Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

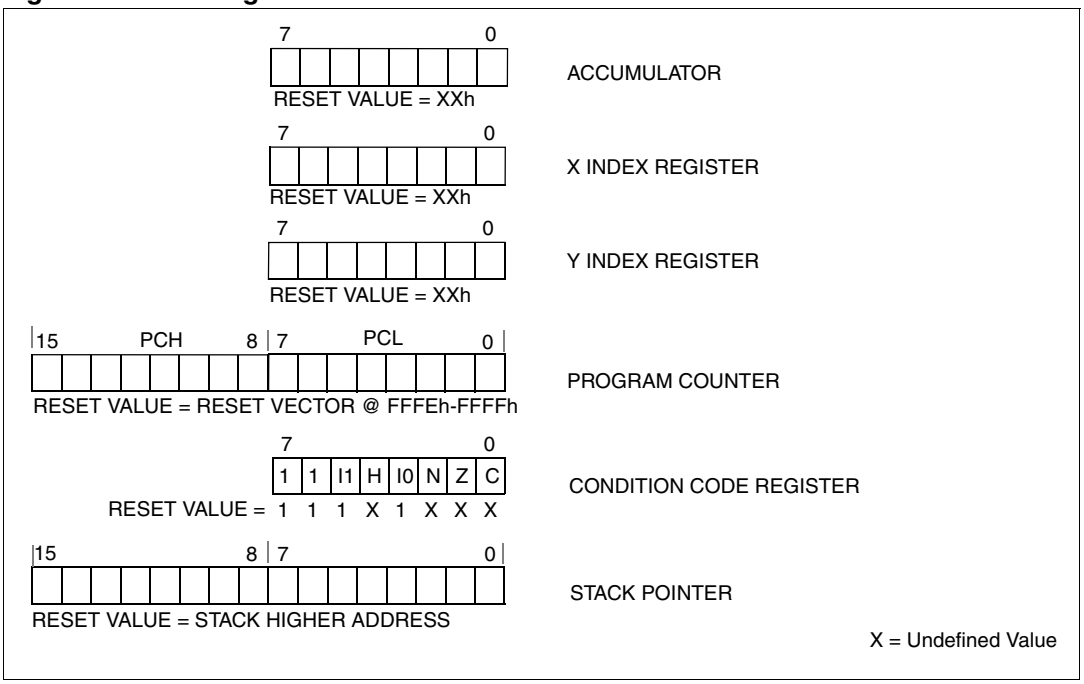
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	I²C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	34
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f344s4t6

List of tables

Figure 11. CPU registers



6.3.1 Condition code register (CC)

Reset value: 111x1xxx

7							0
1	1	I1	H	I0	N	Z	C
Read/Write							

The 8-bit Condition Code register contains the interrupt masks and four flags representative of the result of the instruction just executed. This register can also be handled by the PUSH and POP instructions.

These bits can be individually tested and/or controlled by specific instructions.

Arithmetic management bits

Bit 4 = **H** Half carry.

This bit is set by hardware when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instructions. It is reset by hardware during the same instructions.

0: No half carry has occurred.
1: A half carry has occurred.

This bit is tested using the JRH or JRNH instruction. The H bit is useful in BCD arithmetic subroutines.

Bit 2 = **N** Negative.

This bit is set and cleared by hardware. It is representative of the result sign of the last arithmetic, logical or data manipulation. It's a copy of the result 7^{th} bit.

0: The result of the last operation is positive or null.
1: The result of the last operation is negative
(that is, the most significant bit is a logic 1).

This bit is accessed by the JRMI and JRPL instructions.

Bit 1 = **Z** Zero.

This bit is set and cleared by hardware. This bit indicates that the result of the last arithmetic, logical or data manipulation is zero.

0: The result of the last operation is different from zero.
1: The result of the last operation is zero.

This bit is accessed by the JREQ and JRNE test instructions.

Bit 0 = **C** Carry/borrow.

This bit is set and cleared by hardware and software. It indicates an overflow or an underflow has occurred during the last arithmetic operation.

0: No overflow or underflow has occurred.

1: An overflow or underflow has occurred.

This bit is driven by the SCF and RCF instructions and tested by the JRC and JRNC instructions. It is also affected by the "bit test and branch", shift and rotate instructions.

Interrupt management bits

Bits 5,3 = **I1, I0** Interrupt

The combination of the I1 and I0 bits gives the current interrupt software priority.

Table 6. Interrupt software priority

Priority	I1	I0
Level 0 (main)	1	0
Level 1	0	1
Level 2	0	0
Level 3 (= interrupt disable)	1	1

These two bits are set/cleared by hardware when entering in interrupt. The loaded value is given by the corresponding bits in the interrupt software priority registers (IxSPR). They can be also set/cleared by software with the RIM, SIM, IRET, HALT, WFI and PUSH/POP instructions.

See the interrupt management chapter for more details.

7.3 Multioscillator (MO)

The main clock of the ST7 can be generated by three different source types coming from the multioscillator block:

- an external source
- 4 crystal or ceramic resonator oscillators
- an internal high-accuracy RC oscillator

Each oscillator is optimized for a given frequency range in terms of consumption and is selectable through the option byte. The associated hardware configurations are shown in Table 8. Refer to Section 13: Electrical characteristics for more details.

Caution: The OSC1 and/or OSC2 pins must not be left unconnected. For the purposes of Failure Mode and Effect Analysis, it should be noted that if the OSC1 and/or OSC2 pins are left unconnected, the ST7 main oscillator may start and, in this configuration, could generate an f_{OSC} clock frequency in excess of the allowed maximum (>16 MHz), putting the ST7 in an unsafe/undefined state. The product behavior must therefore be considered undefined when the OSC pins are left unconnected.

7.3.1 External clock source

In this external clock mode, a clock signal (square, sinus or triangle) with ~50% duty cycle has to drive the OSC1 pin while the OSC2 pin is tied to ground.

7.3.2 Crystal/ceramic oscillators

This family of oscillators has the advantage of producing a very accurate rate on the main clock of the ST7. The selection within a list of 4 oscillators with different frequency ranges has to be done by option byte in order to reduce consumption (refer to Section 15.1: Option bytes for more details on the frequency ranges). In this mode of the multi-oscillator, the resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and startup stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

These oscillators are not stopped during the reset phase to avoid losing time in the oscillator startup phase.

8.7 External interrupt control register (EICR)

Reset value: 0000 0000 (00h)

7							0
IS11	IS10	IPB	IS21	IS20	IPA	0	0
Read/Write							

Bits 7:6 = **IS1[1:0]** ei2 and ei3 sensitivity

The interrupt sensitivity, defined using the IS1[1:0] bits, is applied to the following external interrupts:

ei2 (port B3..0)

Table 18. External interrupt sensitivity (ei2)

IS11	IS10	Sensitivity	
		IPB bit =0	IPB bit =1
0	0	Falling edge & low level	Rising edge & high level
0	1	Rising edge only	Falling edge only
1	0	Falling edge only	Rising edge only
1	1	Rising and falling edge	

ei3 (port B4)

Table 19. External interrupt sensitivity (ei3)

IS11	IS10	Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

Bit 5 = **IPB** Interrupt polarity for port B

This bit is used to invert the sensitivity of the port B [3:0] external interrupts. It can be set and cleared by software only when I1 and I0 of the CC register are both set to 1 (level 3).
0: No sensitivity inversion
1: Sensitivity inversion

Table 22. Nested interrupts register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0024h		ei1		ei0		MCC + SI		AWU	
	ISPR0 Reset value	I1_3 1	I0_3 1	I1_2 1	I0_2 1	I1_1 1	I0_1 1	I1_0 1	I0_0 1
0025h		I2C3SNS		I2C3SNS		ei3		ei2	
	ISPR1 Reset value	I1_7 1	I0_7 1	I1_6 1	I0_6 1	I1_5 1	I0_5 1	I1_4 1	I0_4 1
0026h		SCI		TIMER B		TIMER A		SPI	
	ISPR2 Reset value	I1_11 1	I0_11 1	I1_10 1	I0_10 1	I1_9 1	I0_9 1	I1_8 1	I0_8 1
0027h						I2C		AVD	
	ISPR3 Reset value	1	1	1	1	I1_13 1	I0_13 1	I1_12 1	I0_12 1
0028h	**EICR** Reset value	IS11 0	IS10 0	IPB 0	IS21 0	IS20 0	IPA 0	0	0

Figure 33. AWUF Halt timing diagram

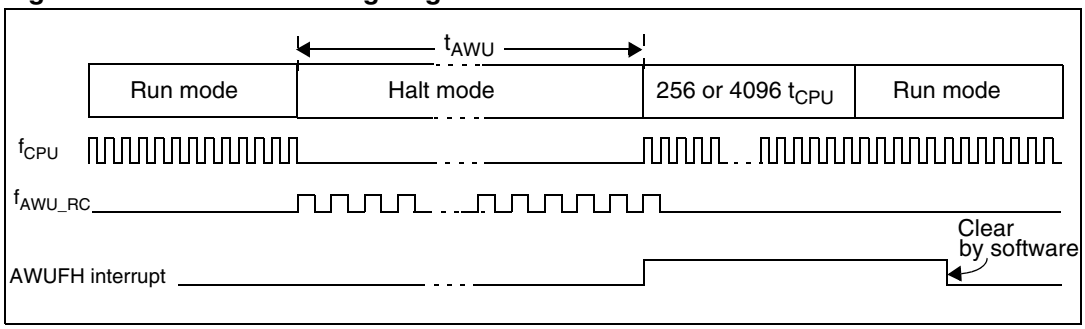
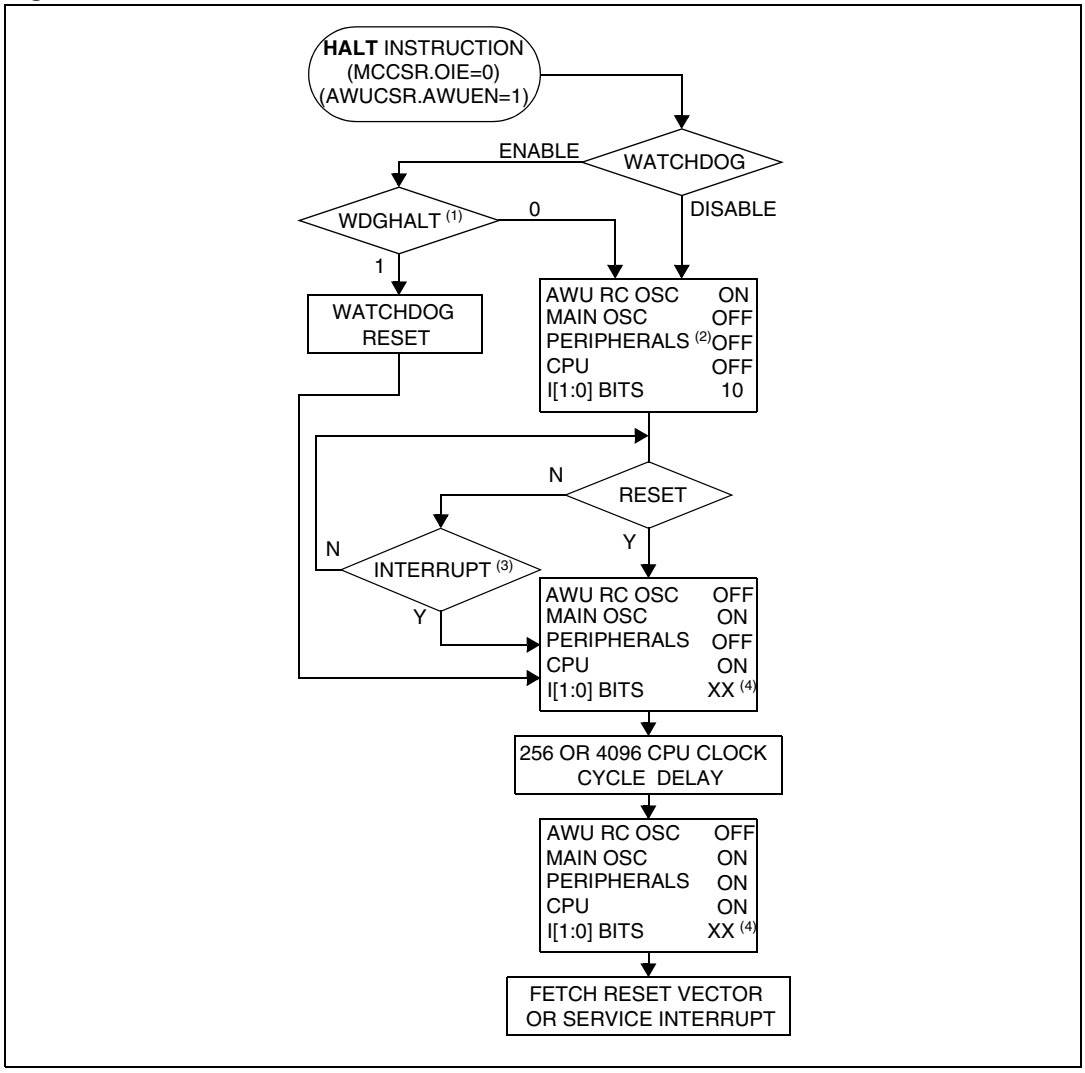


Figure 34. AWUFH mode flowchart



1. WDGHALT is an option bit. See Section 15.1: Option bytes for more details.
2. Peripheral clocked with an external clock source can still be active.
3. Only an AWUFH interrupt and some specific interrupts can exit the MCU from Halt mode (such as external interrupt). Refer to Table 17: Interrupt mapping for more details.
4. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and recovered when the CC register is popped.

11.2.7 Register description

MCC control/status register (MCCSR)

Reset value: 0000 0000 (00h)

7							0
MCO	CP1	CP0	SMS	TB1	TB0	OIE	OIF
Read/Write							

Bit 7 = **MCO** Main clock out selection

This bit enables the MCO alternate function on the PF0 I/O port. It is set and cleared by software.
0: MCO alternate function disabled (I/O pin free for general-purpose I/O)
1: MCO alternate function enabled (f_{CPU} on I/O port)

Note: To reduce power consumption, the MCO function is not active in Active-halt mode.

Bits 6:5 = **CP[1:0]** CPU clock prescaler

These bits select the CPU clock prescaler which is applied in the different slow modes. Their action is conditioned by the setting of the SMS bit. These two bits are set and cleared by software.

Table 41. CPU clock prescaler selection

f_{CPU} in Slow mode	CP1	CP0
f_{OSC2} / 2	0	0
f_{OSC2} / 4	0	1
f_{OSC2} / 8	1	0
f_{OSC2} / 16	1	1

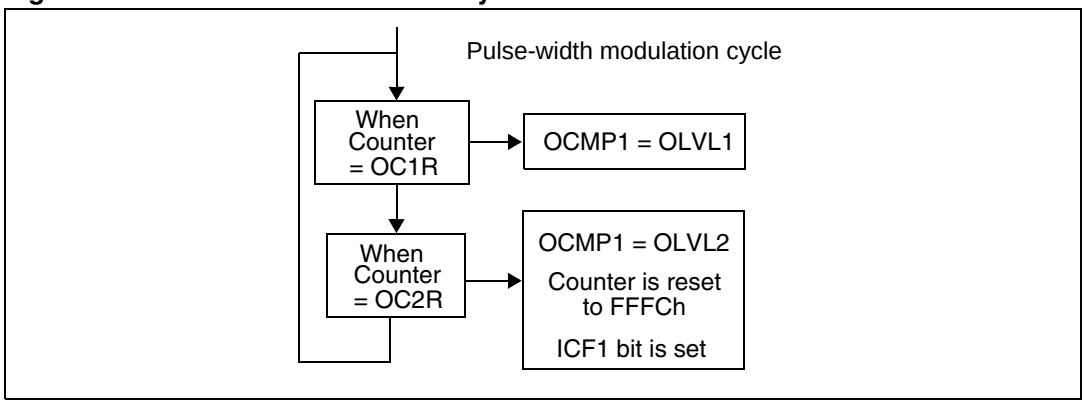
Bit 4 = **SMS** Slow mode select

This bit is set and cleared by software.
0: Normal mode. $f_{CPU} = f_{OSC2}$
1: Slow mode. f_{CPU} is given by CP1, CP0
See Section 9.2: Slow mode and Section 11.1: Window watchdog (WWDG) for more details.

Bits 3:2 = **TB[1:0]** Time base control

These bits select the programmable divider time base. They are set and cleared by software.

Figure 55. Pulse width modulation cycle



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 OCiR register value required for a specific timing application can be calculated using the following formula:

$$\text{OCiR Value} = \frac{t * f_{CPU}}{PRESC} - 5$$

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 50: Clock control bits)

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

$$\text{OCiR} = t * f_{EXT} - 5$$

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 54)

Note: 1 The OCF1 and OCF2 bits cannot be set by hardware in PWM mode; therefore, the Output Compare interrupt is inhibited.

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

3 In PWM mode, the ICAP1 pin cannot be used to perform an input capture because it is disconnected to the timer. The ICAP2 pin can be used to perform an 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 generate an interrupt if ICIE is set.

4 When the pulse-width modulation (PWM) and One-pulse mode (OPM) bits are both set, the PWM mode is the only active one.

11.4 Serial peripheral interface (SPI)

11.4.1 Introduction

The serial peripheral interface (SPI) allows full-duplex, synchronous, serial communication with external devices. An SPI system may consist of a master and one or more slaves, or a system in which devices may be either masters or slaves.

11.4.2 Main features

Full-duplex synchronous transfers (on three lines)

Simplex synchronous transfers (on two lines)

Master or slave operation

6 master mode frequencies (f_{CPU}/4 max.)

f_{CPU}/2 max. slave mode frequency (see the note)

$\overline{SS}$ Management by software or hardware

Programmable clock polarity and phase

End-of-transfer interrupt flag

Write collision, master mode fault and Overrun flags

Note: In slave mode, continuous transmission is not possible at maximum frequency due to the software overhead for clearing status flags and to initiate the next transmission sequence.

11.4.3 General description

Figure 56 on page 116 shows the serial peripheral interface (SPI) block diagram. There are three registers:

SPI Control Register (SPICR)

SPI Control/Status Register (SPICSR)

SPI Data Register (SPIDR)

The SPI is connected to external devices through four pins:

MISO: Master In / Slave Out data

MOSI: Master Out / Slave In data

SCK: Serial Clock out by SPI masters and input by SPI slaves

$\overline{SS}$: Slave select:
This input signal acts as a ‘chip select’ to let the SPI master communicate with slaves individually and to avoid contention on the data lines. Slave $\overline{SS}$ inputs can be driven by standard I/O ports on the master Device.

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

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

11.4.4 Clock phase and clock polarity

Four possible timing relationships may be chosen by software, using the CPOL and CPHA bits (See Figure 60).

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 60 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 and 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 resetting the SPE bit.

SCI clock tolerance

During reception, each bit is sampled 16 times. The majority of the 8th, 9th and 10th samples is considered as the bit value. For a valid bit detection, all the three samples should have the same value otherwise the noise flag (NF) is set. For example: if the 8th, 9th and 10th samples are 0, 1 and 1 respectively, then the bit value is “1”, but the Noise Flag bit is set because the three samples values are not the same.

Consequently, the bit length must be long enough so that the 8th, 9th and 10th samples have the desired bit value. This means the clock frequency should not vary more than 6/16 (37.5%) within one bit. The sampling clock is resynchronized at each start bit, so that when receiving 10 bits (one start bit, 1 data byte, 1 stop bit), the clock deviation must not exceed 3.75%.

Note: The internal sampling clock of the microcontroller samples the pin value on every falling edge. Therefore, the internal sampling clock and the time the application expects the sampling to take place may be out of sync. For example: If the baud rate is 15.625 kbaud (bit length is 64 µs), then the 8th, 9th and 10th samples will be at 28 µs, 32 µs and 36 µs respectively (the first sample starting ideally at 0 µs). But if the falling edge of the internal clock occurs just before the pin value changes, the samples would then be out of sync by ~4 µs. This means the entire bit length must be at least 40 µs (36 µs for the 10th sample + 4 µs for synchronization with the internal sampling clock).

Clock deviation causes

The causes which contribute to the total deviation are:

D_{TRA}: Deviation due to transmitter error (Local oscillator error of the transmitter or the transmitter is transmitting at a different baud rate).

D_{QUANT}: Error due to the baud rate quantization of the receiver.

D_{REC}: Deviation of the local oscillator of the receiver: This deviation can occur during the reception of one complete SCI message assuming that the deviation has been compensated at the beginning of the message.

D_{TCL}: Deviation due to the transmission line (generally due to the transceivers)

All the deviations of the system should be added and compared to the SCI clock tolerance:

$D_{TRA} + D_{QUANT} + D_{REC} + D_{TCL} < 3.75\%$

Noise error causes

See also description of noise error in Receiver.

Start bit: the noise flag (NF) is set during start bit reception if one of the following conditions occurs:

– A valid falling edge is not detected. A falling edge is considered to be valid if the three consecutive samples before the falling edge occurs are detected as '1' and,

new interrupt. Software must ensure either that the SCL line is back at 0 before reading the SR2 register, or be able to correctly handle a second interrupt during the 9th pulse of a transmitted byte.

Note: In both cases, SCL line is not held low; however, the SDA line can remain low if the last bits transmitted are all 0. It is then necessary to release both lines by software. The SCL line is not held low while AF=1 but by other flags (SB or BTF) that are set at the same time.

How to release the SDA / SCL lines

Set and subsequently clear the STOP bit while BTF is set. The SDA/SCL lines are released after the transfer of the current byte.

SMBus compatibility

ST7 I^2C is compatible with SMBus V1.1 protocol. It supports all SMBus addressing modes, SMBus bus protocols and CRC-8 packet error checking. Refer to AN1713: SMBus Slave Driver For ST7 I^2C Peripheral.

Master mode

To switch from default Slave mode to Master mode a Start condition generation is needed.

Start condition

Setting the START bit while the BUSY bit is cleared causes the interface to switch to Master mode (M/SL bit set) and generates a Start condition.

Once the Start condition is sent:

The EVF and SB bits are set by hardware with an interrupt if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the DR register with the Slave address, **holding the SCL line low** (see Figure 69 Transfer sequencing EV5).

Slave address transmission

Then the slave address is sent to the SDA line via the internal shift register.

In 7-bit addressing mode, one address byte is sent.

In 10-bit addressing mode, sending the first byte including the header sequence causes the following event:

The EVF bit is set by hardware with interrupt generation if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the DR register, **holding the SCL line low** (see Figure 69 Transfer sequencing EV9).

Then the second address byte is sent by the interface.

After completion of this transfer (and acknowledge from the slave if the ACK bit is set):

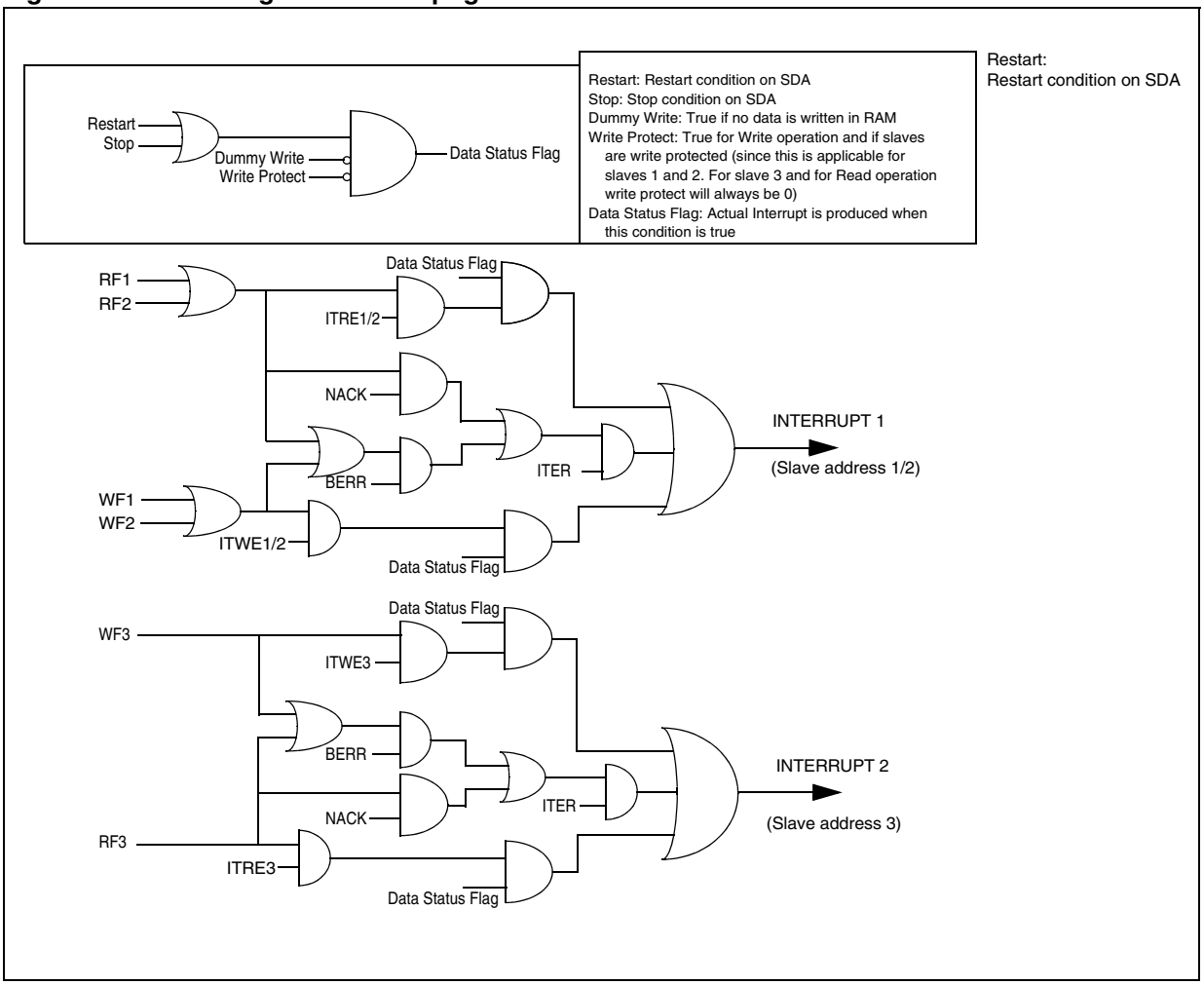
The EVF bit is set by hardware with interrupt generation if the ITE bit is set.

Then the master waits for a read of the SR1 register followed by a write in the CR register (for example set PE bit), **holding the SCL line low** (see Figure 69 Transfer sequencing EV6).

Next the master must enter Receiver or Transmitter mode.

11.7.7 Interrupt generation

Figure 83. Event flags and interrupt generation



Note: Read/Write interrupts are generated only after stop or restart conditions. Figure 83 shows the conditions for the generation of the two interrupts.

Table 69. Interrupt events

Interrupt event	Flag	Enable control bit	Exit from wait	Exit from halt
Interrupt on write to Slave 1	WF1	ITWE1	Yes	No
Interrupt on write to Slave 2	WF2	ITWE1	Yes	No
Interrupt on write to Slave 3	WF3	ITWE2	Yes	No
Interrupt on Read from Slave 1, Slave 2 or Slave 3.	RF1- RF3	ITREx	Yes	No
Errors	BERR, NACK	ITER	Yes	No

Table 79. Instructions supporting direct, indexed, indirect and indirect indexed addressing modes

Long and short instructions	Function
LD	Load
CP	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Addition/subtraction operations
BCP	Bit Compare

Table 80. Short instructions

Short instructions only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
BSET, BRES	Bit Operations
BTJT, BTJF	Bit Test and Jump Operations
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles
CALL, JP	Call or Jump subroutine

12.1.7 Relative Mode (direct, indirect)

This addressing mode is used to modify the PC register value by adding an 8-bit signed offset to it.

Table 81. Relative direct/indirect instructions

Available relative direct/indirect instructions	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two submodes:

Relative (Direct)

The offset follows the opcode.

Relative (Indirect)

The offset is defined in memory, of which the address follows the opcode.

13.4 Internal RC oscillator characteristics

Table 92. Internal RC oscillator characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{RC}	Internal RC oscillator frequency (1)	RCCR = FF (reset value), T_A = 25 °C, V_{DD} = 5 V		625		kHz
		RCCR = RCCR0 (2), T_A = 25 °C, V_{DD} = 5 V		1000		
		RCCR = FF (reset value), T_A = 25 °C, V_{DD} = 3 V		612		
		RCCR = RCCR1(2), T_A = 25 °C, V_{DD} = 3 V		1000		
ACC_{RC}	Accuracy of Internal RC oscillator with RCCR=RCCR0 (2)	T_A = 25 °C, V_{DD} = 5 V	-1		+1	%
		T_A = 25 °C, V_{DD} = 4.5 to 5.5 V (3)	-1		+1	%
		T_A = 25 to +85 °C, V_{DD} = 5 V (3)	-3		+3	%
		T_A = 25 to +85 °C, V_{DD} = 4.5 to 5.5 V (3)	-3.5		+3.5	%
		T_A = -40 to +25 °C, V_{DD} = 4.5 to 5.5 V (3)	-3		+7	%
$I_{DD(RC)}$	RC oscillator current consumption	T_A = 25 °C, V_{DD} = 5 V		600 (3)		µA
$t_{su(RC)}$	RC oscillator setup time	T_A = 25 °C, V_{DD} = 5 V			10 (2)	µs

1. If the RC oscillator clock is selected, to improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100 nF, between the V_{DD} and V_{SS} pins as close as possible to the ST7 device. It is also recommended to perform the calibration on board.
2. See Internal RC oscillator.
3. Expected results. Data based on characterization, not tested in production.

Figure 88. Typical RC frequency vs. RCCR

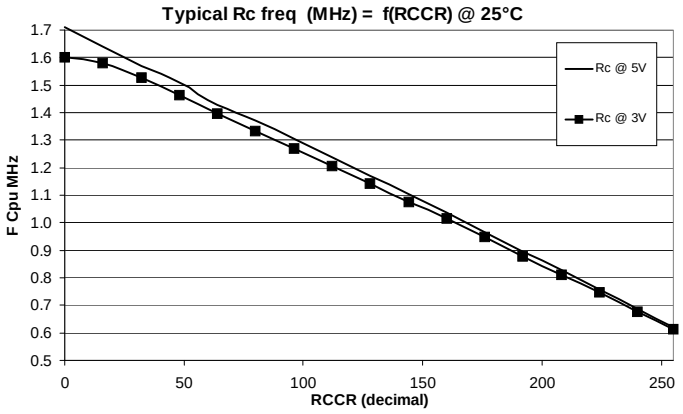


Figure 111. Typical $V_{DD} - V_{OH}$ at V_{DD} = 3 V (std I/Os)

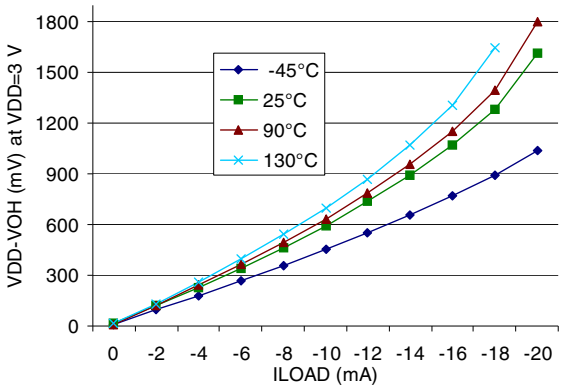


Figure 112. Typical $V_{DD} - V_{OH}$ at V_{DD} = 4 V (std)

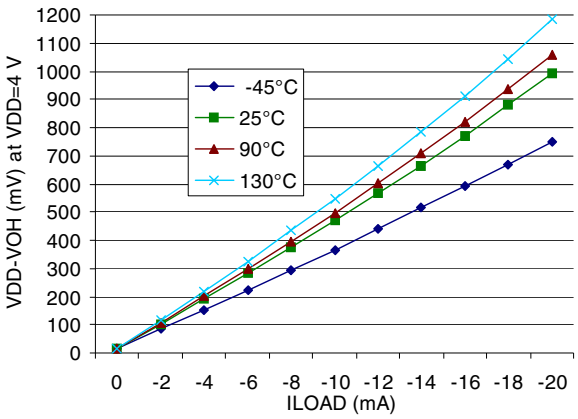


Figure 113. Typical $V_{DD} - V_{OH}$ at V_{DD} = 5 V (std)

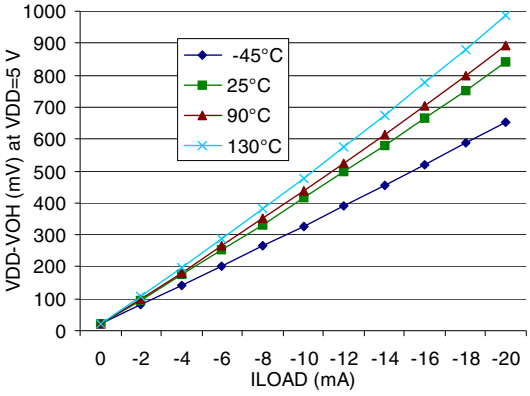
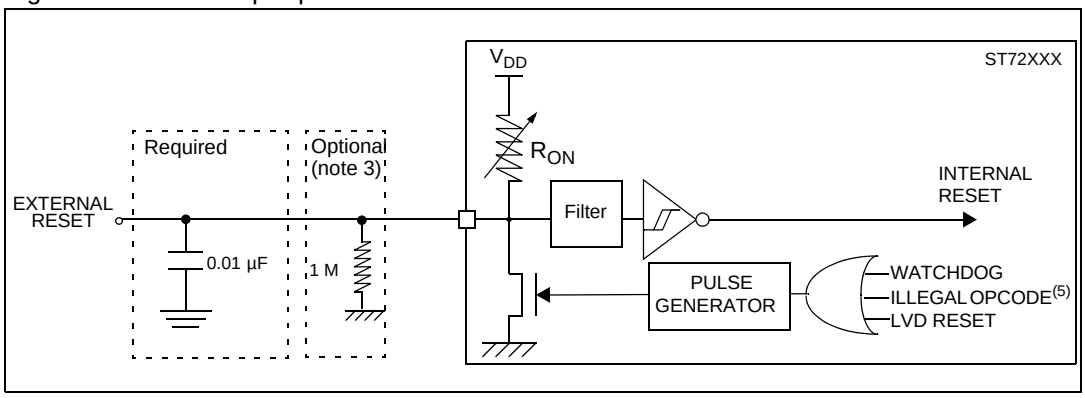
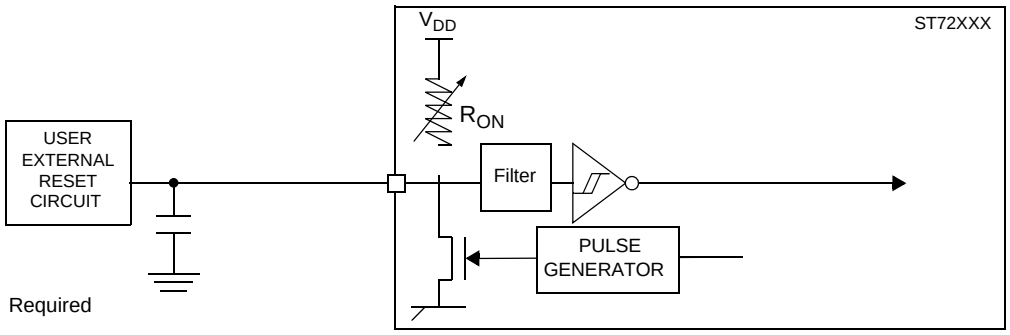


Figure 115. $\overline{\text{RESET}}$ pin protection when LVD is enabled (1)(2)(3)(4)



1. The reset network protects the device against parasitic resets.
   - The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).
   - Whatever the reset source is (internal or external), the user must ensure that the level on the $\overline{\text{RESET}}$ pin can go below the V_{IL} max. level specified in Section 13.10.1. Otherwise the reset will not be taken into account internally.
   - Because the reset circuit is designed to allow the internal reset to be output in the $\overline{\text{RESET}}$ pin, the user must ensure that the current sunk on the $\overline{\text{RESET}}$ pin is less than the absolute maximum value specified for $I_{INJ(RESET)}$ in Table 85.
2. When the LVD is enabled, it is recommended not to connect a pull-up resistor or capacitor. A 10nF pull-down capacitor is required to filter noise on the reset line.
3. In case a capacitive power supply is used, it is recommended to connect a 1M pull-down resistor to the $\overline{\text{RESET}}$ pin to discharge any residual voltage induced by the capacitive effect of the power supply (this will add 5µA to the power consumption of the MCU).
4. Tips when using the LVD:
   1. Check that all recommendations related to the reset circuit have been applied (see notes above)
   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 + 1M pull-down on the $\overline{\text{RESET}}$ pin.
   3. The capacitors connected on the $\overline{\text{RESET}}$ 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."
5. Please refer to Section 12.2.1: Illegal opcode reset for more details on illegal opcode reset conditions.

Figure 116. $\overline{\text{RESET}}$ pin protection when LVD is disabled (1)



1. The reset network protects the device against parasitic resets.
   - The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).
   - Whatever the reset source is (internal or external), the user must ensure that the level on the $\overline{\text{RESET}}$ pin can go below the V_{IL} max. level specified in Section 13.10.1. Otherwise the reset will not be taken into account internally.
   - Because the reset circuit is designed to allow the internal reset to be output in the $\overline{\text{RESET}}$ pin, the user must ensure that the current sunk on the $\overline{\text{RESET}}$ pin is less than the absolute maximum value specified for $I_{INJ(RESET)}$ in Table 85.
2. Please refer to Section 12.2.1: Illegal opcode reset for more details on illegal opcode reset conditions.

15 Device configuration and ordering information

Each device is available for production in user programmable versions (Flash).

ST72F34x Flash devices are shipped to customers with a default content (FFh). This implies that Flash devices have to be configured by the customer using the Option Bytes.

15.1 Option bytes

The four option bytes allow the hardware configuration of the microcontroller to be selected.

The option bytes can be accessed only in programming mode (for example using a standard ST7 programming tool).

15.1.1 Option byte 0

OPT7 = **WDG Halt** Watchdog Reset on Halt

This option bit determines if a reset is generated when entering Halt mode while the Watchdog is active.
0: No Reset generation when entering Halt mode
1: Reset generation when entering Halt mode

OPT6 = **WDG SW** Hardware or Software Watchdog

This option bit selects the watchdog type.
0: Hardware (watchdog always enabled)
1: Software (watchdog to be enabled by software)

OPT5:4 = **LVD[1:0]** Low voltage detection selection

These option bits enable the LVD block with a selected threshold as shown in Table 119.

Table 119. LVD threshold configuration

Configuration	LVD1	LVD0
LVD Off	1	1
Highest voltage threshold (~4.1V)	1	0
Medium voltage threshold (~3.5V)	0	1
Lowest voltage threshold (~2.8V)	0	0

OPT3:2 = **SEC[1:0]** Sector 0 size definition

These option bits indicate the size of sector 0 according to the following table.

Table 120. Size of sector 0

Sector 0 size	SEC1	SEC0
0.5k	0	0
1k	0	1

17 Revision history

Table 126. Document revision history

Date	Revision	Changes
29-April-2006	1	First release on internet
23-Oct-2006	2	Removed references to BGA56 and QFN40 packages TQFP package naming changed to LQFP (Low-profile Quad Flat) Changed number of I/O ports on first page PDVD (Power Down Voltage Detector) replaced by AVD (Auxiliary Voltage Detector) Modified note 3 to Table 4 on page 24 Added PF4 to Figure 3 on page 18 and Figure 4 on page 19 Modified Memory access on page 31 Modified Figure 8, Figure 9 on page 33 and Figure 10 on page 34 Changed RCCR table in Section 7.3 on page 43 (f_{RC}=1 MHz) References to PDVDF, PDVDIE corrected to AVDF, AVDIE: Section 7.6.2 on page 49 Current characteristics Table 85 on page 201 updated General operating conditions table updated, Table 87 on page 202 Data updated in Table 88 on page 202, note replaced Table modified in Table 89 on page 203 Notes adjusted for table in Table 90 on page 203 Modified Section 13.4 on page 204 (for V_{DD}=5V) Table in Table 93 on page 205 modified Updated Table 94 on page 208 Added Table 96 on page 209 and Figure 96 on page 209 Table in Table 101 on page 212 modified Absolute maximum ratings and electrical sensitivity table updated, Section 13.8.3 on page 214 Added note 1 to V_{IL} and V_{IH} in Table 107 on page 215 Table in Table 108 on page 216 modified (for V_{DD}= 3.3V and V_{DD}=2.7V) Modified graphs in Table 108 on page 216 $t_{g(RSTL)in}$ updated in Table 13.10 on page 222 Updated Table 111 on page 224 Updated Table 118 on page 231 Modified default values for option byte 2 and 3 on Option byte 2 on page 234 Added option list on Option byte 2 on page 234 Added Section 15.3: Development tools on page 236 Added known limitations: Section 16.6: In-application programming on page 242, Section 16.7: Programming of EEPROM data on page 243, and Section 16.8: Flash write/erase protection on page 243 Modified Section 16.7 on page 243 Changed status of the document (datasheet instead of preliminary data)