STMicroelectronics - ST72F521R9TC Datasheet

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, I²C, 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	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f521r9tc

Table 4. Hardware register map (continued)

Address	Block	Register label	Register name	Reset status	Remarks
0070h 0071h 0072h	ADC	ADCCSR ADCDRH ADCDRL	Control/Status Register Data High Register Data Low Register	00h 00h 00h	R/W Read Only Read Only
0073h 0074h 0075h 0076h 0077h 0078h 0079h 007Ah 007Bh 007Ch 007Dh	PWM ART	PWMDCR3 PWMDCR2 PWMDCR1 PWMDCR0 PWMCR ARTCSR ARTCAR ARTARR ARTICCSR ARTICR1 ARTICR2	PWM AR Timer Duty Cycle Register 3 PWM AR Timer Duty Cycle Register 2 PWM AR Timer Duty Cycle Register 1 PWM AR Timer Duty Cycle Register 0 PWM AR Timer Control Register Auto-Reload Timer Control/Status Register Auto-Reload Timer Counter Access Register Auto-Reload Timer Auto-Reload Register AR Timer Input Capture Control/Status Reg. AR Timer Input Capture Register 1 AR Timer Input Capture Register 1	00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h	R/W R/W R/W R/W R/W R/W R/W R/W R/W Read Only Read Only
007Eh 007Fh	Reserved Area (2 bytes)				

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.

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

Halt mode recommendations

- Make sure that an external event is available to wake up the microcontroller from Halt mode.
- When using an external interrupt to wake up the microcontroller, re-initialize the corresponding I/O as “Input Pull-up with Interrupt” before executing the HALT instruction. The main reason for this is that the I/O may be wrongly configured due to external interference or by an unforeseen logical condition.
- For the same reason, reinitialize the level sensitiveness of each external interrupt as a precautionary measure.
- The opcode for the HALT instruction is 0x8E. To avoid an unexpected HALT instruction due to a program counter failure, it is advised to clear all occurrences of the data value 0x8E from memory. For example, avoid defining a constant in ROM with the value 0x8E.
- As the HALT instruction clears the interrupt mask in the CC register to allow interrupts, the user may choose to clear all pending interrupt bits before executing the HALT instruction. This avoids entering other peripheral interrupt routines after executing the external interrupt routine corresponding to the wake-up event (reset or external interrupt).

Related documentation

ST7 Keypad Decoding Techniques, Implementing Wake-Up on Keystroke (AN 980)

How to Minimize the ST7 Power Consumption (AN1014)

Using an active RC to wake up the ST7LITE0 from power saving mode (AN1605)

Figure 33. Exact timeout duration (t_{min} and t_{max})

WHERE:

t_{min0} = (LSB + 128) x 64 x t_{OSC2}
t_{max0} = 16384 x t_{OSC2}
t_{OSC2} = 125ns if f_{OSC2}= 8 MHz

CNT = Value of T[5:0] bits in the WDGCR register (6 bits)
MSB and LSB are values from the table below depending on the timebase selected by the TB[1:0] bits in the MCCSR register

TB1 bit (MCCSR reg.)	TB0 bit (MCCSR reg.)	Selected MCCSR timebase	MSB	LSB
0	0	2ms	4	59
0	1	4ms	8	53
1	0	10ms	20	35
1	1	25ms	49	54

To calculate the minimum Watchdog Timeout (t_{min}):

IF $\mathrm{CNT} < \left[\frac{\mathrm{MSB}}{4}\right]$ **THEN** $t_{min} = t_{min0} + 16384 \times \mathrm{CNT} \times t_{osc2}$

ELSE $t_{min} = t_{min0} + \left[16384 \times \left(\mathrm{CNT} - \left[\frac{4\mathrm{CNT}}{\mathrm{MSB}}\right]\right) + (192 + \mathrm{LSB}) \times 64 \times \left[\frac{4\mathrm{CNT}}{\mathrm{MSB}}\right]\right] \times t_{osc2}$

To calculate the maximum Watchdog Timeout (t_{max}):

IF $\mathrm{CNT} \leq \left[\frac{\mathrm{MSB}}{4}\right]$ **THEN** $t_{max} = t_{max0} + 16384 \times \mathrm{CNT} \times t_{osc2}$

ELSE $t_{max} = t_{max0} + \left[16384 \times \left(\mathrm{CNT} - \left[\frac{4\mathrm{CNT}}{\mathrm{MSB}}\right]\right) + (192 + \mathrm{LSB}) \times 64 \times \left[\frac{4\mathrm{CNT}}{\mathrm{MSB}}\right]\right] \times t_{osc2}$

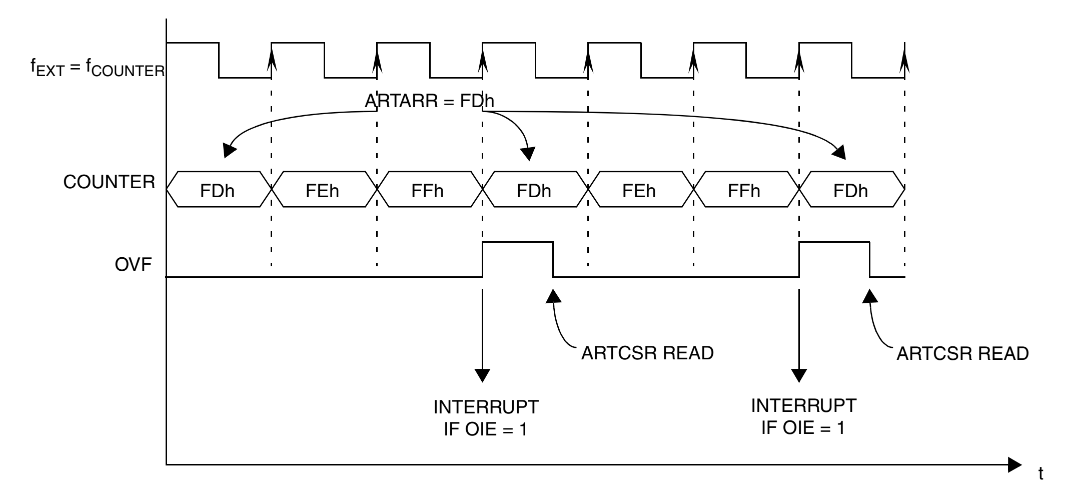
Note: In the above formulae, division results must be rounded down to the next integer value.

Example:

With 2ms timeout selected in MCCSR register

Value of T[5:0] bits in WDGCR register (Hex.)	Min. Watchdog Timeout (ms) t_{min}	Max. Watchdog Timeout (ms) t_{max}
00	1.496	2.048
3F	128	128.552

Figure 39. External event detector example (3 counts)



12.2.8 Input capture function

This mode allows the measurement of external signal pulse widths through ARTICRx registers.

Each input capture can generate an interrupt independently on a selected input signal transition. This event is flagged by a set of the corresponding CFx bits of the Input Capture Control/Status register (ARTICCSR).

These input capture interrupts are enabled through the CIEx bits of the ARTICCSR register.

The active transition (falling or rising edge) is software programmable through the CSx bits of the ARTICCSR register.

The read only input capture registers (ARTICRx) are used to latch the auto-reload counter value when a transition is detected on the ARTICx pin (CFx bit set in ARTICCSR register). After fetching the interrupt vector, the CFx flags can be read to identify the interrupt source.

Note: After a capture detection, data transfer in the ARTICRx register is inhibited until it is read (clearing the CFx bit).

The timer interrupt remains pending while the CFx flag is set when the interrupt is enabled (CIEx bit set). This means that the ARTICRx register has to be read at each capture event to clear the CFx flag.

The timing resolution is given by auto-reload counter cycle time ($1/f_{COUNTER}$).

Note: During Halt mode, if both the input capture and the external clock are enabled, the ARTICRx register value is not guaranteed if the input capture pin and the external clock change simultaneously.

If the timer clock is an external clock, the formula is:

$$\Delta \text{ OC}i\text{R} = \Delta t * f_{EXT}$$

Where:

Δt = Output compare period (in seconds)

f_{CPU} = External timer clock frequency (in hertz)

Clearing the output compare interrupt request (that is, clearing the OCF*i* bit) is done by:

1. Reading the SR register while the OCF*i* bit is set
2. An access (read or write) to the OC*i*LR register

The following procedure is recommended to prevent the OCF*i* bit from being set between the time it is read and the write to the OC*i*R register:

- Write to the OC*i*HR register (further compares are inhibited).
- Read the SR register (first step of the clearance of the OCF*i* bit, which may be already set).
- Write to the OC*i*LR register (enables the output compare function and clears the OCF*i* bit).

Note: 1 After a processor write cycle to the OCiHR register, the output compare function is inhibited until the OCiLR register is also written.

2 If the OCiE bit is not set, the OCMPi pin is a general I/O port and the OLVLi bit will not appear when a match is found but an interrupt could be generated if the OCIE bit is set.

3 In both internal and external clock modes, OCFi and OCMPi are set while the counter value equals the OCiR register value (see Figure 49 on page 110 for an example with $f_{CPU}/2$ *and Figure 50 on page 110 for an example with* $f_{CPU}/4$*). This behavior is the same in OPM or PWM mode.*

4 The output compare functions can be used both for generating external events on the OCMPi pins even if the input capture mode is also used.

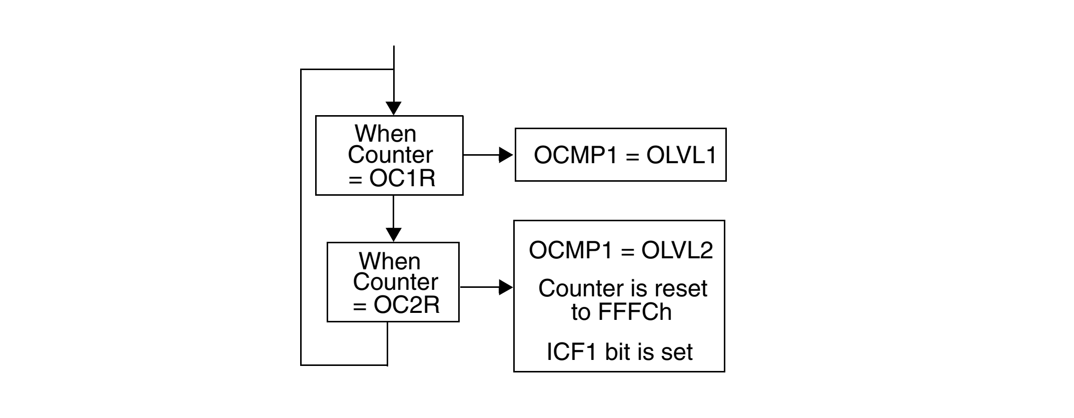
5 The value in the 16-bit OCiR register and the OLVi bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

13.3.5 Forced compare output capability

When the FOLV*i* bit is set by software, the OLVL*i* bit is copied to the OCMP*i* pin. The OLV*i* bit has to be toggled in order to toggle the OCMP*i* pin when it is enabled (OC*i*E bit = 1). The OCF*i* bit is then not set by hardware, and thus no interrupt request is generated.

The FOLVL*i* bits have no effect in both one pulse mode and PWM mode.

Figure 54. Pulse width modulation cycle flowchart



If OLVL1 = 1 and OLVL2 = 0 the length of the positive pulse is the difference between the OC2R and OC1R registers.

If OLVL1 = OLVL2 a continuous signal will be seen on the OCMP1 pin.

The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$OC\textit{i}R\ \text{value} = \frac{t \cdot f_{CPU} - 5}{PRESC}$$

Where:

t = Signal or pulse period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits; see *Table 61: Timer clock selection*)

If the timer clock is an external clock the formula is:

$$OC\textit{i}R = t \cdot f_{EXT} - 5$$

Where:

t = Signal or pulse period (in seconds)

f_{EXT} = External timer clock frequency (in hertz)

The Output Compare 2 event causes the counter to be initialized to FFFCh (see *Figure 53*).

Note: 1 *After a write instruction to the OCiHR register, the output compare function is inhibited until the OCiLR register is also written.*

2 *The OCF1 and OCF2 bits cannot be set by hardware in PWM mode therefore the Output Compare interrupt is inhibited.*

3 *The ICF1 bit is set by hardware when the counter reaches the OC2R value and can produce a timer interrupt if the ICIE bit is set and the I bit is cleared.*

4 *In PWM mode the ICAP1 pin cannot be used to perform input capture because it is disconnected to the timer. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each period and ICF1 can also generates interrupt if ICIE is set.*

5 *When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.*

Table 60. CR2 register description (continued)

Bit	Name	Function
5	OPM	*One Pulse Mode* 0: One Pulse Mode is not active. 1: One Pulse Mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.
4	PWM	*Pulse Width Modulation* 0: PWM mode is not active. 1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.
3:2	CC[1:0]	*Clock Control* The timer clock mode depends on these bits (see *Table 61*).
1	IEDG2	*Input Edge 2* This bit determines which type of level transition on the ICAP2 pin will trigger the capture. 0: A falling edge triggers the capture. 1: A rising edge triggers the capture.
0	EXEDG	*External Clock Edge* This bit determines which type of level transition on the external clock pin EXTCLK will trigger the counter register. 0: A falling edge triggers the counter register. 1: A rising edge triggers the counter register.

Table 61. Timer clock selection

Timer clock	CC1	CC0
f_{CPU} / 4	0	0
f_{CPU} / 2	0	1
f_{CPU} / 8	1	0
External clock (where available)(1)	1	1

1. If the external clock pin is not available, programming the external clock configuration stops the counter.

13.7.3 Control/status register (CSR)

CSR Reset value: xxxx x0xx (xxh)

7	6	5	4	3	2	1	0
ICF1	OCF1	TOF	ICF2	OCF2	TIMD	Reserved	
RO	RO	RO	RO	RO	RW	-	

When an overrun error occurs:

- The OR bit is set.
- The RDR content is not lost.
- The shift register is overwritten.
- An interrupt is generated if the RIE bit is set and the I bit is cleared in the CCR register.

The OR bit is reset by an access to the SCISR register followed by a SCIDR register read operation.

Noise error

Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise. Normal data bits are considered valid if three consecutive samples (8th, 9th, 10th) have the same bit value, otherwise the NF flag is set. In the case of start bit detection, the NF flag is set on the basis of an algorithm combining both valid edge detection and three samples (8th, 9th, 10th). Therefore, to prevent the NF flag getting set during start bit reception, there should be a valid edge detection as well as three valid samples.

When noise is detected in a frame:

- The NF flag is set at the rising edge of the RDRF bit.
- Data is transferred from the Shift register to the SCIDR register.
- No interrupt is generated. However this bit rises at the same time as the RDRF bit which itself generates an interrupt.

The NF flag is reset by a SCISR register read operation followed by a SCIDR register read operation.

During reception, if a false start bit is detected (for example, 8th, 9th, 10th samples are 011, 101, 110), the frame is discarded and the receiving sequence is not started for this frame. There is no RDRF bit set for this frame and the NF flag is set internally (not accessible to the user). This NF flag is accessible along with the RDRF bit when a next valid frame is received.

Note: *If the application Start Bit is not long enough to match the above requirements, then the NF Flag may get set due to the short Start Bit. In this case, the NF flag may be ignored by the application software when the first valid byte is received.*

See also *Noise error causes on page 148*.

16.3 General description

In addition to receiving and transmitting data, this interface converts it from serial to parallel format and vice versa, using either an interrupt or polled handshake. The interrupts are enabled or disabled by software. The interface is connected to the I^{2}C bus by a data pin (SDAI) and by a clock pin (SCLI). It can be connected both with a standard I^{2}C bus and a fast I^{2}C bus. This selection is made by software.

16.3.1 Mode selection

The interface can operate in the four following modes:

- Slave transmitter/receiver
- Master transmitter/receiver

By default, it operates in slave mode.

The interface automatically switches from slave to master after it generates a START condition and from master to slave in case of arbitration loss or a STOP generation, allowing then Multimaster capability.

16.3.2 Communication flow

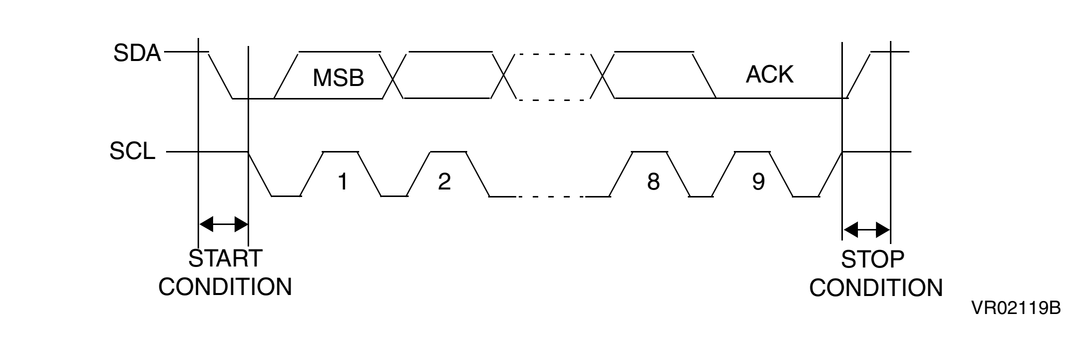
In Master mode, it initiates a data transfer and generates the clock signal. A serial data transfer always begins with a start condition and ends with a stop condition. Both start and stop conditions are generated in master mode by software.

In Slave mode, the interface is capable of recognizing its own address (7- or 10-bit), and the General Call address. The General Call address detection may be enabled or disabled by software.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the start condition contain the address (one in 7-bit mode, two in 10-bit mode). The address is always transmitted in Master mode.

A 9th clock pulse follows the 8 clock cycles of a byte transfer, during which the receiver must send an acknowledge bit to the transmitter. Refer to *Figure 66*.

Figure 66. I^{2}C bus protocol



Acknowledge may be enabled and disabled by software.

The I^{2}C interface address and/or general call address can be selected by software.

The speed of the I^{2}C interface may be selected between standard (up to 100 kHz) and fast I^{2}C (up to 400 kHz).

16.3.3 SDA/SCL line control

Transmitter mode

The interface holds the clock line low before transmission to wait for the microcontroller to write the byte in the data register.

Receiver mode

The interface holds the clock line low after reception to wait for the microcontroller to read the byte in the data register.

The SCL frequency (f_{SCL}) is controlled by a programmable clock divider which depends on the I^2C bus mode.

When the I^2C cell is enabled, the SDA and SCL ports must be configured as floating inputs. In this case, the value of the external pull-up resistor used depends on the application.

When the I^2C cell is disabled, the SDA and SCL ports revert to being standard I/O port pins.

Figure 67. I^2C interface block diagram

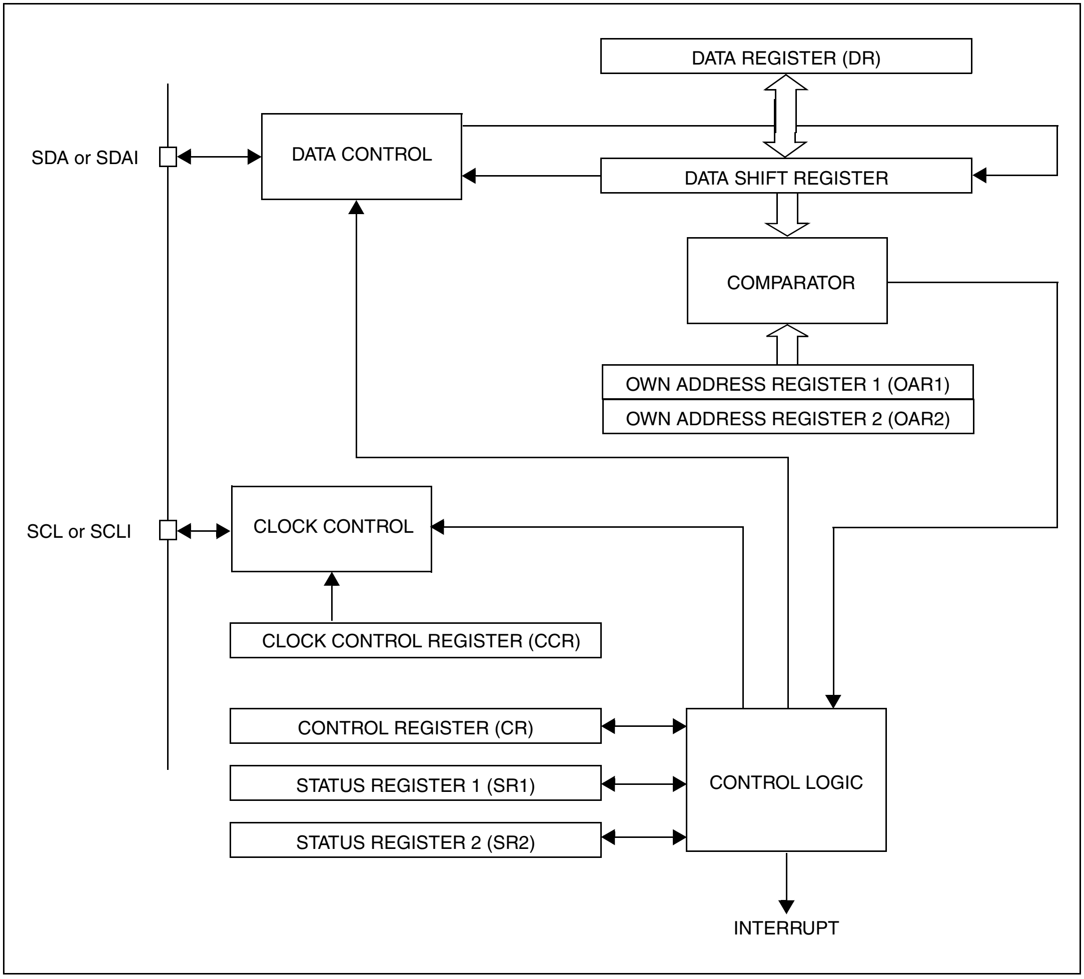
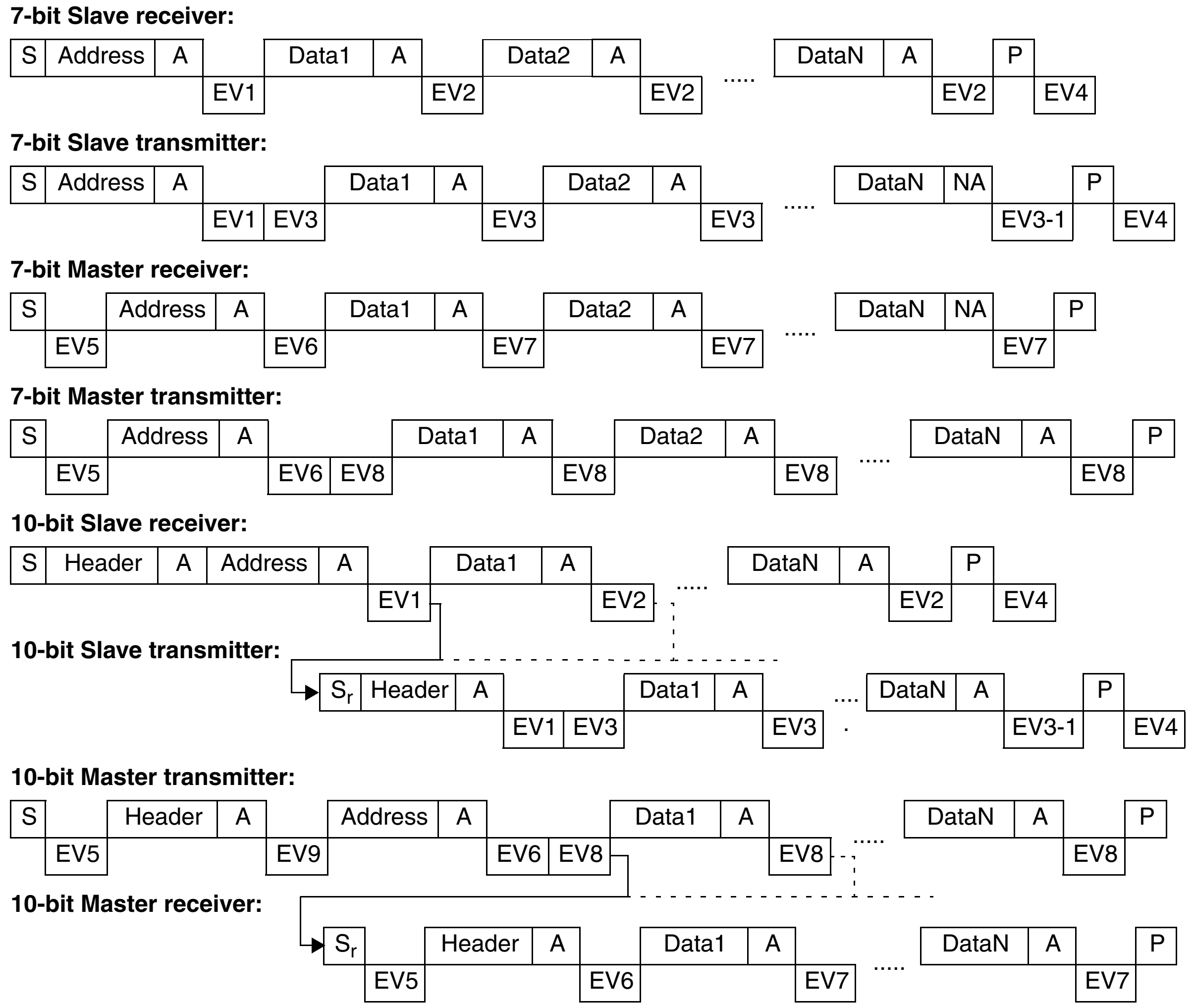


Figure 68. Transfer sequencing


7-bit Slave receiver:
S Address A Data1 A Data2 A ..... DataN A P
EV1 EV2 EV2 EV2 EV4
7-bit Slave transmitter:
S Address A Data1 A Data2 A ..... DataN NA P
EV1 EV3 EV3 EV3 EV3-1 EV4
7-bit Master receiver:
S Address A Data1 A Data2 A ..... DataN NA P
EV5 EV6 EV7 EV7 EV7
7-bit Master transmitter:
S Address A Data1 A Data2 A ..... DataN A P
EV5 EV6 EV8 EV8 EV8 EV8
10-bit Slave receiver:
S Header A Address A Data1 A ..... DataN A P
EV1 EV2 EV2 EV4
10-bit Slave transmitter:
Sr Header A Data1 A .... DataN A P
EV1 EV3 EV3 EV3-1 EV4
10-bit Master transmitter:
S Header A Address A Data1 A ..... DataN A P
EV5 EV9 EV6 EV8 EV8 EV8
10-bit Master receiver:
Sr Header A Data1 A ..... DataN A P
EV5 EV6 EV7 EV7

Legend:

S = Start, Sr = Repeated Start, P = Stop, A = Acknowledge, NA = Non-acknowledge, EVx = Event (with interrupt if ITE = 1)

EV1: EVF = 1, ADSL = 1, cleared by reading SR1 register.

EV2: EVF = 1, BTF = 1, cleared by reading SR1 register followed by reading DR register.

EV3: EVF = 1, BTF = 1, cleared by reading SR1 register followed by writing DR register.

EV3-1: EVF = 1, AF = 1, BTF = 1; AF is cleared by reading SR1 register. BTF is cleared by releasing the lines (STOP = 1, STOP = 0) or by writing DR register (DR = FFh). ***Note:*** *If lines are released by STOP = 1, STOP = 0, the subsequent EV4 is not seen.*

EV4: EVF = 1, STOPF = 1, cleared by reading SR2 register.

EV5: EVF = 1, SB = 1, cleared by reading SR1 register followed by writing DR register.

EV6: EVF = 1, cleared by reading SR1 register followed by writing CR register (for example PE = 1).

EV7: EVF = 1, BTF = 1, cleared by reading SR1 register followed by reading DR register.

EV8: EVF = 1, BTF = 1, cleared by reading SR1 register followed by writing DR register.

EV9: EVF = 1, ADD10 = 1, cleared by reading SR1 register followed by writing DR register.

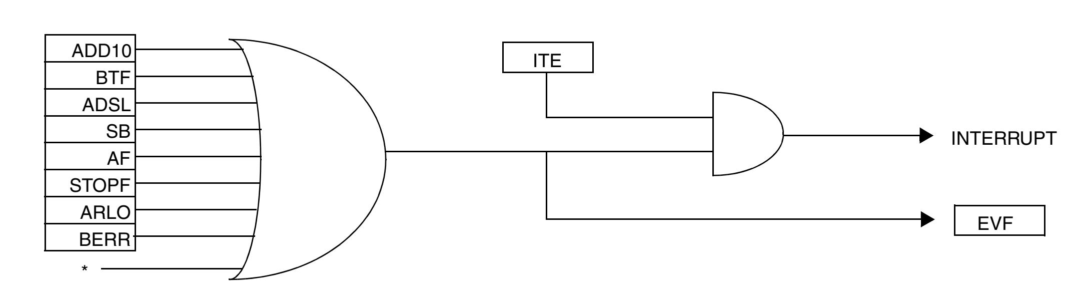
16.5 Low power modes

Table 81. Effect of low power modes on I^2C

Mode	Effect
Wait	No effect on I^2C interface. I^2C interrupts cause the device to exit from Wait mode.
Halt	I^2C registers are frozen. In Halt mode, the I^2C interface is inactive and does not acknowledge data on the bus. The I^2C interface resumes operation when the MCU is woken up by an interrupt with “exit from Halt mode” capability.

16.6 Interrupts

Figure 69. Interrupt control logic diagram



* EVF can also be set by EV6 or an error from the SR2 register.

Table 82. I^2C interrupt control/wake-up capability

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
10-bit Address Sent Event (Master mode)	ADD10	ITE	Yes	No
End of Byte Transfer Event	BTF			
Address Matched Event (Slave mode)	ADSEL			
Start Bit Generation Event (Master mode)	SB			
Acknowledge Failure Event	AF			
Stop Detection Event (Slave mode)	STOPF			
Arbitration Lost Event (Multimaster configuration)	ARLO			
Bus Error Event	BERR			

Note: The I^2C 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 I-bit in the CC register is reset (RIM instruction).

Table 109. CAN register map and reset values

Address (Hex.)	Page	Register label	7	6	5	4	3	2	1	0
5A		CANISR Reset value	RXIF3 0	RXIF2 0	RXIF1 0	TXIF 0	SCIF 0	ORIF 0	TEIF 0	EPND 0
5B		CANICR Reset value	0	ESCI 0	RXIE 0	TXIE 0	SCIE 0	ORIE 0	TEIE 0	ETX 0
5C		CANCSR Reset value	0	BOFF 0	EPSV 0	SRTE 0	NRTX 0	FSYN 0	WKPS 0	RUN 0
5D		CANBRPR Reset value	RJW1 0	RJW0 0	BRP5 0	BRP4 0	BRP3 0	BRP2 0	BRP1 0	BRP0 0
5E		CANBTR Reset value	0	BS22 0	BS21 1	BS20 0	BS13 0	BS12 0	BS11 1	BS10 1
5F		CANPSR Reset value	0	0	0	0	0	PAGE2 0	PAGE1 0	PAGE0 0
60	0	CANLIDHR Reset value	LID10 x	LID9 x	LID8 x	LID7 x	LID6 x	LID5 x	LID4 x	LID3 x
	1 to 3	CANIDHRx Reset value	ID10 x	ID9 x	ID8 x	ID7 x	ID6 x	ID5 x	ID4 x	ID3 x
60, 64	4	CANFHRx Reset value	FIL11 x	FIL10 x	FIL9 x	FIL8 x	FIL7 x	FIL6 x	FIL5 x	FIL4 x
61	0	CANLIDLR Reset value	LID2 x	LID1 x	LID0 x	LRTR x	LDLC3 x	LDLC2 x	LDLC1 x	LDLC0 x
	1 to 3	CANIDLRx Reset value	ID2 x	ID1 x	ID0 x	RTR x	DLC3 x	DLC2 x	DLC1 x	DLC0 x
61, 65	4	CANFLRx Reset value	FIL3 x	FIL2 x	FIL1 x	FIL0 x	0	0	0	0
62 to 69	1 to 3	CANDRx Reset value	MSB x	x	x	x	x	x	x	LSB x
62, 66	4	CANMHRx Reset value	MSK11 x	MSK10 x	MSK9 x	MSK8 x	MSK7 x	MSK6 x	MSK5 x	MSK4 x
63, 67	4	CANMLRx Reset value	MSK3 x	MSK2 x	MSK1 x	MSK0 x	0	0	0	0
6E	0	CANTECR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
6F		CANRECR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
	1 to 3	CANBCSRx Reset value	0	0	0	0	ACC 0	RDY 0	BUSY 0	LOCK 0

20.4 Supply current characteristics

The following current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To obtain the total device consumption, the two current values must be added (except for Halt mode, for which the clock is stopped).

20.4.1 Current consumption

Table 131. Current consumption

Symbol	Parameter	Conditions	Flash devices		ROM devices		Unit
			Typ	Max[1]	Typ	Max[1]	
I_{DD}	Supply current in Run mode[2]	f_{OSC} = 2MHz, f_{CPU} = 1 MHz f_{OSC} = 4 MHz, f_{CPU} = 2 MHz f_{OSC} = 8 MHz, f_{CPU} = 4 MHz f_{OSC} = 16 MHz, f_{CPU} = 8 MHz	1.3 2.0 3.6 7.1	3.0 5.0 8.0 15.0	1.3 2.0 3.6 7.1	2.0 3.0 5.0 10.0	mA
	Supply current in Slow mode[2]	f_{OSC} = 2 MHz, f_{CPU} = 62.5 kHz f_{OSC} = 4 MHz, f_{CPU} = 125 kHz f_{OSC} = 8 MHz, f_{CPU} = 250 kHz f_{OSC} = 16 MHz, f_{CPU} = 500 kHz	600 700 800 1100	2700 3000 3600 4000	600 700 800 1100	1800 2100 2400 3000	µA
	Supply current in Wait mode[2]	f_{OSC} = 2 MHz, f_{CPU} = 1 MHz f_{OSC} = 4 MHz, f_{CPU} = 2 MHz f_{OSC} = 8 MHz, f_{CPU} = 4 MHz f_{OSC} = 16 MHz, f_{CPU} = 8 MHz	1.0 1.5 2.5 4.5	3.0 4.0 5.0 7.0	1.0 1.5 2.5 4.5	1.3 2.0 3.3 6.0	mA
	Supply current in Slow Wait mode[2]	f_{OSC} = 2 MHz, f_{CPU} = 62.5 kHz f_{OSC} = 4 MHz, f_{CPU} = 125 kHz f_{OSC} = 8 MHz, f_{CPU} = 250 kHz f_{OSC} = 16 MHz, f_{CPU} = 500 kHz	580 650 770 1050	1200 1300 1800 2000	70 100 200 350	200 300 600 1200	µA
	Supply current in Halt mode[3]	$-40°C \leq T_A \leq +85°C$	<1	10	<1	10	µA
		$-40°C \leq T_A \leq +125°C$	<1	50	<1	50	
I_{DD}	Supply current in Active Halt mode[4]	f_{OSC} = 2 MHz f_{OSC} = 4 MHz f_{OSC} = 8 MHz f_{OSC} = 16 MHz	80 160 325 650	No max. guaranteed	15 30 60 120	25 50 100 200	µA

1. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
2. Measurements are done in the following conditions:
- Program executed from RAM, CPU running with RAM access. The increase in consumption when executing from Flash is 50%.
- All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load)
- All peripherals in reset state.
- LVD disabled.
- Clock input (OSC1) driven by external square wave.
- In Slow and Slow Wait mode, f_{CPU} is based on f_{OSC} divided by 32.
- To obtain the total current consumption of the device, add the clock source (*Section 20.4.2*) and the peripheral power consumption (*Section 20.4.3*).
3. All I/O pins in push-pull 0 mode (when applicable) with a static value at V_{DD} or VSS (no load), LVD disabled. Data based on characterization results, tested in production at V_{DD} max. and f_{CPU} max.
4. Data based on characterization results, not tested in production. All I/O pins in push-pull 0 mode (when applicable) with a static value at V_{DD} or V_{SS} (no load); clock input (OSC1) driven by external square wave, LVD disabled. To obtain the total current consumption of the device, add the clock source consumption (*Section 20.4.2*).

20.6 Memory characteristics

20.6.1 RAM and hardware registers

Table 140. RAM supply voltage

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{RM}	Data retention mode[1]	Halt mode (or RESET)	1.6			V

[1] Minimum V_{DD} supply voltage without losing data stored in RAM (in Halt mode or under RESET) or in hardware registers (only in Halt mode). Not tested in production.

20.6.2 Flash memory

Table 141. Dual voltage HDFlash memory

Symbol	Parameter	Conditions	Min[1]	Typ	Max[1]	Unit
f_{CPU}	Operating frequency	Read mode	0		8	MHz
		Write / Erase mode	1		8	
V_{PP}	Programming voltage[2]	$4.5V \leq V_{DD} \leq 5.5V$	11.4		12.6	V
I_{DD}	Supply current[3]	Run mode (f_{CPU} = 4 MHz)			3	mA
		Write / Erase		0		
		Power down mode / HALT		1	10	µA
I_{PP}	V_{PP} current[3]	Read (V_{PP} = 12V)			200	
		Write / Erase			30	mA
t_{VPP}	Internal V_{PP} stabilization time			10		µs
t_{RET}	Data retention	T_A = 55°C	20			years
N_{RW}	Write erase cycles	T_A = 85°C	100			cycles
T_{PROG} T_{ERASE}	Programming or erasing temperature range		-40	25	85	°C

[1] Data based on characterization results, not tested in production

[2] V_{PP} must be applied only during the programming or erasing operation and not permanently for reliability reasons.

[3] Data based on simulation results, not tested in production

Warning: Do not connect 12V to V_{PP} before V_{DD} is powered on, as this may damage the device.

Figure 102. Typical V_{OL} versus V_{DD} (standard)

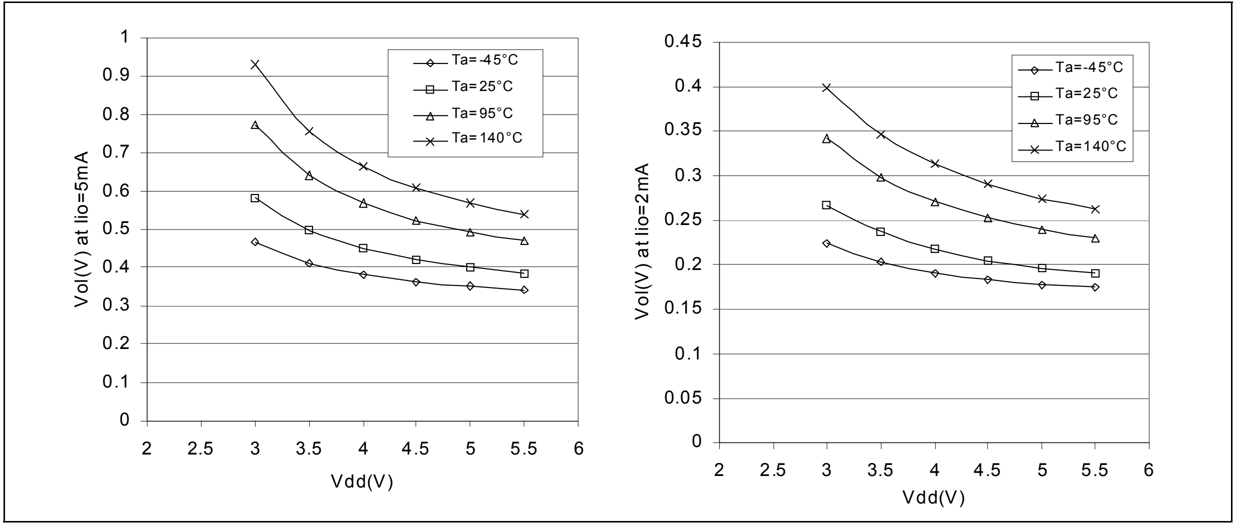


Figure 103. Typical V_{OL} versus V_{DD} (high-sink)

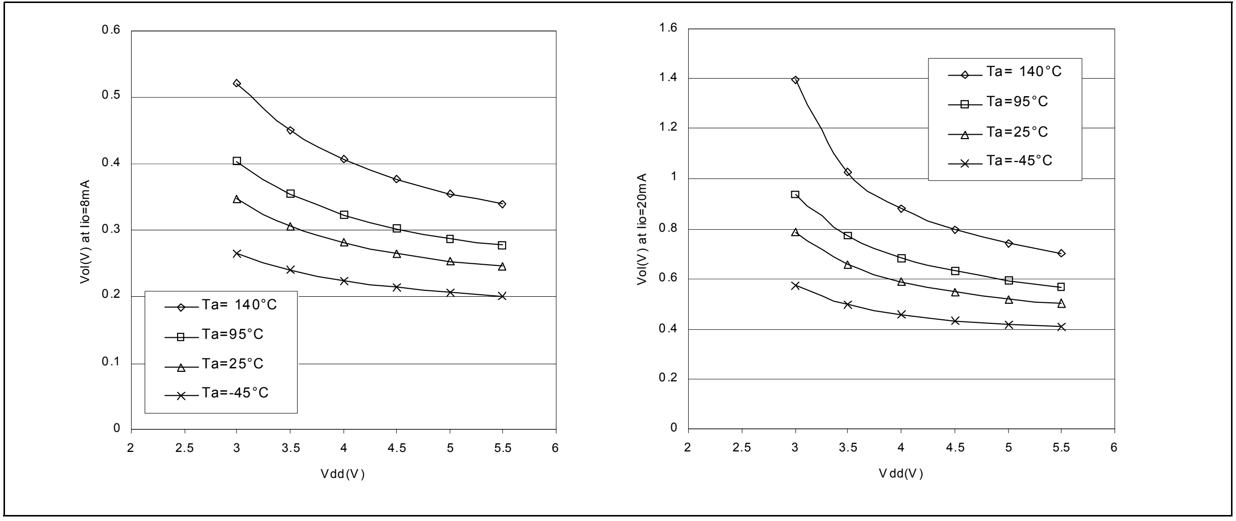


Figure 104. Typical V_{DD} - V_{OH} versus V_{DD}

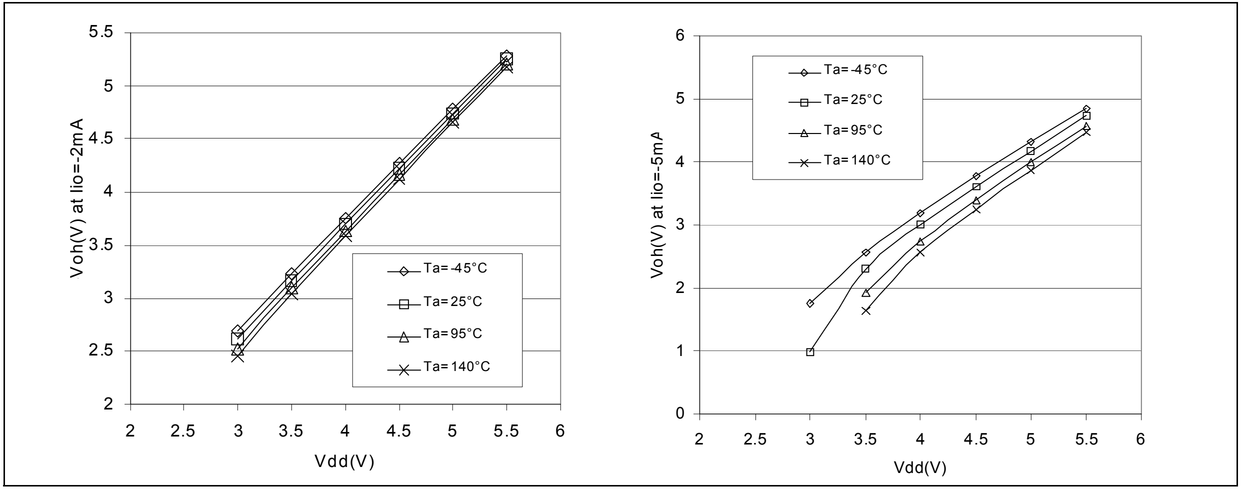
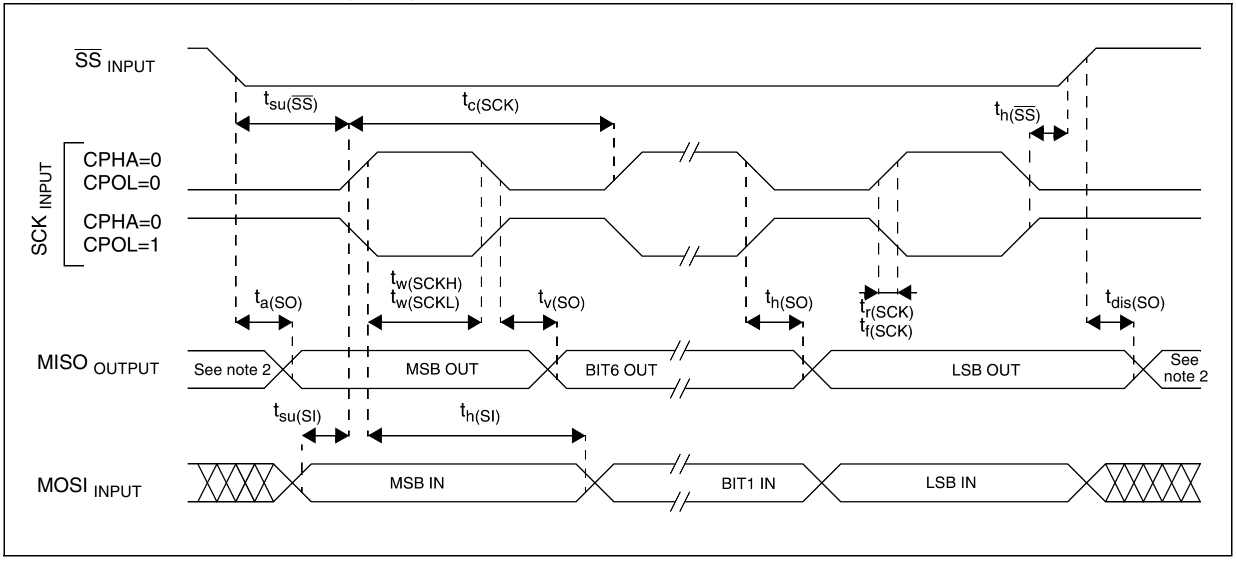
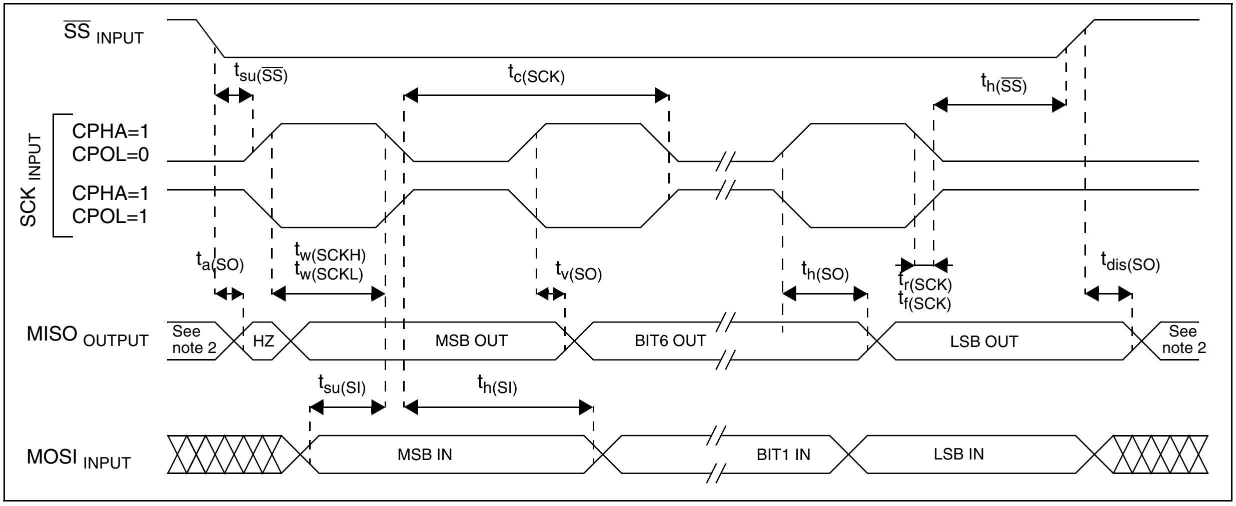


Figure 108. SPI slave timing diagram with CPHA = 0[1]



1. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends on the I/O port configuration.

Figure 109. SPI slave timing diagram with CPHA = 1[1]



1. Measurement points are done at CMOS levels: $0.3xV_{DD}$ and $0.7xV_{DD}$.
2. When no communication is on-going the data output line of the SPI (MOSI in master mode, MISO in slave mode) has its alternate function capability released. In this case, the pin status depends of the I/O port configuration.

20.11.2 I^2C - inter IC control interface

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

Refer to *Section 20.8: I/O port pin characteristics* for more details on the input/output alternate function characteristics (SDAI and SCLI). The ST7 I2C interface meets the requirements of the standard I2C communication protocol described in the following table.

Table 153. I^2C control interface characteristics

Symbol	Parameter	Standard mode I^2C		Fast mode I^2C(1)		Unit
		Min(2)	Max(2)	Min(2)	Max(2)	
$t_{w(SCLL)}$	SCL clock low time	4.7		1.3		µs
$t_{w(SCLH)}$	SCL clock high time	4.0		0.6		
$t_{su(SDA)}$	SDA setup time	250		100		ns
$t_{h(SDA)}$	SDA data hold time	0(3)		0(4)	900(3)	
$t_{r(SDA)}$ $t_{r(SCL)}$	SDA and SCL rise time		1000	$20+0.1C_b$	300	
$t_{f(SDA)}$ $t_{f(SCL)}$	SDA and SCL fall time		300			
$t_{h(STA)}$	START condition hold time	4.0		0.6		µs
$t_{su(STA)}$	Repeated START condition setup time	4.7				
$t_{su(STO)}$	STOP condition setup time	4.0				
$t_{w(STO:STA)}$	STOP to START condition time (bus free)	4.7		1.3		
C_b	Capacitive load for each bus line		400		400	pF

1. At 4 MHz f_{CPU}, maximum I^2C speed (400 kHz) is not achievable. In this case, maximum I^2C speed will be approximately 260 kHz.
2. Data based on standard I^2C protocol requirement, not tested in production.
3. The maximum hold time of the START condition has only to be met if the interface does not stretch the low period of SCL signal.
4. The device must internally provide a hold time of at least 300ns for the SDA signal in order to bridge the undefined region of the falling edge of SCL.

20.12.3 ADC accuracy

Conditions: V_{DD} = 5V[1]

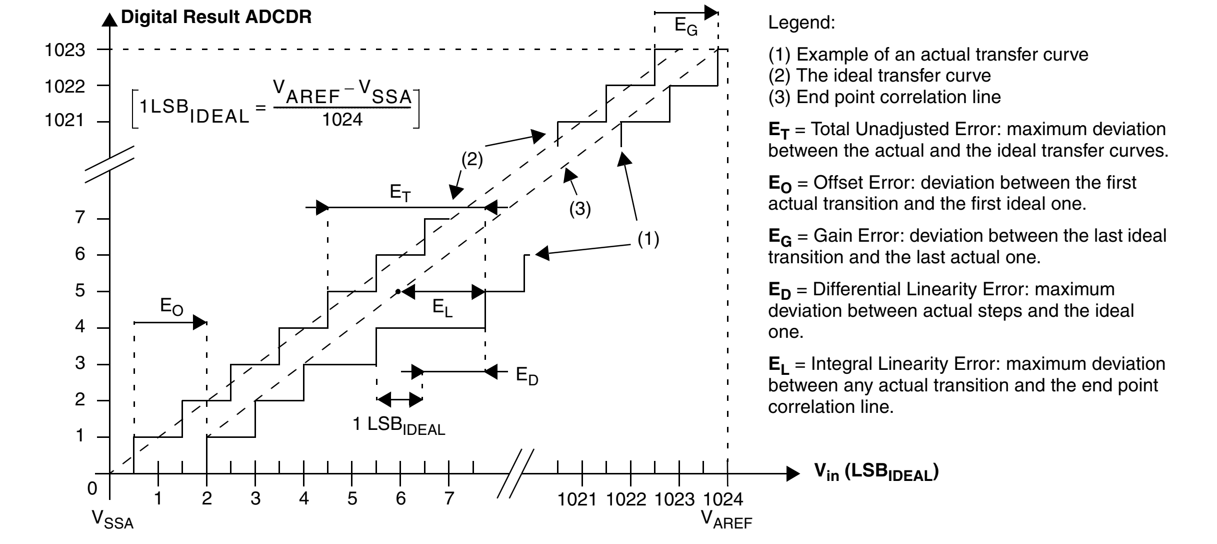
Table 157. ADC accuracy

Symbol	Parameter[1]	Conditions	Typ	Max[2]	Unit
$\|E_T\|$	Total unadjusted error	CPU in run mode @ f_{ADC} 2 MHz	3	4	LSB
$\|E_O\|$	Offset error		2	3	
$\|E_G\|$	Gain error		0.5	3	
$\|E_D\|$	Differential linearity error		1	2	
$\|E_L\|$	Integral linearity error				

1. ADC Accuracy versus Negative Injection Current: Injecting negative current may reduce the accuracy of the conversion being performed on another analog input. The effect of negative injection current on robust pins is specified in *Section 20.12*.
Any positive injection current within the limits specified for $I_{INJ(PIN)}$ and $\Sigma I_{INJ(PIN)}$ in *Section 20.8* does not affect the ADC accuracy.

2. Data based on characterization results, monitored in production to guarantee 99.73% within ± max value from -40°C to 125°C (± 3σ distribution limits).

Figure 116. ADC error classification



Legend:

(1) Example of an actual transfer curve
(2) The ideal transfer curve
(3) End point correlation line

E_T = Total Unadjusted Error: maximum deviation between the actual and the ideal transfer curves.

E_O = Offset Error: deviation between the first actual transition and the first ideal one.

E_G = Gain Error: deviation between the last ideal transition and the last actual one.

E_D = Differential Linearity Error: maximum deviation between actual steps and the ideal one.

E_L = Integral Linearity Error: maximum deviation between any actual transition and the end point correlation line.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

Information in this document supersedes and replaces all information previously supplied.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -
Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com