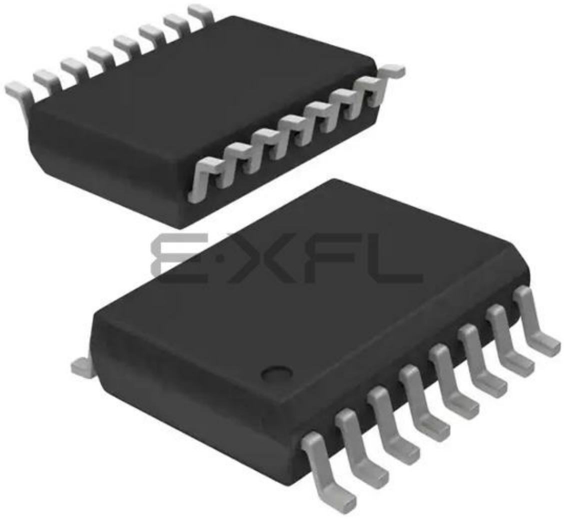


Welcome to E-XFL.COM

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	Surface Mount
Package / Case	16-SOIC (0.295", 7.50mm Width)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7flit15by1m6

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.

POWER SAVING MODES (Cont'd)

9.4 HALT MODE

The HALT mode is the lowest power consumption mode of the MCU. It is entered by executing the 'HALT' instruction when ACTIVE-HALT is disabled (see section 9.5 on page 43 for more details) and when the AWUEN bit in the AWUCSR register is cleared.

The MCU can exit HALT mode on reception of either a specific interrupt (see Table 5, "Interrupt Mapping," on page 37) or a RESET. When exiting HALT mode by means of a RESET or an interrupt, the oscillator is immediately turned on and the 256 or 4096 CPU cycle delay is used to stabilize the oscillator. After the start up delay, the CPU resumes operation by servicing the interrupt or by fetching the reset vector which woke it up (see Figure 26).

When entering HALT mode, the I bit in the CC register is forced to 0 to enable interrupts. Therefore, if an interrupt is pending, the MCU wakes up immediately.

In HALT mode, the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. All peripherals are not clocked except the ones which get their clock supply from another clock generator (such as an external or auxiliary oscillator).

The compatibility of Watchdog operation with HALT mode is configured by the "WDGHALT" option bit of the option byte. The HALT instruction when executed while the Watchdog system is enabled, can generate a Watchdog RESET (see section 15.1 on page 149 for more details).

Figure 25. HALT Timing Overview

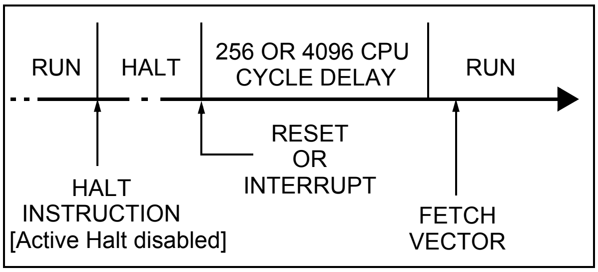
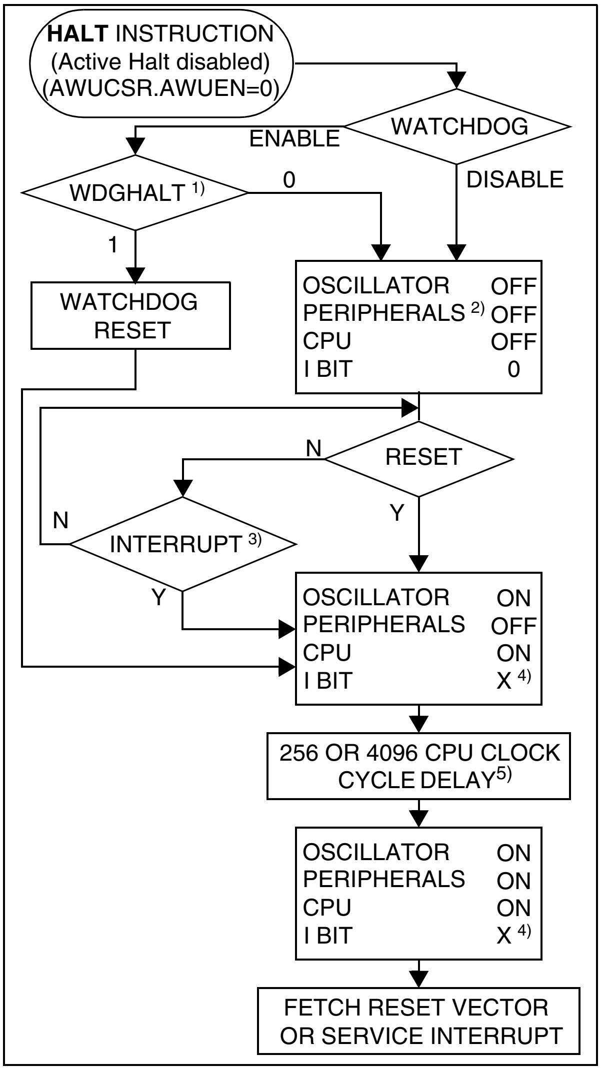


Figure 26. HALT Mode Flow-chart



Notes:

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

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

3. Only some specific interrupts can exit the MCU from HALT mode (such as external interrupt). Refer to Table 5 Interrupt Mapping for more details.

4. Before servicing an interrupt, the CC register is pushed on the stack. The I bit of the CC register is set during the interrupt routine and cleared when the CC register is popped.

5. If the PLL is enabled by option byte, it outputs the clock after a delay of $t_{STARTUP}$ (see Figure 13).

POWER SAVING MODES (Cont'd)

Figure 27. ACTIVE-HALT Timing Overview

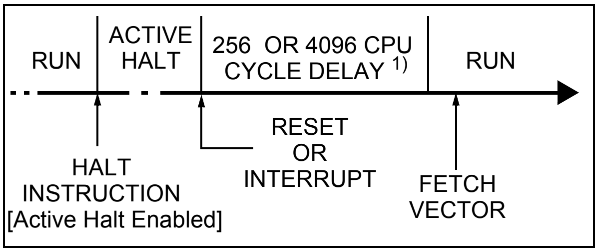
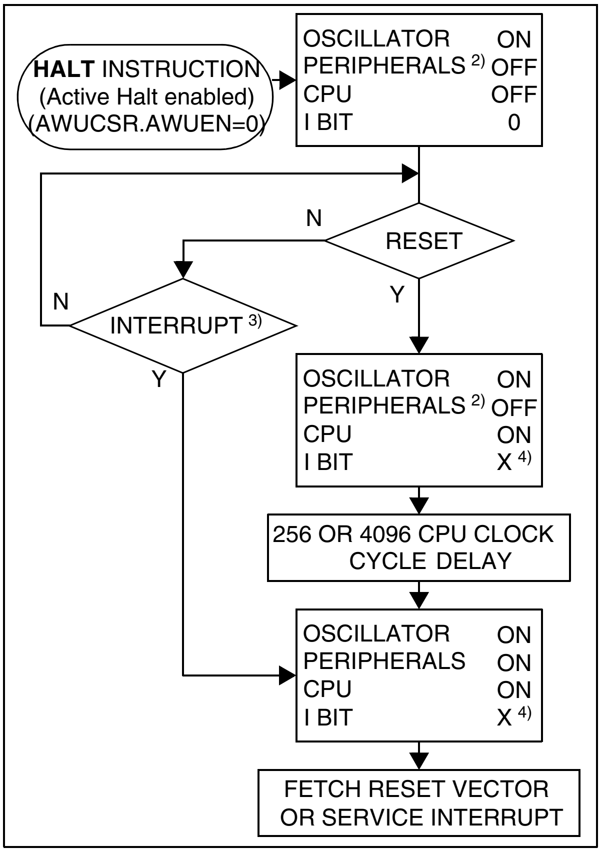


Figure 28. ACTIVE-HALT Mode Flow-chart



Notes:

1. This delay occurs only if the MCU exits ACTIVE-HALT mode by means of a RESET.

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

3. Only the RTC1 interrupt and some specific interrupts can exit the MCU from ACTIVE-HALT mode. Refer to Table 5, "Interrupt Mapping," on page 37 for more details.

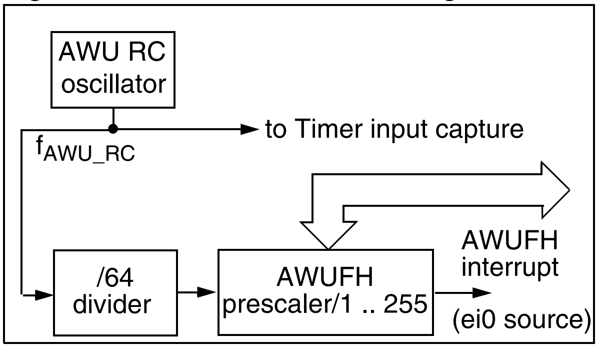
4. Before servicing an interrupt, the CC register is pushed on the stack. The I bit of the CC register is set during the interrupt routine and cleared when the CC register is popped.

9.6 AUTO WAKE UP FROM HALT MODE

Auto Wake Up From Halt (AWUFH) mode is similar to Halt mode with the addition of a specific internal RC oscillator for wake-up (Auto Wake Up from Halt Oscillator). Compared to ACTIVE-HALT mode, AWUFH has lower power consumption (the main clock is not kept running, but there is no accurate realtime clock available.

It is entered by executing the HALT instruction when the AWUEN bit in the AWUCSR register has been set.

Figure 29. AWUFH Mode Block Diagram



As soon as HALT mode is entered, and if the AWUEN bit has been set in the AWUCSR register, the AWU RC oscillator provides a clock signal (f_{AWU_RC}). Its frequency is divided by a fixed divider and a programmable prescaler controlled by the AWUPR register. The output of this prescaler provides the delay time. When the delay has elapsed the AWUF flag is set by hardware and an interrupt wakes-up the MCU from Halt mode. At the same time the main oscillator is immediately turned on and a 256 or 4096 cycle delay is used to stabilize it. After this start-up delay, the CPU resumes operation by servicing the AWUFH interrupt. The AWU flag and its associated interrupt are cleared by software reading the AWUCSR register.

To compensate for any frequency dispersion of the AWU RC oscillator, it can be calibrated by measuring the clock frequency f_{AWU_RC} and then calculating the right prescaler value. Measurement mode is enabled by setting the AWUM bit in the AWUCSR register in Run mode. This connects f_{AWU_RC} to the input capture of the 12-bit Auto-Reload timer, allowing the f_{AWU_RC} to be measured using the main oscillator clock as a reference timebase.

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
0002h	**PAOR** Reset Value	MSB 0	1	0	0	0	0	0	LSB 0
0003h	**PBDR** Reset Value	MSB 1	1	1	1	1	1	1	LSB 1
0004h	**PBDDR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0
0005h	**PBOR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0
0006h	**PCDR** Reset Value	MSB 0	0	0	0	0	0	1	LSB 1
0007h	**PCDDR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 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

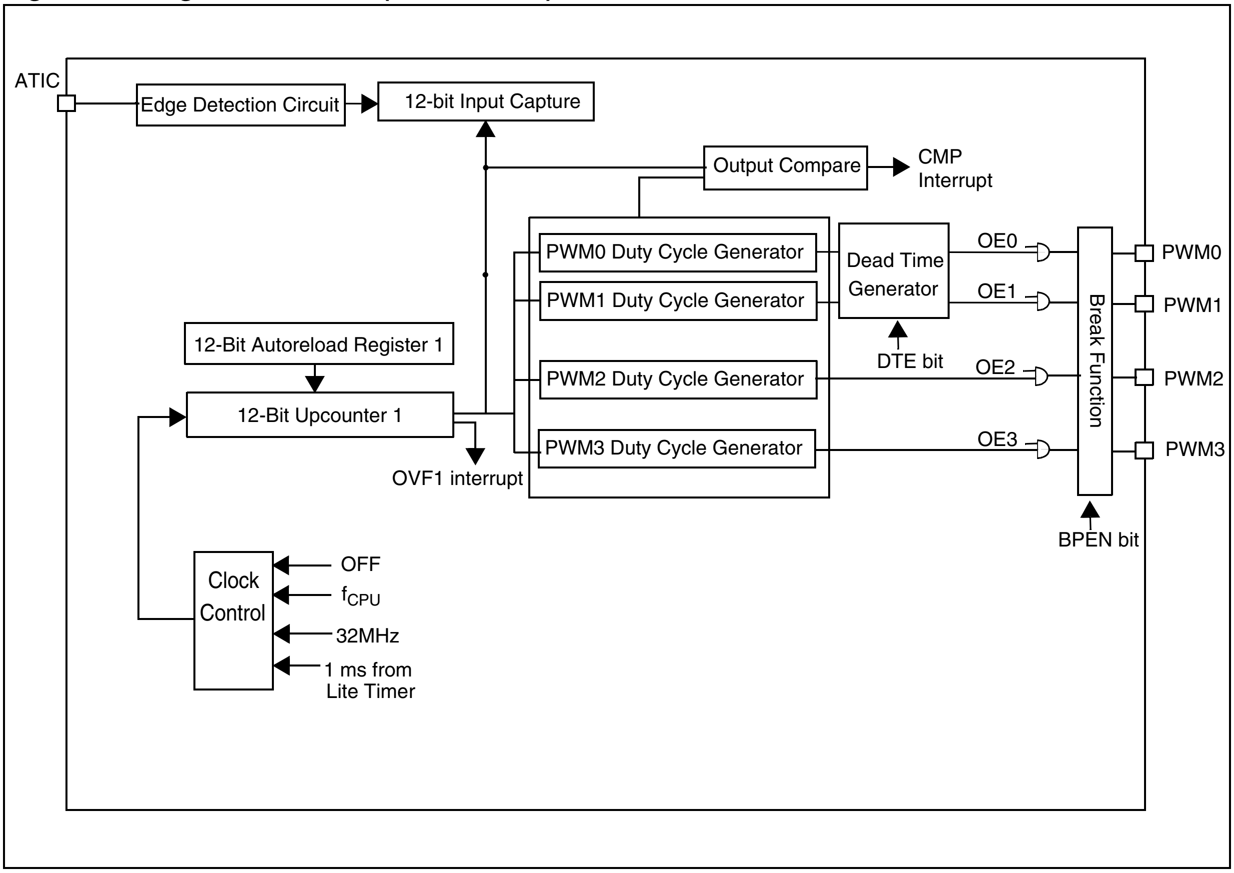
– ATIC/LTIC pins for the Input Capture function

– 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
  – 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

Figure 35. Single Timer Mode (ENCNTR2=0)



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

11.2.3.6 One Pulse Mode

One Pulse Mode can be used to control PWM2/3 signal with an external LTIC pin. This mode is available only in dual timer mode i.e. only for CNTR2, when the OP_EN bit in PWM3CSR register is set.

One Pulse Mode is activated by the external LTIC input. The active edge of the LTIC pin is selected by the OPEDGE bit in the PWM3CSR register.

After getting the active edge of the LTIC pin, CNTR2 is reset (000h) and PWM3 is set to high. CNTR2 starts counting from 000h, when it reaches the active DCR3 value then the PWM3 output goes low. Till this time, any further transitions on the LTIC signal will have no effect. If there are LTIC transitions after CNTR2 reaches the DCR3 value, CNTR2 is reset again and the PWM3 output goes high.

If there is no LTIC active edge then CNTR2 will count till it reaches the ATR2 value, and then it will be reset again and the PWM3 output is set to high. The counter again starts counting from 000h, when it reaches the active DCR3 value the PWM3 output goes low, the counter counts till it reaches the ATR2 value, it resets and the PWM3 output is set to high and it goes on the same way.

The same operation applies for the PWM2 output, but in this case the comparison is done on the DCR2 value.

The OP_EN and OPEDGE bits take effect on the fly and are not synchronized with the CNTR2 overflow.

The OP2/3 bits can be used to inverse the polarity of the PWM2/3 outputs in one-pulse mode. The update of these bits (OP2/3) is synchronized with the CNTR2 overflow, they will be updated if the TRAN2 bit is set.

Notes:

1. If CNTR2 is running at 32 MHz, the time taken from activation of LTIC input and CNTR2 reset is between 2 and 3 t_{CNTR2} cycles, i.e. 66 ns to 99 ns (with 8 MHz f_{cpu}).

2. The Lite Timer input capture interrupt must be disabled while 12-bit ARTimer is in One Pulse Mode. This is to avoid spurious interrupts.

3. The priority of various events affecting PWM3 is as follows:

– Break (Highest priority)

– One-pulse mode with active LTIC edge

– Forced overflow (by FORCE2 bit)

– One-pulse mode without active LTIC edge

– Normal PWM operation. (Lowest priority)

4. It is possible to synchronize the update of DCR2/3 registers and OP2/3 bits with the CNTR2 reset. This is managed by the overflow interrupt which is generated if CNTR2 is reset either due to an ATR match or an active pulse on the LTIC pin.

5. Updating the DCR2/3 registers and OP2/3 bits in one-pulse mode is done dynamically by software using force update (FORCE2 bit in the ATCSR2 register).

6. DCR3 update in this mode is not synchronized with any event. Consequently the next PWM3 cycle just after the change may be longer than expected (refer to Figure 15).

7. In One Pulse Mode the ATR2 value must be greater than the DCR2/3 value for the PWM2/3 outputs. (contrary to normal PWM mode)

8. If there is an active edge on the LTIC pin after the CNTR2 has reset due to an ATR2 match, then the timer gets reset again. The duty cycle may be modified depending on whether the new DCR value is less than or more than the previous value.

9. The TRAN2 bit must be set simultaneously with the FORCE2 bit in the same instruction after a write to the DCR register.

10. The ATR2 value should be changed after an overflow in one pulse mode to avoid an irregular PWM cycle.

11. When exiting from one pulse mode, the OP_EN bit in the PWM3CSR register must be reset first and then the ENCNTR2 bit (if CNTR2 is to be stopped).

How to Enter One Pulse Mode:

1. Load the ATR2H/ATR2L registers with required value.

2. Load the DCR3H/DCR3L registers for PWM3 output. The ATR2 value must be greater than DCR3.

3. Set the OP3 bit in the PWM3CSR register if polarity change is required.

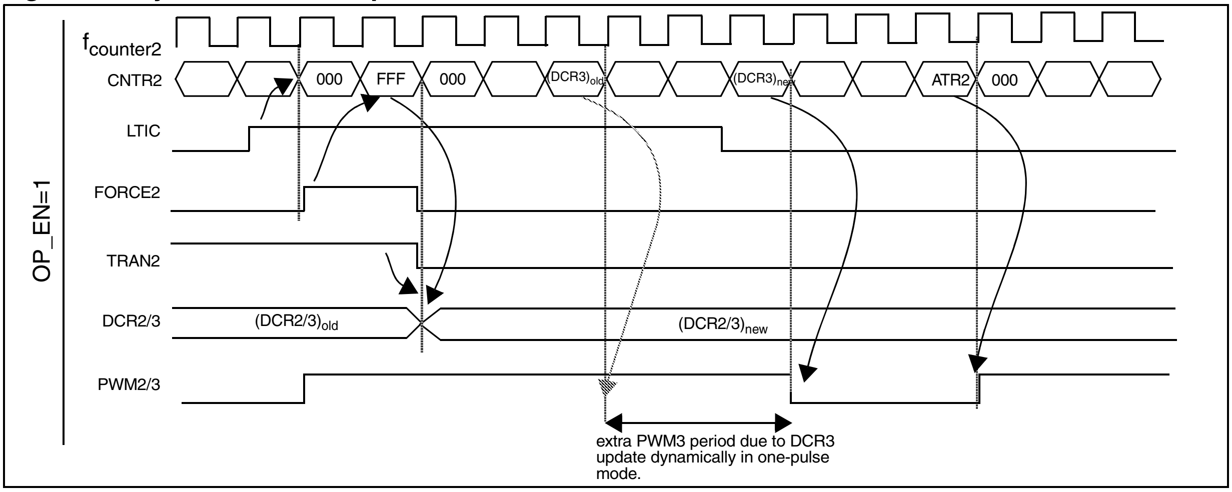
4. Start the CNTR2 counter by setting the ENCNTR2 bit in the ATCSR2 register.

5. Set TRAN2 bit in ATCSR2 to enable transfer.

6. Wait for an overflow event by polling the OVF2 flag in the ATCSR2 register.

7. Select the counter clock using the CK[1:0] bits in the ATCSR register.

Figure 49. Dynamic DCR2/3 update in One Pulse Mode



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

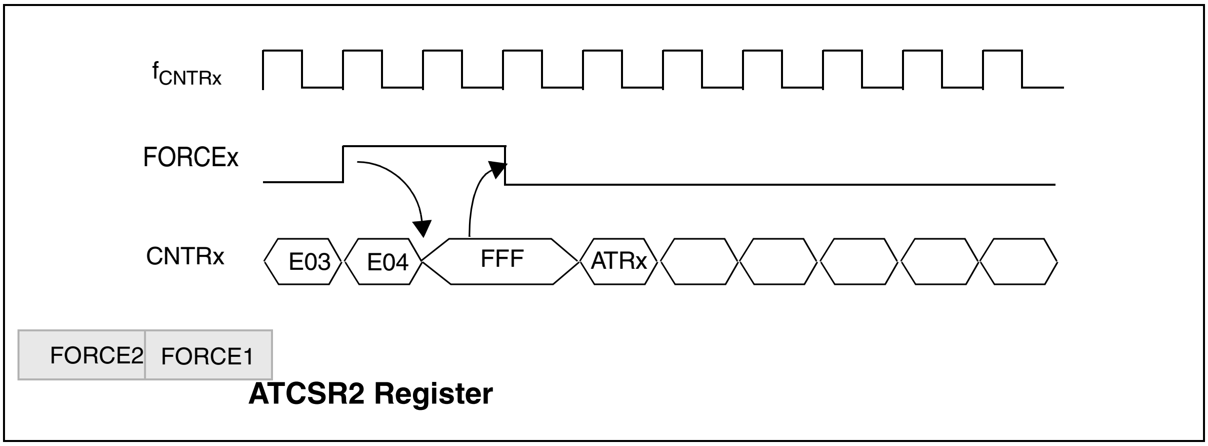
11.2.3.7 Force Update

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 reset 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.

Figure 50. Force Overflow Timing Diagram



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

11.2.6 Register Description

TIMER CONTROL STATUS REGISTER (ATCSR)
Read / Write
Reset Value: 0x00 0000 (x0h)

7							0
0	ICF	ICIE	CK1	CK0	OVF1	OVFIE1	CMPIE

Bit 7 = Reserved.

Bit 6 = **ICF** *Input Capture Flag.*
This bit is set by hardware and cleared by software by reading the ATICR register (a read access to ATICRH or ATICRL will clear this flag). Writing to this bit does not change the bit value.
0: No input capture
1: An input capture has occurred

Bit 5 = **ICIE** *IC Interrupt Enable.*
This bit is set and cleared by software.
0: Input capture interrupt disabled
1: Input capture interrupt enabled

Bits 4:3 = **CK[1:0]** *Counter Clock Selection.*
These bits are set and cleared by software and cleared by hardware after a reset. They select the clock frequency of the counter.

Counter Clock Selection	CK1	CK0
OFF	0	0
32 MHz	1	1
f_{LTIMER} (1 ms timebase @ 8 MHz)	0	1
f_{CPU}	1	0

Bit 2 = **OVF1** *Overflow Flag.*
This bit is set by hardware and cleared by software by reading the ATCSR register. It indicates the transition of the counter1 CNTR1 from FFFh to ATR1 value.
0: No counter overflow occurred
1: Counter overflow occurred

Bit 1 = **OVFIE1** *Overflow Interrupt Enable.*
This bit is read/write by software and cleared by hardware after a reset.
0: Overflow interrupt disabled.
1: Overflow interrupt enabled.

Bit 0 = **CMPIE** *Compare Interrupt Enable*.
This bit is read/write by software and cleared by hardware after a reset. It can be used to mask the interrupt generated when any of the CMPFx bit is set.
0: Output compare interrupt disabled.
1: Output Compare interrupt enabled.

COUNTER REGISTER 1 HIGH (CNTR1H)
Read only
Reset Value: 0000 0000 (00h)

15							8
0	0	0	0	CNTR1_11	CNTR1_10	CNTR1_9	CNTR1_8

COUNTER REGISTER 1 LOW (CNTR1L)
Read only
Reset Value: 0000 0000 (00h)

7							0
CNTR1_7	CNTR1_6	CNTR1_5	CNTR1_4	CNTR1_3	CNTR1_2	CNTR1_1	CNTR1_0

Bits 15:12 = Reserved.

Bits 11:0 = **CNTR1[11:0]** *Counter Value*.
This 12-bit register is read by software and cleared by hardware after a reset. The counter CNTR1 increments continuously as soon as a counter clock is selected. To obtain the 12-bit value, software should read the counter value in two consecutive read operations. As there is no latch, it is recommended to read LSB first. In this case, CNTR1H can be incremented between the two read operations and to have an accurate result when $f_{timer}=f_{CPU}$, special care must be taken when CNTR1L values close to FFh are read.
When a counter overflow occurs, the counter restarts from the value specified in the ATR1 register.

11.5 10-BIT A/D CONVERTER (ADC)

11.5.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 10-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 7 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 7 different sources.

The result of the conversion is stored in a 10-bit Data Register. The A/D converter is controlled through a Control/Status Register.

11.5.2 Main Features

- 10-bit conversion
- Up to 7 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in Figure 60.

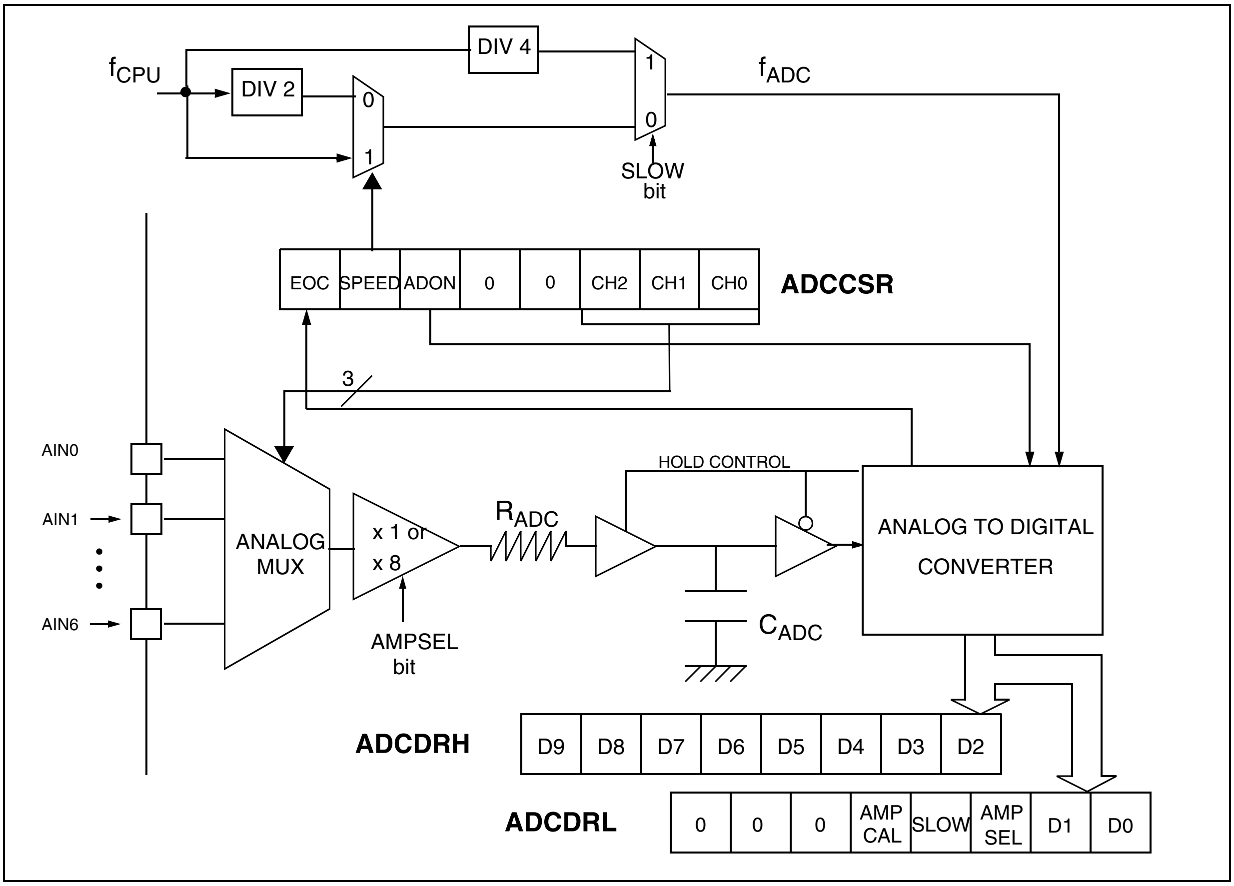
11.5.3 Functional Description

11.5.3.1 Analog Power Supply

V_{DDA} and V_{SSA} are the high and low level reference voltage pins. In some devices (refer to device pin out description) they are internally connected to the V_{DD} and V_{SS} pins.

Conversion accuracy may therefore be impacted by voltage drops and noise in the event of heavily loaded or badly decoupled power supply lines.

Figure 60. ADC Block Diagram



ST7 ADDRESSING MODES (cont'd)

12.1.1 Inherent

All Inherent instructions consist of a single byte. The opcode fully specifies all the required information for the CPU to process the operation.

Inherent Instruction	Function
NOP	No operation
TRAP	S/W Interrupt
WFI	Wait For Interrupt (Low Power Mode)
HALT	Halt Oscillator (Lowest Power Mode)
RET	Subroutine Return
IRET	Interrupt Subroutine Return
SIM	Set Interrupt Mask
RIM	Reset Interrupt Mask
SCF	Set Carry Flag
RCF	Reset Carry Flag
RSP	Reset Stack Pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
MUL	Byte Multiplication
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles

12.1.2 Immediate

Immediate instructions have 2 bytes, the first byte contains the opcode, the second byte contains the operand value.

Immediate Instruction	Function
LD	Load
CP	Compare
BCP	Bit Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Operations

12.1.3 Direct

In Direct instructions, the operands are referenced by their memory address.

The direct addressing mode consists of two submodes:

Direct (Short)

The address is a byte, thus requires only 1 byte after the opcode, but only allows 00 - FF addressing space.

Direct (Long)

The address is a word, thus allowing 64 Kbyte addressing space, but requires 2 bytes after the opcode.

12.1.4 Indexed (No Offset, Short, Long)

In this mode, the operand is referenced by its memory address, which is defined by the unsigned addition of an index register (X or Y) with an offset.

The indirect addressing mode consists of three submodes:

Indexed (No Offset)

There is no offset (no extra byte after the opcode), and allows 00 - FF addressing space.

Indexed (Short)

The offset is a byte, thus requires only 1 byte after the opcode and allows 00 - 1FE addressing space.

Indexed (Long)

The offset is a word, thus allowing 64 Kbyte addressing space and requires 2 bytes after the opcode.

12.1.5 Indirect (Short, Long)

The required data byte to do the operation is found by its memory address, located in memory (pointer).

The pointer address follows the opcode. The indirect addressing mode consists of two submodes:

Indirect (Short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - FF addressing space, and requires 1 byte after the opcode.

Indirect (Long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

INSTRUCTION GROUPS (cont'd)

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
JRULE	Jump if (C + Z = 1)	Unsigned <=							
LD	Load	dst <= src	reg, M	M, reg			N	Z	
MUL	Multiply	X,A = X * A	A, X, Y	X, Y, A	0				0
NEG	Negate (2's compl)	neg $10	reg, M				N	Z	C
NOP	No Operation								
OR	OR operation	A = A + M	A	M			N	Z	
POP	Pop from the Stack	pop reg	reg	M					
		pop CC	CC	M	H	I	N	Z	C
PUSH	Push onto the Stack	push Y	M	reg, CC					
RCF	Reset carry flag	C = 0							0
RET	Subroutine Return								
RIM	Enable Interrupts	I = 0				0			
RLC	Rotate left true C	C <= Dst <= C	reg, M				N	Z	C
RRC	Rotate right true C	C => Dst => C	reg, M				N	Z	C
RSP	Reset Stack Pointer	S = Max allowed							
SBC	Subtract with Carry	A = A - M - C	A	M			N	Z	C
SCF	Set carry flag	C = 1							1
SIM	Disable Interrupts	I = 1				1			
SLA	Shift left Arithmetic	C <= Dst <= 0	reg, M				N	Z	C
SLL	Shift left Logic	C <= Dst <= 0	reg, M				N	Z	C
SRL	Shift right Logic	0 => Dst => C	reg, M				0	Z	C
SRA	Shift right Arithmetic	Dst7 => Dst => C	reg, M				N	Z	C
SUB	Subtraction	A = A - M	A	M			N	Z	C
SWAP	SWAP nibbles	Dst[7..4] <=> Dst[3..0]	reg, M				N	Z	
TNZ	Test for Neg & Zero	tnz lbl1					N	Z	
TRAP	S/W trap	S/W interrupt				1			
WFI	Wait for Interrupt					0			
XOR	Exclusive OR	A = A XOR M	A	M			N	Z	

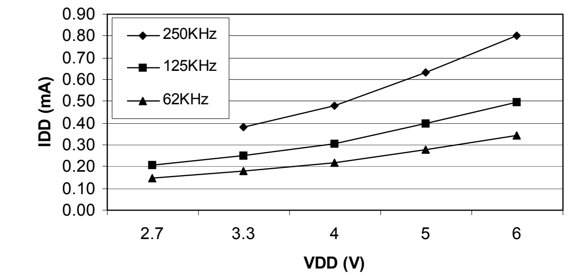
13.3.6 Operating conditions with ADC

T_A = -40 to 125°C, unless otherwise specified

Symbol	Parameter	Typ	Unit
$I_{INJ(ANA)}$ [1)]	Injected current on any analog pin	0	mA

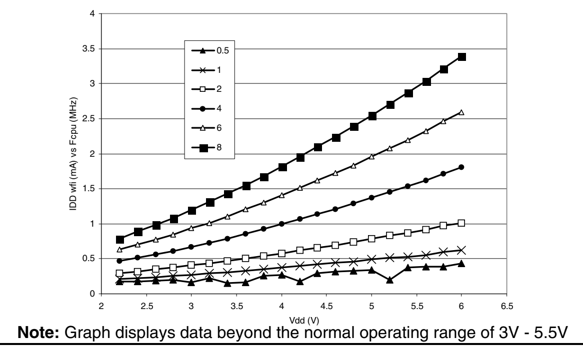
Note:
1. Current injection (negative or positive) not allowed on any analog pin.

Figure 73. Typical I_{DD} in SLOW vs. f_{CPU}



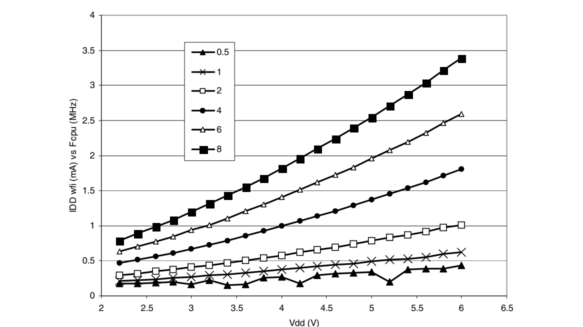
Note: Graph displays data beyond the normal operating range of 3V - 5.5V

Figure 74. Typical I_{DD} in WAIT vs. f_{CPU}



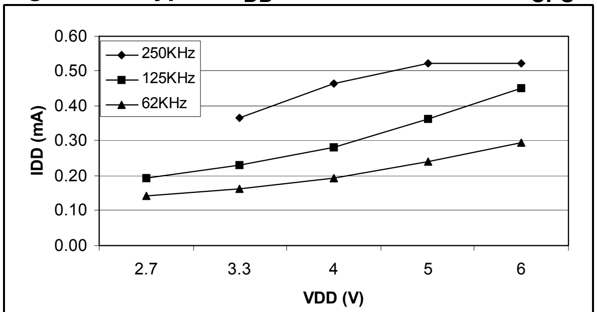
Note: Graph displays data beyond the normal operating range of 3V - 5.5V

Figure 75. Typical I_{DD} in WAIT at f_{CPU}= 8MHz



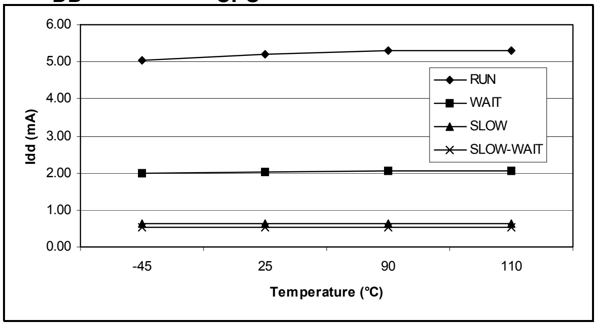
Note: Graph displays data beyond the normal operating range of 3V - 5.5V

Figure 76. Typical I_{DD} in SLOW-WAIT vs. f_{CPU}



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

Figure 77. Typical I_{DD} vs. Temperature at V_{DD} = 5V and f_{CPU} = 8MHz



13.4.2 On-chip peripherals

Symbol	Parameter	Conditions		Typ	Unit
$I_{DD(AT)}$	12-bit Auto-Reload Timer supply current [1]	f_{CPU}=4MHz	V_{DD}=3.0V	150	μA
		f_{CPU}=8MHz	V_{DD}=5.0V	1000	
$I_{DD(SPI)}$	SPI supply current [2]	f_{CPU}=4MHz	V_{DD}=3.0V	50	
		f_{CPU}=8MHz	V_{DD}=5.0V	200	
$I_{DD(ADC)}$	ADC supply current when converting [3]	f_{ADC}=4MHz	V_{DD}=3.0V	250	
			V_{DD}=5.0V	1100	

Notes:

1. Data based on a differential I_{DD} measurement between reset configuration (timer stopped) and a timer running in PWM mode at f_{cpu}=8MHz.

2. Data based on a differential I_{DD} measurement between reset configuration and a permanent SPI master communication (data sent equal to 55h).

3. Data based on a differential I_{DD} measurement between reset configuration and continuous A/D conversions with amplifier disabled.

I/O PORT PIN CHARACTERISTICS (Cont'd)

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

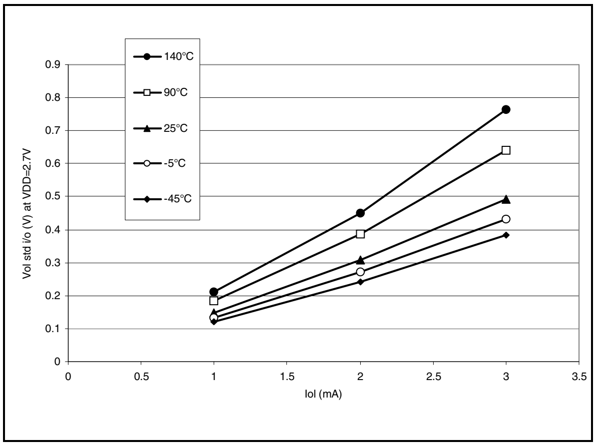


Figure 82. Typical V_{OL} at V_{DD}=3.3V (standard)

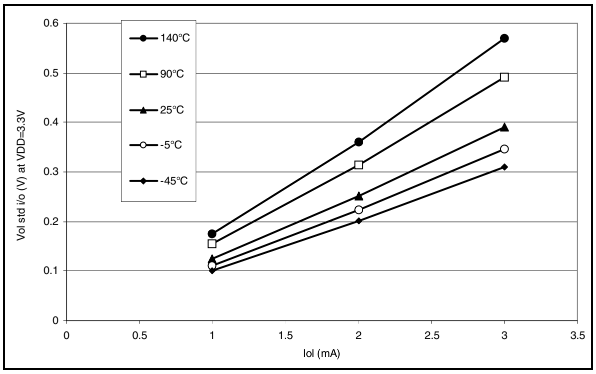


Figure 83. Typical V_{OL} at V_{DD}=5V (standard)

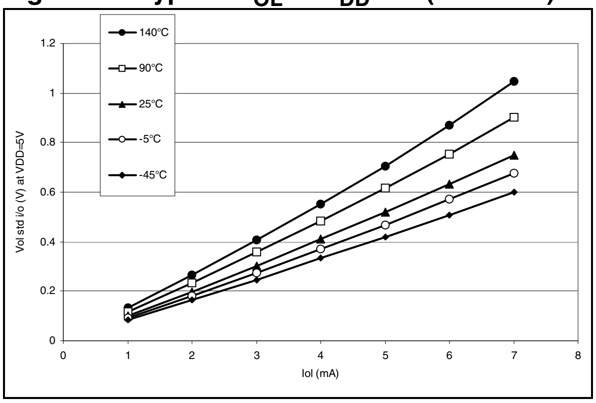


Figure 84. Typical V_{OL} at V_{DD}=2.7V (Port C)

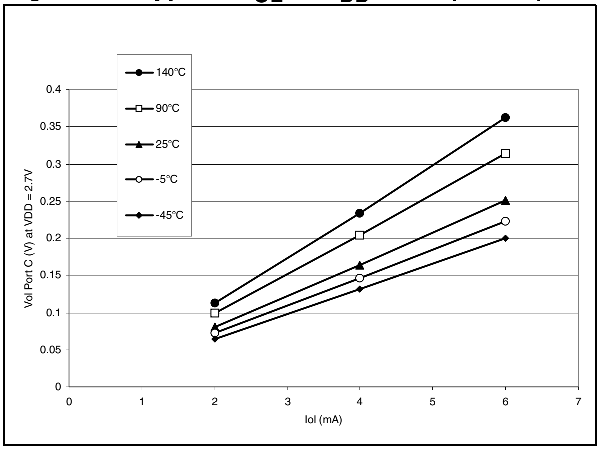


Figure 85. Typical V_{OL} at V_{DD}=3.3V (Port C)

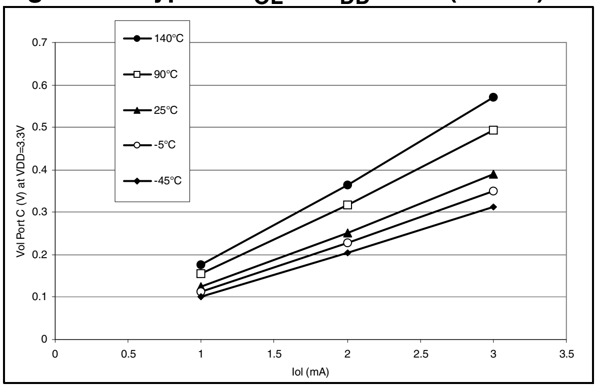
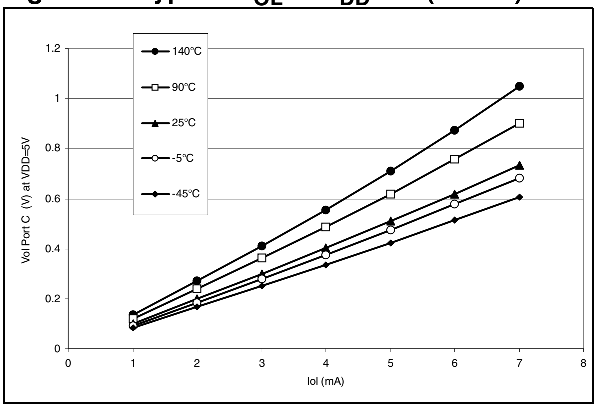


Figure 86. Typical V_{OL} at V_{DD}=5V (Port C)



CONTROL PIN CHARACTERISTICS (Cont'd)

Figure 105. RESET pin protection when LVD is enabled.[1)2)3)4)]

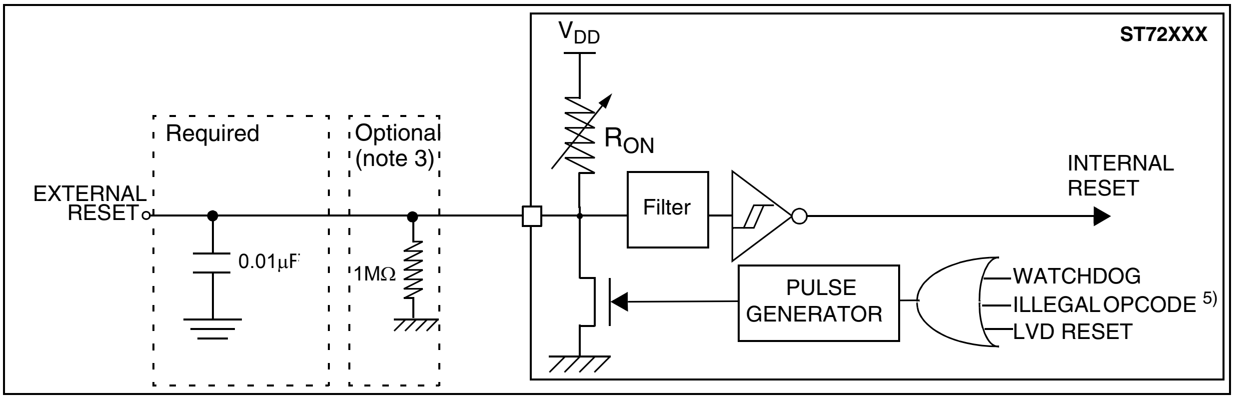
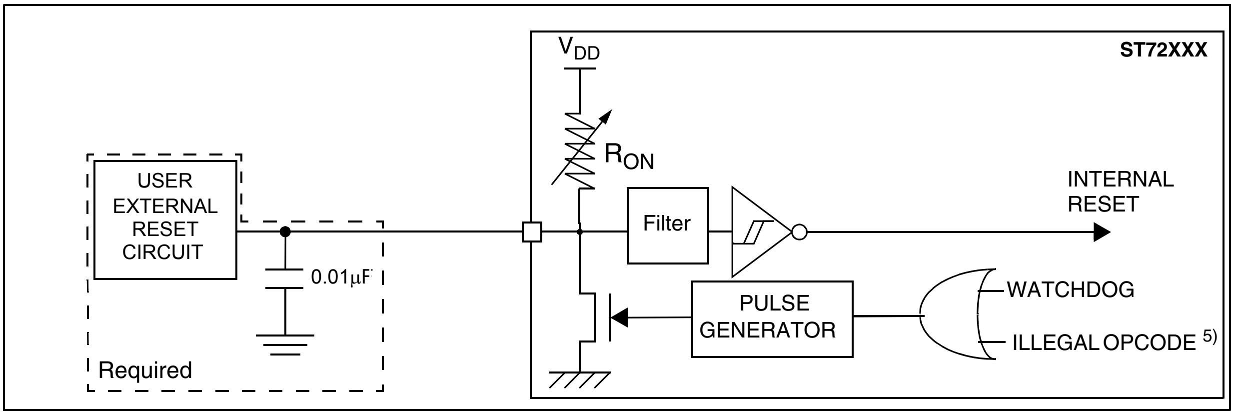


Figure 106. RESET pin protection when LVD is disabled.[1)]



Note 1:
- The reset network protects the device against parasitic resets.
- The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).
- Whatever the reset source is (internal or external), the user must ensure that the level on the RESET pin can go below the V_{IL} max. level specified in section 13.9.1 on page 135. Otherwise the reset will not be taken into account internally.
- Because the reset circuit is designed to allow the internal RESET to be output in the RESET pin, the user must ensure that the current sunk on the RESET pin is less than the absolute maximum value specified for $I_{INJ(RESET)}$ in section 13.2.2 on page 111.

Note 2: When the LVD is enabled, it is recommended not to connect a pull-up resistor or capacitor. A 10nF pull-down capacitor is required to filter noise on the reset line.

Note 3: In case a capacitive power supply is used, it is recommended to connect a 1MΩ pull-down resistor to the RESET pin to discharge any residual voltage induced by the capacitive effect of the power supply (this will add 5µA to the power consumption of the MCU).

Note 4: Tips when using the LVD:
- 1. Check that all recommendations related to ICCCLK and reset circuit have been applied (see caution in Table 1 on page 7 and notes above)
- 2. Check that the power supply is properly decoupled (100nF + 10µF close to the MCU). Refer to AN1709 and AN2017. If this cannot be done, it is recommended to put a 100nF + 1MΩ pull-down on the RESET pin.
- 3. The capacitors connected on the RESET pin and also the power supply are key to avoid any start-up marginality. In most cases, steps 1 and 2 above are sufficient for a robust solution. Otherwise: replace 10nF pull-down on the RESET pin with a 5µF to 20µF capacitor."

Note 5: Please refer to "Illegal Opcode Reset" on page 107 for more details on illegal opcode reset conditions.

14.2 SOLDERING INFORMATION

In accordance with the RoHS European directive, all STMicroelectronics packages have been converted to lead-free technology, named ECOPACK^{TM}.

- ECOPACK^{TM} packages are qualified according to the JEDEC STD-020C compliant soldering profile.
- Detailed information on the STMicroelectronics ECOPACK^{TM} transition program is available on www.st.com/stonline/leadfree/, with specific technical Application notes covering the main technical aspects related to lead-free conversion (AN2033, AN2034, AN2035, AN2036).

Backward and forward compatibility:

The main difference between Pb and Pb-free soldering process is the temperature range.

– ECOPACK^{TM} TQFP, SDIP, SO and QFN20 packages are fully compatible with Lead (Pb) containing soldering process (see application note AN2034)

– TQFP, SDIP and SO Pb-packages are compatible with Lead-free soldering process, nevertheless it's the customer's duty to verify that the Pb-packages maximum temperature (mentioned on the Inner box label) is compatible with their Lead-free soldering temperature.

Table 25. Soldering Compatibility (wave and reflow soldering process)

Package	Plating material devices	Pb solder paste	Pb-free solder paste
SDIP & PDIP	Sn (pure Tin)	Yes	Yes *
QFN	Sn (pure Tin)	Yes	Yes *
TQFP and SO	NiPdAu (Nickel-palladium-Gold)	Yes	Yes *

* Assemblers must verify that the Pb-package maximum temperature (mentioned on the Inner box label) is compatible with their Lead-free soldering process.

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.

Date	Revision	Main changes
27-Nov-06	4	Added QFN20 pinout with new mechanical data (Figure 3 on page 5 and Figure 117 on page 145) Added ST7FLI19BY1M3TR sales type in Table 1, “Supported Flash part numbers,” Modifed “DEVELOPMENT TOOLS” on page 153
23-April-07	5	Added note 1 to Table 1 on page 7 Modified note 1 in section 7.1 on page 23 Added caution to section 7.5.1 on page 28 Modified section 11.2.3.6 on page 67 Modified title of Figure 48 on page 68 and added note 1 Modified Figure 49 on page 69 Modified section 11.5.3.4 on page 97 and added section 11.5.3.5 on page 97 Modified EOC bit description in section 11.5.6 on page 98 Modified V_{FFTB} parameter in section 13.7.1 on page 127 Modified Table 28 on page 153
17-June-08	6	Modified first page Added note 2 in Table 1, “Device Pin Description,” on page 7 Modified WDGRF bit description in section 7.6.4 on page 35 Modified note 1 in section 11.2.3.6 on page 67 Added section 13.3.6 on page 120 Modified CLKSEL option bits description in section 15.1 on page 149 Modified section 15.2 on page 151 and option list