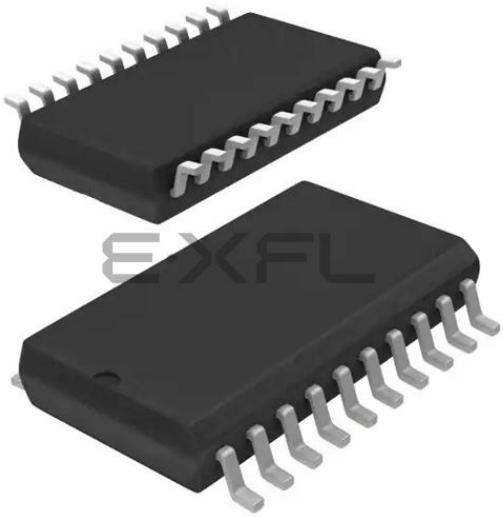


Welcome to E-XFL.COM

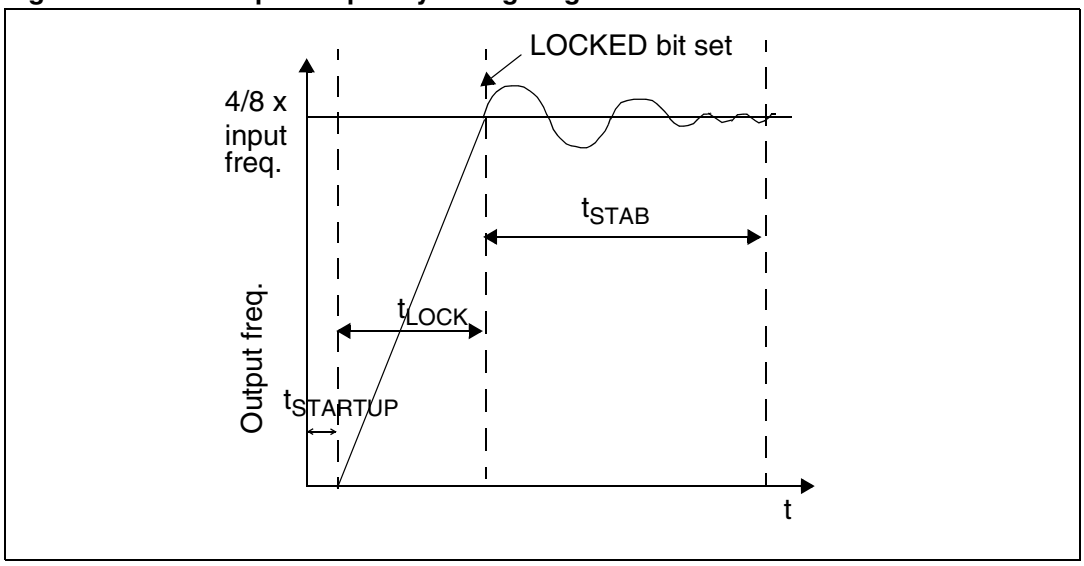
What is "Embedded - Microcontrollers"?

"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	DALI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	15
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SO
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fdalif2m6tr

Figure 11. PLL output frequency timing diagram



When the PLL is started, after reset or wakeup from Halt mode or AWUFH mode, it outputs the clock after a delay of $t_{STARTUP}$.

When the PLL output signal reaches the operating frequency, the LOCKED bit in the SICSCR register is set. Full PLL accuracy (ACC_{PLL}) is reached after a stabilization time of t_{STAB} (see *Figure 11* and *Section 20.3.3: Internal RC oscillator and PLL on page 131*)

Refer to *Section 9.7.4 on page 44* for a description of the LOCKED bit in the SICSR register.

9.4 Register description

9.4.1 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_{OSC2} or $f_{OSC2}/32$.

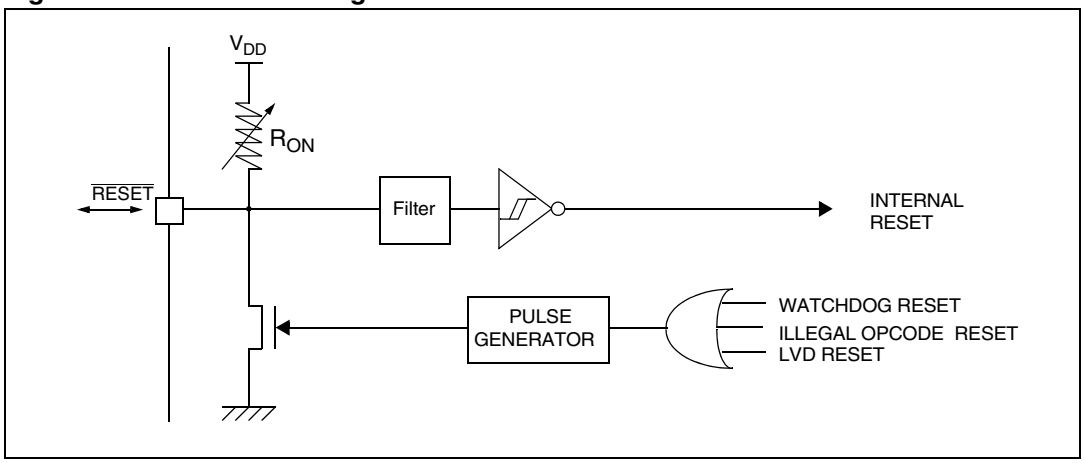
0: Normal mode ($f_{CPU} = f_{OSC2}$
1: Slow mode ($f_{CPU} = f_{OSC2}/32$)

9.6.2 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 15*). This detection is asynchronous and therefore the MCU can enter reset state even in Halt mode.

Figure 14. Reset block diagram



Note: *Illegal Opcode Reset on page 124 for more details on illegal opcode reset conditions.*

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.

9.6.3 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.

9.6.4 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 15*.

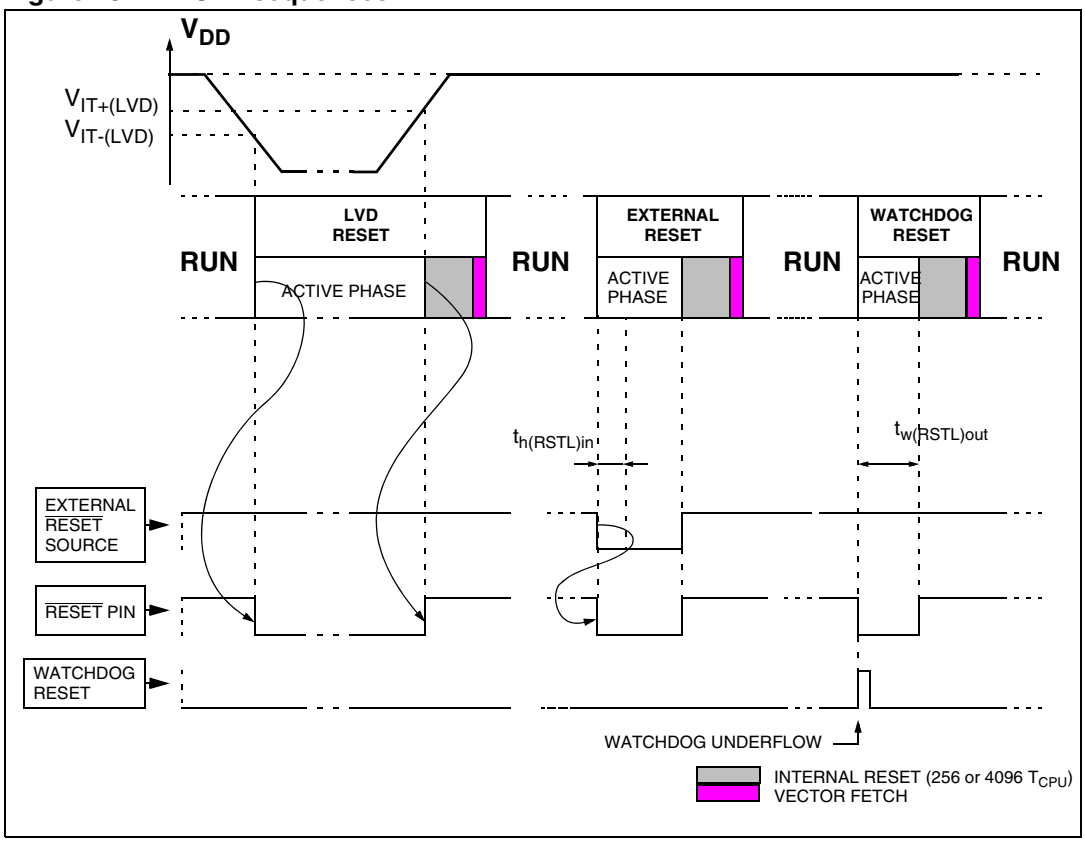
The LVD filters spikes on V_{DD} larger than $t_{g(VDD)}$ to avoid parasitic resets.

9.6.5 Internal watchdog RESET

The RESET sequence generated by a internal Watchdog counter overflow is shown in *Figure 15*.

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 15. RESET sequences



9.7 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 Illegal Opcode Reset on page 124 for further details.

9.7.1 Low voltage detector (LVD)

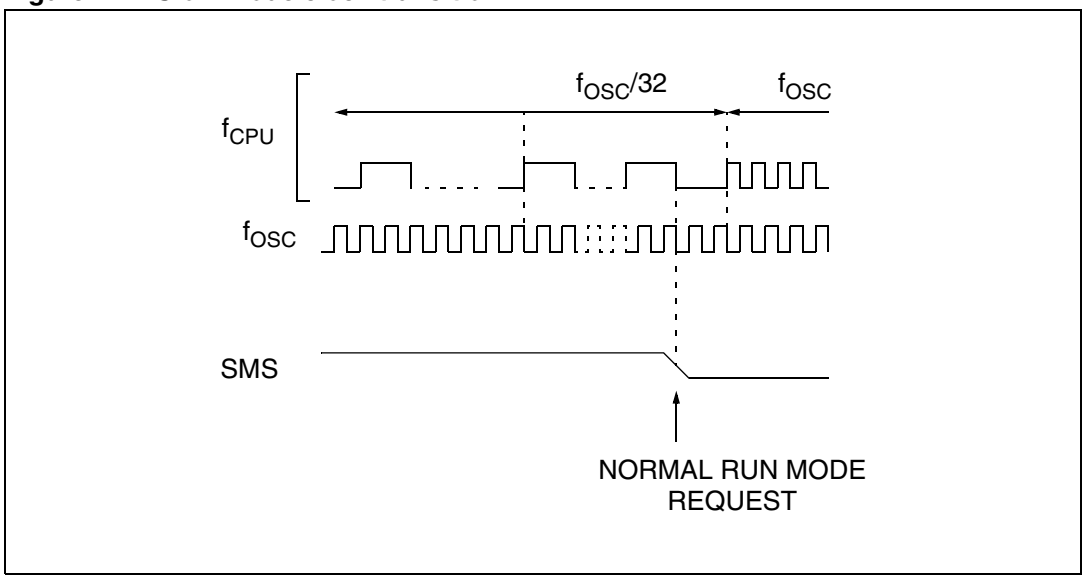
The Low Voltage Detector function (LVD) generates a static reset when the V_{DD} supply voltage is below a $V_{IT-(LVD)}$ reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

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).

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 21. Slow mode clock transition



11.3 Wait mode

Wait mode places the MCU in a low power consumption mode by stopping the CPU.

This power saving mode is selected by calling the 'WFI' instruction.

All peripherals remain active. During Wait mode, the I bit of the CC register is cleared, to enable all interrupts. All other registers and memory remain unchanged. The MCU remains in Wait mode until an interrupt or RESET occurs, whereupon the Program Counter branches to the starting address of the interrupt or Reset service routine.
The MCU will remain in Wait mode until a Reset or an Interrupt occurs, causing it to wake up.

Refer to *Figure 22*.

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.

Table 38. Counter clock frequency

Counter clock selection	CK1	CK0
OFF	0	0
f_{LTIMER} (1 ms timebase @ 8 MHz) (1)	0	1
f_{CPU}	1	0
32 MHz (2)	1	1

1. PWM mode and Output Compare modes are not available at this frequency.
2. ATICR counter may return inaccurate results when read. It is therefore not recommended to use Input Capture mode at this frequency.

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

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

14.6.2 Counter register high (CNTRH)

Read only
Reset Value: 0000 0000 (000h)

15							8
0	0	0	0	CNTR 11	CNTR 10	CNTR9	CNTR8

14.6.3 Counter register low (CNTRL)

Read only
Reset Value: 0000 0000 (000h)

7							0
CNTR7	CNTR6	CNTR5	CNTR4	CNTR3	CNTR2	CNTR1	CNTR0

Bits 15:12 = Reserved.

Bits 11:0 = **CNTR[11:0]** *Counter Value*.
This 12-bit register is read by software and cleared by hardware after a reset. The counter is incremented continuously as soon as a counter clock is selected. To obtain the 12-bit value,

16.4 General description

The DCM is able to receive or transmit a serial DALI signal using a 16-bit shift register, an edge detector, several data/control registers and arbitration logic.

The DCM receives the DALI standard signal from the lighting control network, checks for errors and loads the address/data bytes of a "forward frame" to the corresponding DCMFA/DCMFD registers and sends back the data byte of the "backward frame" (written by software to the DCMBD register) in DALI standard format.

The data rate can be changed by writing in the DCMCLK register (f_{DATA} = 2* f_{DALI}).

$$f_{DATA} = f_{CPU}/[(N+1)*16]$$

The DALI standard data rate f_{DALI} is 1.2 kHz. N is the integer value of the DCMCLK register. Following the above formula, if f_{CPU} is 8 MHz, the integer value of the DCMCLK register is "207". The bi-phase bit period is 833.33 us ±10%.

The polarity of the bi-phase start bit is not configurable. The start bit is a logical ’1’.

The polarity of the 2 stop bits is not configurable. The 2 stop bits are set to high level.

If an error is detected during reception, the frame will be ignored and the DCM will return to Receive state.

16.5 Functional description

The user must write to the DCMCLK register to select the data rate according to the DALI signal frequency.

After Reset, the DCM is in Receive state and waits for the bi-phase start bit (logical ’1’) of the "forward frame".

The DCM checks the data format of the "forward frame" with the 4-bit pre-shift register. If an error occurs during reception, the DCM will skip the data and return to the Receive state.

If there is no error in the "forward frame", the data will be shifted Most Significant Bit-first into the 16-bit shift register. The address byte and the data byte will be loaded to the corresponding DCMFA and DCMFD registers. The DCM will send an interrupt signal by setting the ITF bit in the DCMCSR register.

If the software receives an interrupt signal from the DCM, it reads the DCMFA and DCMFD registers.

Depending on the command, the DCM is able to send back or receive data.

In an interrupt routine, the RTS bit has to be set either before or at the same time as the RTA bit.

If the software asks the DCM to send back a "backward frame", the software must first write to the DCMBD register and switch the DCM to Transmit state by setting the RTS and RTA bits in the DCMCR register during the interrupt routine. The DCMBD register will be shifted out from the 16-bit shift register in DALI format, the Most Significant Bit-first.

When the "backward frame" has been transmitted, the DCM will send an interrupt signal by setting the ITF bit in the DCMCSR register.

0: The DCM is in Transmit state

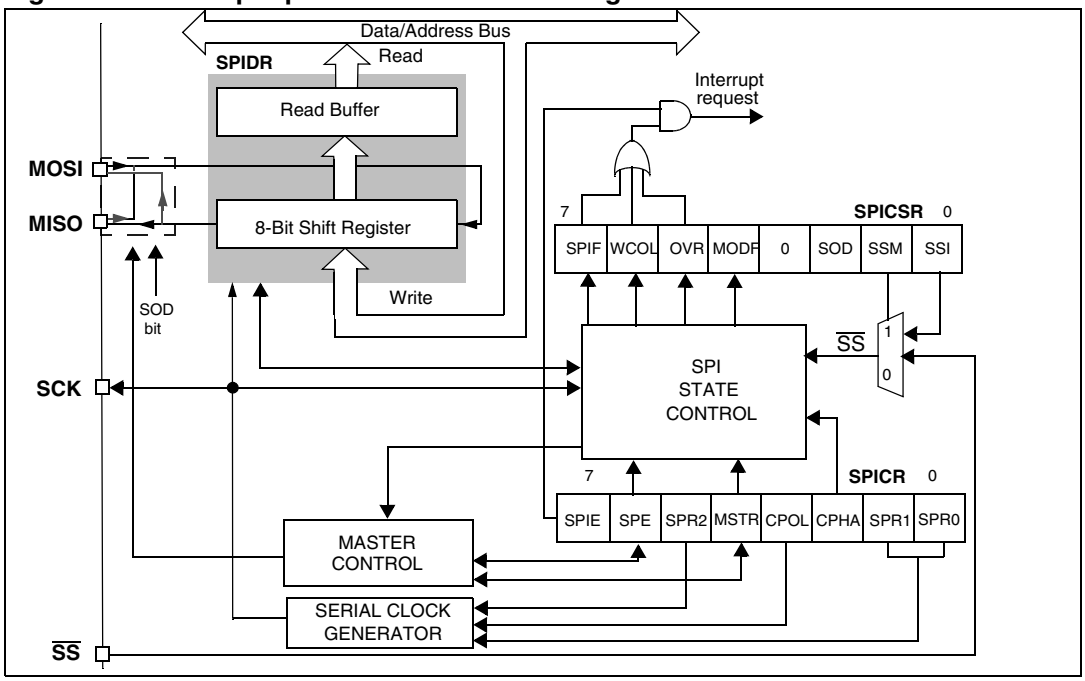
1: The DCM is in Receive state

Bits 3:0 = **DCMCSR[3:0]** *Clock counter value.* (Read only) These bits are set/cleared by hardware and read by software.
The value of the 4-bit sample clock counter (integer range 0 to 15). The clock counter value is loaded in the DCMCSR register when a DALI change of phase signal is detected (edge trigger). Refer to *Figure 44*.

Table 45. Register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0040h	**DCMCLK** Reset Value	CK7 0	CK6 0	CK5 0	CK4 0	CK3 0	CK2 0	CK1 0	CK0 0
0041h	**DCMFA** Reset Value	FA7 0	FA6 0	FA5 0	FA4 0	FA3 0	FA2 0	FA1 0	FA0 0
0042h	**DCMFD** Reset Value	FD7 0	FD6 0	FD5 0	FD4 0	FD3 0	FD2 0	FD1 0	FD0 0
0043h	**DCMBD** Reset Value	BD7 0	BD6 0	BD5 0	BD4 0	BD3 0	BD2 0	BD1 0	BD0 0
0044h	**DCMCR** Reset Value	0	0	0	0	DCME 0	RTA 0	RTS 0	FTS 0
0045h	**DCMCSR** Reset Value	ITE 0	ITF 0	EF 0	RTF 0	CK3 0	CK2 0	CK1 0	CK0 0

Figure 45. Serial peripheral interface block diagram



17.4 Functional description

A basic example of interconnections between a single master and a single slave is illustrated in *Figure 46*.

The MOSI pins are connected together and the MISO pins are connected together. In this way data is transferred serially between master and slave (most significant bit first).

The communication is always initiated by the master. When the master device transmits data to a slave device via MOSI pin, the slave device responds by sending data to the master device via the MISO pin. This implies full duplex communication with both data out and data in synchronized with the same clock signal (which is provided by the master device via the SCK pin).

To use a single data line, the MISO and MOSI pins must be connected at each node (in this case only simplex communication is possible).

Four possible data/clock timing relationships may be chosen (see *Figure 49*) but master and slave must be programmed with the same timing mode.

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

17.4.3 Slave mode operation

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

To operate the SPI in slave mode:

1. Write to the SPICSR register to perform the following actions:
   - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see *Figure 49*).

Note: The slave must have the same CPOL and CPHA settings as the master.

   - Manage the $\overline{SS}$ pin as described in *Section 17.4.1* and *Figure 47*. If CPHA=1 $\overline{SS}$ must be held low continuously. If CPHA=0 $\overline{SS}$ must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.
2. Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

Slave mode transmit sequence

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

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

When data transfer is complete:

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

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

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

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

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an Overrun condition (see Overrun condition (OVR) on page 106).

17.4.4 Clock phase and clock polarity

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits (See *Figure 49*).

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

The combination of the CPOL clock polarity and CPHA (clock phase) bits selects the data capture clock edge

1. A read access to the SPICSR register while the MODF bit is set.
2. A write to the SPICR register.

Note: To avoid any conflicts in an application with multiple slaves, the $\overline{SS}$ pin must be pulled high during the MODF bit clearing sequence. The SPE and MSTR bits may be restored to their original state during or after this clearing sequence.

Hardware does not allow the user to set the SPE and MSTR bits while the MODF bit is set except in the MODF bit clearing sequence.

In a slave device, the MODF bit can not be set, but in a multi master configuration the Device can be in slave mode with the MODF bit set.

The MODF bit indicates that there might have been a multi-master conflict and allows software to handle this using an interrupt routine and either perform to a reset or return to an application default state.

Overrun condition (OVR)

An overrun condition occurs, when the master device has sent a data byte and the slave device has not cleared the SPIF bit issued from the previously transmitted byte.

When an Overrun occurs:

- The OVR bit is set and an interrupt request is generated if the SPIE bit is set.

In this case, the receiver buffer contains the byte sent after the SPIF bit was last cleared. A read to the SPIDR register returns this byte. All other bytes are lost.

The OVR bit is cleared by reading the SPICSR register.

Write collision error (WCOL)

A write collision occurs when the software tries to write to the SPIDR register while a data transfer is taking place with an external device. When this happens, the transfer continues uninterrupted; and the software write will be unsuccessful.

Write collisions can occur both in master and slave mode. See also *Section 17.4.1: Slave select management*.

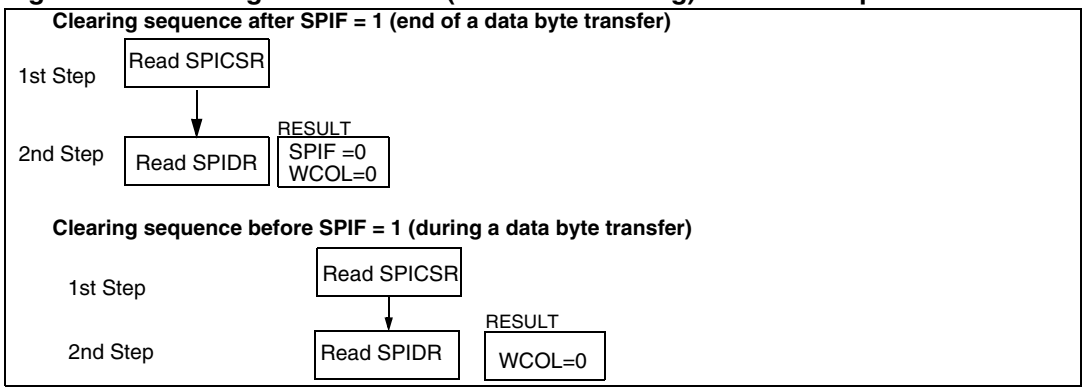
Note: A "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the CPU operation.

The WCOL bit in the SPICSR register is set if a write collision occurs.

No SPI interrupt is generated when the WCOL bit is set (the WCOL bit is a status flag only).

Clearing the WCOL bit is done through a software sequence (see *Figure 50*).

Figure 50. Clearing the WCOL bit (write collision flag) software sequence



Note: Writing to the SPIDR register instead of reading it does not reset the WCOL bit.

17.4.6 Single master and multimaster configurations

There are two types of SPI systems:

- Single Master System
- Multimaster System

Single master system

A typical single master system may be configured, using a device as the master and four devices as slaves (see *Figure 51*).

The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices.

The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices.

Note: To prevent a bus conflict on the MISO line the master allows only one active slave device during a transmission.

For more security, the slave device may respond to the master with the received data byte. Then the master will receive the previous byte back from the slave device if all MISO and MOSI pins are connected and the slave has not written to its SPIDR register.

Other transmission security methods can use ports for handshake lines or data bytes with command fields.

Multimaster system

A multimaster system may also be configured by the user. Transfer of master control could be implemented using a handshake method through the I/O ports or by an exchange of code messages through the serial peripheral interface system.

The multimaster system is principally handled by the MSTR bit in the SPICR register and the MODF bit in the SPICSR register.

Table 57. Immediate instructions

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

19.1.3 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 requiring only one 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.

19.1.4 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 indexed addressing mode consists of three submodes:

Indexed (no offset)

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

Indexed (short)

The offset is a byte, thus requiring only one 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.

19.1.5 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:

Table 63. Current characteristics

Symbol	Ratings	Maximum value	Unit
I_{VDD}	Total current into V_{DD} power lines (source)	75	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink)	150	
I_{IO}	Output current sunk by any standard I/O and control pin	20	
	Output current sunk by any high sink I/O pin	40	
	Output current source by any I/Os and control pin	- 25	
$I_{INJ(PIN)}$ (1) & (2)	Injected current on $\overline{RESET}$ pin	± 5	
	Injected current on OSC1 and OSC2 pins	± 5	
	Injected current on PB0 and PB1 pins (3)	+5	
	Injected current on any other pin(4)	± 5	
$\Sigma I_{INJ(PIN)}$ 2)	Total injected current (sum of all I/O and control pins) (4)	± 20	

1. $I_{INJ(PIN)}$ must never be exceeded. This is implicitly insured if V_{IN} maximum is respected. If V_{IN} maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by $V_{IN}>V_{DD}$ while a negative injection is induced by $V_{IN}<V_{SS}$. For true open-drain pads, there is no positive injection current, and the corresponding V_{IN} maximum must always be respected

2. Negative injection disturbs the analog performance of the device. In particular, it induces leakage currents throughout the device including the analog inputs. To avoid undesirable effects on the analog functions, care must be taken:
- Analog input pins must have a negative injection less than 0.8 mA (assuming that the impedance of the analog voltage is lower than the specified limits)
- Pure digital pins must have a negative injection less than 1.6mA. In addition, it is recommended to inject the current as far as possible from the analog input pins.

3. No negative current injection allowed on PB0 and PB1 pins.

4. When several inputs are submitted to a current injection, the maximum $\Sigma I_{INJ(PIN)}$ is the absolute sum of the positive and negative injected currents (instantaneous values). These results are based on characterization with $\Sigma I_{INJ(PIN)}$ maximum current injection on four I/O port pins of the device.

Table 64. Thermal characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature (see *Table 94 on page 160*)		

Table 68. Operating voltage and startup time

Symbol	Parameter	Min	Typ	Max	Unit
$V_{DD(RC)}$	Internal RC Oscillator operating voltage	2.4		5.5	V
$V_{DD(x4PLL)}$	x4 PLL operating voltage	2.4		3.3	
$V_{DD(x8PLL)}$	x8 PLL operating voltage	3.3		5.5	
$t_{STARTUP}$	PLL Startup time		60		PLL input clock (f_{PLL}) cycles

The RC oscillator and PLL characteristics are temperature-dependent and are grouped in four tables.

Table 69. RC oscillator and PLL characteristics (tested for T_A = -40 to +85° C) @ V_{DD} = 4.5 to 5.5 V

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{RC} 1)	Internal RC oscillator frequency (1)	RCCR = FF (reset value), T_A=25° C, V_{DD}=5 V		760		kHz
		RCCR = RCCR0(2), T_A=25° C, V_{DD}=5 V		1000		
ACC_{RC}	Accuracy of Internal RC oscillator with RCCR=RCCR0(2)	T_A=25° C, V_{DD}=4.5 to 5.5 V	-1		+1	%
		T_A=-40 to +85° C, V_{DD}=5 V	-5		+2	%
		T_A=0 to +85° C, V_{DD}=4.5 to 5.5 V	-2(3)		+2(3)	%
$I_{DD(RC)}$	RC oscillator current consumption	T_A=25° C, V_{DD}=5 V		970(3)		µA
$t_{su(RC)}$	RC oscillator setup time	T_A=25° C, V_{DD}=5 V			10(2)	µs
f_{PLL}	x8 PLL input clock			1(3)		MHz
t_{LOCK}	PLL Lock time(4)			2		ms
t_{STAB}	PLL Stabilization time(4)			4		ms
ACC_{PLL}	x8 PLL Accuracy	f_{RC} = 1 MHz@T_A=25° C, V_{DD}=4.5 to 5.5 V		0.1(5)		%
		f_{RC} = 1 MHz@T_A=-40 to +85° C, V_{DD}=5 V		0.1(5)		%
$t_{w(JIT)}$	PLL jitter period	f_{RC} = 1 MHz		125(6)		µs
JIT_{PLL}	PLL jitter ($\Delta f_{CPU}/f_{CPU}$)			1(6)		%
$I_{DD(PLL)}$	PLL current consumption	T_A=25° C		600(3)		µA

1. If the RC oscillator clock is selected, to improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100nF, between the V_{DD} and V_{SS} pins as close as possible to the ST7 device.
2. See *Section 9.2: Internal RC oscillator adjustment on page 32*
3. Data based on characterization results, not tested in production
4. After the LOCKED bit is set ACC_{PLL} is max. 10% until t_{STAB} has elapsed. See *Figure 11 on page 34*.
5. Averaged over a 4ms period. After the LOCKED bit is set, a period of t_{STAB} is required to reach ACC_{PLL} accuracy.
6. Guaranteed by design.

Figure 65. Typical I_{DD} in Wait vs. f_{CPU}

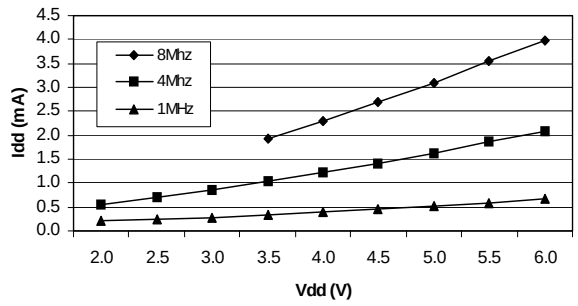


Figure 66. Typical I_{DD} in Slow-Wait vs. f_{CPU}

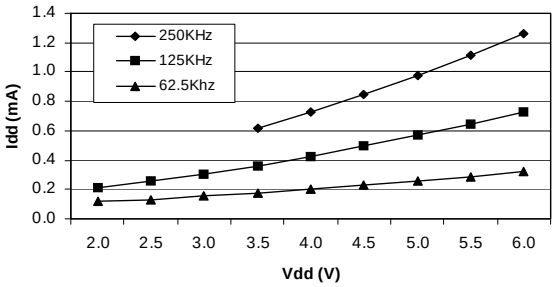


Figure 67. Typical I_{DD} in AWUFH mode at T_A=25°C

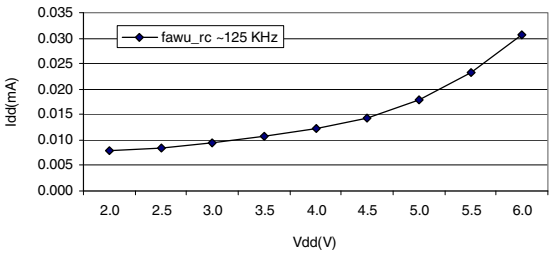


Figure 68. Typical I_{DD} vs. Temperature at V_{DD} = 5 V and f_{CPU} = 8 MHz

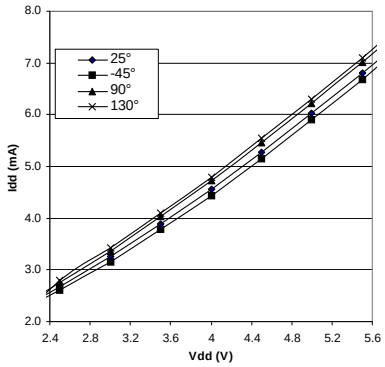


Table 73. On-chip peripherals

Symbol	Parameter	Conditions		Typ	Unit
$I_{DD(AT)}$	12-bit Auto-Reload Timer supply current (1)	f_{CPU}=4 MHz	V_{DD}=3.0 V	300	µA
		f_{CPU}=8 MHz	V_{DD}=5.0 V	1000	
$I_{DD(SPI)}$	SPI supply current (2)	f_{CPU}=4 MHz	V_{DD}=3.0 V	50	
		f_{CPU}=8 MHz	V_{DD}=5.0 V	300	
$I_{DD(ADC)}$	ADC supply current when converting (3)	f_{ADC}=4 MHz	V_{DD}=3.0 V	250	
			V_{DD}=5.0 V	1100	

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 off.

20.8 I/O port pin characteristics

20.8.1 General characteristics

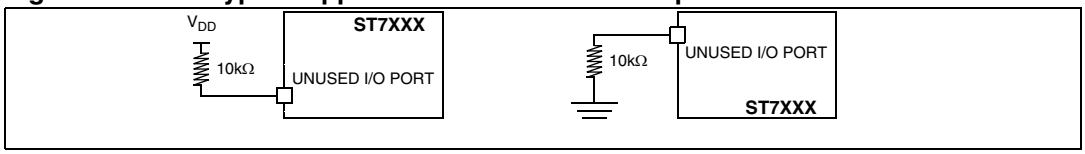
Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Table 85. General characteristics

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

1. Data based on characterization results, not tested in production.
2. Configuration not recommended, all unused pins must be kept at a fixed voltage: using the output mode of the I/O for example or an external pull-up or pull-down resistor (see *Figure 70*). Static peak current value taken at a fixed V_{IN} value, based on design simulation and technology characteristics, not tested in production. This value depends on V_{DD} and temperature values.
3. The R_{PU} pull-up equivalent resistor is based on a resistive transistor (corresponding I_{PU} current characteristics described in *Figure 71*).
4. To generate an external interrupt, a minimum pulse width has to be applied on an I/O port pin configured as an external interrupt source.

Figure 70. Two typical applications with unused I/O pin



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

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

20.11 10-bit ADC characteristics

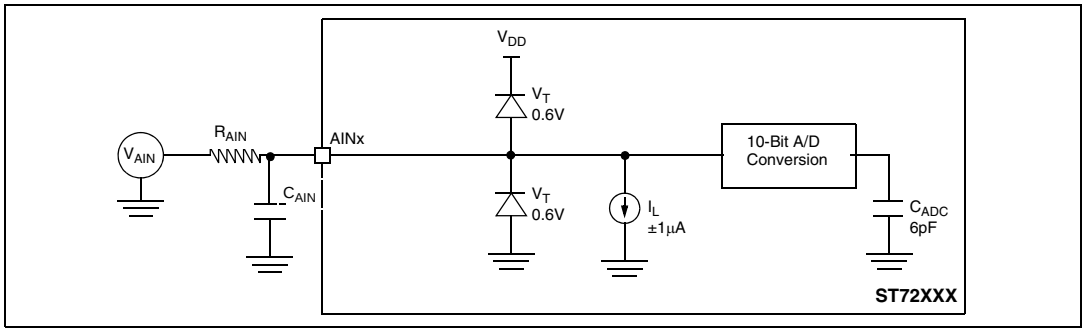
Subject to general operating condition for V_{DD}, f_{OSC}, and T_A unless otherwise specified.

Table 89. ADC characteristics

Symbol	Parameter	Conditions	Min	Typ (1)	Max	Unit
f_{ADC}	ADC clock frequency				4	MHz
V_{AIN}	Conversion voltage range (2)		V_{SSA}		V_{DDA}	V
R_{AIN}	External input resistor				10 (3)	kΩ
C_{ADC}	Internal sample and hold capacitor			6		pF
t_{STAB}	Stabilization time after ADC enable	f_{CPU}=8MHz, f_{ADC}=4MHz	0 (4)			µs
t_{ADC}	Conversion time (Sample+Hold)		3.5			µs
	- Sample capacitor loading time - Hold conversion time		4 10			$1/f_{ADC}$
I_{ADC}	Analog Part				1	mA
	Digital Part				0.2	mA

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. When V_{DDA} and V_{SSA} pins are not available on the pinout, the ADC refers to V_{DD} and V_{SS}.
3. Any added external serial resistor will downgrade the ADC accuracy (especially for resistance greater than 10kΩ). Data based on characterization results, not tested in production.
4. The stabilization time of the AD converter is masked by the first t_{LOAD}. The first conversion after the enable is then always valid.

Figure 92. Typical Application with ADC



22.2 Device ordering information and transfer of customer code

Customer code is made up of the FASTROM contents and the list of the selected options (if any). The FASTROM contents are to be sent on diskette, or by electronic means, with the S19 hexadecimal file generated by the development tool. All unused bytes must be set to FFh. The selected options are communicated to STMicroelectronics using the correctly completed OPTION LIST appended *on page 165*.

Refer to application note AN1635 for information on the counter listing returned by ST after code has been transferred.

The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.

Table 99. Order codes

Order code	Program memory (bytes)	RAM (bytes)	Temp. range	Package
ST7FDALIF2M6	8K Flash	384	-40° C to 85° C	SO20
ST7PDALIF2M6	8K FASTROM	384	-40° C to 85° C	SO20

Note: Contact ST sales office for product availability.

Table 100. Document revision history (continued)

Date	Revision	Changes
19-Nov-2004	2	Reset delay in section 11.1.3 on page 51 changed to 30 µs Altered note 1 for section 13.2.3 on page 101 removing references to RESET Removed sentence relating to an effective change only after overflow for CK[1:0], page 60 MOD00 replaced by 0Ex in Figure 36 on page 57 Added Note 2 related to Exit from Active halt, section 11.2.5 on page 59 Added illegal opcode detection to page 1, section 7.6 on page 29, section 12 on page 94 Clarification of Flash readout protection, section 4.5.1 on page 14 Added note 4 and description relating to Total Percentage in Error and Amplifier Output Offset Variation to the ADC Characteristics subsection and table, page 126 Added note 5 and description relating to Offset Variation in Temperature to ADC Characteristics subsection and table, page 126 FPLL value of 1MHz quoted as Typical instead of a Minimum in section 13.3.4.1 on page 104 Updated FSCK in section 13.10.1 on page 121 to $f_{CPU}/4$ and $f_{CPU}/2$ Corrected f_{CPU} in Slow and slow wait modes in section 13.4.1 on page 108 Max values updated for ADC Accuracy, page 124 Notes indicating that PB4 cannot be used as an external interrupt in HALT mode, section 16.6 on page 138 and Section 8.3 PERIPHERAL INTERRUPTS Changed section 11.5.2 on page 79 Changed section 11.5.3.3 on page 82 Removed “optional” referring to VDD in Figure 4 on page 13 Changed FMP_R option bit description in section 15.1 on page 130 Added “Clearing active interrupts outside interrupt routine” on page 138 Changed “Development Tools” on page 134 Changed Figure 41 on page 70: f_{CPU} instead of 8 MHz f_{CPU}

Table 100. Document revision history (continued)

Date	Revision	Changes
05-Feb-2009	3	Updated *Section 7.5: Access error handling on page 25* Added caution in *Section 9.6: Reset sequence manager (RSM) on page 38* Renamed f_{OSC} to f_{OSC2} in *Figure 12: Clock management block diagram on page 35* Modified *Section 14.6.3: Counter register low (CNTRL) on page 78* Updated description of EOC bit in *Section 18: 10-bit A/D converter (ADC)* and added *Section 18.4: Changing the conversion channel on page 115* Updated EMC characteristics *Section 20.7 on page 141* Updated *Table 63: Current characteristics on page 129* Removed EMC protective circuitry in *Figure 87* and *Figure 88 on page 151* (device works correctly without these components) Replaced soldering information with ECOPACK reference in *Section 21 on page 158* Increased precision of package dimensions in inches to 4 decimal digits in *Table 93 on page 159*.