



Welcome to [E-XFL.COM](http://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.

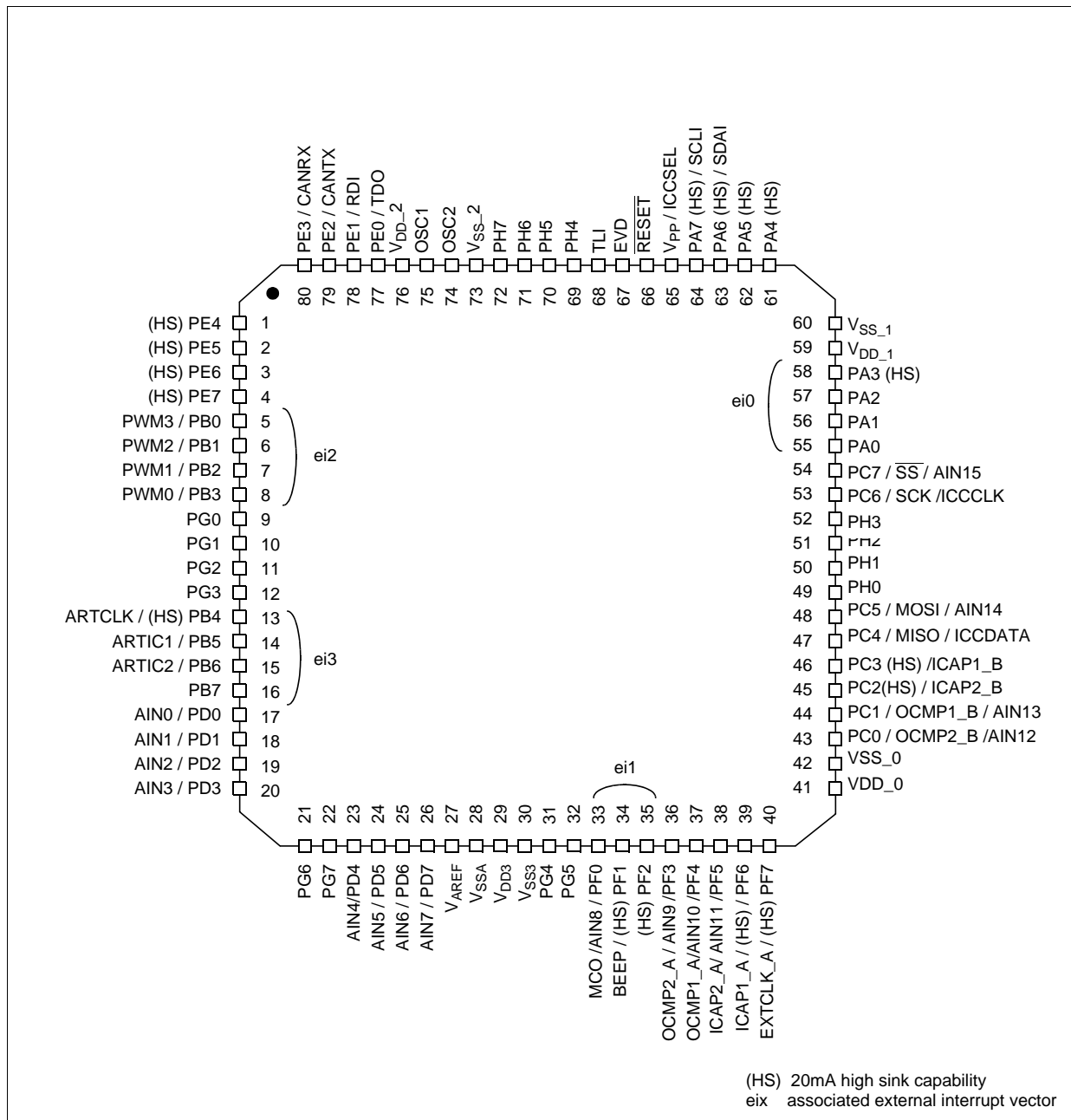
### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, I <sup>2</sup> C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	48
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/stmicroelectronics/st72f521r9t3">https://www.e-xfl.com/product-detail/stmicroelectronics/st72f521r9t3</a>

## 2 PIN DESCRIPTION

Figure 2. 80-Pin TQFP 14x14 Package Pinout



## 6 SUPPLY, RESET AND CLOCK MANAGEMENT

The device includes a range of utility features for securing the application in critical situations (for example in case of a power brown-out), and reducing the number of external components. An overview is shown in [Figure 11](#).

For more details, refer to dedicated parametric section.

### Main features

- Optional PLL for multiplying the frequency by 2 (not to be used with internal RC oscillator)
- Reset Sequence Manager (RSM)
- Multi-Oscillator Clock Management (MO)
  - 5 Crystal/Ceramic resonator oscillators
  - 1 Internal RC oscillator
- System Integrity Management (SI)
  - Main supply Low voltage detection (LVD)
  - Auxiliary Voltage detector (AVD) with interrupt capability for monitoring the main supply or the EVD pin

### 6.1 PHASE LOCKED LOOP

If the clock frequency input to the PLL is in the range 2 to 4 MHz, the PLL can be used to multiply the frequency by two to obtain an  $f_{OSC2}$  of 4 to 8 MHz. The PLL is enabled by option byte. If the PLL is disabled, then  $f_{OSC2} = f_{OSC}/2$ .

**Caution:** The PLL is not recommended for applications where timing accuracy is required. See “PLL Characteristics” on page 177.

Figure 10. PLL Block Diagram

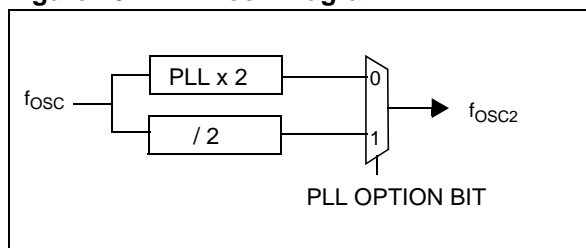
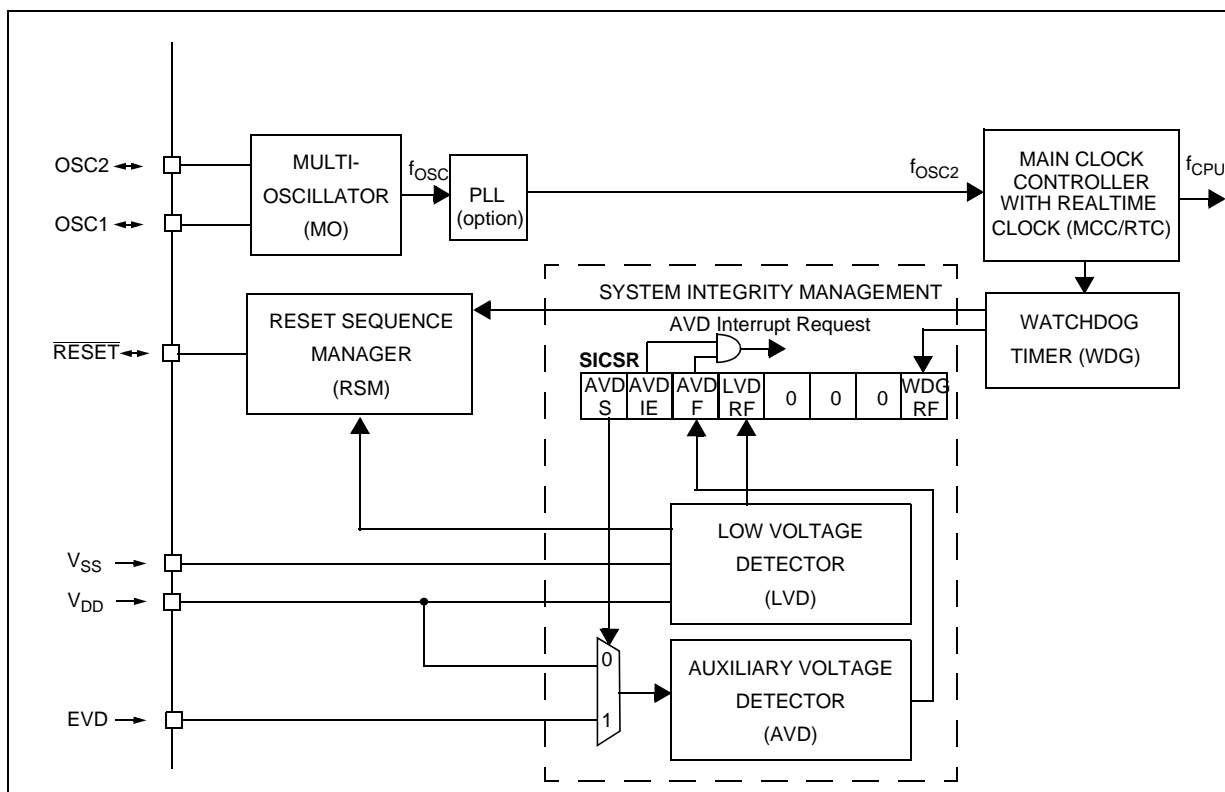


Figure 11. Clock, Reset and Supply Block Diagram



## 8 POWER SAVING MODES

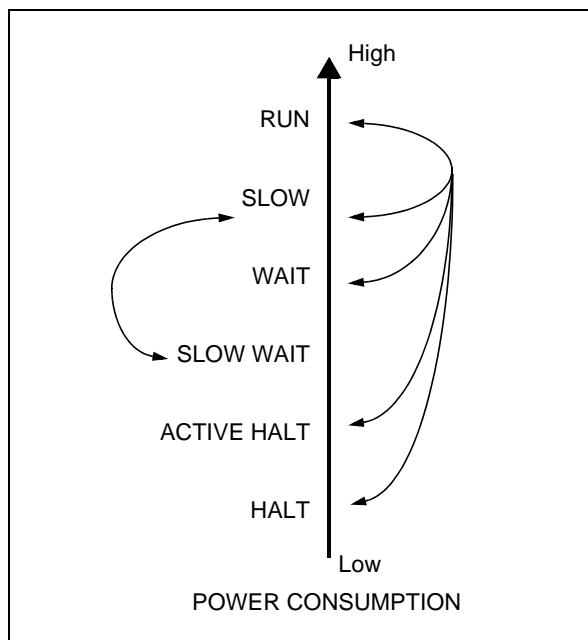
### 8.1 INTRODUCTION

To give a large measure of flexibility to the application in terms of power consumption, four main power saving modes are implemented in the ST7 (see Figure 23): SLOW, WAIT (SLOW WAIT), ACTIVE HALT and 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 23. Power Saving Mode Transitions



### 8.2 SLOW MODE

This mode has two targets:

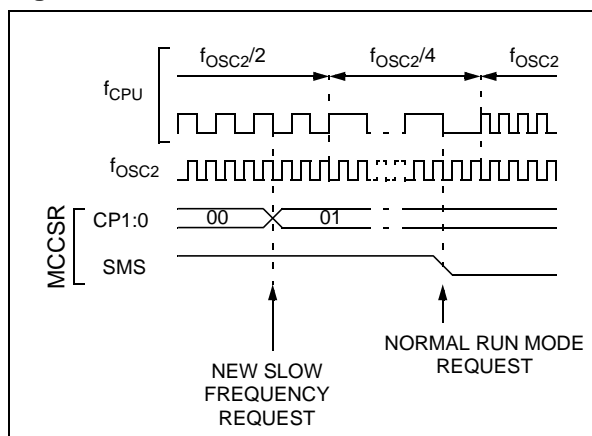
- To reduce power consumption by decreasing the internal clock in the device,
- To adapt the internal clock frequency ( $f_{CPU}$ ) to the available supply voltage.

SLOW mode is controlled by three bits in the MCCSR register: the SMS bit which enables or disables Slow mode and two CPx bits which select the internal slow frequency ( $f_{CPU}$ ).

In this mode, the master clock frequency ( $f_{OSC2}$ ) can be divided by 2, 4, 8 or 16. The CPU and peripherals are clocked at this lower frequency ( $f_{CPU}$ ).

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

Figure 24. SLOW Mode Clock Transitions





16-BIT TIMER (Cont'd)

Figure 49. Output Compare Timing Diagram,  $f_{\text{TIMER}} = f_{\text{CPU}}/2$

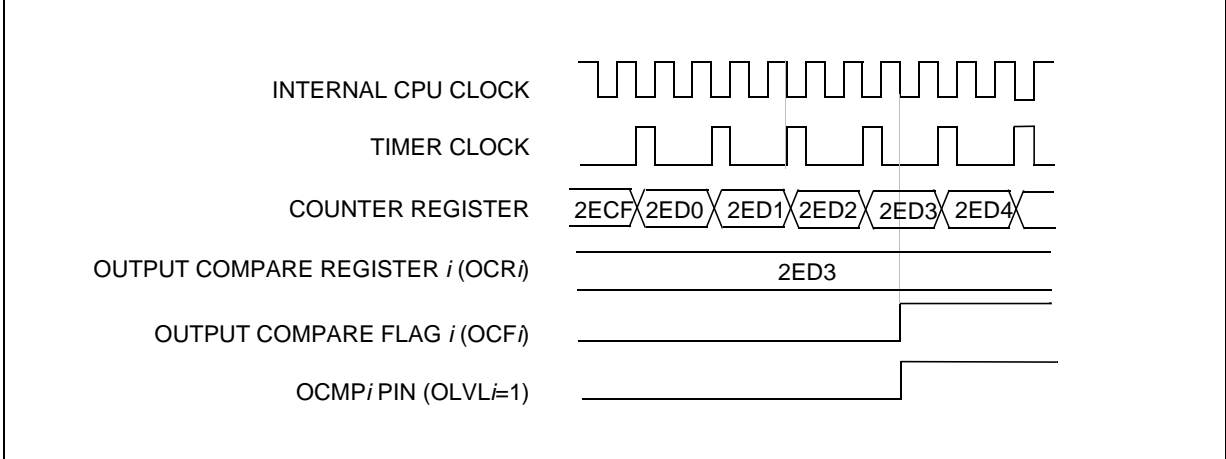
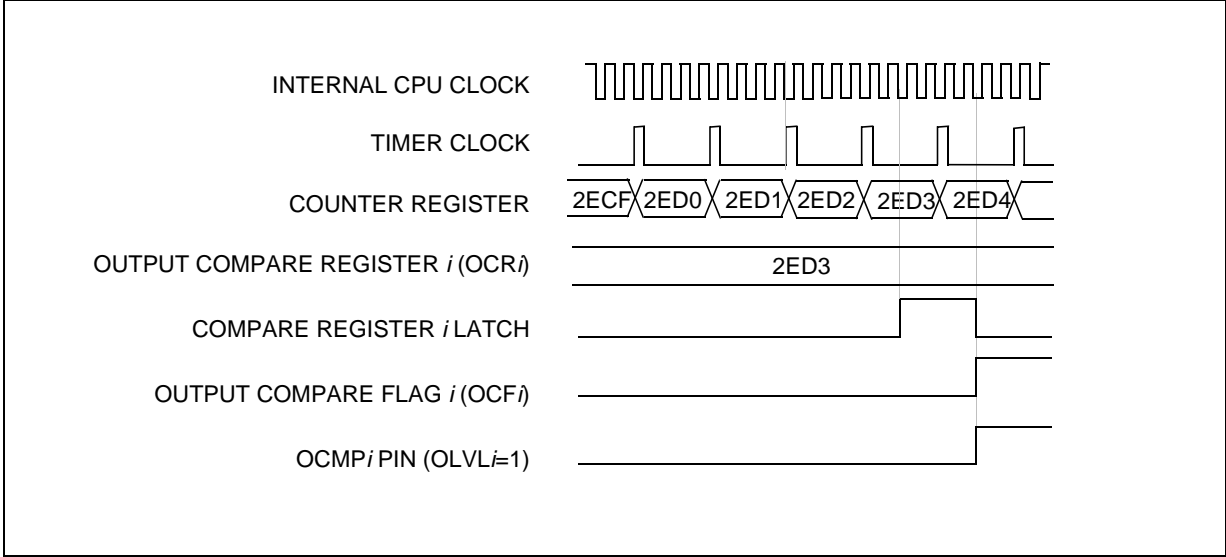


Figure 50. Output Compare Timing Diagram,  $f_{\text{TIMER}} = f_{\text{CPU}}/4$



**SERIAL PERIPHERAL INTERFACE (Cont'd)****10.5.3.2 Slave Select Management**

As an alternative to using the  $\overline{SS}$  pin to control the Slave Select signal, the application can choose to manage the Slave Select signal by software. This is configured by the SSM bit in the SPICSR register (see [Figure 56](#))

In software management, the external  $\overline{SS}$  pin is free for other application uses and the internal  $\overline{SS}$  signal level is driven by writing to the SSI bit in the SPICSR register.

**In Master mode:**

- $\overline{SS}$  internal must be held high continuously

**In Slave Mode:**

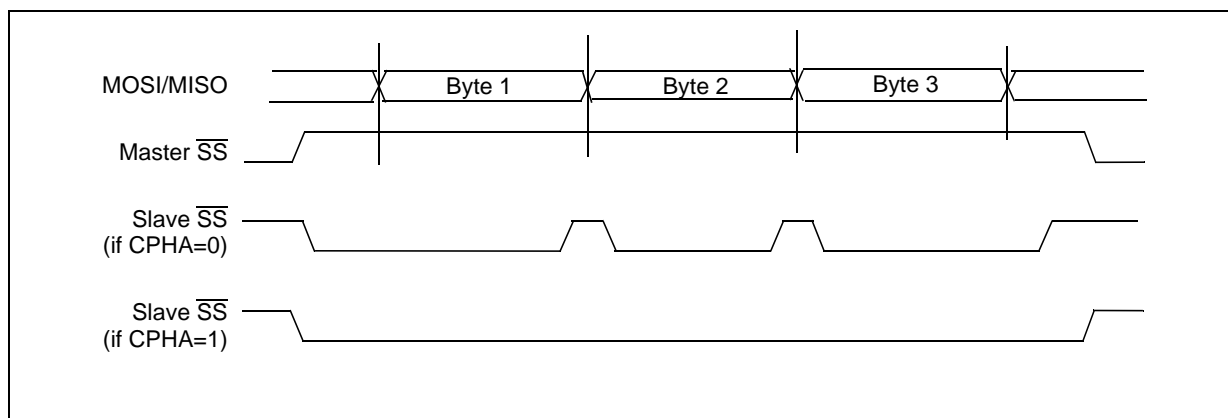
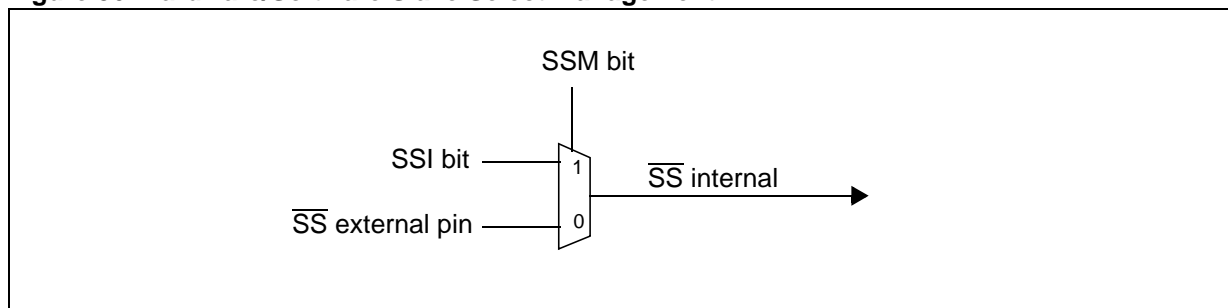
There are two cases depending on the data/clock timing relationship (see [Figure 55](#)):

If CPHA=1 (data latched on 2nd clock edge):

- $\overline{SS}$  internal must be held low during the entire transmission. This implies that in single slave applications the  $\overline{SS}$  pin either can be tied to  $V_{SS}$ , or made free for standard I/O by managing the  $\overline{SS}$  function by software (SSM= 1 and SSI=0 in the in the SPICSR register)

If CPHA=0 (data latched on 1st clock edge):

- $\overline{SS}$  internal must be held low during byte transmission and pulled high between each byte to allow the slave to write to the shift register. If  $\overline{SS}$  is not pulled high, a Write Collision error will occur when the slave writes to the shift register (see [Section 10.5.5.3](#)).

**Figure 55. Generic  $\overline{SS}$  Timing Diagram****Figure 56. Hardware/Software Slave Select Management**

## SERIAL PERIPHERAL INTERFACE (Cont'd)

### 10.5.4 Clock Phase and Clock Polarity

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits (See Figure 57).

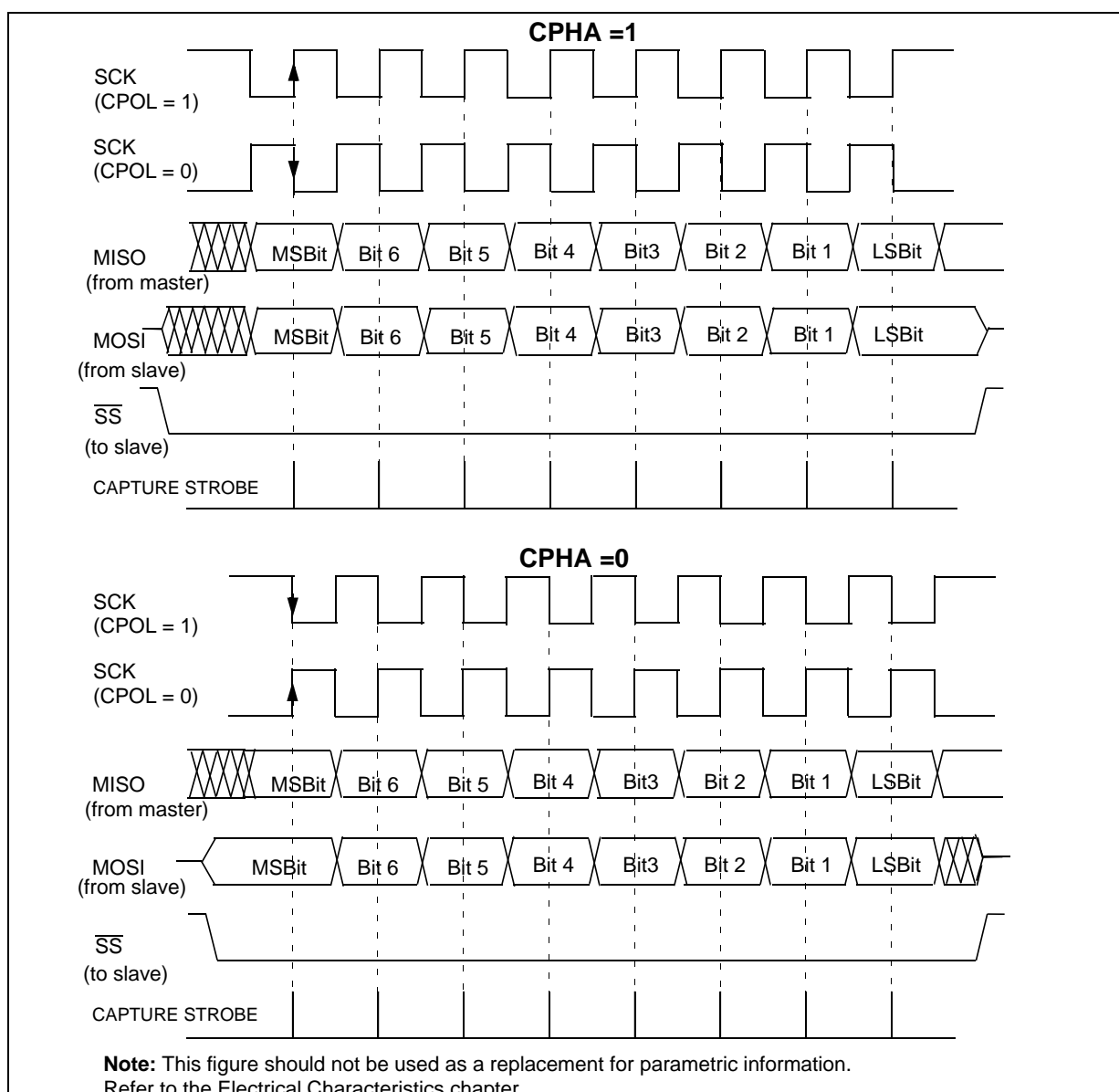
**Note:** The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL=1 or pulling down SCK if CPOL=0).

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

Figure 57, shows an SPI transfer with the four combinations of the CPHA and CPOL bits. The diagram may be interpreted as a master or slave timing diagram where the SCK pin, the MISO pin, the MOSI pin are directly connected between the master and the slave device.

**Note:** If CPOL is changed at the communication byte boundaries, the SPI must be disabled by re-setting the SPE bit.

Figure 57. Data Clock Timing Diagram





**SERIAL PERIPHERAL INTERFACE (Cont'd)****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 58](#)).

0: No write collision occurred

1: A write collision has been detected

Bit 5 = **OVR** *SPI 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 10.5.5.2](#)). An interrupt is generated if SPIE = 1 in 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 10.5.5.1 Master Mode Fault \(MODF\)](#)). An SPI interrupt can be generated if SPIE=1 in the SPICSR register. This bit is cleared by a software sequence (An access to the SPICR 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 10.5.3.2 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

**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 53](#)).

SERIAL COMMUNICATIONS INTERFACE (Cont'd)

10.6.4 Functional Description

The block diagram of the Serial Control Interface, is shown in [Figure 60](#). It contains 6 dedicated registers:

- Two control registers (SCICR1 & SCICR2)
- A status register (SCISR)
- A baud rate register (SCIBRR)
- An extended prescaler receiver register (SCIERP)
- An extended prescaler transmitter register (SCIETPR)

Refer to the register descriptions in [Section 10.6.7](#) for the definitions of each bit.

10.6.4.1 Serial Data Format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the SCICR1 register (see [Figure 60](#)).

The TDO pin is in low state during the start bit.

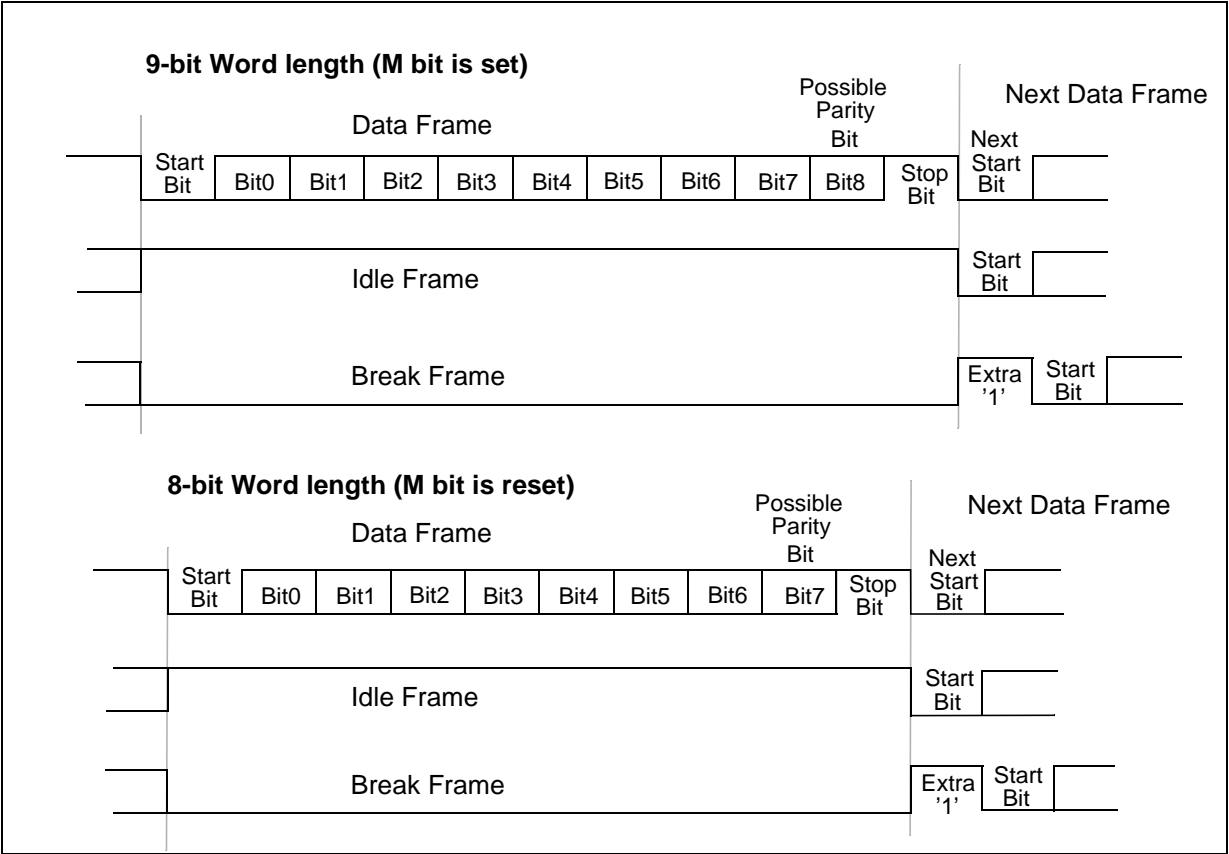
The TDO pin is in high state during the stop bit.

An Idle character is interpreted as an entire frame of “1”s followed by the start bit of the next frame which contains data.

A Break character is interpreted on receiving “0”s for some multiple of the frame period. At the end of the last break frame the transmitter inserts an extra “1” bit to acknowledge the start bit.

Transmission and reception are driven by their own baud rate generator.

Figure 61. Word Length Programming



**I<sup>2</sup>C BUS INTERFACE (Cont'd)****I<sup>2</sup>C OWN ADDRESS REGISTER (OAR1)**

Read / Write

Reset Value: 0000 0000 (00h)

7							0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0

**7-bit Addressing Mode**Bit 7:1 = **ADD[7:1]** *Interface address.*

These bits define the I<sup>2</sup>C bus address of the interface. They are not cleared when the interface is disabled (PE=0).

Bit 0 = **ADD0** *Address direction bit.*

This bit is don't care, the interface acknowledges either 0 or 1. It is not cleared when the interface is disabled (PE=0).

Note: Address 01h is always ignored.

**10-bit Addressing Mode**Bit 7:0 = **ADD[7:0]** *Interface address.*

These are the least significant bits of the I<sup>2</sup>C bus address of the interface. They are not cleared when the interface is disabled (PE=0).

**I<sup>2</sup>C OWN ADDRESS REGISTER (OAR2)**

Read / Write

Reset Value: 0100 0000 (40h)

7							0
FR1	FR0	0	0	0	ADD9	ADD8	0

Bit 7:6 = **FR[1:0]** *Frequency bits.*

These bits are set by software only when the interface is disabled (PE=0). To configure the interface to I<sup>2</sup>C specified delays select the value corresponding to the microcontroller frequency F<sub>CPU</sub>.

f <sub>CPU</sub>	FR1	FR0
< 6 MHz	0	0
6 to 8 MHz	0	1

Bit 5:3 = Reserved

Bit 2:1 = **ADD[9:8]** *Interface address.*

These are the most significant bits of the I<sup>2</sup>C bus address of the interface (10-bit mode only). They are not cleared when the interface is disabled (PE=0).

Bit 0 = Reserved.

## CONTROLLER AREA NETWORK (Cont'd)

## 10.8.3.3 Modes of Operation

The CAN Core unit assumes one of the seven states described below:

- **STANDBY.** Standby mode is entered either on a chip reset or on resetting the RUN bit in the Control/Status Register (CSR). Any on-going transmission or reception operation is not interrupted and completes normally before the Bit Time Logic and the clock prescaler are turned off for minimum power consumption. This state is signalled by the RUN bit being read-back as 0. Once in standby, the only event monitored is the reception of a dominant bit which causes a wake-up interrupt if the SCIE bit of the Interrupt Control Register (ICR) is set.

The STANDBY mode is left by setting the RUN

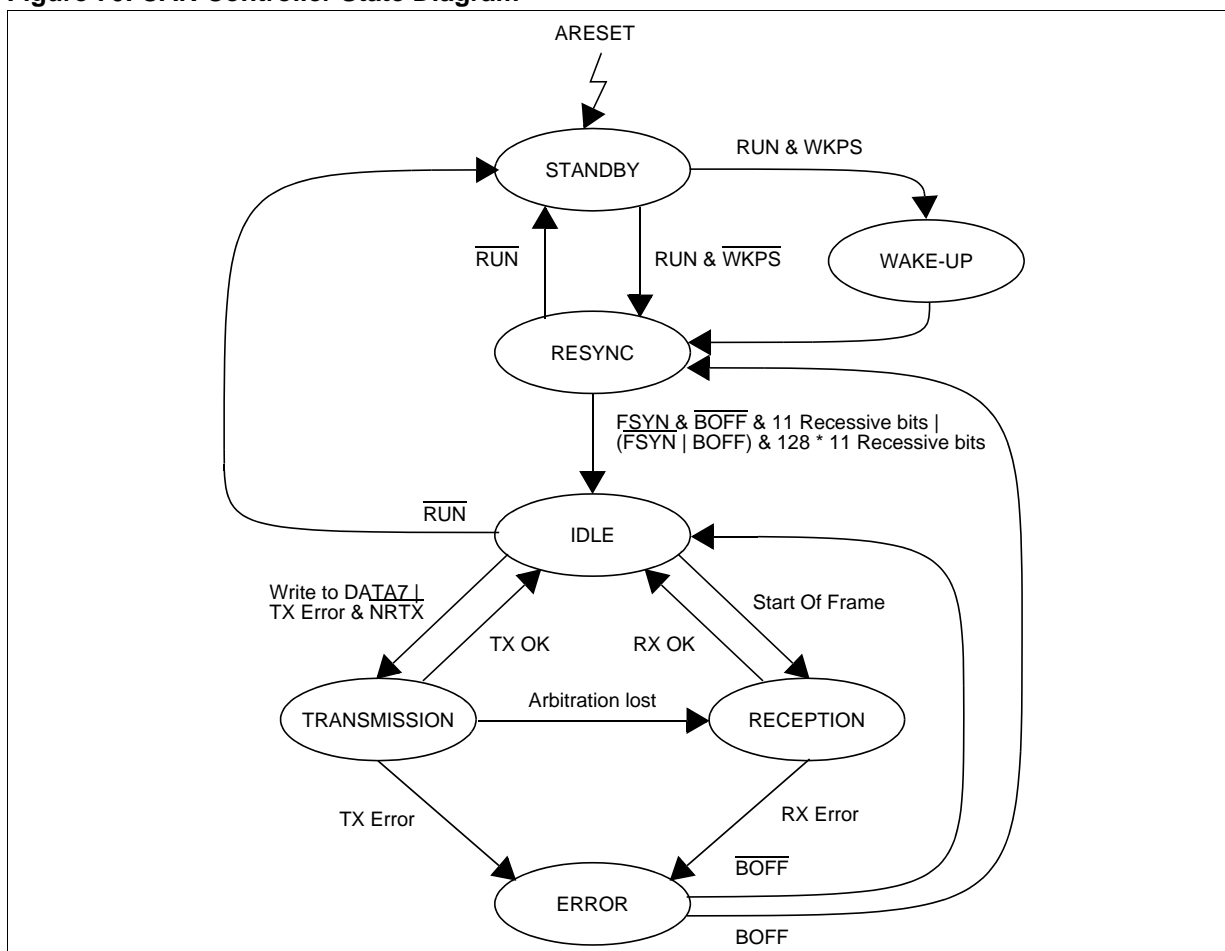
bit. If the WKPS bit is set in the CSR register, then the controller passes through WAKE-UP otherwise it enters RESYNC directly.

It is important to note that the wake-up mechanism is software-driven and therefore carries a significant time overhead. All messages received after the wake-up bit and before the controller is set to run and has completed synchronization are ignored.

**Note:** Standby mode is not entered on resetting the RUN bit in the Control/Status register (CSR) if the CANRX pin is shorted to GND.

- **WAKE-UP.** The CAN bus line is forced to dominant for one bit time signalling the wake-up condition to all other bus members.

Figure 70. CAN Controller State Diagram



**CONTROLLER AREA NETWORK (Cont'd)****IDENTIFIER LOW REGISTERS (IDLRx)**

Read/Write

Reset Value: Undefined

7							0
ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0

**ID[2:0]** are the least significant 3 bits of the 11-bit message identifier.

**RTR** is the Remote Transmission Request bit. It is set to indicate a remote frame and reset to indicate a data frame.

**DLC[3:0]** is the Data Length Code. It gives the number of bytes in the data field of the message. The valid range is 0 to 8.

**DATA REGISTERS (DATA0-7x)**

Read/Write

Reset Value: Undefined

7							0
DATA 7	DATA 6	DATA 5	DATA 4	DATA 3	DATA 2	DATA 1	DATA 0

**DATA[7:0]** is a message data byte. Up to eight such bytes may be part of a message. Writing to byte DATA7 initiates a transmit request and should always be done even when DATA7 is not part of the message.

**BUFFER CONTROL/STATUS REGs. (BCSRx)**

Read/Write

Reset Value: 00h

7							0
0	0	0	0	ACC	RDY	BUSY	LOCK

Bit 3 = **ACC** *Acceptance Code*

— Read Only

Set by hardware with the id of the highest priority filter which accepted the message stored in the buffer.

ACC = 0: Match for Filter/Mask0. Possible match for Filter/Mask1.

ACC = 1: No match for Filter/Mask0 and match for Filter/Mask1.

Reset by hardware when either RDY or RXIF gets reset.

Bit 2 = **RDY** *Message Ready*

— Read/Clear

Set by hardware to signal that a new error-free message is available (LOCK = 0) or that a transmission request is pending (LOCK = 1).

Cleared by software when LOCK = 0 to release the buffer and to clear the corresponding RXIF bit in the Interrupt Status Register.

Cleared by hardware when LOCK = 1 to indicate that the transmission request has been serviced or cancelled.

Bit 1 = **BUSY** *Busy Buffer*

— Read Only

Set by hardware when the buffer is being filled (LOCK = 0) or emptied (LOCK = 1) and reset after the 2nd intermission bit.

Reset by hardware when the buffer is not accessed by the CAN core for transmission nor reception purposes.

Bit 0 = **LOCK** *Lock Buffer*

— Read/Set/Clear

Set by software to lock a buffer. No more message can be received into the buffer thus preserving its content and making it available for transmission.

Cleared by software to make the buffer available for reception. Cancels any pending transmission request.

Cleared by hardware once a message has been successfully transmitted provided the early transmit interrupt mode is on. Left untouched otherwise.

Note that in order to prevent any message corruption or loss of context, LOCK cannot be set nor reset while BUSY is set. Trying to do so will result in LOCK not changing state.

## 10.9 10-BIT A/D CONVERTER (ADC)

### 10.9.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 10-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 16 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 16 different sources.

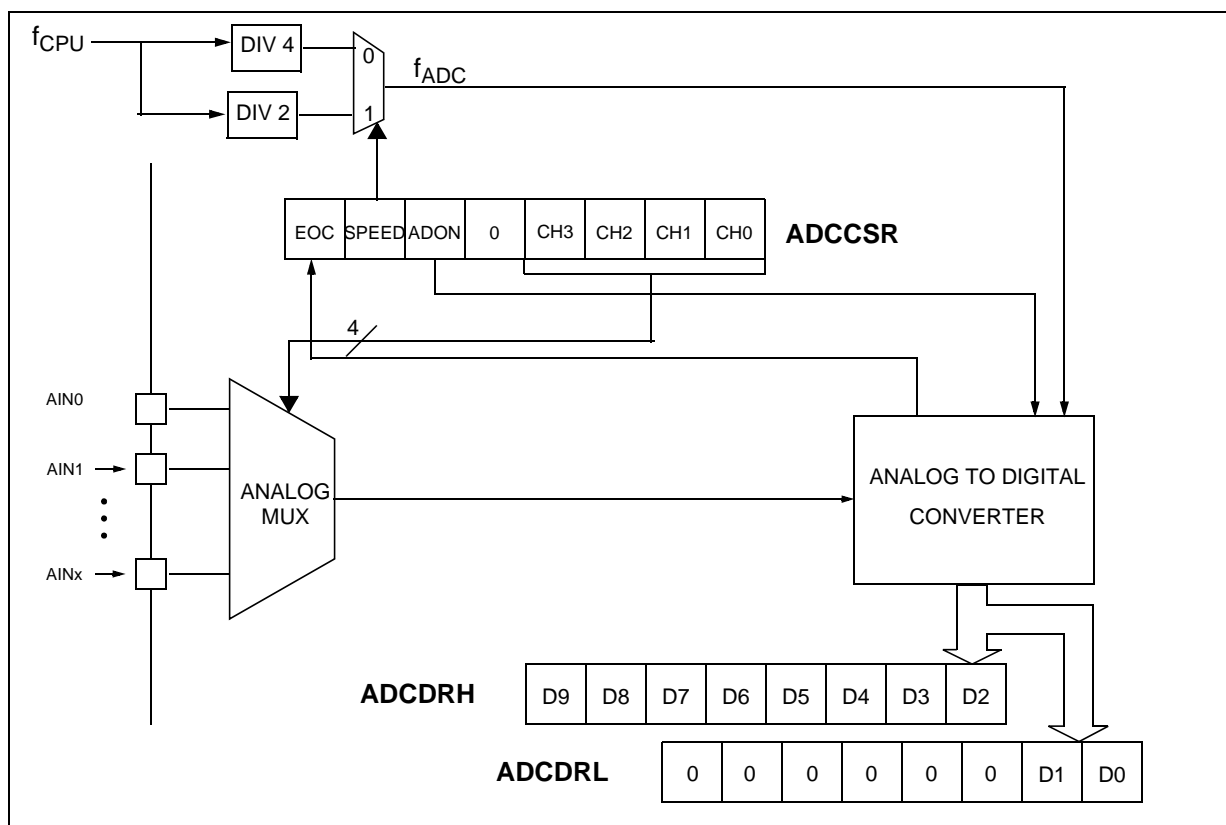
The result of the conversion is stored in a 10-bit Data Register. The A/D converter is controlled through a Control/Status Register.

### 10.9.2 Main Features

- 10-bit conversion
- Up to 16 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in [Figure 83](#).

**Figure 83. ADC Block Diagram**



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

### 10.9.3 Functional Description

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  $V_{AREF}$  (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 allotted time.

#### 10.9.3.1 A/D Converter Configuration

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 CS[3:0] bits to assign the analog channel to convert.

#### 10.9.3.2 Starting the Conversion

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 resets the EOC bit.

To read the 10 bits, perform the following steps:

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

**Note:** The data is not latched, so both the low and the high data register must be read before the next conversion is complete, so it is recommended to disable interrupts while reading the conversion result.

To read only 8 bits, perform the following steps:

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

#### 10.9.3.3 Changing the conversion channel

The application can change channels during conversion. When software modifies the CH[3:0] 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.

#### 10.9.4 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.

#### 10.9.5 Interrupts

None.

**SUPPLY CURRENT CHARACTERISTICS** (Cont'd)**12.4.2 Supply and Clock Managers**

The previous current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To get the total device consumption, the two current values must be added (except for HALT mode).

Symbol	Parameter	Conditions	Typ	Max	Unit
I <sub>DD(RCINT)</sub>	Supply current of internal RC oscillator		625		μA
I <sub>DD(RES)</sub>	Supply current of resonator oscillator <sup>1)</sup> & <sup>2)</sup>		see <a href="#">section 12.5.3 on page 174</a>		
I <sub>DD(PLL)</sub>	PLL supply current	V <sub>DD</sub> = 5V	360		
I <sub>DD(LVD)</sub>	LVD supply current	V <sub>DD</sub> = 5V	150	300	

**Notes:**

- 1.. Data based on characterization results done with the external components specified in [Section 12.5.3](#), not tested in production.
2. As the oscillator is based on a current source, the consumption does not depend on the voltage.



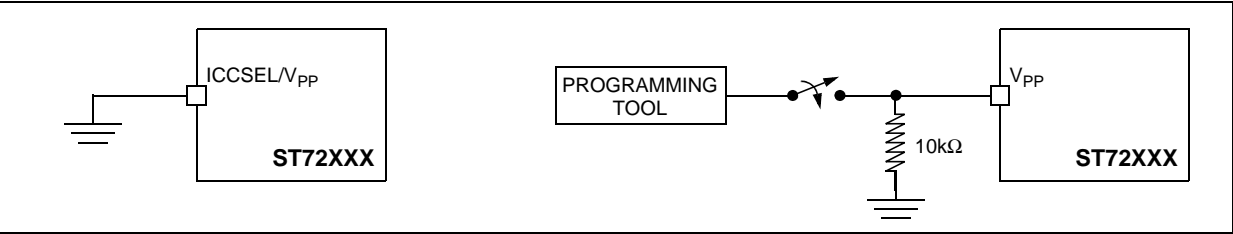
CONTROL PIN CHARACTERISTICS (Cont'd)

12.9.2 ICCSEL/V<sub>PP</sub> Pin

Subject to general operating conditions for V<sub>DD</sub>, f<sub>CPU</sub>, and T<sub>A</sub> unless otherwise specified.

Symbol	Parameter	Conditions	Min	Max	Unit
V <sub>IL</sub>	Input low level voltage <sup>1)</sup>	FLASH versions	V <sub>SS</sub>	0.2	V
		ROM versions	V <sub>SS</sub>	0.3xV <sub>DD</sub>	
V <sub>IH</sub>	Input high level voltage <sup>1)</sup>	FLASH versions	V <sub>DD</sub> -0.1	12.6	
		ROM versions	0.7xV <sub>DD</sub>	V <sub>DD</sub>	
I <sub>L</sub>	Input leakage current	V <sub>IN</sub> =V <sub>SS</sub>		±1	µA

Figure 105. Two typical Applications with ICCSEL/V<sub>PP</sub> Pin <sup>2)</sup>



Notes:

- 1. Data based on design simulation and/or technology characteristics, not tested in production.
- 2. When ICC mode is not required by the application ICCSEL/V<sub>PP</sub> pin must be tied to V<sub>SS</sub>.

**ST72521 DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)**

OPT2 = Reserved, must be kept at default value.

OPT1= **PKG0** *Package selection bit 0*

This option bit is used to select the package (see table in PKG1 option bit description).

OPT0= **FMP\_R** *Flash memory read-out protection*  
Read-out protection, when selected, provides a protection against Program Memory content extraction and against write access to Flash memory.

Erasing the option bytes when the FMP\_R option is selected causes the whole user memory to be erased first, and the device can be reprogrammed. Refer to [Section 4.3.1](#) and the ST7 Flash Programming Reference Manual for more details.

**Note:** Readout protection is not supported if LVD is enabled.

0: Read-out protection enabled

1: Read-out protection disabled

**OPTION BYTE 1**

OPT7= **PKG1** *Package selection bit 1*

This option bit, with the PKG0 bit, selects the package.

Version	Selected Package	PKG 1	PKG 0
M	TQFP80	1	1
(A)R	TQFP64	1	0

**Note:** On the chip, each I/O port has 8 pads. Pads that are not bonded to external pins are in input pull-up configuration after reset. The configuration of these pads must be kept at reset state to avoid added current consumption.

OPT6 = **RSTC** *RESET clock cycle selection*

This option bit selects the number of CPU cycles applied during the RESET phase and when exiting HALT mode. For resonator oscillators, it is advised to select 4096 due to the long crystal stabilization time.

0: Reset phase with 4096 CPU cycles

1: Reset phase with 256 CPU cycles

OPT5:4 = **OSCTYPE[1:0]** *Oscillator Type*

These option bits select the ST7 main clock source type.

Clock Source	OSCTYPE	
	1	0
Resonator Oscillator	0	0
Reserved	0	1
Internal RC Oscillator	1	0
External Source	1	1

OPT3:1 = **OSCRANGE[2:0]** *Oscillator range*

When the resonator oscillator type is selected, these option bits select the resonator oscillator current source corresponding to the frequency range of the used resonator. Otherwise, these bits are used to select the normal operating frequency range.

Typ. Freq. Range		OSCRANGE		
		2	1	0
LP	1~2MHz	0	0	0
MP	2~4MHz	0	0	1
MS	4~8MHz	0	1	0
HS	8~16MHz	0	1	1

OPT0 = **PLLOFF** *PLL activation*

This option bit activates the PLL which allows multiplication by two of the main input clock frequency. The PLL must not be used with the internal RC oscillator or with external clock source. The PLL is guaranteed only with an input frequency between 2 and 4MHz.

0: PLL x2 enabled

1: PLL x2 disabled

**CAUTION:** the PLL can be enabled only if the "OSC RANGE" (OPT3:1) bits are configured to "MP - 2~4MHz". Otherwise, the device functionality is not guaranteed.

## DEVICE CONFIGURATION AND ORDERING INFORMATION (Cont'd)

Table 30. Orderable Flash Device Types

Part Number	Version	Package	Flash Memory (Kbytes)	Temp. Range
ST72F521AR6TC	Automotive	TQFP64 10 x 10	32	-40°C +125°C
ST72F521AR9TC			60	
ST72F521R6TC		TQFP64 14 x 14	32	
ST72F521R9TC			60	
ST72F521M9TC		TQFP80	60	
ST72F521AR6T3	Standard	TQFP64 10 x 10	32	-40°C +125°C
ST72F521AR9T3			60	
ST72F521R6T3		TQFP64 14 x 14	32	
ST72F521R9T3			60	
ST72F521M9T3		TQFP80	60	
ST72F521AR6T6	Standard	TQFP64 10 x 10	32	-40°C +85°C
ST72F521AR9T6			60	
ST72F521R6T6		TQFP64 14 x 14	32	
ST72F521R9T6			60	
ST72F521M9T6		TQFP80	60	

Table 33. ST7 Application Notes

IDENTIFICATION	DESCRIPTION
<b>GENERAL PURPOSE</b>	
AN1476	LOW COST POWER SUPPLY FOR HOME APPLIANCES
AN1526	ST7FLITE0 QUICK REFERENCE NOTE
AN1709	EMC DESIGN FOR ST MICROCONTROLLERS
AN1752	ST72324 QUICK REFERENCE NOTE
<b>PRODUCT EVALUATION</b>	
AN 910	PERFORMANCE BENCHMARKING
AN 990	ST7 BENEFITS VERSUS INDUSTRY STANDARD
AN1077	OVERVIEW OF ENHANCED CAN CONTROLLERS FOR ST7 AND ST9 MCUS
AN1086	U435 CAN-DO SOLUTIONS FOR CAR MULTIPLEXING
AN1103	IMPROVED B-EMF DETECTION FOR LOW SPEED, LOW VOLTAGE WITH ST72141
AN1150	BENCHMARK ST72 VS PC16
AN1151	PERFORMANCE COMPARISON BETWEEN ST72254 & PC16F876
AN1278	LIN (LOCAL INTERCONNECT NETWORK) SOLUTIONS
<b>PRODUCT MIGRATION</b>	
AN1131	MIGRATING APPLICATIONS FROM ST72511/311/214/124 TO ST72521/321/324
AN1322	MIGRATING AN APPLICATION FROM ST7263 REV.B TO ST7263B
AN1365	GUIDELINES FOR MIGRATING ST72C254 APPLICATIONS TO ST72F264
AN1604	HOW TO USE ST7MDT1-TRAIN WITH ST72F264
<b>PRODUCT OPTIMIZATION</b>	
AN 982	USING ST7 WITH CERAMIC RESONATOR
AN1014	HOW TO MINIMIZE THE ST7 POWER CONSUMPTION
AN1015	SOFTWARE TECHNIQUES FOR IMPROVING MICROCONTROLLER EMC PERFORMANCE
AN1040	MONITORING THE VBUS SIGNAL FOR USB SELF-POWERED DEVICES
AN1070	ST7 CHECKSUM SELF-CHECKING CAPABILITY
AN1181	ELECTROSTATIC DISCHARGE SENSITIVE MEASUREMENT
AN1324	CALIBRATING THE RC OSCILLATOR OF THE ST7FLITE0 MCU USING THE MAINS
AN1502	EMULATED DATA EEPROM WITH ST7 HDFLASH MEMORY
AN1529	EXTENDING THE CURRENT & VOLTAGE CAPABILITY ON THE ST7265 VDDF SUPPLY
AN1530	ACCURATE TIMEBASE FOR LOW-COST ST7 APPLICATIONS WITH INTERNAL RC OSCILLATOR
AN1605	USING AN ACTIVE RC TO WAKEUP THE ST7LITE0 FROM POWER SAVING MODE
AN1636	UNDERSTANDING AND MINIMIZING ADC CONVERSION ERRORS
AN1828	PIR (PASSIVE INFRARED) DETECTOR USING THE ST7FLITE05/09/SUPERLITE
AN1946	SENSORLESS BLDC MOTOR CONTROL AND BEMF SAMPLING METHODS WITH ST7MC
AN1971	ST7LITE0 MICROCONTROLLED BALLAST
<b>PROGRAMMING AND TOOLS</b>	
AN 978	ST7 VISUAL DEVELOP SOFTWARE KEY DEBUGGING FEATURES
AN 983	KEY FEATURES OF THE COSMIC ST7 C-COMPILER PACKAGE
AN 985	EXECUTING CODE IN ST7 RAM
AN 986	USING THE INDIRECT ADDRESSING MODE WITH ST7
AN 987	ST7 SERIAL TEST CONTROLLER PROGRAMMING
AN 988	STARTING WITH ST7 ASSEMBLY TOOL CHAIN
AN 989	GETTING STARTED WITH THE ST7 HIWARE C TOOLCHAIN
AN1039	ST7 MATH UTILITY ROUTINES
AN1064	WRITING OPTIMIZED HIWARE C LANGUAGE FOR ST7
AN1071	HALF DUPLEX USB-TO-SERIAL BRIDGE USING THE ST72611 USB MICROCONTROLLER

## KNOWN LIMITATIONS (Cont'd)

### 15.2 ALL FLASH DEVICES

#### 15.2.1 Internal RC Oscillator with LVD

The internal RC can only be used if LVD is enabled.

#### 15.2.2 I/O behaviour during ICC mode entry sequence

##### Symptom

In 80-pin devices (Flash), both Port G and H are forced to output push-pull during ICC mode entry sequence. 80-pin ROM devices are not impacted by this issue.

##### Details

To enable programming of all flash sectors, the device must leave USER mode and be configured in ICC mode. Once in ICC mode, the ICC protocol enables an ST7 microcontroller to communicate with an external controller (such as a PC). ICC mode is entered by applying 39 pulses on the IC-CDATA signal during reset. To enter ICC mode, the device goes through other modes, some

modes are critical because the I/Os PG[7:0] and PH[7:0] are forced to output push-pull.

##### Impact on the Application

The PG and PH I/O ports are forced to output push-pull during three pulses on ICCDATA. In certain circumstances, this behaviour can lead to a short-circuit between the I/O signals and  $V_{DD}$ ,  $V_{SS}$  or an output signal of another application component.

In addition, switching these I/Os to output mode can cause the application to leave reset state, disturbing the ICC communication and preventing the user from programming the flash.

#### 15.2.3 Read-out protection with LVD

The LVD is not supported if Readout protection is enabled.