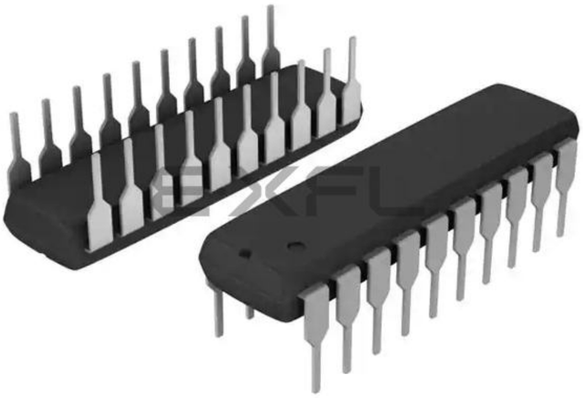
STMicroelectronics - ST7FLITE20F2B6 Datasheet



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	SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	15
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
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	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7flite20f2b6

List of figures

Port and control configuration:

- Input:
  – float = floating, wpu = weak pull-up, int = interrupt, ana = analog
- Output:
  – OD = open drain
  – PP = push-pull

The RESET configuration of each pin is shown in bold which is valid as long as the device is in reset state.

Table 2. Device pin description

Pin No.		Pin name	Type	Level		Port / Control						Main function (after reset)	Alternate function
						Input				Output			
SO20	DIP20			Input	Output	float	wpu	int	ana	OD	PP		
1	16	V_{SS}	S	-	-	-	-	-	-	-	-	Ground	
2	17	V_{DD}	S	-	-	-	-	-	-	-	-	Main power supply	
3	18	$\overline{\text{RESET}}$	I/O	C_T	-	-	X	-	-	X	-	Top priority non maskable interrupt (active low)	
4	19	PB0/AIN0/$\overline{\text{SS}}$	I/O	C_T		**X**	ei3		X	X	X	**Port B0**	ADC analog input 0 or SPI Slave Select (active low)(1)
5	20	PB1/AIN1/SCK	I/O	C_T		**X**			X	X	X	**Port B1**	ADC analog input 1 or SPI Serial Clock(1)
6	1	PB2/AIN2/MISO	I/O	C_T		**X**			X	X	X	**Port B2**	ADC analog input 2 or SPI Master in/ Slave out data
7	2	PB3/AIN3/MOSI	I/O	C_T		**X**	ei2		X	X	X	**Port B3**	ADC analog input 3 or SPI Master out / Slave in data
8	3	PB4/AIN4/CLKIN	I/O	C_T		**X**			X	X	X	**Port B4**	ADC analog input 4 or external clock input
9	4	PB5/AIN5	I/O	C_T		**X**			X	X	X	**Port B5**	ADC analog input 5
10	5	PB6/AIN6	I/O	C_T		**X**			X	X	X	**Port B6**	ADC analog input 6
11	6	PA7	I/O	C_T	HS	**X**	ei1		-	X	X	**Port A7**	-
12	7	PA6 /MCO/ ICCCLK/BREAK	I/O	C_T		X	ei1		-	X	X	**Port A6**	Main clock output or in circuit communication clock or external BREAK(2)
13	8	PA5 /ATPWM3/ ICCDATA	I/O	C_T	HS	**X**	ei1		-	X	X	**Port A5**	Auto-reload timer PWM3 or In circuit communication data
14	9	PA4/ATPWM2	I/O	C_T	HS	**X**			-	X	X	**Port A4**	Auto-reload timer PWM2

Table 2. Device pin description (continued)

Pin No.		Pin name	Type	Level		Port / Control						Main function (after reset)	Alternate function
						Input				Output			
SO20	DIP20			Input	Output	float	wpu	int	ana	OD	PP		
15	10	PA3/ATPWM1	I/O	C_T	HS	**X**	ei0		-	X	X	**Port A3**	Auto-reload timer PWM1
16	11	PA2/ATPWM0	I/O	C_T	HS	**X**			-	X	X	**Port A2**	Auto-reload timer PWM0
17	12	PA1/ATIC	I/O	C_T	HS	**X**			-	X	X	**Port A1**	Auto-reload timer input capture
18	13	PA0/LTIC	I/O	C_T	HS	**X**			-	X	X	**Port A0**	Lite timer input capture
19	14	OSC2	O	–	–	-	-	-	-	-	-	Resonator oscillator inverter output	
20	15	OSC1/CLKIN	I	–	–	-	-	-	-	-	-	Resonator oscillator inverter input or external clock input	

1. No negative current injection allowed on this pin. For details (refer to*Table 58: Current characteristics*).

2. During normal operation this 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. In the application, even if the pin is configured as output, any reset will put it back in input pull-up.

3 Register & memory map

As shown in *Figure 4*, the MCU is able of addressing 64K bytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, 384 bytes of RAM, 256 bytes of data EEPROM and 8 Kbytes of user program memory. The RAM space includes up to 128 bytes for the stack from 180h to 1FFh.

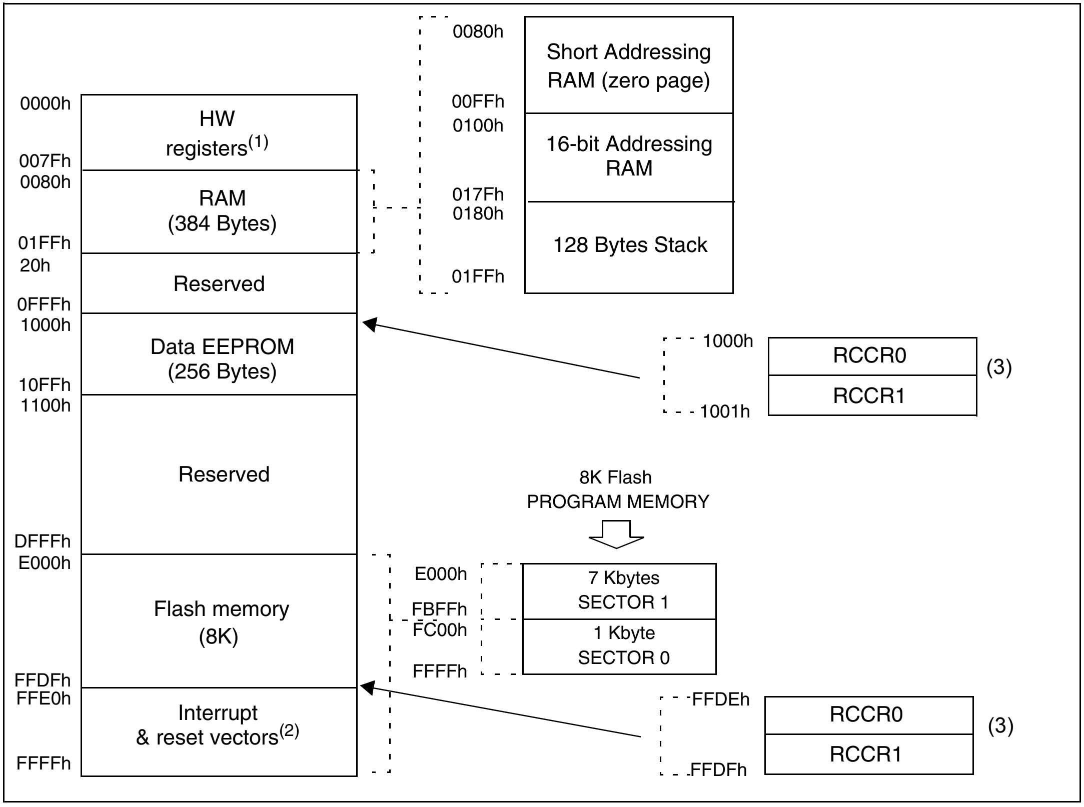
The highest address bytes contain the user reset and interrupt vectors.

The Flash memory contains two sectors (see *Figure 4*) mapped in the upper part of the ST7 addressing space so the reset and interrupt vectors are located in Sector 0 (F000h-FFFFh).

The size of Flash Sector 0 and other device options are configurable by Option byte (refer to *Section 15: Device configuration*).

Note: Memory locations marked as "Reserved" must never be accessed. Accessing a reserved area can have unpredictable effects on the device.

Figure 4. Memory map



1. See*Table 3: Hardware register map*
2. See *Table 12: Interrupt mapping*
3. See *Section 7.1: Internal RC oscillator adjustment*

4 Flash program memory

4.1 Introduction

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

The XFlash devices can be programmed off-board (plugged in a programming tool) or on-board using in-circuit programming or in-application programming.

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

4.2 Main features

- ICP (in-circuit programming)
- IAP (in-application programming)
- ICT (in-circuit testing) for downloading and executing user application test patterns in RAM
- Sector 0 size configurable by option byte
- Read-out and write protection

4.3 Programming modes

The ST7 can be programmed in three different ways:

- Insertion in a programming tool. In this mode, Flash sectors 0 and 1, option byte row and data EEPROM (if present) can be programmed or erased.
- In-circuit programming. In this mode, Flash sectors 0 and 1, option byte row and data EEPROM (if present) can be programmed or erased without removing the device from the application board.
- In-application programming. In this mode, sector 1 and data EEPROM (if present) can be programmed or erased without removing the device from the application board and while the application is running.

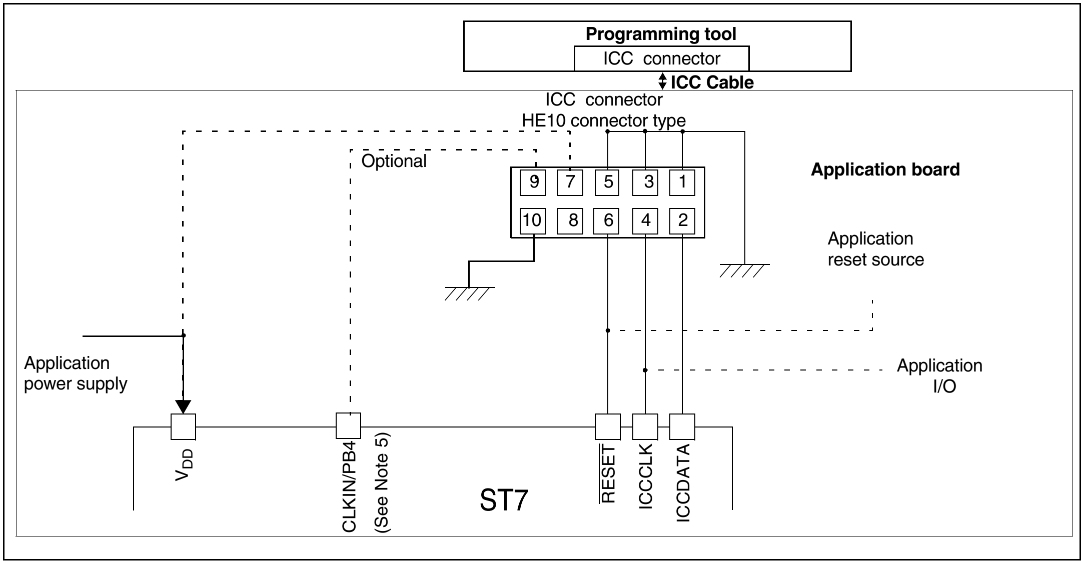
4.3.1 In-circuit programming (ICP)

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

1. Switch the ST7 to ICC mode (in-circuit communications). This is done by driving a specific signal sequence on the ICCCLK/DATA pins while the RESET pin is pulled low. When the ST7 enters ICC mode, it fetches a specific RESET vector which points to the ST7 System Memory containing the ICC protocol routine. This routine enables the ST7 to receive bytes from the ICC interface.
2. Download ICP driver code in RAM from the ICCDATA pin.
3. Execute ICP driver code in RAM to program the Flash memory.

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. In the application, even if the pin is configured as output, any reset will put it back in input pull-up.

Figure 5. Typical ICC interface



4.5 Memory protection

There are two different types of memory protection: Read-Out Protection and Write/Erase Protection which can be applied individually.

4.5.1 Read-out protection

Read-out protection, when selected provides a protection against program memory content extraction and against write access to Flash memory. Even if no protection can be considered as totally unbreakable, the feature provides a very high level of protection for a general purpose microcontroller. Both program and data E^2 memory are protected.

In flash devices, this protection is removed by reprogramming the option. In this case, both program and data E^2 memory are automatically erased and the device can be reprogrammed.

Read-out protection selection depends on the device type:

- In Flash devices it is enabled and removed through the FMP_R bit in the option byte.
- In ROM devices it is enabled by mask option specified in the Option List.

4.5.2 Flash write/erase protection

Write/erase protection, when set, makes it impossible to both overwrite and erase program memory. It does not apply to E^2 data. Its purpose is to provide advanced security to applications and prevent any change being made to the memory content.

Caution: Once set, Write/erase protection can never be removed. A write-protected flash device is no longer reprogrammable.

Read operation (E2LAT=0)

The EEPROM can be read as a normal ROM location when the E2LAT bit of the EECSR register is cleared.

On this device, Data EEPROM can also be used to execute machine code. Take care not to write to the Data EEPROM while executing from it. This would result in an unexpected code being executed.

Write operation (E2LAT=1)

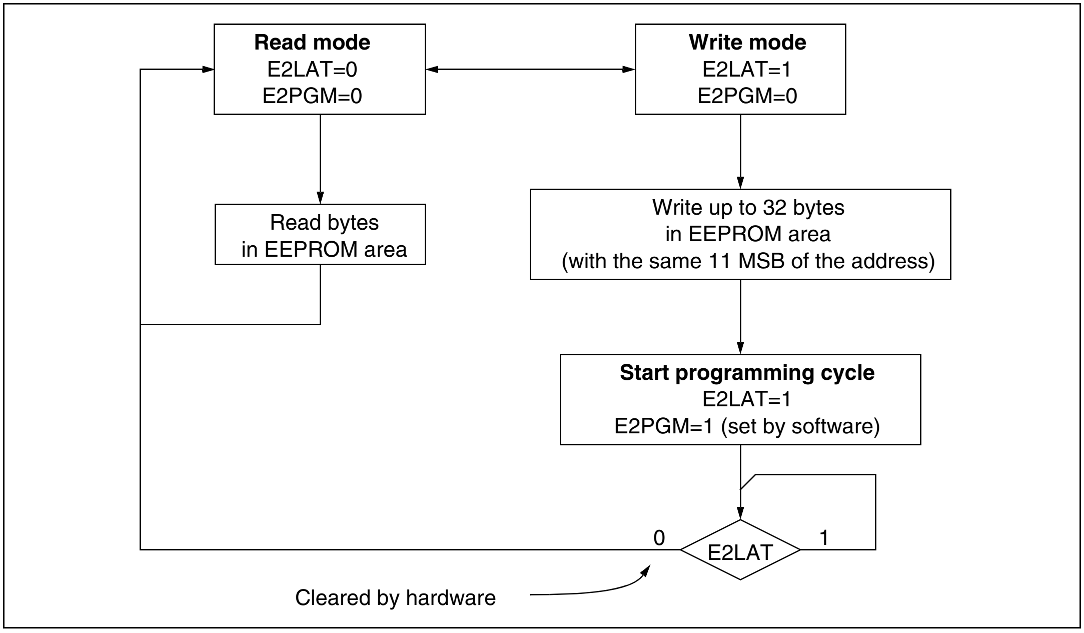
To access the write mode, the E2LAT bit has to be set by software (the E2PGM bit remains cleared). When a write access to the EEPROM area occurs, the value is latched inside the 32 data latches according to its address.

When PGM bit is set by the software, all the previous bytes written in the data latches (up to 32) are programmed in the EEPROM cells. The effective high address (row) is determined by the last EEPROM write sequence. To avoid wrong programming, the user must take care that all the bytes written between two programming sequences have the same high address: only the five Least Significant Bits of the address can change.

At the end of the programming cycle, the PGM and LAT bits are cleared simultaneously.

Note: *Care should be taken during the programming cycle. Writing to the same memory location will over-program the memory (logical AND between the two write access data result) because the data latches are only cleared at the end of the programming cycle and by the falling edge of the E2LAT bit.*
It is not possible to read the latched data.
This note is illustrated by the Figure 9: Data EEPROM programming cycle.

Figure 7. Data EEPROM programming flowchart



Note: See Section 13: Electrical characteristics for more information on the frequency and accuracy of the RC oscillator.

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.

These two bytes are systematically programmed by ST, including on FASTROM devices. Consequently, customers intending to use FASTROM service must not use these two bytes.

RCCR0 and RCCR1 calibration values will be erased if the Read-out protection bit is reset after it has been set. See *Section 4.5.1: Read-out protection*.

Caution: If the voltage or temperature conditions change in the application, the frequency may need to be recalibrated.
Refer to application note AN1324 for information on how to calibrate the RC frequency using an external reference signal.

7.2 Phase locked loop (PLL)

The PLL can be used to multiply a 1 MHz frequency from the RC oscillator or the external clock by 4 or 8 to obtain f_{OSC} of 4 or 8 MHz. The PLL is enabled and the multiplication factor of 4 or 8 is selected by 2 option bits.

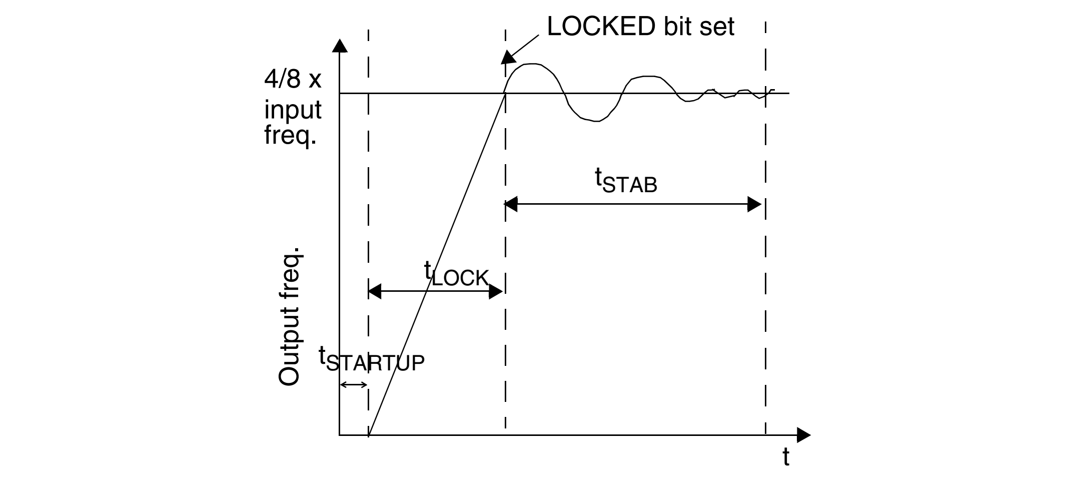
- The x4 PLL is intended for operation with V_{DD} in the 2.4 V to 3.3 V range
- The x8 PLL is intended for operation with V_{DD} in the 3.3 V to 5.5 V range

Note: Refer to Section 15.1: Option bytes for the option byte description.

If the PLL is disabled and the RC oscillator is enabled, then f_{OSC} = 1 MHz.

If both the RC oscillator and the PLL are disabled, f_{OSC} is driven by the external clock.

Figure 12. PLL output frequency timing diagram



When the PLL is started, after reset or wakeup from HALT mode or AWUF 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 12* below and *Figure 64: RC oscillator and PLL characteristics (tested*

External interrupt selection register (EISR)

Read/Write

Reset Value: 0000 1100 (0Ch)

7							0
IS31	IS30	IS21	IS20	IS11	IS10	IS01	IS00

- Bits 7:6 = **ei3[1:0]** *ei3 pin selection*
  These bits are written by software. They select the Port B I/O pin used for the ei3 external interrupt according to the table below.

Table 14. External interrupt I/O pin ei3[1:0] selection

ei31	ei30	I/O pin
0	0	PB0 [1]
0	1	PB1
1	0	PB2

1. Reset state

- Bits 5:4 = **ei2[1:0]** *ei2 pin selection*
  These bits are written by software. They select the Port B I/O pin used for the ei2 external interrupt according to the table below.

Table 15. External interrupt I/O pin ei2[1:0] selection

ei21	ei20	I/O pin
0	0	PB3 [1]
0	1	PB4 [2]
1	0	PB5
1	1	PB6

1. Reset state
2. PB4 cannot be used as an external interrupt in HALT mode.

- Bit 3:2 = **ei1[1:0]** *ei1 pin selection*
  These bits are written by software. They select the Port A I/O pin used for the ei1 external interrupt according to the table below.

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. This downcounter is freerunning: it counts down even if the watchdog is disabled. The value to be stored in the CR register must be between FFh and C0h (see *Table 31: Watchdog timing*):

- The WDGA bit is set (watchdog enabled)
- The T6 bit is set to prevent generating an immediate reset
- The T[5:0] bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.

The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).

If the watchdog is activated, the HALT instruction will generate a Reset.

Table 31. Watchdog timing[1]

f_{CPU} = 8 MHz		
WDG counter code	**min [ms]**	**max [ms]**
C0h	1	2
FFh	127	128

[1] The timing variation is due to the unknown status of the prescaler when writing to the CR register.

Note: The number of CPU clock cycles applied during the reset phase (256 or 4096) must be taken into account in addition to these timings.

11.1.4 Hardware watchdog option

If Hardware Watchdog is selected by option byte, the watchdog is always active and the WDGA bit in the CR is not used.

Refer to the Option Byte description in *Section 15: Device configuration*.

Using HALT mode or ACTIVE-HALT mode with the WDG (WDGHALT option)

If Halt mode with Watchdog is enabled by option byte (No watchdog reset on HALT instruction), it is recommended before executing the HALT instruction to refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller. Same behavior in active-halt mode.

11.1.5 Interrupts

None.

1: Master mode. The function of the SCK pin changes from an input to an output and the functions of the MISO and MOSI pins are reversed.

- Bit 3 = **CPOL** *Clock polarity*
  This bit is set and cleared by software. This bit determines the idle state of the serial Clock. The CPOL bit affects both the master and slave modes.
  0: SCK pin has a low level idle state
  1: SCK pin has a high level idle state

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

- Bit 2 = **CPHA** *Clock Phase*
  This bit is set and cleared by software.
  0: The first clock transition is the first data capture edge.
  1: The second clock transition is the first capture edge.

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

- Bits 1:0 = **SPR[1:0]** *Serial Clock Frequency*
  These bits are set and cleared by software. Used with the SPR2 bit, they select the baud rate of the SPI serial clock SCK output by the SPI in master mode.

Note: These 2 bits have no effect in slave mode.

Table 42. SPI master mode SCK frequency

Serial Clock	SPR2	SPR1	SPR0
$f_{CPU}/4$	1	0	0
$f_{CPU}/8$	0	0	0
$f_{CPU}/16$	0	0	1
$f_{CPU}/32$	1	1	0
$f_{CPU}/64$	0	1	0
$f_{CPU}/128$	0	1	1

Control/status register (SPICSR)

Read/Write (some bits are 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.

12 Instruction set

12.1 ST7 addressing modes

The ST7 Core features 17 different addressing modes which can be classified in seven main groups:

Table 48. Addressing mode groups

Addressing mode	Example
Inherent	nop
Immediate	ld A,#$55
Direct	ld A,$55
Indexed	ld A,($55,X)
Indirect	ld A,([$55],X)
Relative	jrne loop
Bit operation	bset byte,#5

The ST7 Instruction set is designed to minimize the number of bytes required per instruction: To do so, most of the addressing modes may be subdivided in two submodes called long and short:

- Long addressing mode is more powerful because it can use the full 64 Kbyte address space, however it uses more bytes and more CPU cycles.
- Short addressing mode is less powerful because it can generally only access page zero (0000h00FFh range), but the instruction size is more compact, and faster. All memory to memory instructions use short addressing modes only (CLR, CPL, NEG, BSET, BRES, BTJT, BTJF, INC, DEC, RLC, RRC, SLL, SRL, SRA, SWAP).

The ST7 Assembler optimizes the use of long and short addressing modes.

Table 49. ST7 addressing mode overview

Mode			Syntax	Destination/ source	Pointer address (Hex.)	Pointer size (Hex.)	Length (bytes)
Inherent	–	–	nop	–	–	–	+ 0
Immediate	–	–	ld A,#$55	–	–	–	+ 1
Short	Direct	–	ld A,$10	00..FF	–	–	+ 1
Long	Direct	–	ld A,$1000	0000..FFFF	–	–	+ 2
No offset	Direct	Indexed	ld A,(X)	00..FF	–	–	+ 0 (with x register) + 1 (with Y register)
Short	Direct	Indexed	ld A,($10,X)	00..1FE	–	–	+ 1

Table 64. 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 7.1: Internal RC oscillator adjustment*.
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 12: PLL output frequency timing diagram*.
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.

Table 70. Resonator characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{CrOSC}	Crystal Oscillator Frequency[1]	–	2	–	16	MHz
C_{L1} C_{L2}	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator (R_S)	–	See *Table 72: Resonator performances*			pF

1. When PLL is used, please refer to the *Section 13.3.4: Internal RC oscillator and PLL* and *Section 7: Supply, reset and clock management* (f_{CrOSC} min. is 8 Mhz with PLL).

Table 71: Resonator performances

Supplier	f_{CrOSC} [MHz]	Typical ceramic resonators[1]		CL1[2] [pF]	CL2[2] [pF]	Rd [Ω]	Supply voltage range [V]	Temperature range [°C]
		Type[3]	Reference					
Murata	2	SMD	CSTCC2M00G56-R0	(47)	(47)	0	2.4V to 5.5V	-40 to 85
	4	SMD	CSTCR4M00G53-R0	(15)	(15)	0		
		LEAD	CSTLS4M00G53-B0	(15)	(15)	0		
	8	SMD	CSTCE8M00G52-R0	(10)	(10)	0		
		LEAD	CSTLS8M00G53-B0	(15)	(15)	0		
	16	SMD	CSTCE16M0V51-R0	(5)	(5)	0	3.3V to 5.5V	
		LEAD	CSTLS16M0X53-B0	(15)	(15)	0	4.5V to 5.5V	
		LEAD	CSALS16M0X55-B0	7	7	1.5k	3.8V to 5.5V	

1. Resonator characteristics given by the ceramic resonator manufacturer. For more information on these resonators, please consult www.murata.com
2. () means load capacitor built in resonator.
3. SMD = -R0: Plastic tape package (∅=180mm).
   LEAD = -B0: Bulk

Figure 74. Typical V_{OL} at V_{DD} = 5V (high-sink)

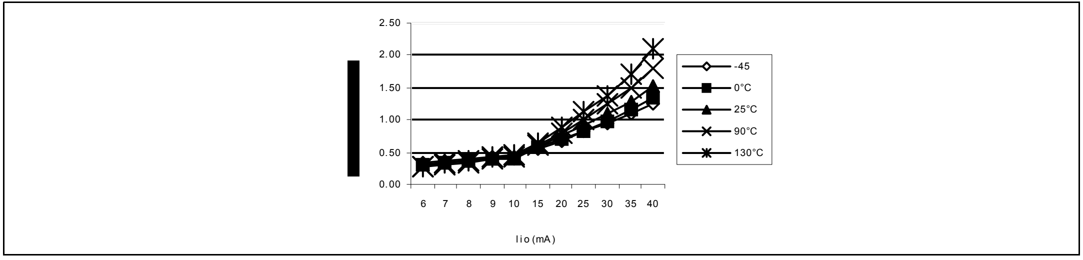


Figure 75. Typical V_{OL} at V_{DD} = 3V (high-sink)

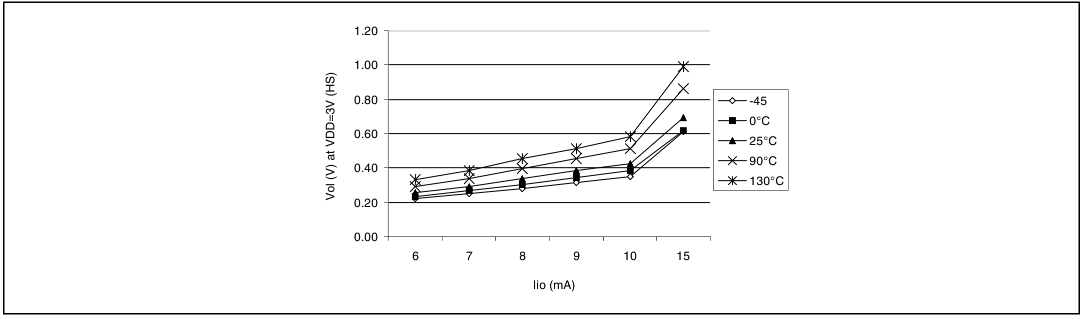


Figure 76. Typical V_{DD}-V_{OH} at V_{DD} = 2.4V

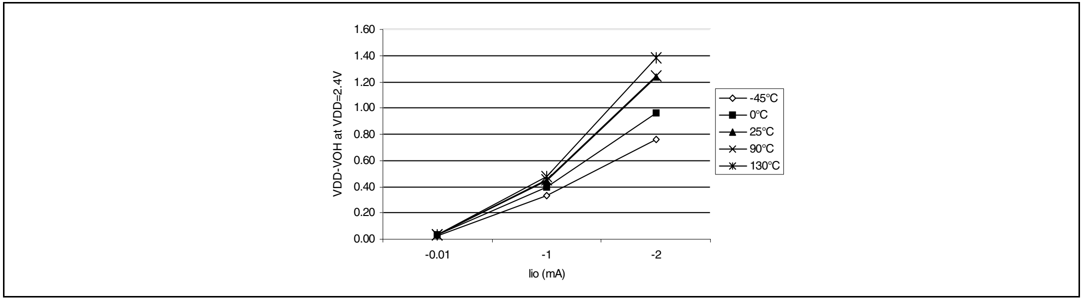
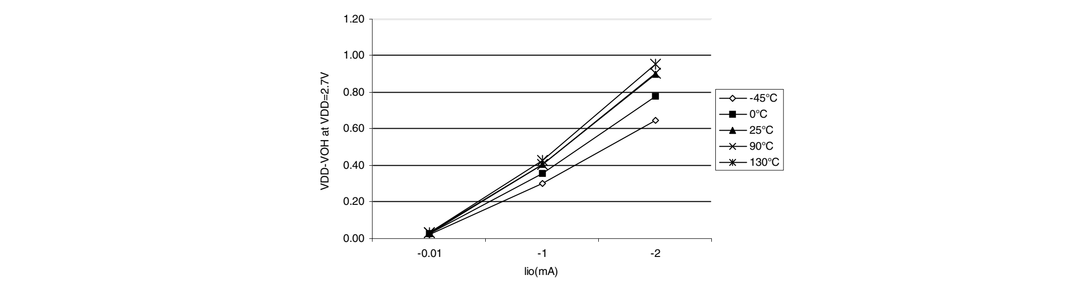


Figure 77. Typical V_{DD}-V_{OH} at V_{DD} = 2.7V



1. Check that all recommendations related to ICCCLK and reset circuit have been applied (see *Table 2: Device pin description*)
2. Check that the power supply is properly decoupled (100nF + 10µF close to the MCU). Refer to AN1709 and AN2017. If this cannot be done, it is recommended to put a 100nF + 1MΩ pull-down on the $\overline{\text{RESET}}$ pin.
3. The capacitors connected on the RESET pin and also the power supply are key to avoid any startup marginality. In most cases, steps 1 and 2 above are sufficient for a robust solution. Otherwise: replace 10nF pull-down on the $\overline{\text{RESET}}$ pin with a 5µF to 20µF capacitor."

Note: Please refer to Section 12.2.1: Illegal opcode reset for more details.

13.10 Communication interface characteristics

13.10.1 Serial peripheral interface (SPI)

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

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

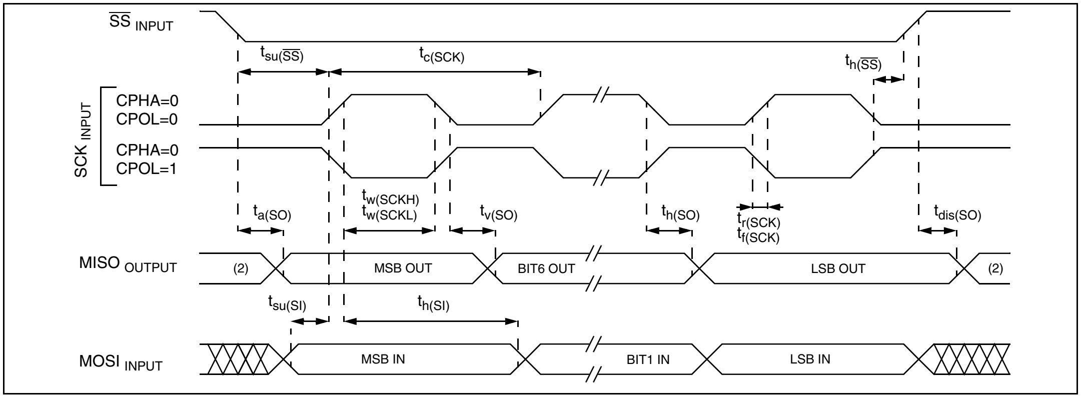
Table 83. Serial peripheral interface (SPI)[1]

Symbol	Parameter	Conditions	Min	Max	Unit
$f_{SCK} = 1/t_{c(SCK)}$	SPI clock frequency	Master f_{CPU} = 8MHz	$f_{CPU}/128 = 0.0625$	$f_{CPU}/4 = 2$	MHz
		Slave f_{CPU} = 8MHz	0	$f_{CPU}/2 = 4$	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time	–	see I/O port pin description		
$t_{su(\overline{SS})}$[2]	$\overline{\text{SS}}$ setup time[3]	Slave	$(4 \times T_{CPU}) + 150$	–	ns
$t_{h(\overline{SS})}$[2]	$\overline{\text{SS}}$ hold time	Slave	120	–	
$t_{w(SCKH)}$ $t_{w(SCKL)}$	SCK high and low time	Master Slave	100 90	–	
$t_{su(MI)}$ $t_{su(SI)}$	Data input setup time	Master Slave	100 100	–	
$t_{h(MI)}$ $t_{h(SI)}$	Data input hold time	Master Slave	100 100	–	
$t_{a(SO)}$	Data output access time	Slave	0	120	
$t_{dis(SO)}$	Data output disable time	Slave	–	240	
$t_{v(SO)}$	Data output valid time	Slave (after enable edge)	–	120	
$t_{h(SO)}$	Data output hold time		0	–	
$t_{v(MO)}$	Data output valid time	Master (after enable edge)	–	120	
$t_{h(MO)}$	Data output hold time		0	–	

1. Subject to general operating conditions for V_{DD}, f_{OSC}, and T_A unless otherwise specified.
2. Data based on design simulation and/or characterization results, not tested in production.

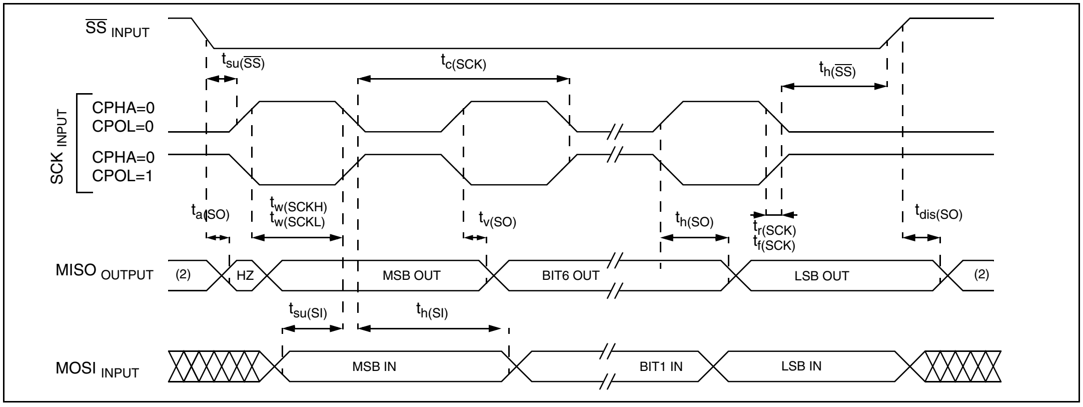
3. Depends on f_{CPU}. For example, if f_{CPU} = 8MHz, then T_{CPU} = 1/ f_{CPU} = 125ns and $t_{SU(\overline{SS})}$ = 550ns

Figure 86. 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 87. 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 on the I/O port configuration.

Table 98. ST7LITE2 FASTROM microcontroller option list

Customer
Address

Contact
Phone N^{o}
Reference/FASTROM code (assigned by STMicroelectronics)
FASTROM code must be sent in .S19 format. .Hex extension cannot be processed.

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

FASTROM device	8K
SO20:	[ ] ST7PLITE20F2M6
	[ ] ST7PLITE25F2M6
	[ ] ST7PLITE29F2M6
	[ ] ST7FLITE29F2M7
DIP20:	[ ] ST7PLITE20F2B6
	[ ] ST7PLITE25F2B6
	[ ] ST7PLITE29F2B6

Note: Addresses 1000h, 1001h, FFDEh and FFDFh are reserved areas for ST to program RCCR0 and RCCR1 (see Section 7.1: Internal RC oscillator adjustment).

Conditioning (do not specify for DIP package)

[ ] Tape & Reel[ ] Tube
Special marking: [ ] No [ ] Yes "_ _ _ _ _ _ _ _ _ "
Authorized characters are letters, digits, '.', '-', '/' and spaces only.
Maximum character count:
DIP20/S020 (8 char. max) : _ _ _ _ _ _ _ _ _
Watchdog Selection:[ ] Software Activation[ ] Hardware Activation
Watchdog Reset on HALT:[ ] Reset[ ] No Reset
LVD Reset[ ] Disabled[ ] Enabled
[ ] Highest threshold
[ ] Medium threshold
[ ] Lowest threshold
Sector 0 size:[ ] 0.5K[ ] 1K [ ] 2K[ ] 4K
Read-out Protection:[ ] Disabled[ ] Enabled
FLASH write Protection:[ ] Disabled[ ] Enabled
Clock Source Selection:[ ] Resonator:
[ ] VLP: Very Low power resonator (32 to 100 kHz)
[ ] LP: Low power resonator (1 to 2 MHz)
[ ] MP: Medium power resonator (2 to 4 MHz)
[ ] MS: Medium speed resonator (4 to 8 MHz)
[ ] HS: High speed resonator (8 to 16 MHz)
[ ] External Clock:
[ ] On OSC1

Note: Not all configurations are available. See Table 96 for authorized option byte combinations.

16 Important notes

16.1 Execution of BTJX instruction

When testing the address $FF with the "BTJT" or "BTJF" instructions, the CPU may perform an incorrect operation when the relative jump is negative and performs an address page change.

To avoid this issue, including when using a C compiler, it is recommended to never use address $00FF as a variable (using the linker parameter for example).

16.2 ADC conversion spurious results

Spurious conversions occur with a rate lower than 50 per million. Such conversions happen when the measured voltage is just between 2 consecutive digital values.

Workaround

A software filter should be implemented to remove erratic conversion results whenever they may cause unwanted consequences.

16.3 A/D converter accuracy for first conversion

When the ADC is enabled after being powered down (for example when waking up from HALT, ACTIVE-HALT or setting the ADON bit in the ADCCSR register), the first conversion (8-bit or 10-bit) accuracy does not meet the accuracy specified in the datasheet.

Workaround

In order to have the accuracy specified in the datasheet, the first conversion after a ADC switch-on has to be ignored.

16.4 Negative injection impact on ADC accuracy

Injecting a negative current on an analog input pins significantly reduces the accuracy of the AD Converter. Whenever necessary, the negative injection should be prevented by the addition of a Schottky diode between the concerned I/Os and ground.

Injecting a negative current on digital input pins degrades ADC accuracy especially if performed on a pin close to ADC channel in use.

16.5 Clearing active interrupts outside interrupt routine

When an active interrupt request occurs at the same time as the related flag or interrupt mask is being cleared, the CC register may be corrupted.