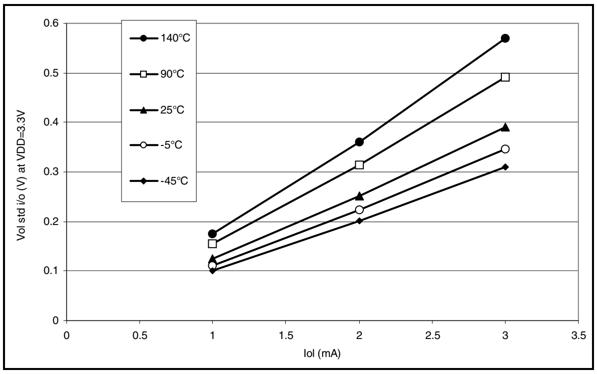


Figure 82. Typical V_{OL} at V_{DD}=3.3V (standard)

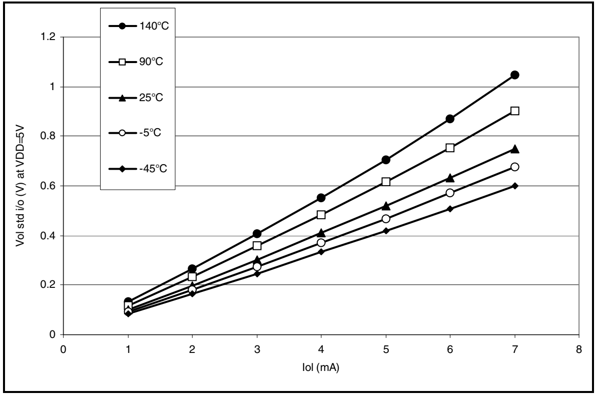


Figure 83. Typical V_{OL} at V_{DD}=5V (standard)

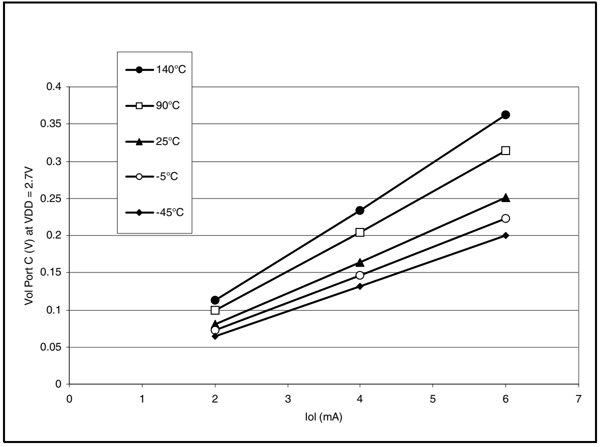


Figure 84. Typical V_{OL} at V_{DD}=2.7V (Port C)

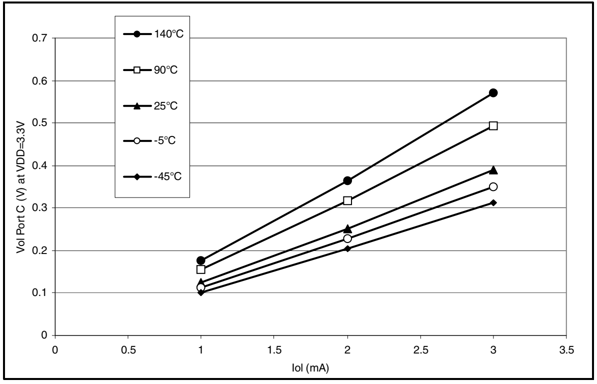


Figure 85. Typical V_{OL} at V_{DD}=3.3V (Port C)

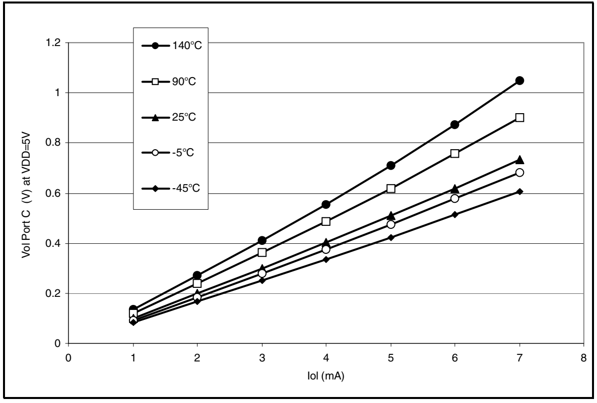


Figure 86. Typical V_{OL} at V_{DD}=5V (Port C)

I/O PORT PIN CHARACTERISTICS (Cont'd)

Figure 87. Typical V_{OL} at V_{DD}=2.7V (High-sink)

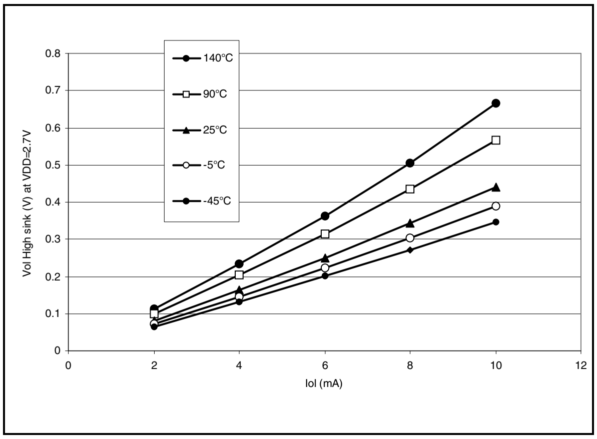


Figure 88. Typical V_{OL} at V_{DD}=3.3V (High-sink)

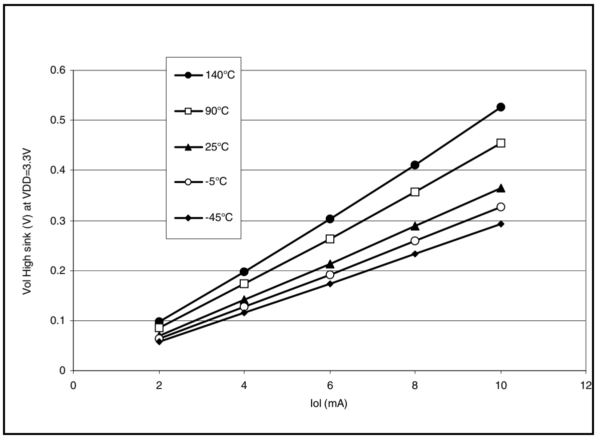


Figure 89. Typical V_{OL} at V_{DD}=5V (High-sink)

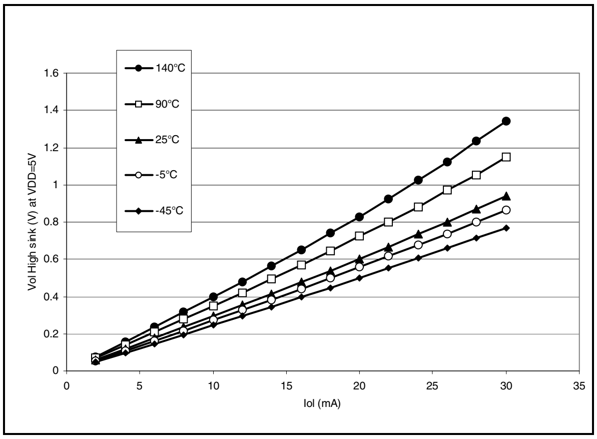


Figure 90. Typical V_{OL} vs. V_{DD} (standard I/Os)

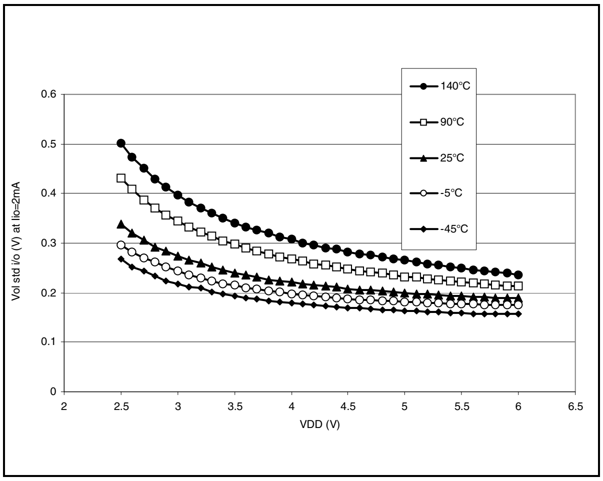


Figure 91. Typical V_{OL} vs V_{DD} (High-sink)

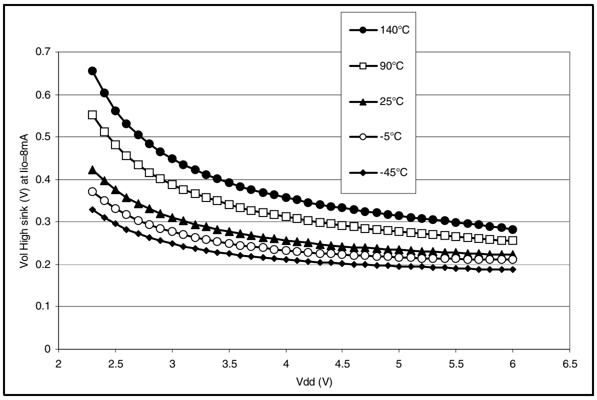
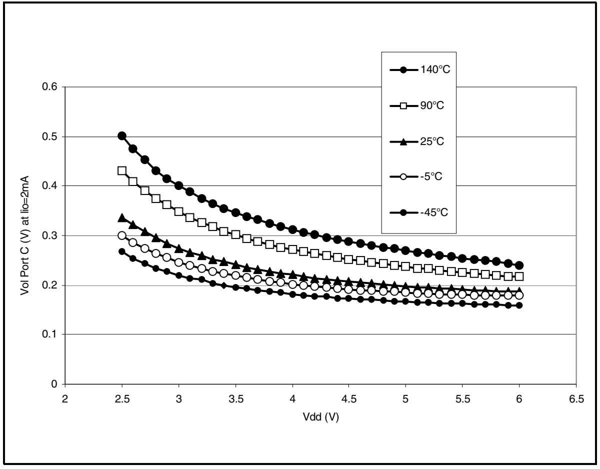


Figure 92. Typical V_{OL} vs V_{DD} (Port C)



13.12 ANALOG COMPARATOR CHARACTERISTICS

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
V_{DDA}	Supply range		4.5		5.5	V
V_{IN}	Comparator input voltage range		0		V_{DDA}	V
Temp	Temperature range		-40		125	°C
V_{offset}	Comparator offset error			20		mV
$I_{DD(CMP)}$	Analog Comparator Consumption			120		µA
	Analog Comparator Consumption during power-down			200		pA
t_{propag}	Comparator propagation delay			40		ns
$t_{startup}$	Startup filter duration			500[2]		ns
t_{stab}	Stabilisation time			500		ns

13.13 PROGRAMMABLE INTERNAL VOLTAGE REFERENCE CHARACTERISTICS

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
V_{DDA}	Supply range		4	5	5.5	V
Temp	Temperature range		-40	27	125	°C
$I_{DD(VOLTREF)}$	Internal Voltage Reference Consumption			50		µA
	Internal Voltage Reference Consumption during power-down			200		pA
$t_{startup}$	Startup duration			1[2]		µs

13.14 CURRENT BIAS CHARACTERISTICS (for Comparator and Internal Voltage Reference)

Symbol	Parameter	Conditions	Min	Typ [1]	Max	Unit
V_{DDA}	Supply range		4.5	5	5.5	V
Temp	Temperature range		-40	27	125	°C
IDD (Bias)	Bias Consumption in run mode			50		µA
	Bias Consumption during power-down			36		pA
$t_{startup}$	Startup time			1[2]		µs

Notes:

1. Unless otherwise specified, typical data are based on T_A=25°C and V_{DD}-V_{SS}=5V. They are given only as design guidelines and are not tested.

2. Since startup time for internal voltage reference and bias is 1 µs, comparator correct output should not be expected before 1 µs during startup.

15.3 DEVELOPMENT TOOLS

Development tools for the ST7 microcontrollers include a complete range of hardware systems and software tools from STMicroelectronics and third-party tool suppliers. The range of tools includes solutions to help you evaluate microcontroller peripherals, develop and debug your application, and program your microcontrollers.

15.3.1 Starter kits

ST offers complete, affordable **starter kits**. Starter kits are complete hardware/software tool packages that include features and samples to help you quickly start developing your application.

15.3.2 Development and debugging tools

Application development for ST7 is supported by fully optimizing **C Compilers** and the **ST7 Assembler-Linker** toolchain, which are all seamlessly integrated in the ST7 integrated development environments in order to facilitate the debugging and fine-tuning of your application. The Cosmic C Compiler is available in a free version that outputs up to 16KBytes of code.

The range of hardware tools includes full-featured **ST7-EMU3 series emulators**, cost effective **ST7-DVP3 series emulators** and the low-cost **RLink** in-circuit debugger/programmer. These tools are supported by the **ST7 Toolset** from STMicroelectronics, which includes the STVD7 integrated development environment (IDE) with high-level language debugger, editor, project manager and integrated programming interface.

15.3.3 Programming tools

During the development cycle, the **ST7-DVP3** and and **ST7-EMU3 series emulators** and the **RLink** provide in-circuit programming capability for programming the Flash microcontroller on your application board.

ST also provides a low-cost dedicated in-circuit programmer, the **ST7-STICK**, as well as **ST7 Socket Boards** which provide all the sockets required for programming any of the devices in a specific ST7 sub-family on a platform that can be used with any tool with in-circuit programming capability for ST7.

For production programming of ST7 devices, ST's third-party tool partners also provide a complete range of gang and automated programming solutions, which are ready to integrate into your production environment.

15.3.4 Order Codes for Development and Programming Tools

Table 28 below lists the ordering codes for the ST7LITE1xB development and programming tools. For additional ordering codes for spare parts and accessories, refer to the online product selector at www.st.com/mcu.

15.3.5 Order codes for ST7LITE1xB development tools

Table 28. Development tool order codes for the ST7LITE1xB family

MCU	In-circuit Debugger, RLink Series[1]		Emulator		Programming Tool	
ST7FLIT1xBF0 ST7FLIT1xBF1 ST7FLIT1xBY0 ST7FLIT1xBY1	**Starter Kit without Demo Board**	**Starter Kit with Demo Board**	**DVP Series**	**EMU Series**	**In-circuit Programmer**	**ST Socket Boards and EPBs**
	STX-RLINK[2]	ST7FLITE-SK/RAIS[2]	ST7MDT10-DVP3[4]	ST7MDT10-EMU3	STX-RLINK ST7-STICK[3][5]	ST7SB10-123[3]

Notes:

1. Available from ST or from Raisonance, www.raisonance.com
2. USB connection to PC
3. Add suffix /EU, /UK or /US for the power supply for your region
4. Includes connection kit for DIP16/SO16 only. See "How to order an EMU or DVP" in ST product and tool selection guide for connection kit ordering information
5. Parallel port connection to PC

16 REVISION HISTORY

Date	Revision	Main changes
20-Dec-05	1	Initial release on internet
20-July-06	2	Added reset default state in bold for RESET, PC0 and PC1 in Table 1, “Device Pin Description,” on page 7 Changed note below Figure 9 on page 17 and the last paragraph of “ACCESS ERROR HANDLING” on page 18 Modified note 3 in Table 2, “Hardware Register Map,” on page 10, changed LTICR reset value and replaced h by b for LTCSR1, ATCSR and SICSR reset values Added note to Figure 14 on page 26 Modified caution in section 7.2 on page 23 Added note 2 in “EXTERNAL INTERRUPT CONTROL REGISTER (EICR)” on page 38 and changed “External Interrupt Function” on page 48 Removed references to true open drain in Table 8 on page 50, Table 9 on page 51 and notes Replaced Auto reload timer 3 by Auto reload timer 4 in section 11.2 on page 57 Modified the BA bit description in the BREAKCR register in section 11.2.6 on page 70 Changed order of Section 11.3.3.2 and section 11.3.3.3 on page 80 and removed two paragraphs before section 11.3.4 on page 81 Modified Section 11.3.3.2 Modified bit names in the description of LTARR and LTCNTR registers in section 11.3.6 on page 81 Added important note in section 11.6.3 on page 100 and added note to CHYST bit description in section 11.6.4 on page 102 Modified CINV bit description in section 11.6.4 on page 102 and Figure 62 on page 101 Changed LTCSR2 reset values in Table 2 on page 10 and in section 11.3.6 on page 81 Modified section 13.2.2 on page 111 (I_{IO} values) Modified Section 13.3.1 and section 13.3.2 on page 112 Removed Vt_{POR} min value in section 13.3.3.1 on page 113 Modified section 13.3.4 on page 114 Modified section 13.3.5 on page 114 Modified section 13.3.5.1 on page 115 and section 13.3.5.2 on page 117 Modified temperature range in section 13.3.5.3 on page 119 Modified section 13.4.1 on page 121 Added note in section 13.5.3 on page 124 Removed figures “PLLx4 and PLLx8 Output vs CLKIN frequency” Updated section 13.5.4 on page 125 Modified section 13.6 on page 126 Modified Section 13.7.1 and section 13.7.2 on page 127 Modified section 13.10.1 on page 137 ($t_{su(\overline{SS})}$, $t_{v(MO)}$ and $t_{h(MO)}$) Modified Figure 108 (CPHA=1) and Figure 109 on page 138 ($t_{v(MO)}$, $t_{h(MO)}$) Removed empty figure “Typical I_{PU} vs. V_{DD} with V_{IN}=V_{SS}” in section 13.8.1 on page 129 and modified note 3 Modified temperature range in section 13.8.2 on page 130 Modified section 13.9.1 on page 135 Added “related Application notes” in section 13.11 on page 139 Removed EMC protection circuitry in Figure 106 on page 136 (device works correctly without these components) Modified ADC accuracy table in section 13.11 on page 139 Modified temperature range in Section 13.12, Section 13.13 and section 13.14 on page 143 Modified Table 27 on page 150 Added note 3 to E_D and E_L in Table “ADC Accuracy with VDD=5.0V” on page 140 Modified section 14.2 on page 148 Modified section 15.2 on page 151 (part numbers in QFN20 package) Updated section 15.3 on page 153
15-Sept-06	3	Removed QFN20 pinout and mechanical data. Modified description of CNTR[11:0] bits in section 11.2.6 on page 72 Added “External Clock Source” on page 124 and Figure 78 on page 124 Modified Table 1.