



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

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	2KB (2K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
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	Surface Mount
Package / Case	16-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7flit19by0m6

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Table 2. Hardware Register Map

Address	Block	Register Label	Register Name	Reset Status	Remarks
0000h 0001h 0002h	Port A	PADR PADDR PAOR	Port A Data Register Port A Data Direction Register Port A Option Register	FFh ¹⁾ 00h 40h	R/W R/W R/W
0003h 0004h 0005h	Port B	PBDR PBDDR PBOR	Port B Data Register Port B Data Direction Register Port B Option Register	FFh ¹⁾ 00h 00h	R/W R/W R/W ²⁾
0006h 0007h	Port C	PCDR PCDDR	Port C Data Register Port C Data Direction Register	0xh 00h	R/W R/W
0008h 0009h 000Ah 000Bh 000Ch	LITE TIMER 2	LTCSR2 LTARR LTCNTR LTCSR1 LTICR	Lite Timer Control/Status Register 2 Lite Timer Auto-reload Register Lite Timer Counter Register Lite Timer Control/Status Register 1 Lite Timer Input Capture Register	00h 00h 00h 0X00 0000b 00h	R/W R/W Read Only R/W Read Only
000Dh 000Eh 000Fh 0010h 0011h 0012h 0012h 0013h 0014h 0015h 0016h 0017h 0018h 0017h 0018h 0017h 0018h 0012h 001Ch 001Ch 001Ch 001Ch 0020h 0021h 0022h 0023h 0025h 0026h	AUTO- RELOAD TIMER 2	ATCSR CNTRH CNTRL ATRH ATRL PWMCR PWM0CSR PWM1CSR PWM1CSR PWM2CSR PWM2CSR DCR0H DCR0L DCR1H DCR0L DCR1H DCR1L DCR2H DCR2L DCR3H DCR3L ATICRH ATICRH ATICRL ATCSR2 BREAKCR ATR2H ATR2L DTGR BREAKEN	Timer Control/Status Register Counter Register High Counter Register Low Auto-Reload Register High Auto-Reload Register Low PWM Output Control Register PWM 0 Control/Status Register PWM 1 Control/Status Register PWM 2 Control/Status Register PWM 3 Control/Status Register PWM 0 Duty Cycle Register High PWM 0 Duty Cycle Register Low PWM 1 Duty Cycle Register Low PWM 1 Duty Cycle Register High PWM 2 Duty Cycle Register Low PWM 2 Duty Cycle Register Low PWM 3 Duty Cycle Register Low PWM 3 Duty Cycle Register Low PWM 3 Duty Cycle Register Low Input Capture Register High Input Capture Register High Input Capture Register Low Timer Control/Status Register 2 Break Control Register Auto-Reload Register 2 Low Dead Time Generation Register Break Enable Register	0X00 0000b 00h 00h 00h 00h 00h 00h 00h 00h 00	R/W Read Only Read Only R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W
0027h to 002Bh		L	Reserved area (5 bytes)		I
002Ch	Comparator Voltage Reference	VREFCR	Internal Voltage Reference Control Reg- ister	00h	R/W
002Dh	Comparator	CMPCR	Comparator and Internal Reference Con- trol Register	00h	R/W
002Eh	WDG	WDGCR	Watchdog Control Register	7Fh	R/W



5 DATA EEPROM

5.1 INTRODUCTION

<u>ل</u>رک

The Electrically Erasable Programmable Read Only Memory can be used as a non volatile backup for storing data. Using the EEPROM requires a basic access protocol described in this chapter.

5.2 MAIN FEATURES

- Up to 32 Bytes programmed in the same cycle
- EEPROM mono-voltage (charge pump)
- Chained erase and programming cycles
- Internal control of the global programming cycle duration
- WAIT mode management
- Readout protection



Figure 7. EEPROM Block Diagram

7.4 MULTI-OSCILLATOR (MO)

The main clock of the ST7 can be generated by four different source types coming from the multioscillator block (1 to 16MHz):

- an external source
- 5 different configurations for crystal or ceramic resonator oscillators
- an internal high frequency RC oscillator

Each oscillator is optimized for a given frequency range in terms of consumption and is selectable through the option byte. The associated hardware configurations are shown in Table 4. Refer to the electrical characteristics section for more details.

External Clock Source

In this external clock mode, a clock signal (square, sinus or triangle) with ~50% duty cycle has to drive the OSC1 pin while the OSC2 pin is tied to ground.

Note: when the Multi-Oscillator is not used, PB4 is selected by default as external clock.

Crystal/Ceramic Oscillators

In this mode, with a self-controlled gain feature, oscillator of any frequency from 1 to 16MHz can be placed on OSC1 and OSC2 pins. This family of oscillators has the advantage of producing a very accurate rate on the main clock of the ST7. In this mode of the multi-oscillator, the resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

These oscillators are not stopped during the RESET phase to avoid losing time in the oscillator start-up phase.

Internal RC Oscillator

57/

In this mode, the tunable 1%RC oscillator is used as main clock source. The two oscillator pins have to be tied to ground if dedicately using for oscillator else can be found as general purpose IO. The calibration is done through the RCCR[7:0] and SICSR[6:5] registers.

Table 4. ST7 Clock Sources



SYSTEM INTEGRITY MANAGEMENT (Cont'd)

Figure 19. Reset and Supply Management Block Diagram



47/

8 INTERRUPTS

The ST7 core may be interrupted by one of two different methods: Maskable hardware interrupts as listed in the "interrupt mapping" table and a nonmaskable software interrupt (TRAP). The Interrupt processing flowchart is shown in Figure 1.

The maskable interrupts must be enabled by clearing the I bit in order to be serviced. However, disabled interrupts may be latched and processed when they are enabled (see external interrupts subsection).

Note: After reset, all interrupts are disabled.

When an interrupt has to be serviced:

- Normal processing is suspended at the end of the current instruction execution.
- The PC, X, A and CC registers are saved onto the stack.
- The I bit of the CC register is set to prevent additional interrupts.
- The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to the Interrupt Mapping table for vector addresses).

The interrupt service routine should finish with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

Note: As a consequence of the IRET instruction, the I bit is cleared and the main program resumes.

Priority Management

By default, a servicing interrupt cannot be interrupted because the I bit is set by hardware entering in interrupt routine.

In the case when several interrupts are simultaneously pending, an hardware priority defines which one will be serviced first (see the Interrupt Mapping table).

Interrupts and Low Power Mode

All interrupts allow the processor to leave the WAIT low power mode. Only external and specifically mentioned interrupts allow the processor to leave the HALT low power mode (refer to the "Exit from HALT" column in the Interrupt Mapping table).

8.1 NON MASKABLE SOFTWARE INTERRUPT

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It is serviced according to the flowchart in Figure 1.

8.2 EXTERNAL INTERRUPTS

External interrupt vectors can be loaded into the PC register if the corresponding external interrupt occurred and if the I bit is cleared. These interrupts allow the processor to leave the HALT low power mode.

The external interrupt polarity is selected through the miscellaneous register or interrupt register (if available).

An external interrupt triggered on edge will be latched and the interrupt request automatically cleared upon entering the interrupt service routine.

Caution: The type of sensitivity defined in the Miscellaneous or Interrupt register (if available) applies to the ei source. In case of a NANDed source (as described in the I/O ports section), a low level on an I/O pin, configured as input with interrupt, masks the interrupt request even in case of rising-edge sensitivity.

8.3 PERIPHERAL INTERRUPTS

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

- The I bit of the CC register is cleared.
- The corresponding enable bit is set in the control register.

If any of these two conditions is false, the interrupt is latched and thus remains pending.

Clearing an interrupt request is done by:

- Writing "0" to the corresponding bit in the status register or
- Access to the status register while the flag is set followed by a read or write of an associated register.

Note: The clearing sequence resets the internal latch. A pending interrupt (that is, waiting for being enabled) will therefore be lost if the clear sequence is executed.

- select rising edge
- reset the interrupt mask with the RIM instruction (in cases where a pin level change could occur)

10.2.2 Output Modes

Setting the DDRx bit selects output mode. Writing to the DR bits applies a digital value to the I/O through the latch. Reading the DR bits returns the previously stored value.

If an OR bit is available, different output modes can be selected by software: push-pull or opendrain. Refer to I/O Port Implementation section for configuration.

DR Value and Output Pin Status

DR	Push-Pull	Open-Drain
0	V _{OL}	V _{OL}
1	V _{OH}	Floating

10.2.3 Alternate Functions

57/

Many ST7s I/Os have one or more alternate functions. These may include output signals from, or input signals to, on-chip peripherals. The Device Pin Description table describes which peripheral signals can be input/output to which ports.

A signal coming from an on-chip peripheral can be output on an I/O. To do this, enable the on-chip peripheral as an output (enable bit in the peripheral's control register). The peripheral configures the I/O as an output and takes priority over standard I/ O programming. The I/O's state is readable by addressing the corresponding I/O data register.

Configuring an I/O as floating enables alternate function input. It is not recommended to configure an I/O as pull-up as this will increase current consumption. Before using an I/O as an alternate input, configure it without interrupt. Otherwise spurious interrupts can occur.

Configure an I/O as input floating for an on-chip peripheral signal which can be input and output.

Caution:

I/Os which can be configured as both an analog and digital alternate function need special attention. The user must control the peripherals so that the signals do not arrive at the same time on the same pin. If an external clock is used, only the clock alternate function should be employed on that I/O pin and not the other alternate function.

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
00006	PAOR	MSB							LSB
00020	Reset Value	0	1	0	0	0	0	0	0
0000	PBDR	MSB							LSB
0003h	Reset Value	1	1	1	1	1	1	1	1
000.4h	PBDDR	MSB							LSB
0004h	Reset Value	0	0	0	0	0	0	0	0
00054	PBOR	MSB							LSB
00050	Reset Value	0	0	0	0	0	0	0	0
00000	PCDR	MSB							LSB
00060	Reset Value	0	0	0	0	0	0	1	1
00076	PCDDR	MSB							LSB
0007h	Reset Value	0	0	0	0	0	0	0	0

10.8 MULTIPLEXED INPUT/OUTPUT PORTS

OSC1/PC0 are multiplexed on one pin (pin20) and OSC2/PC1 are multiplexed on another pin (pin 19).



11.2 DUAL 12-BIT AUTORELOAD TIMER 4 (AT4)

11.2.1 Introduction

The 12-bit Autoreload Timer can be used for general-purpose timing functions. It is based on one or two free-running 12-bit upcounters with an input capture register and four PWM output channels. There are 7 external pins:

- Four PWM outputs
- ATIC/LTIC pins for the Input Capture function
- BREAK pin for forcing a break condition on the PWM outputs

11.2.2 Main Features

- Single Timer or Dual Timer mode with two 12-bit upcounters (CNTR1/CNTR2) and two 12-bit autoreload registers (ATR1/ATR2)
- Maskable overflow interrupts
- PWM mode

47/

Figure 35. Single Timer Mode (ENCNTR2=0)

- Generation of four independent PWMx signals
- Dead time generation for Half bridge driving mode with programmable dead time
- Frequency 2 kHz 4 MHz (@ 8 MHz f_{CPU})
- Programmable duty-cycles
- Polarity control
- Programmable output modes
- Output Compare Mode
- Input Capture Mode
 - 12-bit input capture register (ATICR)
 - Triggered by rising and falling edges
 - Maskable IC interrupt
 - Long range input capture
- Internal/External Break control
- Flexible Clock control
- One Pulse mode on PWM2/3
- Force Update









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

57/

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.

59/159

11.2.3.7 Force Update

In order not to wait for the counter_x overflow to load the value into active DCRx registers, a programmable counter_x overflow is provided. For both counters, a separate bit is provided which when set, make the counters start with the overflow value, i.e. FFFh. After overflow, the counters start counting from their respective auto reload register values.

These bits are FORCE1 and FORCE2 in the ATCSR2 register. FORCE1 is used to force an overflow on Counter 1 and, FORCE2 is used for Counter 2. These bits are set by software and re-

Figure 50. Force Overflow Timing Diagram

set by hardware after the respective counter overflow event has occurred.

This feature can be used at any time. All related features such as PWM generation, Output Compare, Input Capture, One-pulse (refer to Figure 15. Dynamic DCR2/3 update in One Pulse Mode) can be used this way.





Table 14. Register Map and Reset Values

57

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0D	ATCSR Reset Value	0	ICF 0	ICIE 0	CK1 0	CK0 0	OVF1 0	OVFIE1 0	CMPIE 0
0E	CNTR1H Reset Value	0	0	0	0	CNTR1_11 0	CNTR1_10 0	CNTR1_9 0	CNTR1_8 0
0F	CNTR1L Reset Value	CNTR1_7 0	CNTR1_8 0	CNTR1_7 0	CNTR1_6 0	CNTR1_3 0	CNTR1_2 0	CNTR1_1 0	CNTR1_0 0
10	ATR1H Reset Value	0	0	0	0	ATR11 0	ATR10 0	ATR9 0	ATR8 0
11	ATR1L Reset Value	ATR7 0	ATR6 0	ATR5 0	ATR4 0	ATR3 0	ATR2 0	ATR1 0	ATR0 0
12	PWMCR Reset Value	0	OE3 0	0	OE2 0	0	OE1 0	0	OE0 0
13	PWM0CSR Reset Value	0	0	0	0	0	0	OP0 0	CMPF0 0
14	PWM1CSR Reset Value	0	0	0	0	0	0	OP1 0	CMPF1 0
15	PWM2CSR Reset Value	0	0	0	0	0	0	OP2 0	CMPF2 0
16	PWM3CSR Reset Value	0	0	0	0	OP_EN 0	OPEDGE 0	OP3 0	CMPF3 0
17	DCR0H Reset Value	0	0	0	0	DCR11 0	DCR10 0	DCR9 0	DCR8 0
18	DCR0L Reset Value	DCR7 0	DCR6 0	DCR5 0	DCR4 0	DCR3 0	DCR2 0	DCR1 0	DCR0 0
19	DCR1H Reset Value	0	0	0	0	DCR11 0	DCR10 0	DCR9 0	DCR8 0
1A	DCR1L Reset Value	DCR7 0	DCR6 0	DCR5 0	DCR4 0	DCR3 0	DCR2 0	DCR1 0	DCR0 0
1B	DCR2H Reset Value	0	0	0	0	DCR11 0	DCR10 0	DCR9 0	DCR8 0
1C	DCR2L Reset Value	DCR7 0	DCR6 0	DCR5 0	DCR4 0	DCR3 0	DCR2 0	DCR1 0	DCR0 0
1D	DCR3H Reset Value	0	0	0	0	DCR11 0	DCR10 0	DCR9 0	DCR8 0
1E	DCR3L Reset Value	DCR7 0	DCR6 0	DCR5 0	DCR4 0	DCR3 0	DCR2 0	DCR1 0	DCR0 0
1F	ATICRH Reset Value	0	0	0	0	ICR11 0	ICR10 0	ICR9 0	ICR8 0
20	ATICRL Reset Value	ICR7 0	ICR6 0	ICR5 0	ICR4 0	ICR3 0	ICR2 0	ICR1 0	ICR0 0

11.3 LITE TIMER 2 (LT2)

11.3.1 Introduction

The Lite Timer can be used for general-purpose timing functions. It is based on two free-running 8-bit upcounters and an 8-bit input capture register.

11.3.2 Main Features

Realtime Clock

47/

– One 8-bit upcounter 1 ms or 2 ms timebase period (@ 8 MHz $f_{OSC})$

Figure 51. Lite Timer 2 Block Diagram

- One 8-bit upcounter with autoreload and programmable timebase period from 4µs to 1.024ms in 4µs increments (@ 8 MHz f_{OSC})
- 2 Maskable timebase interrupts
- Input Capture
 - 8-bit input capture register (LTICR)
 - Maskable interrupt with wake-up from Halt mode capability



LITE TIMER (Cont'd)

Bit 6 = **ICF** *Input Capture Flag* This bit is set by hardware and cleared by software by reading the LTICR register. Writing to this bit does not change the bit value. 0: No input capture 1: An input capture has occurred

Note: After an MCU reset, software must initialize the ICF bit by reading the LTICR register

Bit 5 = **TB** *Timebase period selection* This bit is set and cleared by software. 0: Timebase period = $t_{OSC} * 8000 (1ms @ 8 MHz)$ 1: Timebase period = $t_{OSC} * 16000 (2ms @ 8 MHz)$

Bit 4 = **TB1IE** *Timebase Interrupt enable* This bit is set and cleared by software. 0: Timebase (TB1) interrupt disabled 1: Timebase (TB1) interrupt enabled

Bit 3 =**TB1F** *Timebase Interrupt Flag* This bit is set by hardware and cleared by software reading the LTCSR register. Writing to this bit has no effect.

0: No counter overflow

1: A counter overflow has occurred

Bits 2:0 = Reserved

LITE TIMER INPUT CAPTURE REGISTER (LTICR) Read only

Reset Value: 0000 0000 (00h)

7

ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0

0

Bits 7:0 = ICR[7:0] Input Capture Value

These bits are read by software and cleared by hardware after a reset. If the ICF bit in the LTCSR is cleared, the value of the 8-bit up-counter will be captured when a rising or falling edge occurs on the LTIC pin.



ANALOG COMPARATOR (Cont'd)

Figure 61. Analog Comparator and Internal Voltage Reference



Figure 62. Analog Comparator

\$7



EMC CHARACTERISTICS (Cont'd)

13.7.3 Absolute Maximum Ratings (Electrical Sensitivity)

Based on two different tests (ESD and LU) using specific measurement methods, the product is stressed in order to determine its performance in terms of electrical sensitivity. For more details, refer to the application note AN1181.

13.7.3.1 Electro-Static Discharge (ESD)

Electro-Static Discharges (a positive then a negative pulse separated by 1 second) are applied to the pins of each sample according to each pin combination. The sample size depends on the number of supply pins in the device (3 parts*(n+1) supply pin). Two models can be simulated: Human Body Model and Machine Model. This test conforms to the JESD22-A114A/A115A standard.

Absolute Maximum Ratings

Symbol	Ratings	Conditions	Maximum value ¹⁾	Unit
V _{ESD(HBM)}	Electro-static discharge voltage (Human Body Model)	T _A =+25°C	8000	V
V _{ESD(MM)}	Electro-static discharge voltage (Machine Model)	T _A =+25°C	400	v

Note:

1. Data based on characterization results, not tested in production.

13.7.3.2 Static Latch-Up

 LU: 3 complementary static tests are required on 6 parts to assess the latch-up performance. A supply overvoltage (applied to each power supply pin) and a current injection (applied to each input, output and configurable I/O pin) are performed on each sample. This test conforms to the EIA/JESD 78 IC latch-up standard. For more details, refer to the application note AN1181.

Electrical Sensitivities

Symbol	Parameter	Conditions	Class
111	Static latch-up class	T _A =+25°C	А
20		T _A =+85°C	A

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)











57

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
Voffset	Comparator offset error			20		mV
	Analog Comparator Consumption			120		μA
I _{DD(CMP)}	Analog Comparator Consumption during power-down			200		pА
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
	Internal Voltage Reference Consumption			50		μA
IDD(VOLTREF)	Internal Voltage Reference Consumption during power-down			200		pА
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
	Bias Consumption in run mode			50		μA
IDD (Bias)	Bias Consumption during power- down			36		pА
t _{startup}	Startup time			1 ²⁾		μs

Notes:

<u>(۲</u>

1. Unless otherwise specified, typical data are based on $T_A=25^{\circ}C$ and $V_{DD}-V_{SS}=5V$. They are given only as design guide-lines and are not tested.

2. Since startup time for internal voltage reference and bias is 1 μ s, comparator correct output should not be expected before 1 μ s 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 thirdparty 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 lan-

guage 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⁾		Emula	ator	Programming Tool		
ST7FLIT1xBF0 ST7FLIT1xBF1	Starter Kit without Demo Board	Starter Kit with Demo Board	DVP Series	EMU Series	In-circuit Programmer	ST Socket Boards and EPBs	
ST7FLIT1xBY1	STX-RLINK ²⁾	ST7FLITE- SK/RAIS ²⁾	ST7MDT10- DVP3 ⁴⁾	ST7MDT10- EMU3	STX-RLINK ST7-STICK ³⁾⁵⁾	ST7SB10- 123 ³⁾	

Notes:

57/

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

Table 29. ST7 Application Notes

IDENTIFICATION	DESCRIPTION
AN1039	ST7 MATH UTILITY ROUTINES
AN1064	WRITING OPTIMIZED HIWARE C LANGUAGE FOR ST7
AN1071	HALF DUPLEX USB-TO-SERIAL BRIDGE USING THE ST72611 USB MICROCONTROLLER
AN1106	TRANSLATING ASSEMBLY CODE FROM HC05 TO ST7
AN1179	PROGRAMMING ST7 FLASH MICROCONTROLLERS IN REMOTE ISP MODE (IN-SITU PRO- GRAMMING)
AN1446	USING THE ST72521 EMULATOR TO DEBUG A ST72324 TARGET APPLICATION
AN1477	EMULATED DATA EEPROM WITH XFLASH MEMORY
AN1478	PORTING AN ST7 PANTA PROJECT TO CODEWARRIOR IDE
AN1527	DEVELOPING A USB SMARTCARD READER WITH ST7SCR
AN1575	ON-BOARD PROGRAMMING METHODS FOR XFLASH AND HDFLASH ST7 MCUS
AN1576	IN-APPLICATION PROGRAMMING (IAP) DRIVERS FOR ST7 HDFLASH OR XFLASH MCUS
AN1577	DEVICE FIRMWARE UPGRADE (DFU) IMPLEMENTATION FOR ST7 USB APPLICATIONS
AN1601	SOFTWARE IMPLEMENTATION FOR ST7DALI-EVAL
AN1603	USING THE ST7 USB DEVICE FIRMWARE UPGRADE DEVELOPMENT KIT (DFU-DK)
AN1635	ST7 CUSTOMER ROM CODE RELEASE INFORMATION
AN1754	DATA LOGGING PROGRAM FOR TESTING ST7 APPLICATIONS VIA ICC
AN1796	FIELD UPDATES FOR FLASH BASED ST7 APPLICATIONS USING A PC COMM PORT
AN1900	HARDWARE IMPLEMENTATION FOR ST7DALI-EVAL
AN1904	ST7MC THREE-PHASE AC INDUCTION MOTOR CONTROL SOFTWARE LIBRARY
AN1905	ST7MC THREE-PHASE BLDC MOTOR CONTROL SOFTWARE LIBRARY
SYSTEM OPTIMIZATION	
AN1711	SOFTWARE TECHNIQUES FOR COMPENSATING ST7 ADC ERRORS
AN1827	IMPLEMENTATION OF SIGMA-DELTA ADC WITH ST7FLITE05/09
AN2009	PWM MANAGEMENT FOR 3-PHASE BLDC MOTOR DRIVES USING THE ST7FMC
AN2030	BACK EMF DETECTION DURING PWM ON TIME BY ST7MC

