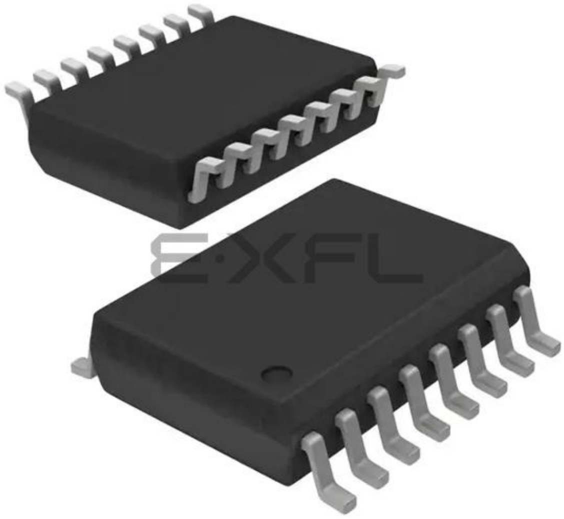


Welcome to E-XFL.COM

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	2KB (2K 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/st7flit15by0m6

Address	Block	Register Label	Register Name	Reset Status	Remarks
0002Fh	FLASH	FCSR	Flash Control/Status Register	00h	R/W
00030h	EEPROM	EECSR	Data EEPROM Control/Status Register	00h	R/W
0031h 0032h 0033h	SPI	SPIDR SPICR SPICSR	SPI Data I/O Register SPI Control Register SPI Control Status Register	xxh 0xh 00h	R/W R/W R/W
0034h 0035h 0036h	ADC	ADCCSR ADCDRH ADCDRL	A/D Control Status Register A/D Data Register High A/D Amplifier Control/Data Low Register	00h xxh 0xh	R/W Read Only R/W
0037h	ITC	EICR	External Interrupt Control Register	00h	R/W
0038h	MCC	MCCSR	Main Clock Control/Status Register	00h	R/W
0039h 003Ah	Clock and Reset	RCCR SICSR	RC oscillator Control Register System Integrity Control/Status Register	FFh 0110 0xx0b	R/W R/W
003Bh	PLL clock select	PLLTST	PLL test register	00h	R/W
003Ch	ITC	EISR	External Interrupt Selection Register	0Ch	R/W
003Dh to 0048h	Reserved area (12 bytes)				
0049h 004Ah	AWU	AWUPR AWUCSR	AWU Prescaler Register AWU Control/Status Register	FFh 00h	R/W R/W
004Bh 004Ch 004Dh 004Eh 004Fh 0050h 0051h	DM3)	DMCR DMSR DMBK1H DMBK1L DMBK2H DMBK2L DMCR2	DM Control Register DM Status Register DM Breakpoint Register 1 High DM Breakpoint Register 1 Low DM Breakpoint Register 2 High DM Breakpoint Register 2 Low DM Control Register 2	00h 00h 00h 00h 00h 00h 00h	R/W R/W R/W R/W R/W R/W R/W
0052h to 007Fh	Reserved area (46 bytes)				

Legend: x=undefined, R/W=read/write

Notes:

1. The contents of the I/O port DR registers are readable only in output configuration. In input configuration, the values of the I/O pins are returned instead of the DR register contents.

2. The bits associated with unavailable pins must always keep their reset value.

3. For a description of the Debug Module registers, see ICC protocol reference manual.

CPU REGISTERS (Cont'd)

STACK POINTER (SP)

Read/Write

Reset Value: 01FFh

15							8
0	0	0	0	0	0	0	1

7							0
1	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see Figure 12).

Since the stack is 128 bytes deep, the 9 most significant bits are forced by hardware. Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP6 to SP0 bits are set) which is the stack higher address.

The least significant byte of the Stack Pointer (called S) can be directly accessed by a LD instruction.

Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

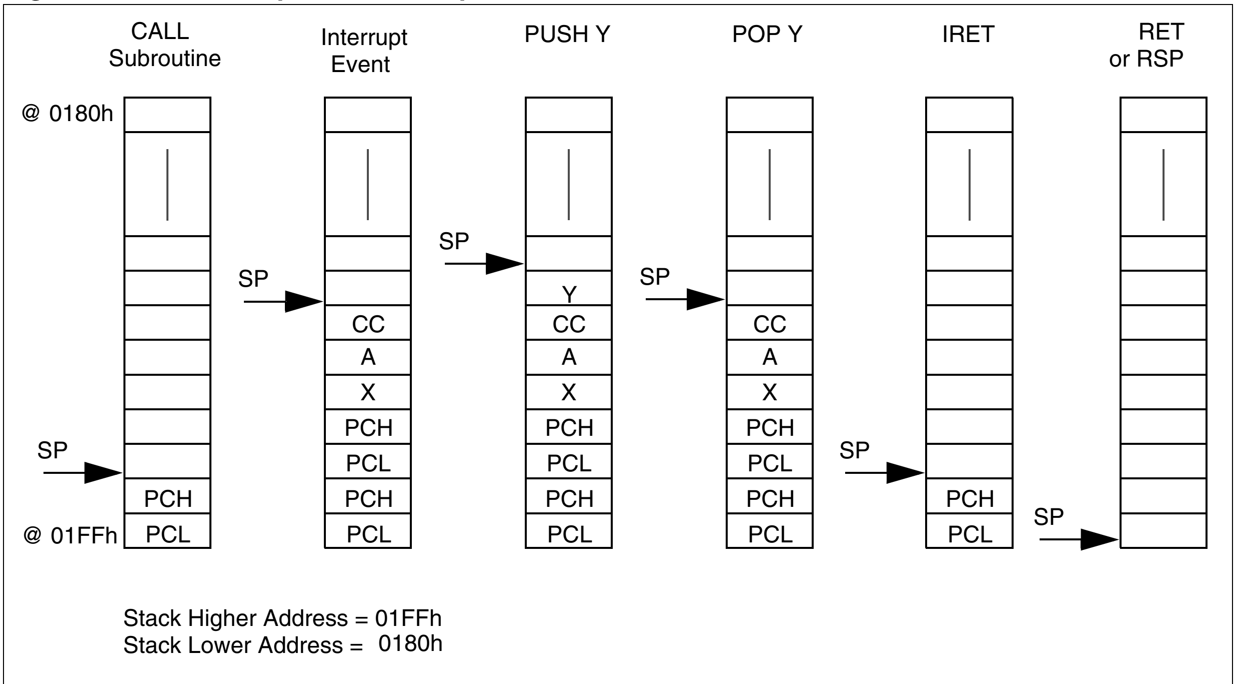
The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in Figure 12.

– When an interrupt is received, the SP is decremented and the context is pushed on the stack.

– On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.

Figure 12. Stack Manipulation Example



7.5 RESET SEQUENCE MANAGER (RSM)

7.5.1 Introduction

The reset sequence manager includes three RESET sources as shown in Figure 16:

- External $\overline{\text{RESET}}$ source pulse
- Internal LVD RESET (Low Voltage Detection)
- Internal WATCHDOG RESET

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

These sources act on the $\overline{\text{RESET}}$ pin and it is always kept low during the delay phase.

The RESET service routine vector is fixed at addresses FFFEh-FFFFh in the ST7 memory map.

The basic RESET sequence consists of 3 phases as shown in Figure 15:

- Active Phase depending on the RESET source
- 256 or 4096 CPU clock cycle delay (see table below)
- RESET vector fetch

Caution: When the ST7 is unprogrammed or fully erased, the Flash is blank and the RESET vector is not programmed. For this reason, it is recommended to keep the RESET pin in low state until programming mode is entered, in order to avoid unwanted behavior.

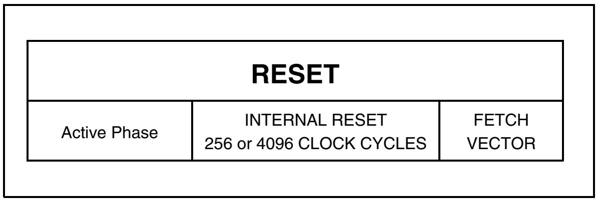
The 256 or 4096 CPU clock cycle delay allows the oscillator to stabilise and ensures that recovery has taken place from the Reset state. The shorter or longer clock cycle delay is automatically selected depending on the clock source chosen by option byte:

The RESET vector fetch phase duration is 2 clock cycles.

Clock Source	CPU clock cycle delay
Internal RC Oscillator	256
External clock (connected to CLKIN pin)	256
External Crystal/Ceramic Oscillator (connected to OSC1/OSC2 pins)	4096

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

Figure 15. RESET Sequence Phases



7.5.2 Asynchronous External $\overline{\text{RESET}}$ pin

The $\overline{\text{RESET}}$ pin is both an input and an open-drain output with integrated R_{ON} weak pull-up resistor. This pull-up has no fixed value but varies in accordance with the input voltage. It can be pulled low by external circuitry to reset the device. See Electrical Characteristic section for more details.

A RESET signal originating from an external source must have a duration of at least $t_{h(RSTL)in}$ in order to be recognized (see Figure 17). This detection is asynchronous and therefore the MCU can enter reset state even in HALT mode.

7.6 SYSTEM INTEGRITY MANAGEMENT (SI)

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

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

7.6.1 Low Voltage Detector (LVD)

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

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

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

– $V_{IT+(LVD)}$when V_{DD} is rising

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

The LVD function is illustrated in Figure 18.

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

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

– under full software control

– in static safe reset

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

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

Notes:

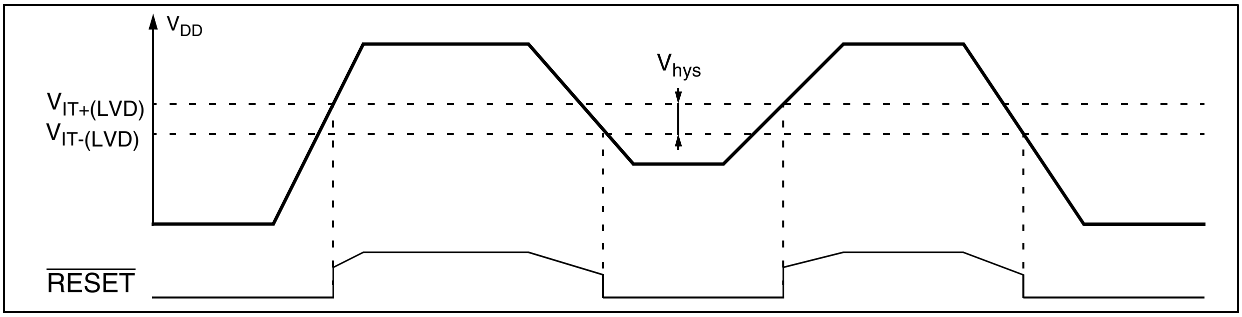
The LVD allows the device to be used without any external RESET circuitry.

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

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

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

Figure 18. Low Voltage Detector vs Reset



9 POWER SAVING MODES

9.1 INTRODUCTION

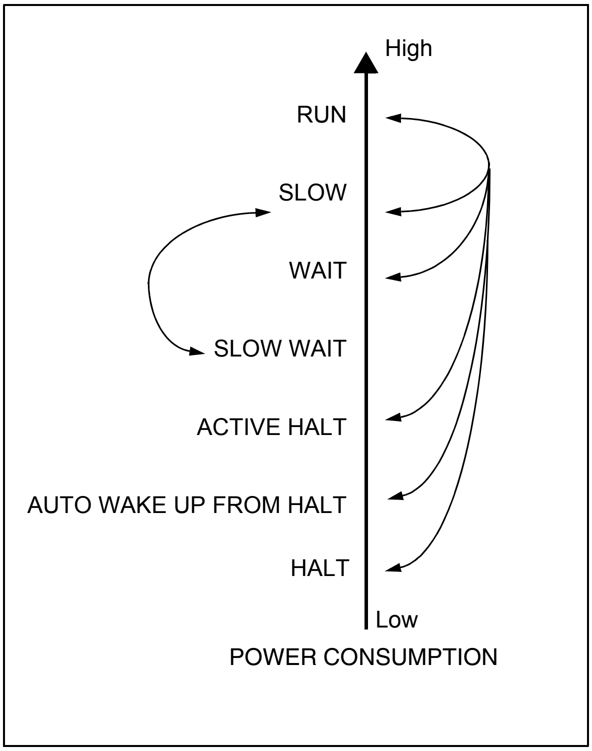
To give a large measure of flexibility to the application in terms of power consumption, five main power saving modes are implemented in the ST7 (see Figure 22):

- Slow
- Wait (and Slow-Wait)
- Active Halt
- Auto Wake up From Halt (AWUFH)
- Halt

After a RESET the normal operating mode is selected by default (RUN mode). This mode drives the device (CPU and embedded peripherals) by means of a master clock which is based on the main oscillator frequency divided or multiplied by 2 (f_{OSC2}).

From RUN mode, the different power saving modes may be selected by setting the relevant register bits or by calling the specific ST7 software instruction whose action depends on the oscillator status.

Figure 22. Power Saving Mode Transitions



9.2 SLOW MODE

This mode has two targets:

– To reduce power consumption by decreasing the internal clock in the device,

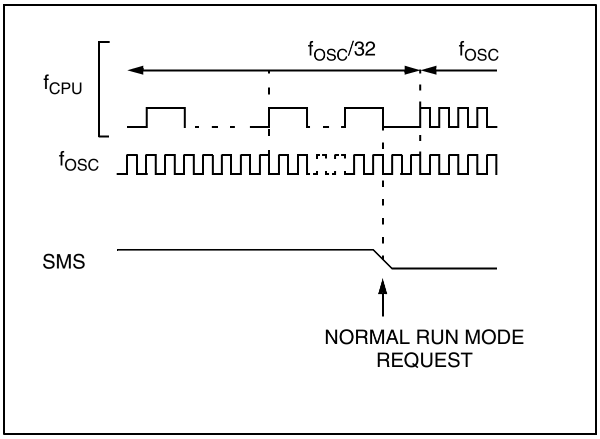
– To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

SLOW mode is controlled by the SMS bit in the MCCSR register which enables or disables Slow mode.

In this mode, the oscillator frequency is divided by 32. The CPU and peripherals are clocked at this lower frequency.

Note: SLOW-WAIT mode is activated when entering WAIT mode while the device is already in SLOW mode.

Figure 23. SLOW Mode Clock Transition



– 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 open-drain. 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

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.

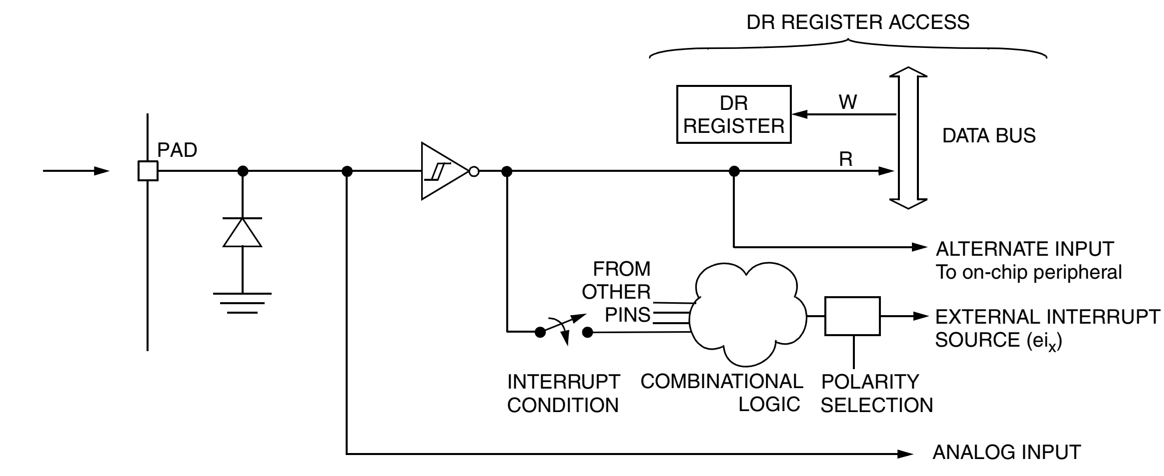
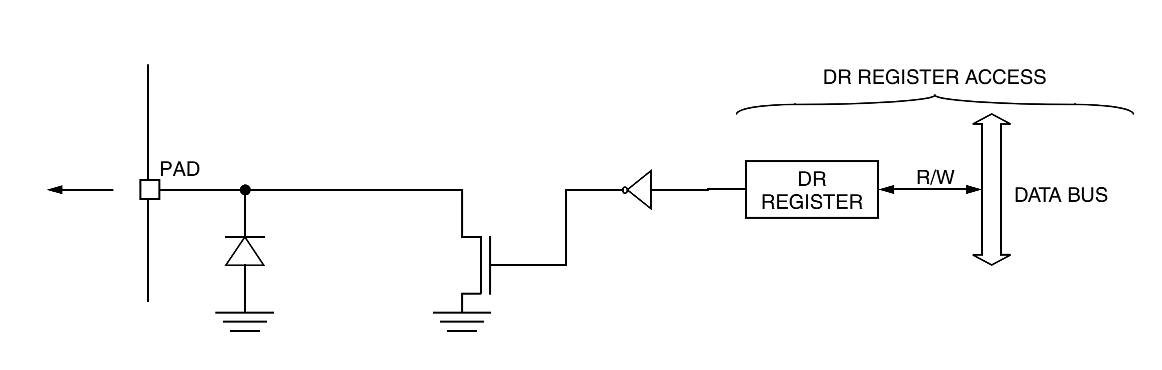
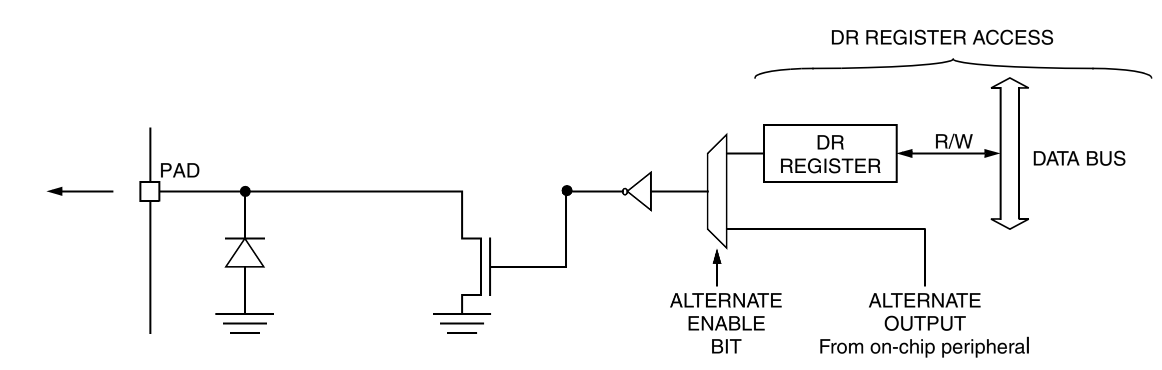
Configuring an I/O as floating enables alternate function input. It is not recommended to configure an I/O as pull-up as this will increase current consumption. Before using an I/O as an alternate input, configure it without interrupt. Otherwise spurious interrupts can occur.

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.

I/O PORTS (Cont'd)

Table 9. I/O Configurations

<table>
<tr><th></th><th>Hardware Configuration</th></tr>
<tr><td>INPUT [1)]</td><td>
</td></tr>
<tr><td>OPEN-DRAIN OUTPUT [2)]</td><td>
</td></tr>
<tr><td>PUSH-PULL OUTPUT [2)]</td><td>
</td></tr>
</table>

Notes:

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

11 ON-CHIP PERIPHERALS

11.1 WATCHDOG TIMER (WDG)

11.1.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the T6 bit becomes cleared.

11.1.2 Main Features

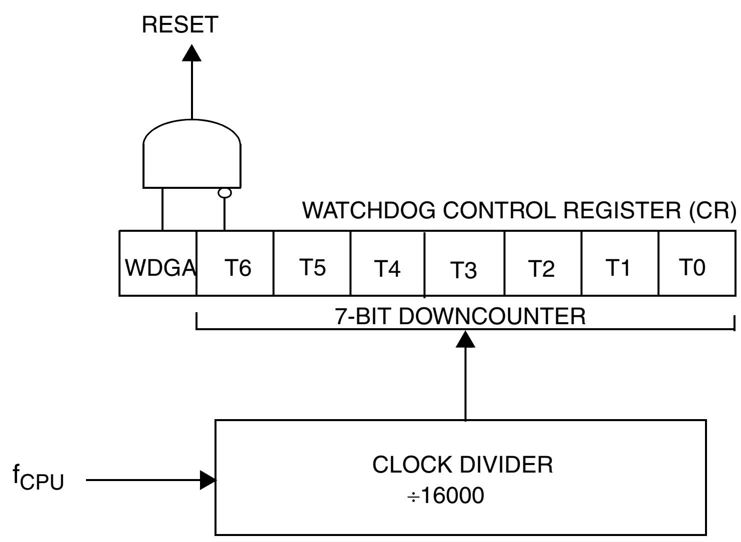
- Programmable free-running downcounter (64 increments of 16000 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) when the T6 bit reaches zero
- Optional reset on HALT instruction (configurable by option byte)
- Hardware Watchdog selectable by option byte

11.1.3 Functional Description

The counter value stored in the CR register (bits T[6:0]), is decremented every 16000 machine cycles, and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T[6:0]) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for typically 30µs.

Figure 34. Watchdog Block Diagram



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

11.2.3 Functional Description

11.2.3.1 PWM Mode

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

PWM Frequency

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

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

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

Following the above formula,

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

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

Notes:

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

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

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

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

Duty Cycle

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

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

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

The maximum available resolution for the PWMx duty cycle is:

$$\text{Resolution} = 1 / (4096 - ATR)$$

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

At reset, the counter starts counting from 0.

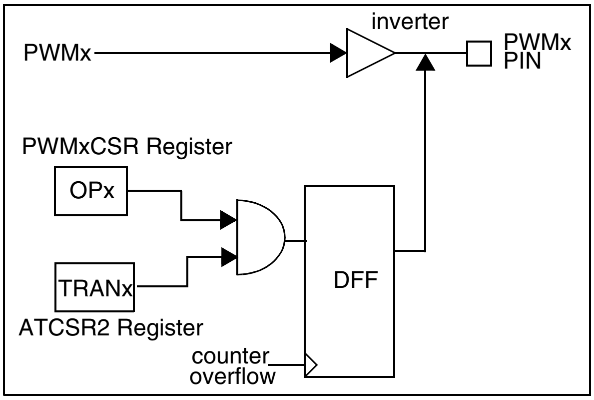
When a upcounter overflow occurs (OVF event), the preloaded Duty cycle values are transferred to the active Duty Cycle registers and the PWMx signals are set to a high level. When the upcounter matches the active DCRx value the PWMx signals are set to a low level. To obtain a signal on a PWMx pin, the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

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

Polarity Inversion

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

Figure 37. PWM Polarity Inversion



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

Output Control

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

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

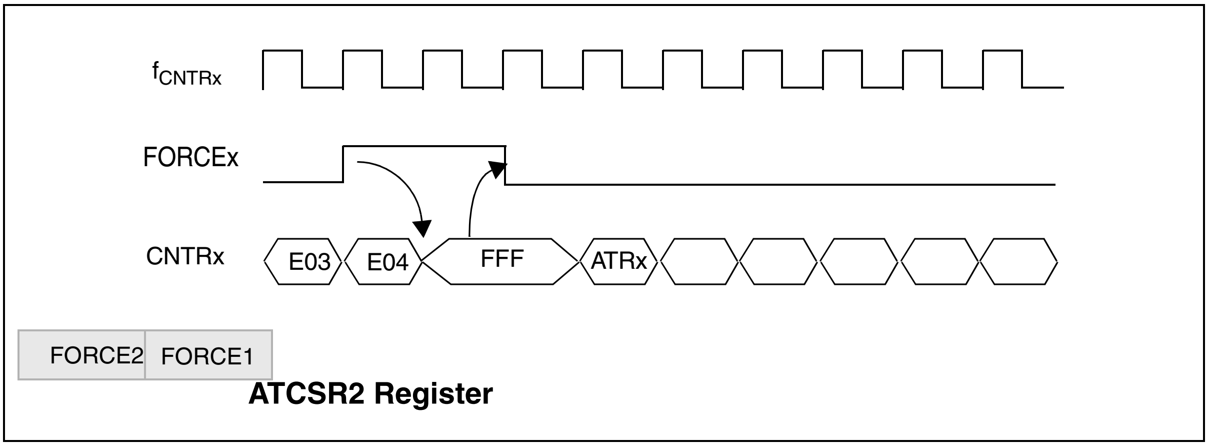
11.2.3.7 Force Update

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



SERIAL PERIPHERAL INTERFACE (cont'd)

11.4.6 Low Power Modes

Mode	Description
WAIT	No effect on SPI. SPI interrupt events cause the device to exit from WAIT mode.
HALT	SPI registers are frozen. In HALT mode, the SPI is inactive. SPI operation resumes when the device is woken up by an interrupt with "exit from HALT mode" capability. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetching). If several data are received before the wake-up event, then an overrun error is generated. This error can be detected after the fetch of the interrupt routine that woke up the Device.

11.4.6.1 Using the SPI to wake up the device from Halt mode

In slave configuration, the SPI is able to wake up the device from HALT mode through a SPIF interrupt. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetch). If multiple data transfers have been performed before software clears the SPIF bit, then the OVR bit is set by hardware.

Note: When waking up from HALT mode, if the SPI remains in Slave mode, it is recommended to perform an extra communications cycle to bring the SPI from HALT mode state to normal state. If the SPI exits from Slave mode, it returns to normal state immediately.

Caution: The SPI can wake up the device from HALT mode only if the Slave Select signal (external $\overline{SS}$ pin or the SSI bit in the SPICSR register) is low when the device enters HALT mode. So, if Slave selection is configured as external (see Section 0.1.3.2), make sure the master drives a low level on the $\overline{SS}$ pin when the slave enters HALT mode.

11.4.7 Interrupts

<table>
<tr><th>Interrupt Event</th><th>Event Flag</th><th>Enable Control Bit</th><th>Exit from Wait</th><th>Exit from Halt</th></tr>
<tr><td>SPI End of Transfer Event</td><td>SPIF</td><td rowspan="3">SPIE</td><td rowspan="3">Yes</td><td>Yes</td></tr>
<tr><td>Master Mode Fault Event</td><td>MODF</td><td rowspan="2">No</td></tr>
<tr><td>Overrun Error</td><td>OVR</td></tr>
</table>

Note: The SPI interrupt events are connected to the same interrupt vector (see Interrupts chapter). They generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

SERIAL PERIPHERAL INTERFACE (cont'd)

SPI CONTROL/STATUS REGISTER (SPICSR)

Read/Write (some bits Read Only)
Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	OVR	MODF	-	SOD	SSM	SSI

Bit 7 = **SPIF** *Serial Peripheral Data Transfer Flag (Read only)*
This bit is set by hardware when a transfer has been completed. An interrupt is generated if SPIE = 1 in the SPICR register. It is cleared by a software sequence (an access to the SPICSR register followed by a write or a read to the SPIDR register).
0: Data transfer is in progress or the flag has been cleared.
1: Data transfer between the device and an external device has been completed.

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

Bit 6 = **WCOL** *Write Collision status (Read only)*
This bit is set by hardware when a write to the SPIDR register is done during a transmit sequence. It is cleared by a software sequence (see Figure 6).
0: No write collision occurred
1: A write collision has been detected

Bit 5 = **OVR** S*PI Overrun error (Read only)*
This bit is set by hardware when the byte currently being received in the shift register is ready to be transferred into the SPIDR register while SPIF = 1 (See Section 0.1.5.2). An interrupt is generated if SPIE = 1 in the SPICR register. The OVR bit is cleared by software reading the SPICSR register.
0: No overrun error
1: Overrun error detected

Bit 4 = **MODF** *Mode Fault flag (Read only)*
This bit is set by hardware when the $\overline{SS}$ pin is pulled low in master mode (see Section 0.1.5.1 Master Mode Fault (MODF)). An SPI interrupt can be generated if SPIE = 1 in the SPICR register. This bit is cleared by a software sequence (An access to the SPICSR register while MODF = 1 followed by a write to the SPICR register).
0: No master mode fault detected
1: A fault in master mode has been detected

Bit 3 = Reserved, must be kept cleared.

Bit 2 = **SOD** *SPI Output Disable*
This bit is set and cleared by software. When set, it disables the alternate function of the SPI output (MOSI in master mode / MISO in slave mode)
0: SPI output enabled (if SPE = 1)
1: SPI output disabled

Bit 1 = **SSM** *$\overline{SS}$ Management*
This bit is set and cleared by software. When set, it disables the alternate function of the SPI $\overline{SS}$ pin and uses the SSI bit value instead. See Section 0.1.3.2 Slave Select Management.
0: Hardware management ($\overline{SS}$ managed by external pin)
1: Software management (internal $\overline{SS}$ signal controlled by SSI bit. External $\overline{SS}$ pin free for general-purpose I/O)

Bit 0 = **SSI** *$\overline{SS}$ Internal Mode*
This bit is set and cleared by software. It acts as a 'chip select' by controlling the level of the $\overline{SS}$ slave select signal when the SSM bit is set.
0: Slave selected
1: Slave deselected

SPI DATA I/O REGISTER (SPIDR)

Read/Write
Reset Value: Undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

The SPIDR register is used to transmit and receive data on the serial bus. In a master device, a write to this register will initiate transmission/reception of another byte.

Notes: During the last clock cycle the SPIF bit is set, a copy of the received data byte in the shift register is moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read.

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

Warning: A write to the SPIDR register places data directly into the shift register for transmission.

A read to the SPIDR register returns the value located in the buffer and not the content of the shift register (see Figure 1).

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
ADC	Add with Carry	A = A + M + C	A	M	H		N	Z	C
ADD	Addition	A = A + M	A	M	H		N	Z	C
AND	Logical And	A = A . M	A	M			N	Z	
BCP	Bit compare A, Memory	tst (A . M)	A	M			N	Z	
BRES	Bit Reset	bres Byte, #3	M						
BSET	Bit Set	bset Byte, #3	M						
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	M						C
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	M						C
CALL	Call subroutine								
CALLR	Call subroutine relative								
CLR	Clear		reg, M				0	1	
CP	Arithmetic Compare	tst(Reg - M)	reg	M			N	Z	C
CPL	One Complement	A = FFH-A	reg, M				N	Z	1
DEC	Decrement	dec Y	reg, M				N	Z	
HALT	Halt					0			
IRET	Interrupt routine return	Pop CC, A, X, PC			H	I	N	Z	C
INC	Increment	inc X	reg, M				N	Z	
JP	Absolute Jump	jp [TBL.w]							
JRA	Jump relative always								
JRT	Jump relative								
JRF	Never jump	jrf *							
JRIH	Jump if ext. interrupt = 1								
JRIL	Jump if ext. interrupt = 0								
JRH	Jump if H = 1	H = 1 ?							
JRNH	Jump if H = 0	H = 0 ?							
JRM	Jump if I = 1	I = 1 ?							
JRNM	Jump if I = 0	I = 0 ?							
JRMI	Jump if N = 1 (minus)	N = 1 ?							
JRPL	Jump if N = 0 (plus)	N = 0 ?							
JREQ	Jump if Z = 1 (equal)	Z = 1 ?							
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?							
JRC	Jump if C = 1	C = 1 ?							
JRNC	Jump if C = 0	C = 0 ?							
JRULT	Jump if C = 1	Unsigned <							
JRUGE	Jump if C = 0	Jmp if unsigned >=							
JRUGT	Jump if (C + Z = 0)	Unsigned >							

13.3.4 Auxiliary Voltage Detector (AVD) Thresholds

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

Symbol	Parameter	Conditions	Typ	Unit
$V_{IT+(AVD)}$	1=>0 AVDF flag toggle threshold (V_{DD} rise)	High Threshold Med. Threshold Low Threshold	4.50 4.00 3.35	V
$V_{IT-(AVD)}$	0=>1 AVDF flag toggle threshold (V_{DD} fall)	High Threshold Med. Threshold Low Threshold	4.40 3.85 3.20	
V_{hys}	AVD voltage threshold hysteresis	$V_{IT+(AVD)}$-$V_{IT-(AVD)}$	170	mV
ΔV_{IT-}	Voltage drop between AVD flag set and LVD reset activation	V_{DD} fall	0.15	V

13.3.5 Internal RC Oscillator and PLL

The ST7 internal clock can be supplied by an internal RC oscillator and PLL (selectable by option byte).

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{DD(RC)}$	Internal RC Oscillator operating voltage	Refer to operating range of V_{DD} with T_A, section 13.3.1 on page 112	2.7		5.5	V
$V_{DD(x4PLL)}$	x4 PLL operating voltage		2.7		3.7	
$V_{DD(x8PLL)}$	x8 PLL operating voltage		3.3		5.5	
$t_{STARTUP}$	PLL Startup time			60		PLL input clock (f_{PLL}) cycles

ADC CHARACTERISTICS (Cont'd)

ADC Accuracy with V_{DD}=5.0V

Symbol	Parameter	Conditions	Typ	Max	Unit
E_T	Total unadjusted error	f_{CPU}=8MHz, f_{ADC}=4MHz	4	6 [1)]	LSB
E_O	Offset error		3	5 [1)]	
E_G	Gain Error		0.5	4 [1)]	
E_D	Differential linearity error [3)]		1.5	3 [2)]	
E_L	Integral linearity error [3)]		1.5	3 [2)]	

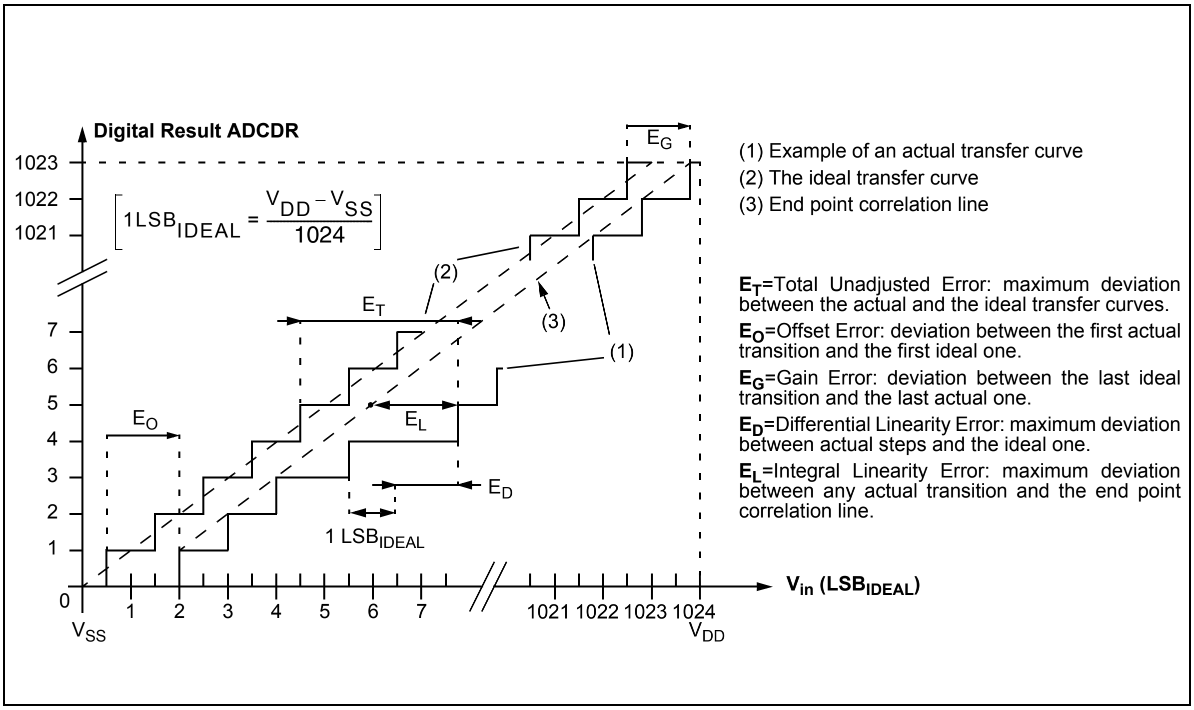
Notes:

1. Data based on characterization results. Not tested in production.

2. Injecting negative current on any of the analog input pins significantly reduces the accuracy of any conversion being performed on any analog input.
Analog pins can be protected against negative injection by adding a Schottky diode (pin to ground). Injecting negative current on digital input pins degrades ADC accuracy especially if performed on a pin close to the analog input pins.
Any positive injection current within the limits specified for I_{INJ}(PIN) and ΣI_{INJ}(PIN) in Section 13.8 does not affect the ADC accuracy.

3. Data based on characterization results over the whole temperature range, monitored in production.

Figure 111. ADC Accuracy Characteristics with amplifier disabled



ADC CHARACTERISTICS (Cont'd)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{DD(AMP)}$	Amplifier operating voltage		3.6		5.5	V
V_{IN}	Amplifier input voltage[4]	V_{DD}=3.6V	0		350	mV
		V_{DD}=5V	0		500	
V_{OFFSET} [1]	Amplifier output offset voltage[5]	V_{DD}=5V		200		mV
V_{STEP} [1]	Step size for monotonicity[3]	V_{DD}=3.6V	3.5			mV
		V_{DD}=5V	4.89			
Linearity [1]	Output Voltage Response		Linear			
Gain factor [1]	Amplified Analog input Gain[2]			8		
Vmax [1]	Output Linearity Max Voltage	V_{INmax} = 430mV, V_{DD}=5V		3.65		V
Vmin [1]	Output Linearity Min Voltage			200		mV

Notes:

1. Data based on characterization results over the whole temperature range, not tested in production.

2. For precise conversion results it is recommended to calibrate the amplifier at the following two points:

– offset at V_{INmin} = 0V

– gain at full scale (for example V_{IN}=430mV)

3. Monotonicity guaranteed if V_{IN} increases or decreases in steps of min. 5mV.

4. Please refer to the Application Note AN1830 for details of TE% vs Vin.

5. Refer to the offset variation in temperature below

Amplifier output offset variation

The offset is quite sensitive to temperature variations. In order to ensure a good reliability in measurements, the offset must be recalibrated periodically i.e. during power on or whenever the device is reset depending on the customer application and during temperature variation. The table below gives the typical offset variation over temperature:

Typical Offset Variation (LSB)				UNIT
-45	-20	+25	+90	°C
-12	-7	-	+13	LSB

13.12 ANALOG COMPARATOR CHARACTERISTICS

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

13.13 PROGRAMMABLE INTERNAL VOLTAGE REFERENCE CHARACTERISTICS

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

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

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

Notes:

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

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

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.