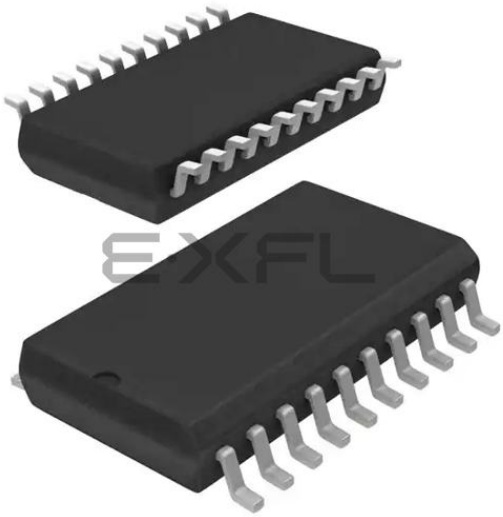


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	15
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	A/D 7x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SO
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7flite29f2m7tr

List of figures

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.

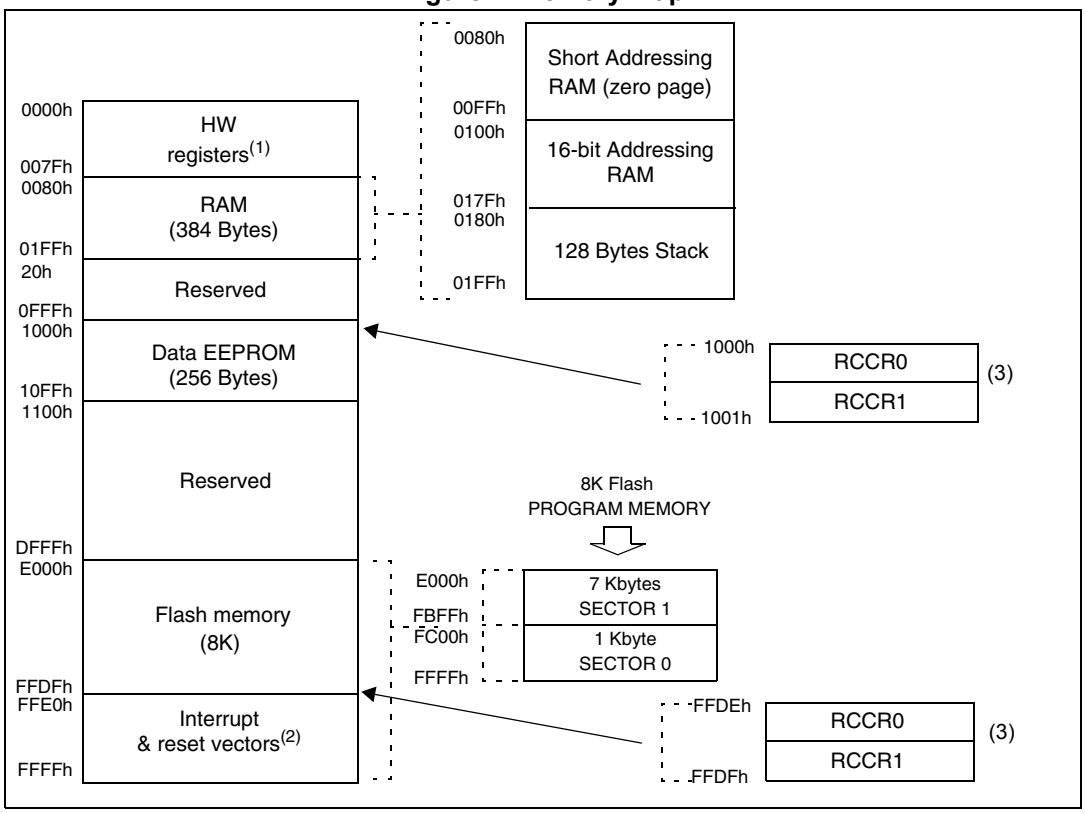
The highest address bytes contain the user reset and interrupt vectors.

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*

Write/erase protection is enabled through the FMP_W bit in the option byte.

4.6 Related documentation

For details on Flash programming and ICC protocol, refer to the ST7 Flash Programming Reference Manual and to the ST7 ICC Protocol Reference Manual.

4.7 Register description

Flash control/status register (FCSR)

Read / Write

Reset value: 0000 0000 (00h)

1st RASS Key: 0101 0110 (56h)

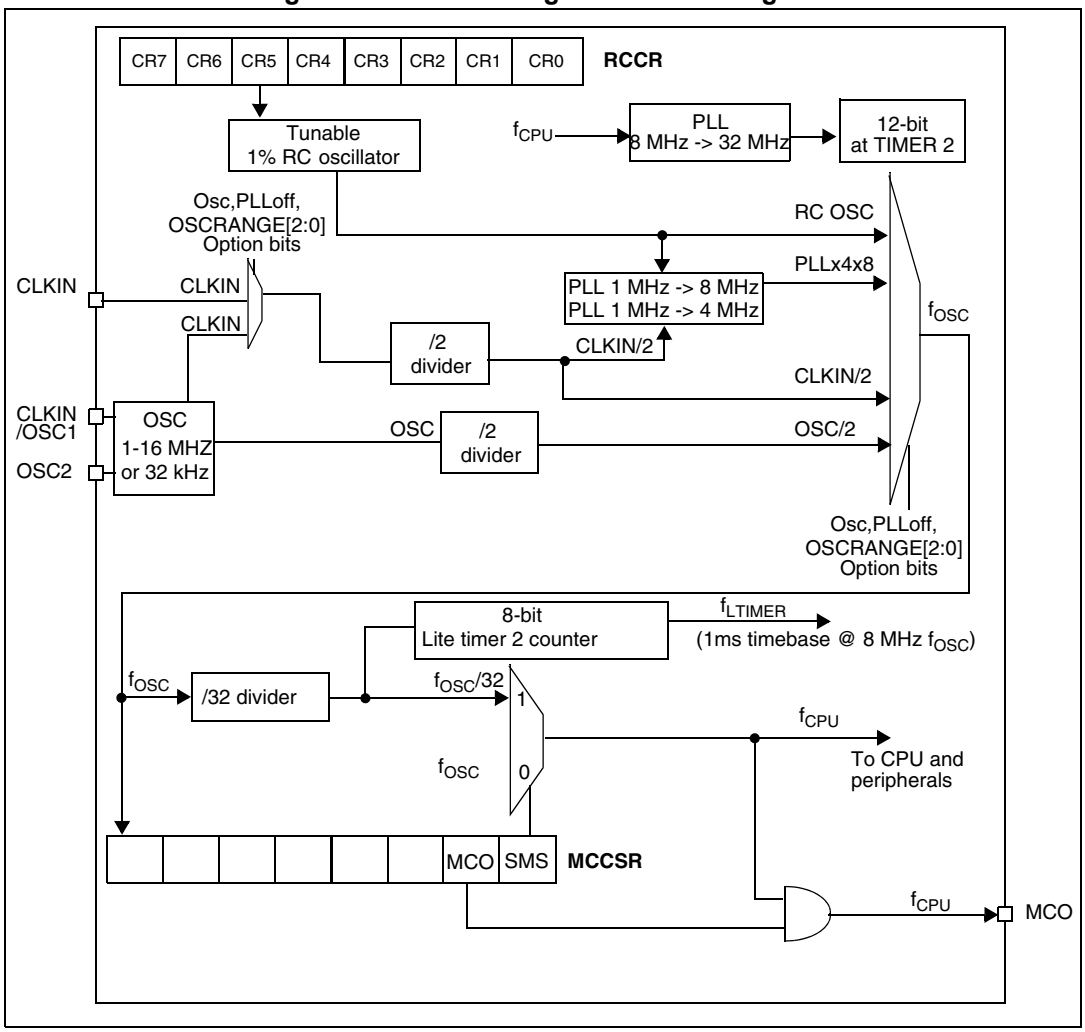
2nd RASS Key: 1010 1110 (AEh)

7							0
0	0	0	0	0	OPT	LAT	PGM

Note: This register is reserved for programming using ICP, IAP or other programming methods. It controls the XFlash programming and erasing operations.

When an EPB or another programming tool is used (in socket or ICP mode), the RASS keys are sent automatically.

Figure 13. Clock management block diagram



7.4 Multi-oscillator (MO)

The main clock of the ST7 can be generated by four different source types coming from the multioscillator block (1 to 16MHz or 32kHz):

- an external source
- 5 crystal or ceramic resonator oscillators
- an internal high frequency 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 7*.

Note: Refer to Section 13: Electrical characteristics for more details.

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.

7.5 Reset sequence manager (RSM)

7.5.1 Introduction

The reset sequence manager includes three RESET sources as shown in *Figure 15: Reset block diagram*:

- External $\overline{\text{RESET}}$ source pulse
- Internal LVD RESET (low voltage detection)
- Internal WATCHDOG RESET

Note: A reset can also be triggered following the detection of an illegal opcode or prebyte code. Refer to Section 12.2.1: Illegal opcode reset for further details.

These sources act on the $\overline{\text{RESET}}$ pin and it is always kept low during the delay phase.

The RESET service routine vector is fixed at addresses FFFEh-FFFFh in the ST7 memory map.

The basic RESET sequence consists of 3 phases as shown in *Figure 14*:

- Active Phase depending on the RESET source
- 256 or 4096 CPU clock cycle delay (see table below)
- RESET vector fetch.

The 256 or 4096 CPU clock cycle delay allows the oscillator to stabilise and ensures that recovery has taken place from the Reset state. The shorter or longer clock cycle delay is automatically selected depending on the clock source chosen by option byte:

Table 8. CPU clock cycle delay

Clock source	CPU clock cycle delay
Internal RC oscillator	256
External clock (connected to CLKIN pin)	256
External crystal/ceramic oscillator (connected to OSC1/OSC2 pins)	4096

The RESET vector fetch phase duration is 2 clock cycles.

If the PLL is enabled by option byte, it outputs the clock after an additional delay of $t_{STARTUP}$ (see *Figure 12: PLL output frequency timing diagram*).

Figure 14. RESET sequence phases

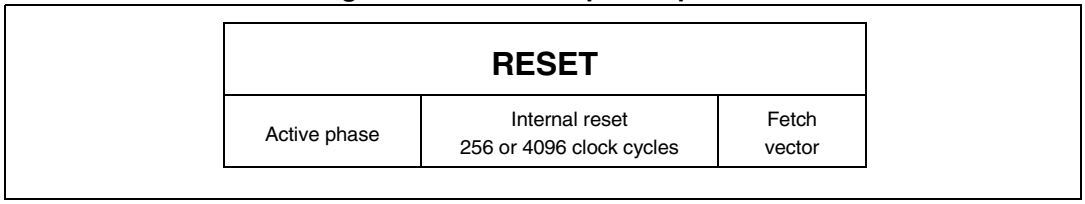
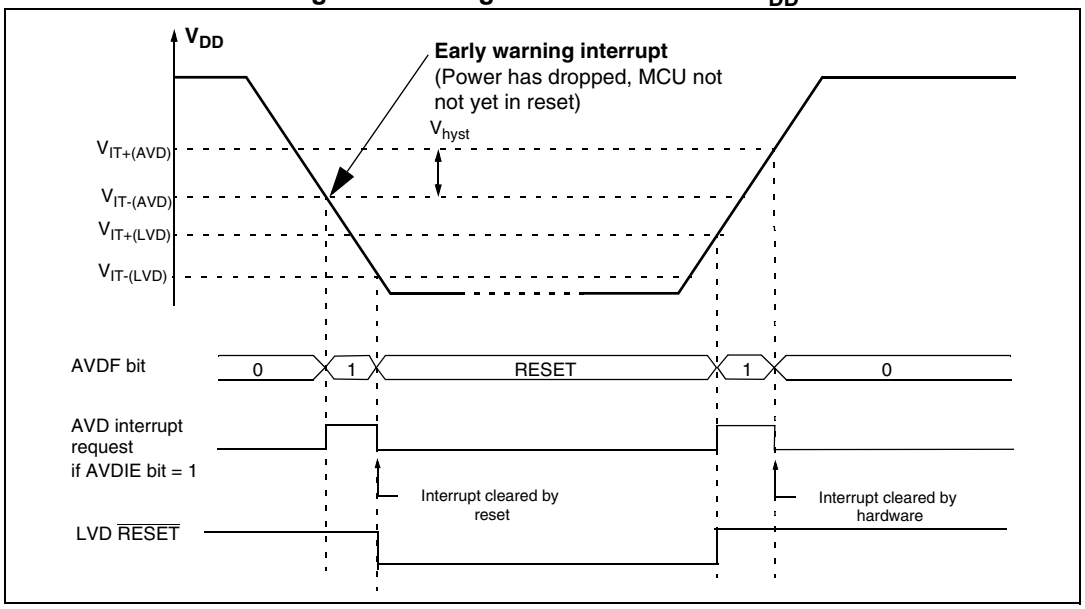


Figure 19. Using the AVD to monitor V_{DD}



7.6.3 Low power modes

Table 9. Effect of low power modes on SI

Mode	Description
WAIT	No effect on SI. AVD interrupts cause the device to exit from WAIT mode.
HALT	The SICSR register is frozen. The AVD remains active.

Interrupts

The AVD interrupt event generates an interrupt if the corresponding enable control bit (AVDIE) is set and the interrupt mask in the CC register is reset (RIM instruction).

Table 10. Interrupt control bits

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
AVD event	AVDF	AVDIE	Yes	No

9 Power saving modes

9.1 Introduction

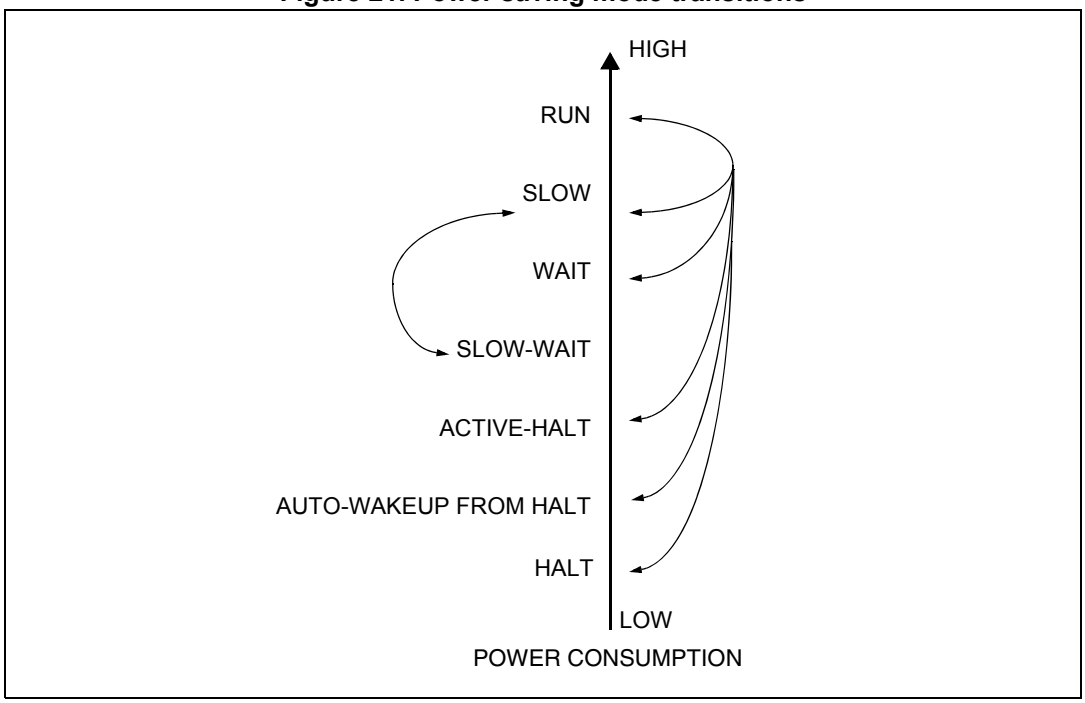
To give a large measure of flexibility to the application in terms of power consumption, five main power saving modes are implemented in the ST7 (see *Figure 21*):

- Slow
- Wait (and Slow-Wait)
- Active Halt
- Auto Wake up From Halt (AWUF)
- Halt

After a RESET the normal operating mode is selected by default (Run mode). This mode drives the device (CPU and embedded peripherals) by means of a master clock which is based on the main oscillator frequency divided or multiplied by 2 (f_{OSC2}).

From RUN mode, the different power saving modes may be selected by setting the relevant register bits or by calling the specific ST7 software instruction whose action depends on the oscillator status.

Figure 21. Power saving mode transitions



9.2 SLOW mode

This mode has two targets:

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

interrupt). Refer to *Table 12: 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.
5. If the PLL is enabled by option byte, it outputs the clock after an additional delay of $t_{STARTUP}$ (see *Figure 12: PLL output frequency timing diagram*).

9.6.1 Register description

AWUF control/status register (CR)

Read/Write

Reset value: 0000 0000 (0Ch)

7							0
0	0	0	0	0	AWUF	AWUM	AWUEN

- Bits 7:3 = Reserved
- Bit 2 = **AWUF** *Auto-Wakeup Flag*
  This bit is set by hardware when the AWU module generates an interrupt and cleared by software on reading AWUCSR. Writing to this bit does not change its value.
  0: No AWU interrupt occurred
  1: AWU interrupt occurred
- Bit 1= **AWUM** *Auto-Wakeup Measurement*
  This bit enables the AWU RC oscillator and connects its output to the input capture of the 12-bit Auto-Reload timer. This allows the timer to be used to measure the AWU RC oscillator dispersion and then compensate this dispersion by providing the right value in the AWUPRE register.
  0: Measurement disabled
  1: Measurement enabled
- Bit 0 = **AWUEN** *Auto Wake Up From Halt Enabled*
  This bit enables the Auto Wake Up From Halt feature:
  once HALT mode is entered, the AWUF wakes up the microcontroller after a time delay dependent on the AWU prescaler value. It is set and cleared by software.
  0: AWUF (Auto Wake Up From Halt) mode disabled
  1: AWUF (Auto Wake Up From Halt) mode enabled

10.2.3 Alternate functions

Many ST7s I/Os have one or more alternate functions. These may include output signals from, or input signals to, on-chip peripherals. The Device Pin Description table describes which peripheral signals can be input/output to which ports.

A signal coming from an on-chip peripheral can be output on an I/O. To do this, enable the on-chip peripheral as an output (enable bit in the peripheral's control register). The peripheral configures the I/O as an output and takes priority over standard I/O programming. The I/O's state is readable by addressing the corresponding I/O data register.

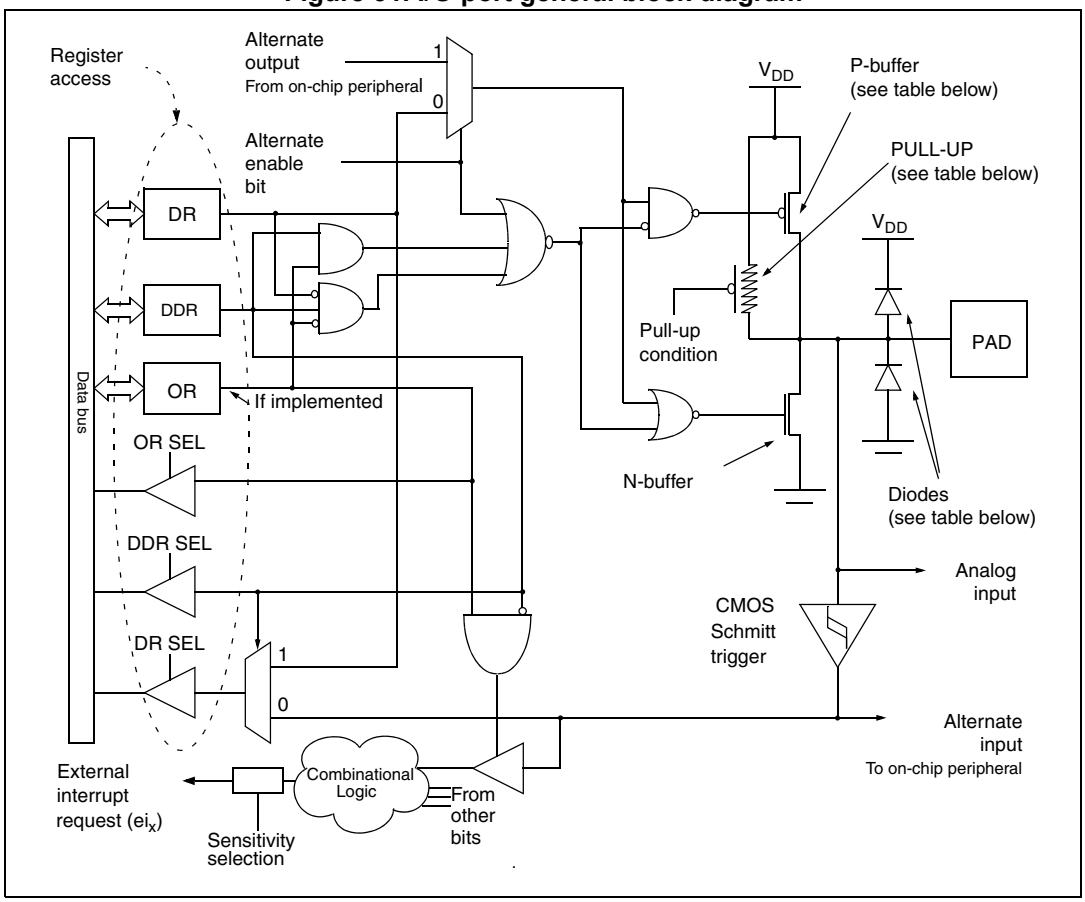
Configuring an I/O as floating enables alternate function input. It is not recommended to configure an I/O as pull-up as this will increase current consumption.

Before using an I/O as an alternate input, configure it without interrupt. Otherwise spurious interrupts can occur.

Configure an I/O as input floating for an on-chip peripheral signal which can be input and output.

Caution: I/Os which can be configured as both an analog and digital alternate function need special attention.
The user must control the peripherals so that the signals do not arrive at the same time on the same pin. If an external clock is used, only the clock alternate function should be employed on that I/O pin and not the other alternate function.

Figure 31. I/O port general block diagram



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.

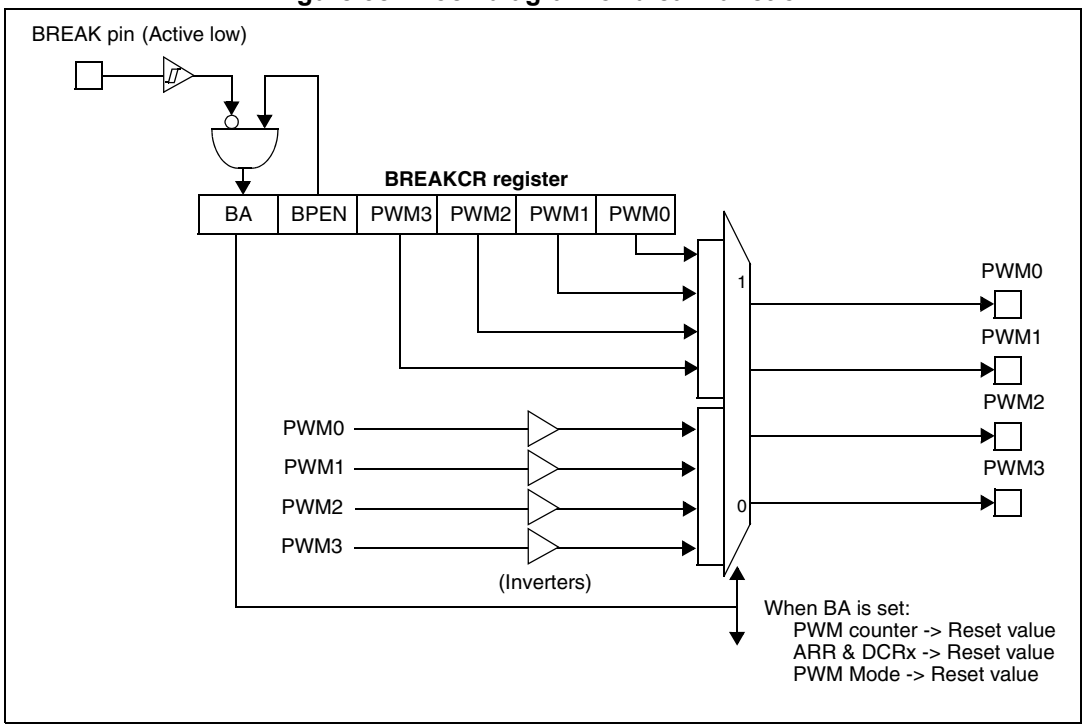
11.1.5 Interrupts

None.

Software can set the BA bit to activate the break function without using the BREAK pin.

- When the break function is activated (BA bit =1):
  - the break pattern (PWM[3:0] bits in the BREAKCR) is forced directly on the PWMx output pins (after the inverter),
  - the 12-bit PWM counter is set to its reset value,
  - the ARR, DCRx and the corresponding shadow registers are set to their reset values,
  - the PWMCR register is reset.
- When the break function is deactivated after applying the break (BA bit goes from 1 to 0 by software):
  - the control of PWM outputs is transferred to the port registers.

Figure 38. Block diagram of break function



Note: The BREAK pin value is latched by the BA bit.

Input capture

The 12-bit ATICR register is used to latch the value of the 12-bit free running upcounter after a rising or falling edge is detected on the ATIC pin.

When an input capture occurs, the ICF bit is set and the ATICR register contains the value of the free running upcounter. An IC interrupt is generated if the ICIE bit is set. The ICF bit is reset by reading the ATICR register when the ICF bit is set. The ATICR is a read only register and always contains the free running upcounter value which corresponds to the most recent input capture. Any further input capture is inhibited while the ICF bit is set.

Autoreload register (ATRL)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
ATR7	ATR6	ATR5	ATR4	ATR3	ATR2	ATR1	ATR0

PWM output control register (PWMCR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	OE3	0	OE2	0	OE1	0	OE0

- Bits 7:0 = **OE[3:0]** *PWMx output enable*
  These bits are set and cleared by software and cleared by hardware after a reset.
  0: PWM mode disabled. PWMx output alternate function disabled: I/O pin free for general purpose I/O after an overflow event.
  1: PWM mode enabled

PWMx control status register (PWMxCSR)

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	OPx	OE0

- Bits 7:2 = Reserved, must be kept cleared
- Bit 1 = **OPx** *PWMx Output Polarity*
  This bit is read/write by software and cleared by hardware after a reset. This bit selects the polarity of the PWM signal.
  0: The PWM signal is not inverted
  1: The PWM signal is inverted
- Bit 0 = **CMPFx** PWMx *Compare Flag*
  This bit is set by hardware and cleared by software by reading the PWMxCSR register. It indicates that the upcounter value matches the DCRx register value.
  0: Upcounter value does not match DCR value.
  1: Upcounter value matches DCR value

Break control register (BREAKCR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	BA	BPEN	PWM3	PWM2	PWM1	PWM0

- Bits 7:6 = Reserved. Forced by hardware to 0.
- Bit 5 = **BA** *Break Active*
  This bit is read/write by software, cleared by hardware after reset and set by hardware when the BREAK pin is low. It activates/deactivates the Break function.
  0: Break not active
  1: Break active
- Bit 4 = **BPEN** *Break pin enable*
  This bit is read/write by software and cleared by hardware after Reset.
  0: Break pin disabled
  1: Break pin enabled
- Bits 3:0 = **PWM[3:0]** *Break pattern*
  These bits are read/write by software and cleared by hardware after a reset. They are used to force the four PWMx output signals into a stable state when the Break function is active.

PWMx duty cycle register high (DCRxH)

Read / Write

Reset Value: 0000 0000 (00h)

15							8
0	0	0	0	DCR11	DCR10	DCR9	DCR8

PWMx duty cycle register low (DCRxL)

Read / Write

Reset value: 0000 0000 (00h)

7							0
DCR7	DCR6	DCR5	DCR4	DCR3	DCR2	DCR1	DCR0

- Bits 15:12 = Reserved
- Bits 11:0 = **DCR[11:0]** *PWMx duty cycle value*
  This 12-bit value is written by software. It defines the duty cycle of the corresponding PWM output signal (see *Figure 36*).
  In PWM mode (OEx=1 in the PWMCR register) the DCR[11:0] bits define the duty cycle of the PWMx output signal (see *Figure 36*). In Output Compare mode, they define the value to be compared with the 12-bit upcounter value.

0: Timebase period = t_{OSC} * 8000 (1 ms @ 8 MHz)
1: Timebase period = t_{OSC} * 16000 (2 ms @ 8 MHz)

- Bit 4 = **TB1IE** *Timebase interrupt enable*
  This bit is set and cleared by software.
  0: Timebase (TB1) interrupt disabled
  1: Timebase (TB1) interrupt enabled
- Bit 3 = **TB1F** *Timebase interrupt flag*
  This bit is set by hardware and cleared by software reading the LTCSR register. Writing to this bit has no effect.
  0: No counter overflow
  1: A counter overflow has occurred
- Bit 2:0 = reserved.

Lite timer input capture register (LTICR)

Read only

Reset Value: 0000 0000 (00h)

7							0
ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0

- Bits 7:0 = **ICR[7:0]** *Input capture value*
  These bits are read by software and cleared by hardware after a reset. If the ICF bit in the LTCSR is cleared, the value of the 8-bit up-counter will be captured when a rising or falling edge occurs on the LTIC pin.

Table 39. Lite timer register map and reset values

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
08	**LTCSR2**							TB2IE	TB2F
	Reset Value	0	0	0	0	0	0	0	0
09	**LTARR**	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
	Reset Value	0	0	0	0	0	0	0	0
0A	**LTCNTR**	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
	Reset Value	0	0	0	0	0	0	0	0
0B	**LTCSR1**	ICIE	ICF	TB	TB1IE	TB1F			
	Reset Value	0	x	0	0	0	0	0	0
0C	**LTICR**	ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0
	Reset Value	0	0	0	0	0	0	0	0

1. The MODF bit is set and an SPI interrupt request is generated if the SPIE bit is set.
2. The SPE bit is reset. This blocks all output from the Device and disables the SPI peripheral.
3. The MSTR bit is reset, thus forcing the Device into slave mode.

Clearing the MODF bit is done through a software sequence:

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

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

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

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

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

Overrun condition (OVR)

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

When an overrun occurs:

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

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

The OVR bit is cleared by reading the SPICSR register.

Write collision error (WCOL)

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

Write collisions can occur both in master and slave mode. See also *Slave select management on page 93*.

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

The WCOL bit in the SPICSR register is set if a write collision occurs. No SPI interrupt is generated when the WCOL bit is set (the WCOL bit is a status flag only).

Clearing the WCOL bit is done through a software sequence (see *Figure 47: Clearing the WCOL bit (write collision flag) software sequence*).

Table 69. Auto Wakeup from Halt Oscillator (AWU)

Symbol	Parameter(1)	Conditions	Min	Typ	Max	Unit
f_{AWU}	AWU Oscillator Frequency	–	50	125	250	kHz
t_{RCSRT}	AWU Oscillator startup time	–	–	–	50	µs

1. Guaranteed by design.

13.5.1 Crystal and ceramic resonator oscillators

The ST7 internal clock can be supplied with eight different Crystal/Ceramic resonator oscillators. All the information given in this paragraph are based on characterization results with specified typical external components. In the application, 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 start-up stabilization time. Refer to the crystal/ceramic resonator manufacturer for more details (frequency, package, accuracy...).

14 Package characteristics

14.1 Package mechanical data

Figure 93. 20-pin plastic small outline package, 300-mil width

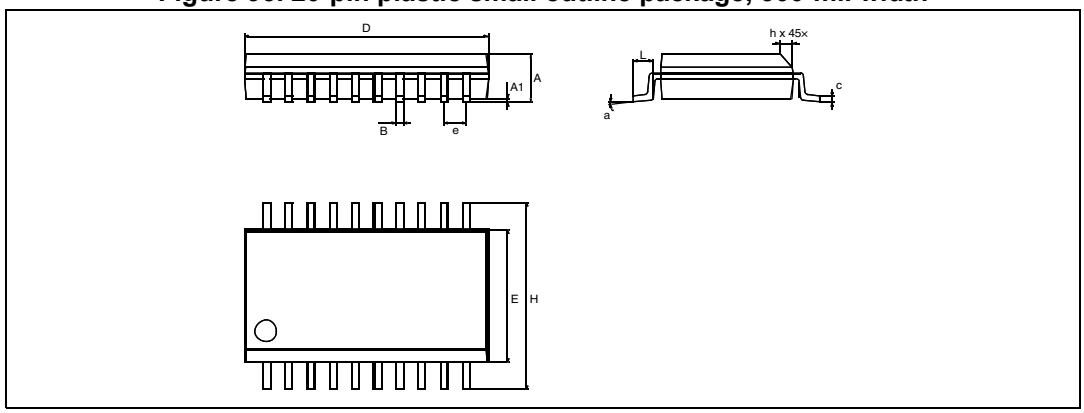


Table 88. Small outline package characteristics

Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	2.35	–	2.65	0.093	–	0.104
A1	0.10	–	0.30	0.004	–	0.012
B	0.33	–	0.51	0.013	–	0.020
C	0.23	–	0.32	0.009	–	0.013
D(1)	12.60	–	13.00	0.496	–	0.512
E	7.40	–	7.60	0.291	–	0.299
e	–	1.27	–	–	0.050	–
H	10.00	–	10.65	0.394	–	0.419
h	0.25	–	0.75	0.010	–	0.030
α	0°	–	8°	0°	–	8°
L	0.40	–	1.27	0.016	–	0.050
	Number of Pins					
N	20					

1. *Dimension "D" does not include mold flash, protrusions or gate burrs. Mold flash, protrusions or gate burrs shall not exceed 0.15mm per side*

Table 90. Thermal characteristics

Symbol	Ratings		Value	Unit
R_{thJA}	Package thermal resistance (junction to ambient)	SO20 DIP20	125 63	°C/W
T_{Jmax}	Maximum junction temperature(1)		150	°C
P_{Dmax}	Power dissipation(2)		500	mW

1. The maximum chip-junction temperature is based on technology characteristics.
2. The maximum power dissipation is obtained from the formula $P_D = (T_J - T_A) / R_{thJA}$.
The power dissipation of an application can be defined by the user with the formula: $P_D=P_{INT}+P_{PORT}$ where P_{INT} is the chip internal power ($I_{DD}xV_{DD}$) and P_{PORT} is the port power dissipation depending on the ports used in the application.

14.2 Soldering information

In accordance with the RoHS European directive, all STMicroelectronics packages have been converted to lead-free technology, named ECOPACK™.

- ECOPACK™ packages are qualified according to the JEDEC STD-020C compliant soldering profile.
- Detailed information on the STMicroelectronics ECOPACK™ transition program is available on www.st.com/stonline/leadfree/, with specific technical Application notes covering the main technical aspects related to lead-free conversion (AN2033, AN2034, AN2035, AN2036).

Backward and forward compatibility

The main difference between Pb and Pb-free soldering process is the temperature range.

- ECOPACK™ TQFP, SDIP and SO packages are fully compatible with Lead (Pb) containing soldering process (see application note AN2034)
- TQFP, SDIP and SO Pb-packages are compatible with Lead-free soldering process, nevertheless it's the customer's duty to verify that the Pb packages maximum temperature (mentioned on the Inner box label) is compatible with their Leadfree soldering temperature.

Table 91. Soldering compatibility (wave and reflow soldering process)

Package	Plating material devices	Pb solder paste	Pb-free solder paste
SDIP & PDIP	Sn (pure Tin)	Yes	Yes(1)
TQFP and SO	NiPdAu (Nickel-palladium-Gold)	Yes	Yes(1)

1. Assemblers must verify that the Pb-package maximum temperature (mentioned on the Inner box label) is compatible with their Lead-free soldering process.