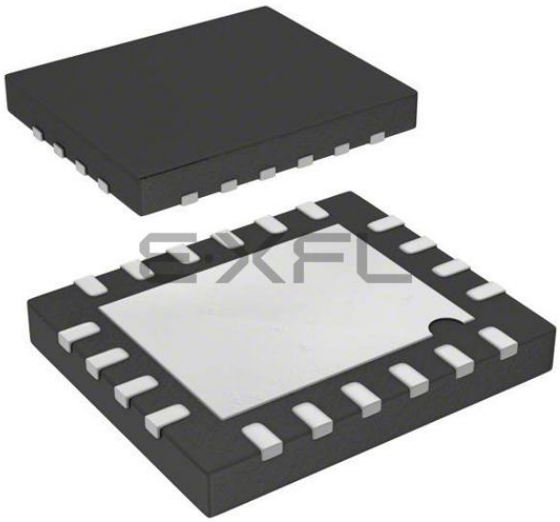


Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	17
Program Memory Size	4KB (4K 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 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-VQFN Exposed Pad
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fli19bf1u6tr

Table of Contents

4 FLASH PROGRAM MEMORY

4.1 Introduction

The ST7 single voltage extended Flash (XFlash) is a non-volatile memory that can be electrically erased and programmed either on a byte-by-byte basis or up to 32 bytes in parallel.

The XFlash devices can be programmed off-board (plugged in a programming tool) or on-board using In-Circuit Programming or In-Application Programming.

The array matrix organisation allows each sector to be erased and reprogrammed without affecting other sectors.

4.2 Main Features

- ICP (In-Circuit Programming)
- IAP (In-Application Programming)
- ICT (In-Circuit Testing) for downloading and executing user application test patterns in RAM
- Sector 0 size configurable by option byte
- Read-out and write protection

4.3 PROGRAMMING MODES

The ST7 can be programmed in three different ways:

– Insertion in a programming tool. In this mode, FLASH sectors 0 and 1, option byte row and data EEPROM (if present) can be programmed or erased.

– In-Circuit Programming. In this mode, FLASH sectors 0 and 1, option byte row and data EEPROM (if present) can be programmed or erased without removing the device from the application board.

– In-Application Programming. In this mode, sector 1 and data EEPROM (if present) can be programmed or erased without removing the device from the application board and while the application is running.

4.3.1 In-Circuit Programming (ICP)

ICP uses a protocol called ICC (In-Circuit Communication) which allows an ST7 plugged on a printed circuit board (PCB) to communicate with an external programming device connected via cable. ICP is performed in three steps:

Switch the ST7 to ICC mode (In-Circuit Communications). This is done by driving a specific signal sequence on the ICCCLK/DATA pins while the RESET pin is pulled low. When the ST7 enters ICC mode, it fetches a specific RESET vector which points to the ST7 System Memory containing the ICC protocol routine. This routine enables the ST7 to receive bytes from the ICC interface.

– Download ICP Driver code in RAM from the ICCDATA pin

– Execute ICP Driver code in RAM to program the FLASH memory

Depending on the ICP Driver code downloaded in RAM, FLASH memory programming can be fully customized (number of bytes to program, program locations, or selection of the serial communication interface for downloading).

4.3.2 In Application Programming (IAP)

This mode uses an IAP Driver program previously programmed in Sector 0 by the user (in ICP mode).

This mode is fully controlled by user software. This allows it to be adapted to the user application, (user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored etc.)
IAP mode can be used to program any memory areas except Sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

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.

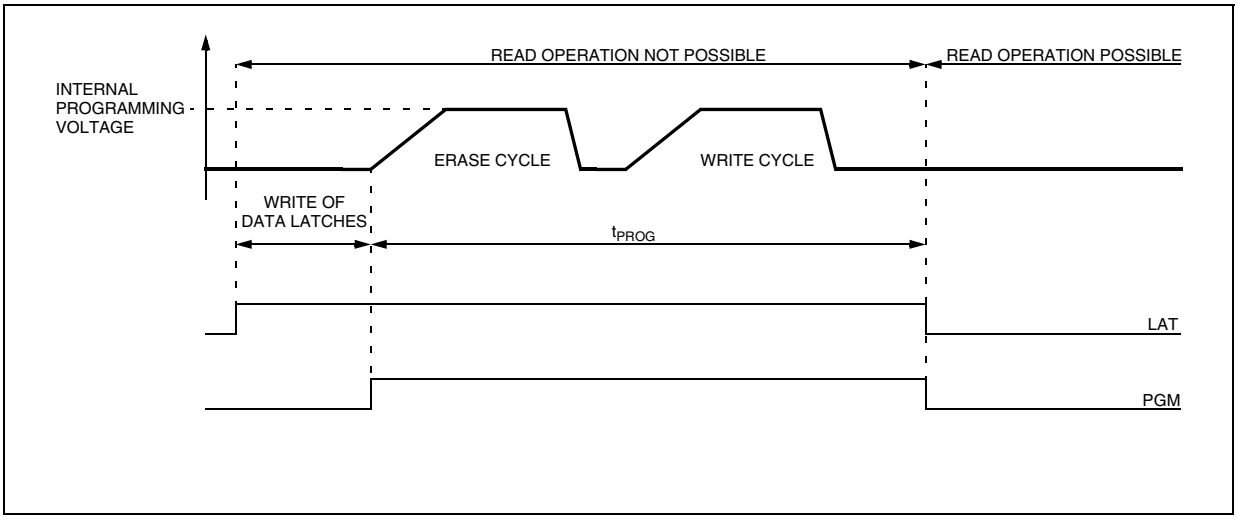
5.6 Data EEPROM Read-out Protection

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



6 CENTRAL PROCESSING UNIT

6.1 INTRODUCTION

This CPU has a full 8-bit architecture and contains six internal registers allowing efficient 8-bit data manipulation.

6.2 MAIN FEATURES

- 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes
- Two 8-bit index registers
- 16-bit stack pointer
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

6.3 CPU REGISTERS

The six CPU registers shown in Figure 1 are not present in the memory mapping and are accessed by specific instructions.

Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

Index Registers (X and Y)

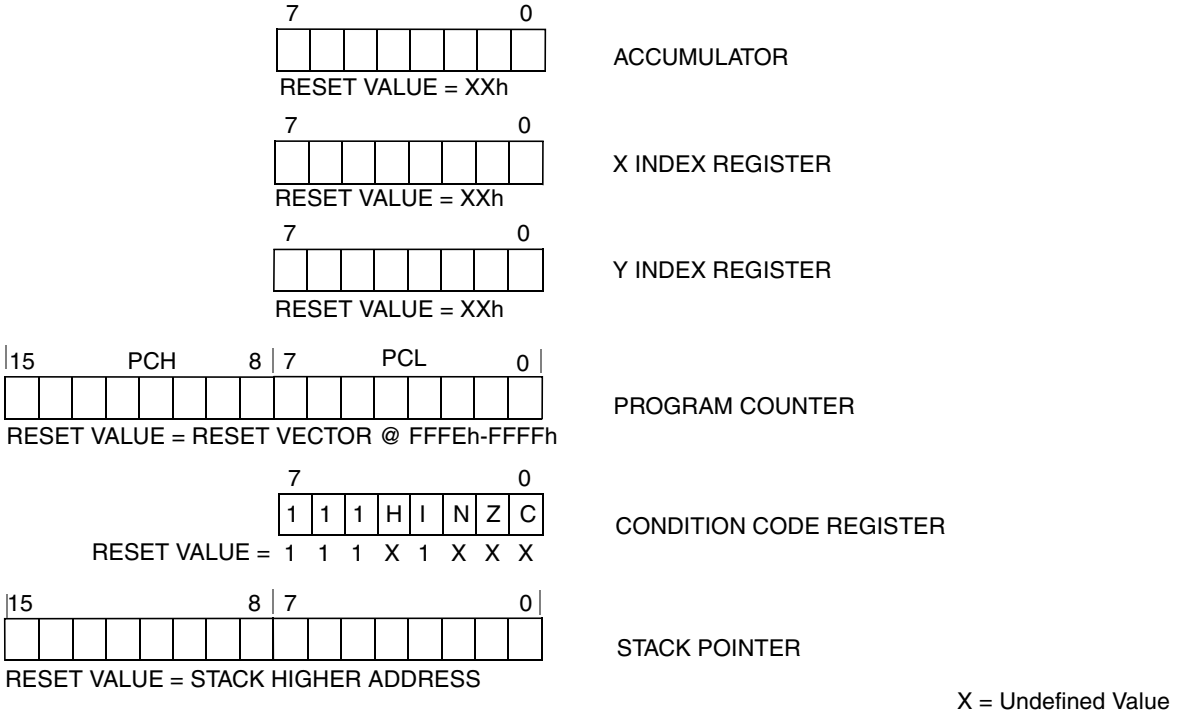
In indexed addressing modes, these 8-bit registers are used to create either effective addresses or temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures (not pushed to and popped from the stack).

Program Counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

Figure 11. CPU Registers



CPU REGISTERS (cont'd)

CONDITION CODE REGISTER (CC)

Read/Write

Reset Value: 111x1xxx

7							0
1	1	1	H	I	N	Z	C

The 8-bit Condition Code register contains the interrupt mask and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions.

These bits can be individually tested and/or controlled by specific instructions.

Bit 4 = **H** *Half carry*

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. It is reset by hardware during the same instructions.
0: No half carry has occurred.
1: A half carry has occurred.

This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Bit 3 = **I** *Interrupt mask*

This bit is set by hardware when entering in interrupt or by software to disable all interrupts except the TRAP software interrupt. This bit is cleared by software.
0: Interrupts are enabled.
1: Interrupts are disabled.

This bit is controlled by the RIM, SIM and IRET instructions and is tested by the JRM and JRNM instructions.

Note: Interrupts requested while I is set are latched and can be processed when I is cleared. By default an interrupt routine is not interruptible because the I bit is set by hardware at the start of the routine and reset by the IRET instruction at the end of the routine. If the I bit is cleared by software in the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine.

Bit 2 = **N** *Negative*

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It is a copy of the 7^{th} bit of the result.
0: The result of the last operation is positive or null.
1: The result of the last operation is negative (that is, the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = **Z** *Zero*

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.
0: The result of the last operation is different from zero.
1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

Bit 0 = **C** *Carry/borrow*

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.
0: No overflow or underflow has occurred.
1: An overflow or underflow has occurred.

This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the "bit test and branch", shift and rotate instructions.

RESET SEQUENCE MANAGER (Cont'd)

The $\overline{\text{RESET}}$ pin is an asynchronous signal which plays a major role in EMS performance. In a noisy environment, it is recommended to follow the guidelines mentioned in the electrical characteristics section.

7.5.3 External Power-On RESET

If the LVD is disabled by option byte, to start up the microcontroller correctly, the user must ensure by means of an external reset circuit that the reset signal is held low until V_{DD} is over the minimum level specified for the selected f_{OSC} frequency.

A proper reset signal for a slow rising V_{DD} supply can generally be provided by an external RC network connected to the $\overline{\text{RESET}}$ pin.

7.5.4 Internal Low Voltage Detector (LVD) RESET

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

- Power-On RESET
- Voltage Drop RESET

The device $\overline{\text{RESET}}$ pin acts as an output that is pulled low when $V_{DD}<V_{IT+}$ (rising edge) or $V_{DD}<V_{IT-}$ (falling edge) as shown in Figure 17.

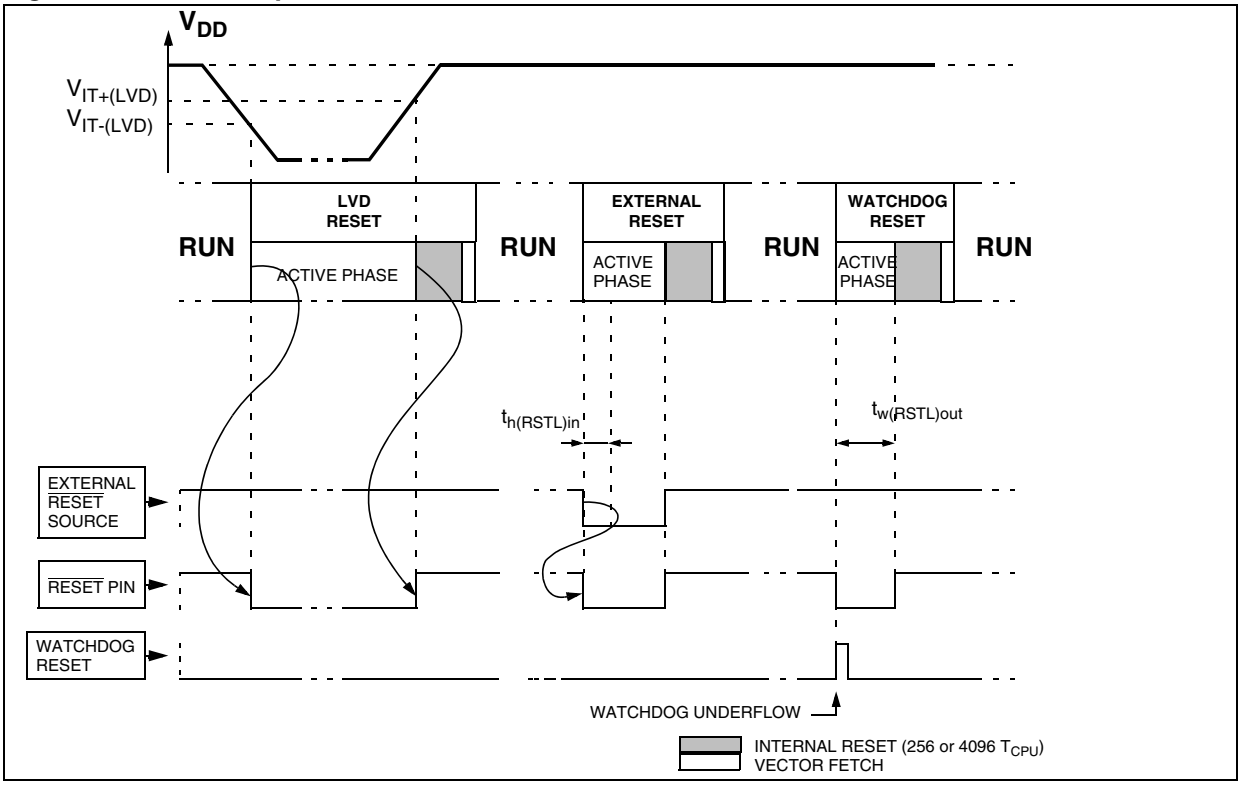
The LVD filters spikes on V_{DD} larger than $t_{g(VDD)}$ to avoid parasitic resets.

7.5.5 Internal Watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow is shown in Figure 17.

Starting from the Watchdog counter underflow, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.

Figure 17. RESET Sequences



POWER SAVING MODES (Cont'd)

Similarities with Halt mode

The following AWUFH mode behaviour is the same as normal Halt mode:

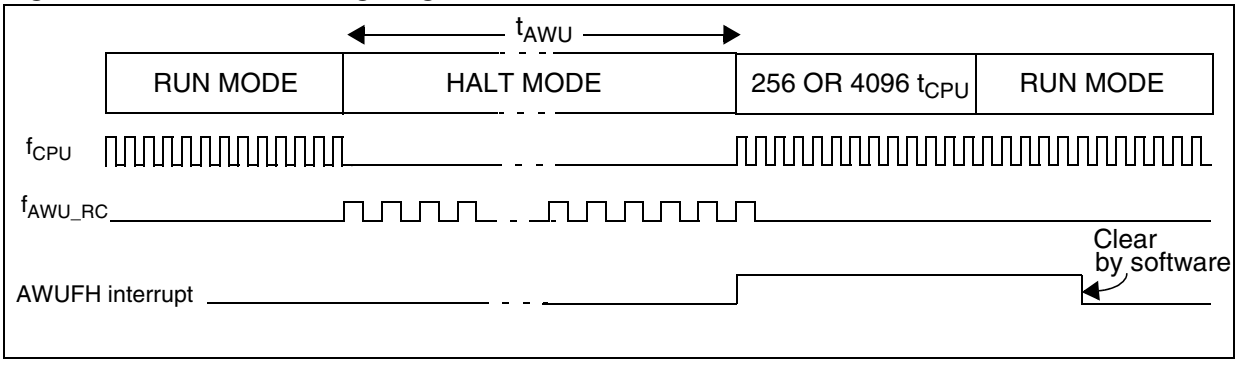
– The MCU can exit AWUFH mode by means of any interrupt with exit from Halt capability or a reset (see Section 9.4 HALT MODE).

– When entering AWUFH 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 AWUFH mode, the main oscillator is turned off causing all internal processing to be stopped, including the operation of the on-chip peripherals. None of the peripherals are clocked except those which get their clock supply from another clock generator (such as an external or auxiliary oscillator like the AWU oscillator).

– The compatibility of Watchdog operation with AWUFH mode is configured by the WDGHALT option bit in the option byte. Depending on this setting, the HALT instruction when executed while the Watchdog system is enabled, can generate a Watchdog RESET.

Figure 30. AWUF Halt Timing Diagram



I/O PORTS (Cont'd)

Analog alternate function

Configure the I/O as floating input to use an ADC input. The analog multiplexer (controlled by the ADC registers) switches the analog voltage present on the selected pin to the common analog rail, connected to the ADC input.

Analog Recommendations

Do not change the voltage level or loading on any I/O while conversion is in progress. Do not have clocking pins located close to a selected analog pin.

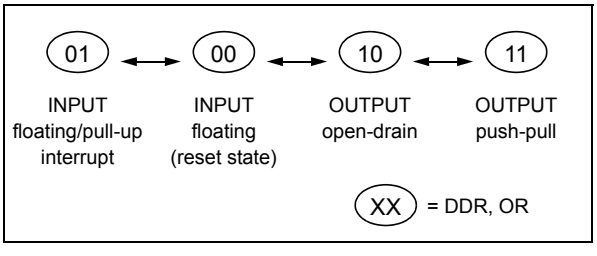
WARNING: The analog input voltage level must be within the limits stated in the absolute maximum ratings.

10.3 I/O PORT IMPLEMENTATION

The hardware implementation on each I/O port depends on the settings in the DDR and OR registers and specific I/O port features such as ADC input or open drain.

Switching these I/O ports from one state to another should be done in a sequence that prevents unwanted side effects. Recommended safe transitions are illustrated in Figure 33. Other transitions are potentially risky and should be avoided, since they may present unwanted side-effects such as spurious interrupt generation.

Figure 33. Interrupt I/O Port State Transitions



10.4 UNUSED I/O PINS

Unused I/O pins must be connected to fixed voltage levels. Refer to Section 13.8.

10.5 LOW POWER MODES

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

10.6 INTERRUPTS

The external interrupt event generates an interrupt if the corresponding configuration is selected with DDR and OR registers and if the I bit in the CC register is cleared (RIM instruction).

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
External interrupt on selected external event	-	DDRx ORx	Yes	Yes

Related Documentation

AN 970: SPI Communication between ST7 and EEPROM

AN1045: S/W implementation of I2C bus master

AN1048: Software LCD driver

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

11.2.3.3 Break Function

The break function can be used to perform an emergency shutdown of the application being driven by the PWM signals.

The break function is activated by the external BREAK pin or internal comparator output. This can be selected by using the BRSEL bit in BREAKCR Register. In order to use the break function it must be previously enabled by software setting the BPEN bit in the BREAKCR register.

The Break active level can be programmed by the BREDGE bit in the BREAKCR register. When an active level is detected on the BREAK pin, the BA bit is set and the break function is activated. In this case, the PWM signals are forced to BREAK value if respective OEx bit is set in PWMCR register.

Software can set the BA bit to activate the break function without using the BREAK pin. The BREN1 and BREN2 bits in the BREAKEN Register are used to enable the break activation on the 2 counters respectively. In Dual Timer Mode, the break for PWM2 and PWM3 is enabled by the BREN2 bit. In Single Timer Mode, the BREN1 bit enables the break for all PWM channels.

When a break function is activated (BA bit =1 and BREN1/BREN2 =1):

– The break pattern (PWM[3:0] bits in the BREAKCR) is forced directly on the PWMx output pins if respective OEx is set. (after the inverter).

– The 12-bit PWM counter CNTR1 is put to its reset value, i.e. 00h (if BREN1 = 1).

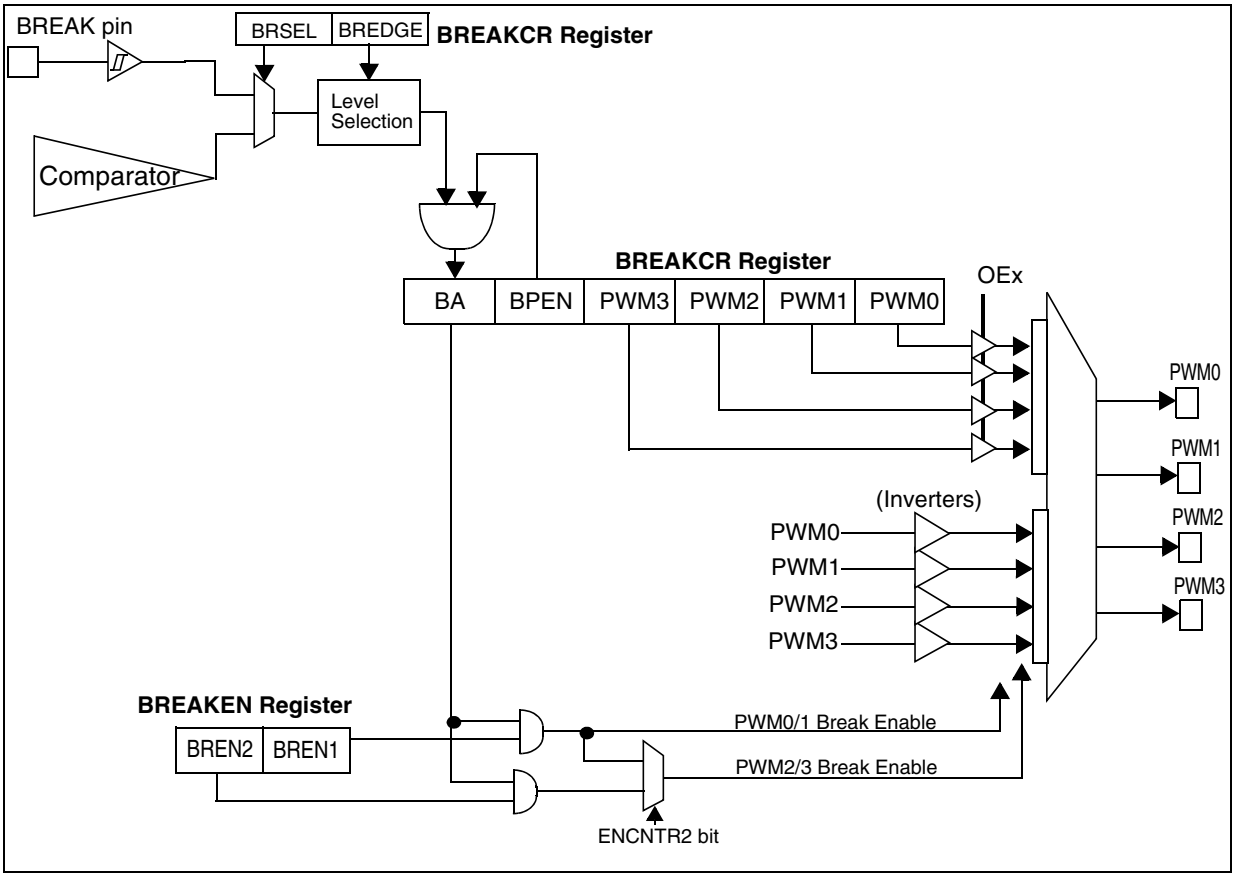
– The 12-bit PWM counter CNTR2 is put to its reset value,i.e. 00h (if BREN2 = 1).

– ATR1, ATR2, Preload and Active DCRx are put to their reset values.

– Counters stop counting.

When the break function is deactivated after applying the break (BA bit goes from 1 to 0 by software), Timer takes the control of PWM ports.

Figure 41. Block Diagram of Break Function



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

Bits 11:0 = **ICR[11:0]** *Input Capture Data*.
This is a 12-bit register which is readable by software and cleared by hardware after a reset. The ATICR register contains captured the value of the 12-bit CNTR1 register when a rising or falling edge occurs on the ATIC or LTIC pin (depending on

ICS). Capture will only be performed when the ICF flag is cleared.

BREAK ENABLE REGISTER (BREAKEN)

Read/Write
Reset Value: 0000 0011 (03h)

7							0
0	0	0	0	0	0	BREN2	BREN1

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

Bit 1 = **BREN2** *Break Enable for Counter 2*
This bit is read/write by software. It enables the break functionality for Counter2 if BA bit is set in BREAKCR. It controls PWM2/3 if ENCNTR2 bit is set.
0: No Break applied for CNTR2
1: Break applied for CNTR2

Bit 0 = **BREN1** *Break Enable for Counter 1*
This bit is read/write by software. It enables the break functionality for Counter1. If BA bit is set, it controls PWM0/1 by default, and controls PWM2/3 also if ENCNTR2 bit is reset.
0: No Break applied for CNTR1
1: Break applied for CNTR1

TIMER CONTROL REGISTER2 (ATCSR2)
Read/Write
Reset Value: 0000 0011 (03h)

7							0
FORCE 2	FORCE 1	ICS	OVFIE2	OVF2	ENCNT R2	TRAN2	TRAN1

Bit 7 = **FORCE2** *Force Counter 2 Overflow*
This bit is read/set by software. When set, it loads FFFh in the CNTR2 register. It is reset by hardware one CPU clock cycle after counter 2 overflow has occurred.

0 : No effect on CNTR2
1 : Loads FFFh in CNTR2

Note: This bit must not be reset by software

Bit 6 = **FORCE1** *Force Counter 1 Overflow*
This bit is read/set by software. When set, it loads FFFh in CNTR1 register. It is reset by hardware one CPU clock cycle after counter 1 overflow has occurred.

0 : No effect on CNTR1
1 : Loads FFFh in CNTR1

Note: This bit must not be reset by software

Bit 5 = **ICS** *Input Capture Shorted*
This bit is read/write by software. It allows the AT-timer CNTR1 to use the LTIC pin for long input capture.
0 : ATIC for CNTR1 input capture
1 : LTIC for CNTR1 input capture

Bit 4 = **OVFIE2** *Overflow interrupt 2 enable*
This bit is read/write by software and controls the overflow interrupt of counter2.
0: Overflow interrupt disabled.
1: Overflow interrupt enabled.

Bit 3 = **OVF2** *Overflow Flag.*
This bit is set by hardware and cleared by software by reading the ATCSR2 register. It indicates the transition of the counter2 from FFFh to ATR2 value.
0: No counter overflow occurred
1: Counter overflow occurred

Bit 2 = **ENCNTR2** *Enable counter2* for PWM2/3

This bit is read/write by software and switches the PWM2/3 operation to the CNTR2 counter. If this bit is set, PWM2/3 will be generated using CNTR2.
0: PWM2/3 is generated using CNTR1.
1: PWM2/3 is generated using CNTR2.

Note: Counter 2 gets frozen when the ENCNTR2 bit is reset. When ENCNTR2 is set again, the counter will restart from the last value.

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

Bit 1= **TRAN2** *Transfer enable2*
This bit is read/write by software, cleared by hardware after each completed transfer and set by hardware after reset. It controls the transfers on CNTR2.

It allows the value of the Preload DCRx registers to be transferred to the Active DCRx registers after the next overflow event.

The OPx bits are transferred to the shadow OPx bits in the same way.

Notes:

1. DCR2/3 transfer will be controlled using this bit if ENCNTR2 bit is set.
2. This bit must not be reset by software

Bit 0 = **TRAN1** *Transfer enable* 1
This bit is read/write by software, cleared by hardware after each completed transfer and set by hardware after reset. It controls the transfers on CNTR1. It allows the value of the Preload DCRx registers to be transferred to the Active DCRx registers after the next overflow event.

The OPx bits are transferred to the shadow OPx bits in the same way.

Notes:

1. DCR0,1 transfers are always controlled using this bit.

2. DCR2/3 transfer will be controlled using this bit if ENCNTR2 is reset.

3.This bit must not be reset by software

AUTORELOAD REGISTER2 (ATR2H)
Read / Write
Reset Value: 0000 0000 (00h)

15							8
0	0	0	0	ATR11	ATR10	ATR9	ATR8

AUTORELOAD REGISTER (ATR2L)
Read / Write
Reset Value: 0000 0000 (00h)

7							0
ATR7	ATR6	ATR5	ATR4	ATR3	ATR2	ATR1	ATR0

Bits 11:0 = **ATR2[11:0]** *Autoreload Register 2.*
This is a 12-bit register which is written by software. The ATR2 register value is automatically loaded into the upcounter CNTR2 when an overflow of CNTR2 occurs. The register value is used to set the PWM2/PWM3 frequency when ENCNTR2 is set.

DEAD TIME GENERATOR REGISTER (DTGR)

Read/Write
Reset Value: 0000 0000 (00h)

7							0
DTE	DT6	DT5	DT4	DT3	DT2	DT1	DT0

Bit 7 = **DTE** *Dead Time Enable*
This bit is read/write by software. It enables a dead time generation on PWM0/PWM1.
0: No Dead time insertion.
1: Dead time insertion enabled.

Bits 6:0 = **DT[6:0]** *Dead Time Value*

These bits are read/write by software. They define the dead time inserted between PWM0/PWM1. Dead time is calculated as follows:

Dead Time = DT[6:0] x Tcounter1

Note:

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

10-BIT A/D CONVERTER (ADC) (Cont'd)

11.5.3.2 Input Voltage Amplifier

The input voltage can be amplified by a factor of 8 by enabling the AMPSEL bit in the ADCDRL register.

When the amplifier is enabled, the input range is 0V to $V_{DD}/8$.

For example, if V_{DD} = 5V, then the ADC can convert voltages in the range 0V to 430mV with an ideal resolution of 0.6mV (equivalent to 13-bit resolution with reference to a V_{SS} to V_{DD} range).

For more details, refer to the Electrical characteristics section.

Note: The amplifier is switched on by the ADON bit in the ADCCSR register, so no additional start-up time is required when the amplifier is selected by the AMPSEL bit.

11.5.3.3 Digital A/D Conversion Result

The conversion is monotonic, meaning that the result never decreases if the analog input does not and never increases if the analog input does not.

If the input voltage (V_{AIN}) is greater than V_{DDA} (high-level voltage reference) then the conversion result is FFh in the ADCDRH register and 03h in the ADCDRL register (without overflow indication).

If the input voltage (V_{AIN}) is lower than V_{SSA} (low-level voltage reference) then the conversion result in the ADCDRH and ADCDRL registers is 00 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDRH and ADCDRL registers. The accuracy of the conversion is described in the Electrical Characteristics Section.

R_{AIN} is the maximum recommended impedance for an analog input signal. If the impedance is too high, this will result in a loss of accuracy due to leakage and sampling not being completed in the alloted time.

11.5.3.4 A/D Conversion

The analog input ports must be configured as input, no pull-up, no interrupt. Refer to the «I/O ports» chapter. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the ADCCSR register:

– Select the CH[2:0] bits to assign the analog channel to convert.

ADC Conversion mode

In the ADCCSR register:

Set the ADON bit to enable the A/D converter and to start the conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete:

– The EOC bit is set by hardware.
– The result is in the ADCDR registers.

A read to the ADCDRH or a write to any bit of the ADCCSR register resets the EOC bit.

To read the 10 bits, perform the following steps:

1. Poll the EOC bit
2. Read ADCDRL
3. Read ADCDRH. This clears EOC automatically.

To read only 8 bits, perform the following steps:

1. Poll EOC bit
2. Read ADCDRH. This clears EOC automatically.

11.5.3.5 Changing the conversion channel

The application can change channels during conversion.

When software modifies the CH[2:0] bits in the ADCCSR register, the current conversion is stopped, the EOC bit is cleared, and the A/D converter starts converting the newly selected channel.

11.5.4 Low Power Modes

Note: The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed and between single shot conversions.

Mode	Description
WAIT	No effect on A/D Converter
HALT	A/D Converter disabled. After wakeup from Halt mode, the A/D Converter requires a stabilization time t_{STAB} (see Electrical Characteristics) before accurate conversions can be performed.

11.5.5 Interrupts

None.

ANALOG COMPARATOR (Cont'd)

Bit 4 = **CMPIF** *Comparator Interrupt Flag*

This bit is set by hardware when interrupt is generated at the rising edge (CINV = 0) or falling edge (CINV = 1) of comparator output. This bit is cleared by reading the CMPCR register. Writing to this bit does not change the value.
0 : Comparator interrupt flag cleared
1 : Comparator interrupt flag set and can generate interrupt if CMPIE is set.

Bit 3 : **CMPIE** *Comparator Interrupt Enable*

This bit is set or reset by software and cleared by hardware reset. This bit enables or disables the interrupt generation depending on interrupt flag
0: Interrupt not generated
1: Interrupt generated if interrupt flag is set

Note:

This bit should be set to enable interrupt only after the comparator has been switched ON, i.e. when CMPON is set.

Once CMPON bit is set, it is recommended to wait the specified stabilization time before setting CMPIE bit in order to avoid a spurious interrupt (see section 13.12 on page 143).

Bit 2 : **CMP** *Comparator Output*

This bit is set or reset by software and cleared by hardware reset. It stores the value of comparator output.

Bit 1 = **COUT** *Comparator Output Enable on Port*
This bit is set or cleared by software. When this bit is set, the comparator output is available on PA7 port.
0 : Comparator output not connected to PA7
1 : Comparator output connected to PA7

Bit 0 : **CMPON** *Comparator ON/OFF*
This bit is set or cleared by software and reset by hardware reset. This bit is used to switch ON/OFF the comparator, internal voltage reference and current bias which provides 4µA current to both.
0: Comparator, Internal Voltage Reference, Bias OFF (in power-down state).
1: Comparator, Internal Voltage Reference, Bias ON

Note: For the comparator interrupt generation, it takes 250ns delay from comparator output change to rising or falling edge of interrupt generated.

Table 21. Analog Comparator Register Map and Reset Values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
002Ch	**VREFCR**	VCEXT	VCBGR	VR3	VR2	VR1	VR0	-	-
	Reset Value	0	0	0	0	0	0	0	0
002Dh	**CMPCR**	CHYST	-	CINV	CMPIF	CMPIE	CMP	COUT	CMPON
	Reset value	1	0	0	0	0	0	0	0

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.

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

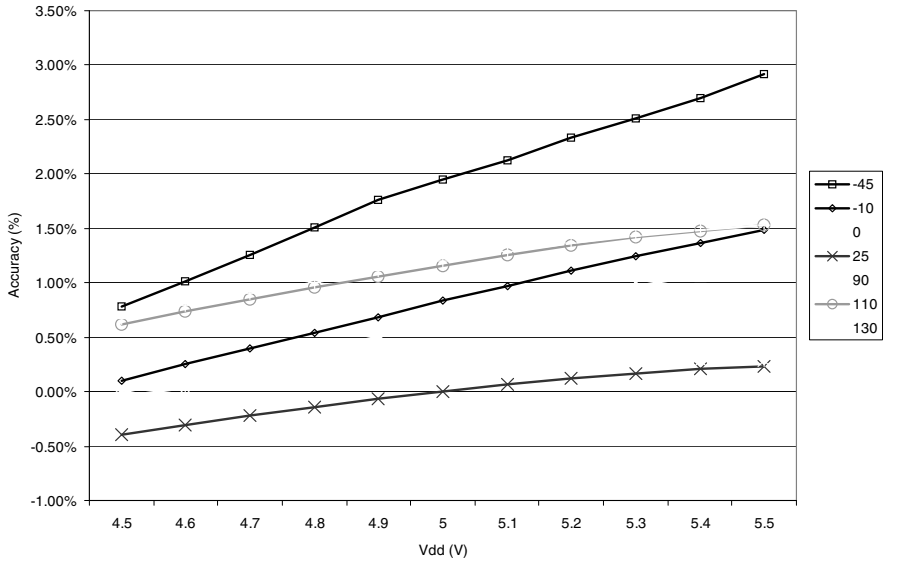
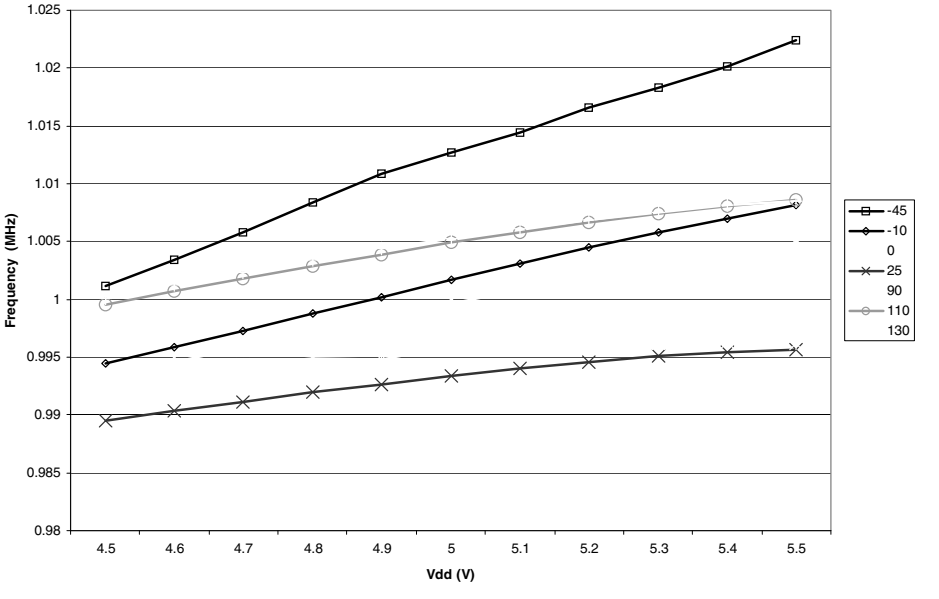


Figure 67. Typical RCCR0 vs V_{DD} and Temperature



13.4.2 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 93. Typical V_{DD}-V_{OH} at V_{DD}=2.7V

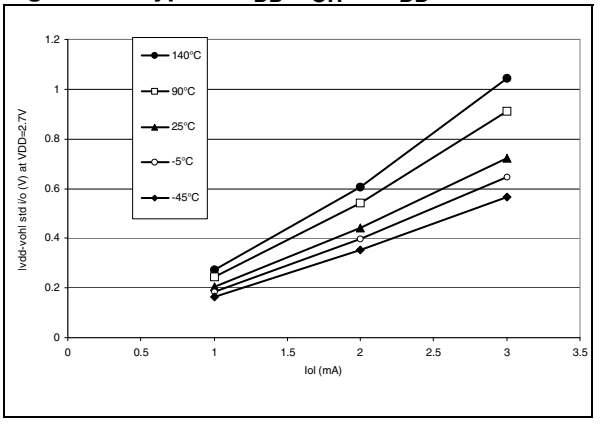


Figure 94. Typical V_{DD}-V_{OH} at V_{DD}=3.3V

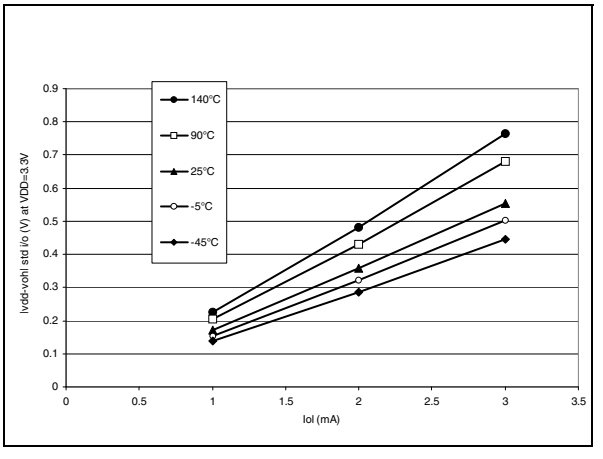


Figure 95. Typical V_{DD}-V_{OH} at V_{DD}=5V

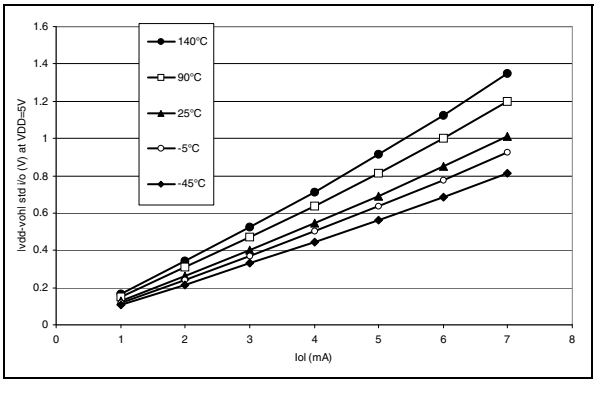


Figure 96. Typical V_{DD}-V_{OH} at V_{DD}=2.7V (HS)

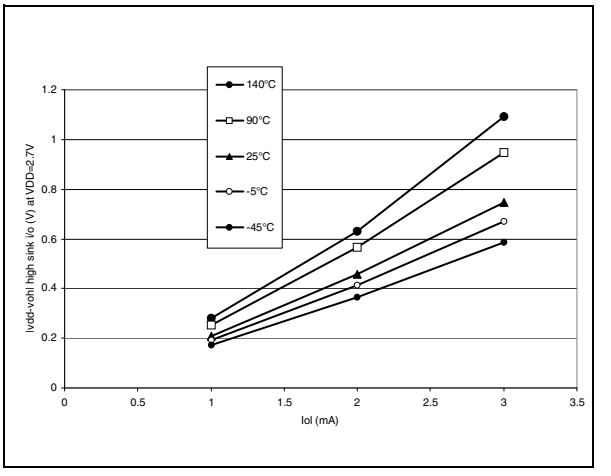


Figure 97. Typical V_{DD}-V_{OH} at V_{DD}=3.3V (HS)

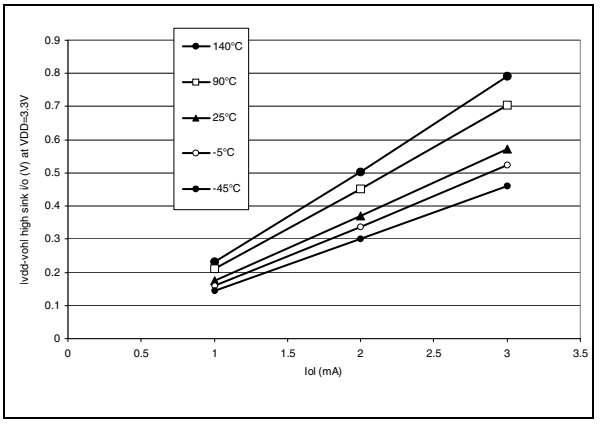
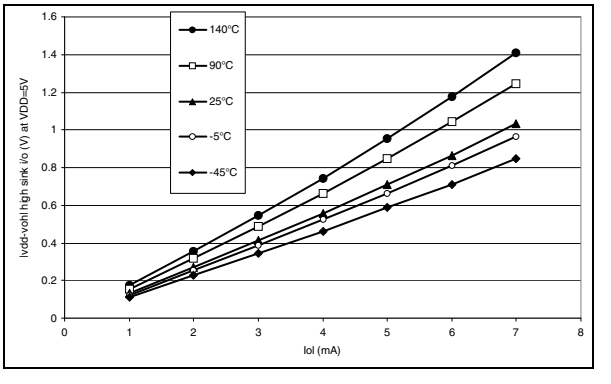


Figure 98. Typical V_{DD}-V_{OH} at V_{DD}=5V (HS)



I/O PORT PIN CHARACTERISTICS (Cont'd)

Figure 99. Typical V_{DD}-V_{OH} at V_{DD}=2.7V (Port C)

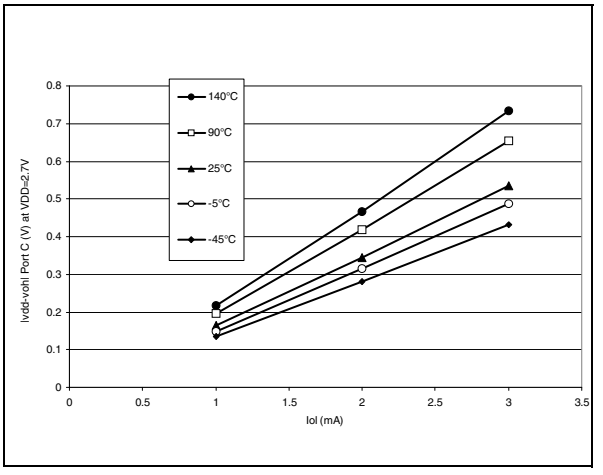


Figure 100. Typical V_{DD}-V_{OH} at V_{DD}=3.3V (Port C)

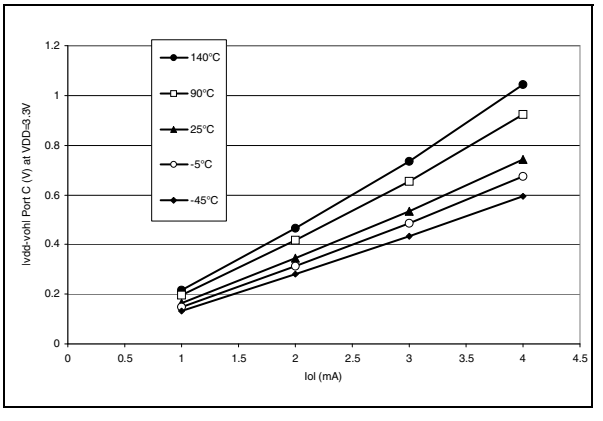


Figure 101. Typical V_{DD}-V_{OH} at V_{DD}=5V (Port C)

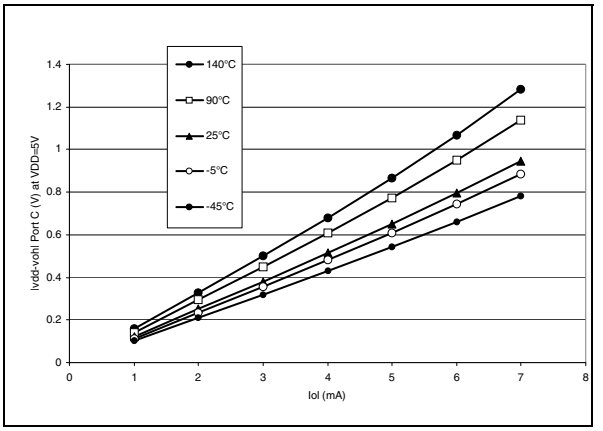


Figure 102. Typical V_{DD}-V_{OH} vs. V_{DD} (Standard)

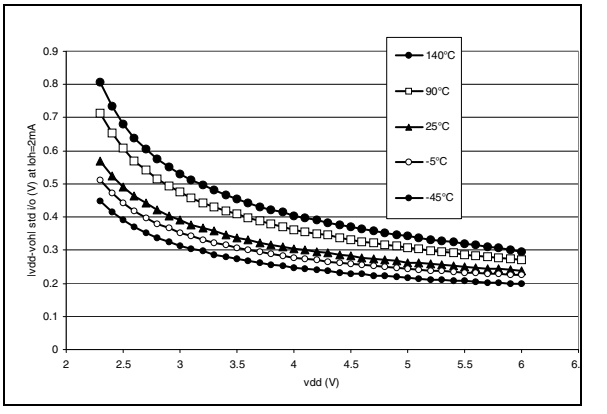


Figure 103. Typical V_{DD}-V_{OH} vs. V_{DD} (High Sink)

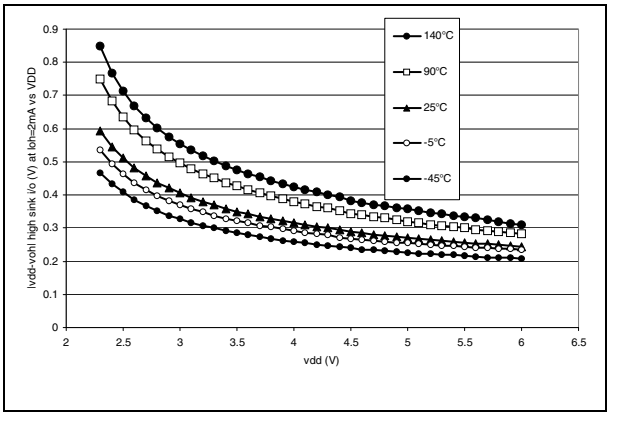
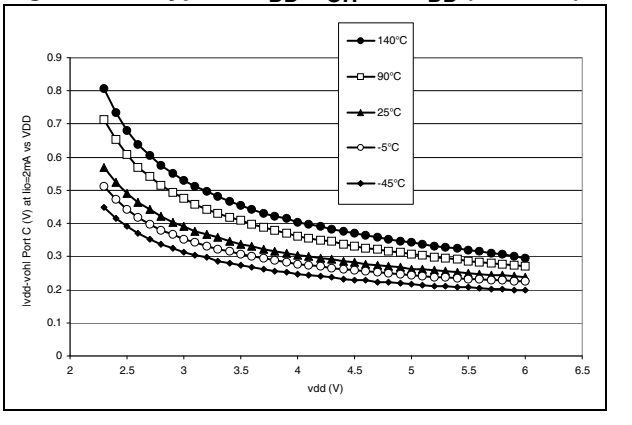


Figure 104. Typical V_{DD}-V_{OH} vs. V_{DD} (PORT C)



13.10 COMMUNICATION INTERFACE CHARACTERISTICS

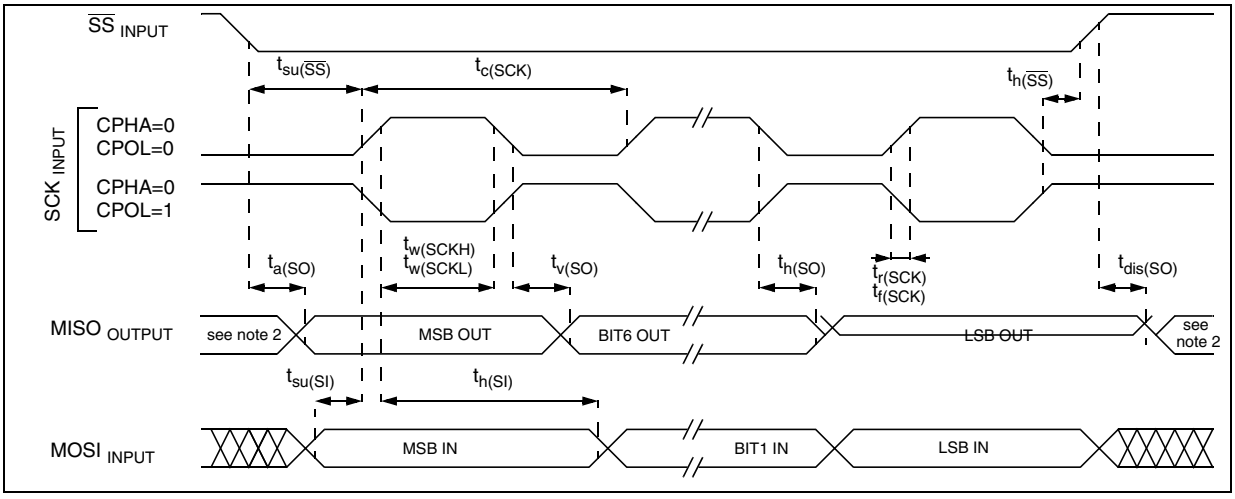
13.10.1 SPI - Serial Peripheral Interface

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

Refer to I/O port characteristics for more details on the input/output alternate function characteristics ($\overline{SS}$, SCK, MOSI, MISO).

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master f_{CPU}=8MHz	$f_{CPU}/128$ 0.0625	$f_{CPU}/4$ 2	MHz
		Slave f_{CPU}=8MHz	0	$f_{CPU}/2$ 4	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time		see I/O port pin description		
$t_{su(\overline{SS})}$ [1]	$\overline{SS}$ setup time [4]	Slave	$(4 \times T_{CPU}) + 50$		ns
$t_{h(\overline{SS})}$ [1]	$\overline{SS}$ hold time	Slave	120		
$t_{w(SCKH)}$ [1] $t_{w(SCKL)}$ [1]	SCK high and low time	Master Slave	100 90		
$t_{su(MI)}$ [1] $t_{su(SI)}$ [1]	Data input setup time	Master Slave	100 100		
$t_{h(MI)}$ [1] $t_{h(SI)}$ [1]	Data input hold time	Master Slave	100 100		
$t_{a(SO)}$ [1]	Data output access time	Slave	0	120	
$t_{dis(SO)}$ [1]	Data output disable time	Slave		240	
$t_{v(SO)}$ [1]	Data output valid time	Slave (after enable edge)		120	
$t_{h(SO)}$ [1]	Data output hold time		0		
$t_{v(MO)}$ [1]	Data output valid time	Master (after enable edge)		120	
$t_{h(MO)}$ [1]	Data output hold time		0		

Figure 107. SPI Slave Timing Diagram with CPHA=0 [3]



Notes:

1. Data based on design simulation, not tested in production.

2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.

3. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.

4. Depends on f_{CPU}. For example, if f_{CPU}=8MHz, then $T_{CPU} = 1/f_{CPU}$ =125ns and $t_{su(\overline{SS})}$=550ns

ST7LITE1xB FASTROM microcontroller option list

Customer . . . . . .
Address . . . . . .
. . . . . .
Contact . . . . . .
Phone No . . . . . .
Reference/FASTROM Code*: . . . . . .
*FASTROM code name is assigned by STMicroelectronics.
FASTROM code must be sent in .S19 format. .Hex extension cannot be processed.

Device Type/Memory Size/Package (check only one option):

FASTROM DEVICE:	2K	4K
VFQFPN20:	[ ] ST7PLIT19BF0Ux	[ ] ST7PLIT19BF1Ux
SO20:	[ ] ST7PLIT19BF0Mx	[ ] ST7PLIT19BF1Mx
PDIP20:	[ ] ST7PLIT19BF0Bx	[ ] ST7PLIT19BF1Bx
SO16:	[ ] ST7PLIT19BY0Mx	[ ] ST7PLIT19BY1Mx
PDIP16:	[ ] ST7PLIT19BY0Bx	[ ] ST7PLIT19BY1Bx

Warning: Addresses DEE0h, DEE1h, DEE2h and DEE3h are reserved areas for ST to program RCCR0 and RCCR1 (see section 7.1 on page 23).

Conditioning (check only one option, do not specify for DIP package) :

VFQFPN	[ ] Tape & Reel	[ ] Tray
SO	[ ] Tape & Reel	[ ] Tube

Special marking: [ ] No [ ] Yes "_ _ _ _ _ _ _ _ "
Authorized characters are letters, digits, '.', '-', '/' and spaces only.
Maximum character count: 8 char. max _ _ _ _ _ _ _ _

Temperature range: [ ] -40°C to +85°C [ ] -40°C to +125°C

Watchdog selection (WDG_SW): [ ] Software activation [ ] Hardware activation

Watchdog reset on Halt (WDG_HALT): [ ] Reset [ ] No Reset

LVD reset (LVD): [ ] Disabled [ ] Enabled
[ ] Highest threshold
[ ] Medium threshold
[ ] Lowest threshold

Sector 0 size (SEC): [ ] 0.5K [ ] 1K [ ] 2K [ ] 4K

Readout protection (FMP_R): [ ] Disabled [ ] Enabled

Flash write protection (FMP_W): [ ] Disabled [ ] Enabled

RC oscillator (OSC) : [ ] Disabled [ ] Enabled

Clock source selection (CKSEL):
(if OSC disabled) [ ] External crystal / ceramic resonator:
[ ] External Clock on PB4
[ ] External Clock on PC0

PLL (PLLOFF): [ ] Disabled [ ] Enabled

PLL factor (PLLx4x8): [ ] PLLx4 [ ] PLLx8

PLL32 (PLL32OFF): [ ] Disabled [ ] Enabled

Comments : . . . . . .
Supply operating range in the application : . . . . . .
Notes . . . . . .
Date : . . . . . .
Signature : . . . . . .

Important note : Not all configurations are available. See Table 27 on page 150 for authorized option byte combinations.