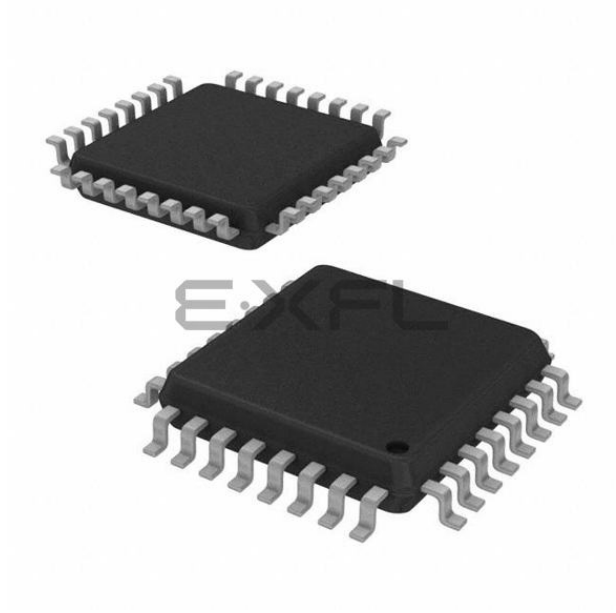


Welcome to E-XFL.COM

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	SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f324bk2t3tr

3 Register and memory map

As shown in *Figure 4* the MCU is capable of addressing 64 Kbytes of memories and I/O registers.

The available memory locations consist of 128 bytes of register locations, up to 1024 bytes of RAM and up to 32 Kbytes of user program memory. The RAM space includes up to 256 bytes for the stack from 0100h to 01FFh.

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

Caution: Never access memory locations marked as 'Reserved'. Accessing a reserved area can have unpredictable effects on the device.

Figure 4. Memory map

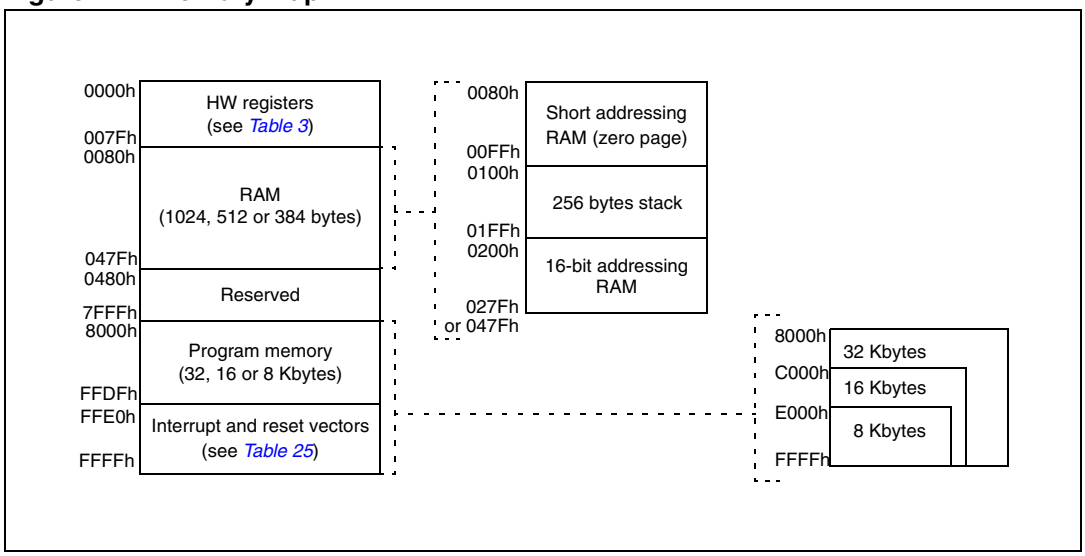
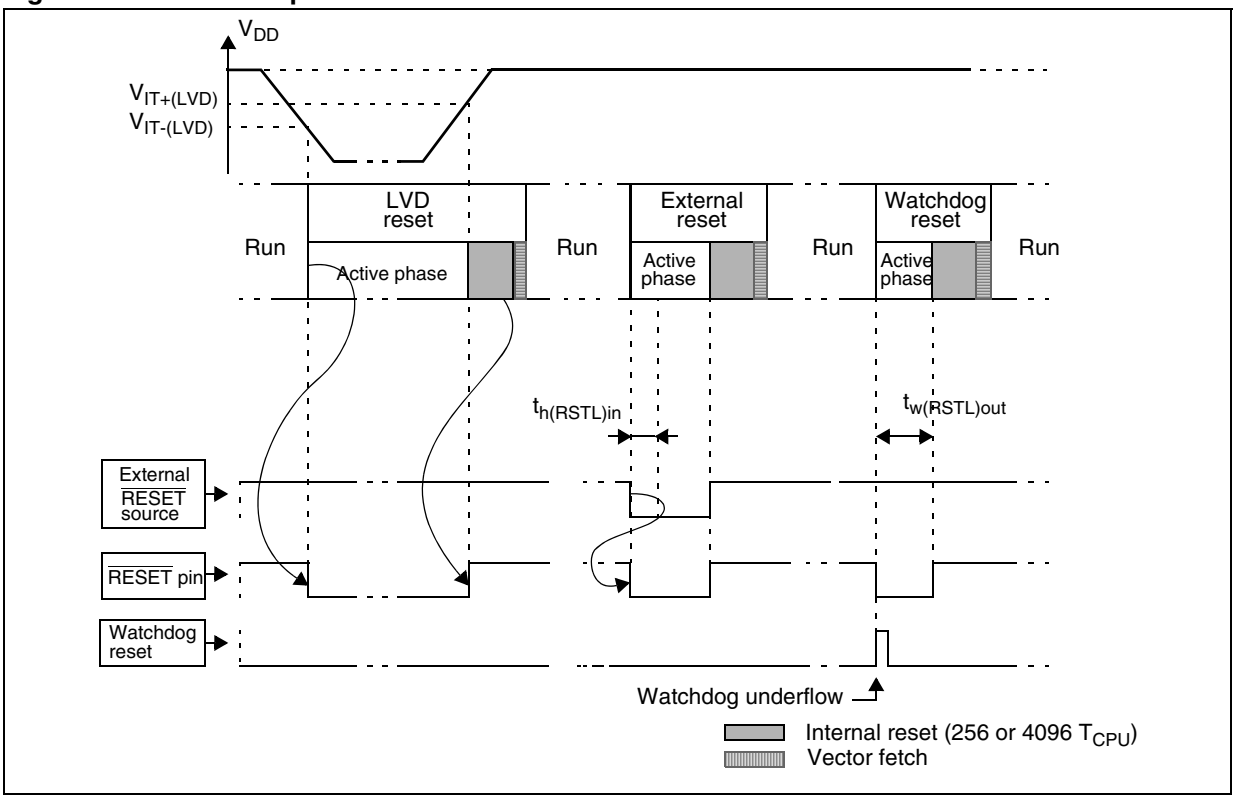


Table 3. Hardware register map

Address	Block	Register label	Register name	Reset status(1)	Remarks(1)
0000h 0001h 0002h	Port A(2)	PADR PADDR PAOR	Port A data register Port A data direction register Port A option register	00h(3) 00h 00h	R/W R/W R/W
0003h 0004h 0005h	Port B(1)	PBDR PBDDR PBOR	Port B data register Port B data direction register Port B option register	00h(2) 00h 00h	R/W R/W R/W
0006h 0007h 0008h	Port C	PCDR PCDDR PCOR	Port C data register Port C data direction register Port C option register	00h(2) 00h 00h	R/W R/W R/W
0009h 000Ah 000Bh	Port D(1)	PDADR PDDDR PDOR	Port D data register Port D data direction register Port D option register	00h(2) 00h 00h	R/W R/W R/W
000Ch 000Dh 000Eh	Port E(1)	PEDR PEDDR PEOR	Port E data register Port E data direction register Port E option register	00h(2) 00h 00h	R/W R/W(1) R/W(1)

Figure 13. RESET sequences



6.5 System integrity management (SI)

The system integrity management block contains the LVD and auxiliary voltage detector (AVD) functions. It is managed by the SICSR register.

6.5.1 LVD (low voltage detector)

The LVD function generates a static reset when the V_{DD} supply voltage is below a V_{IT-} reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The V_{IT-} reference value for a voltage drop is lower than the V_{IT+} reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

The LVD reset circuitry generates a reset when V_{DD} is below:

- V_{IT+} when V_{DD} is rising
- V_{IT-} when V_{DD} is falling

The LVD function is illustrated in *Figure 13*.

The voltage threshold can be configured by option byte to be low, medium or high.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is above V_{IT-}, the MCU can only be in two modes:

- under full software control
- in static safe reset

In these conditions, secure operation is always ensured for the application without the need for external reset hardware.

During an LVD reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

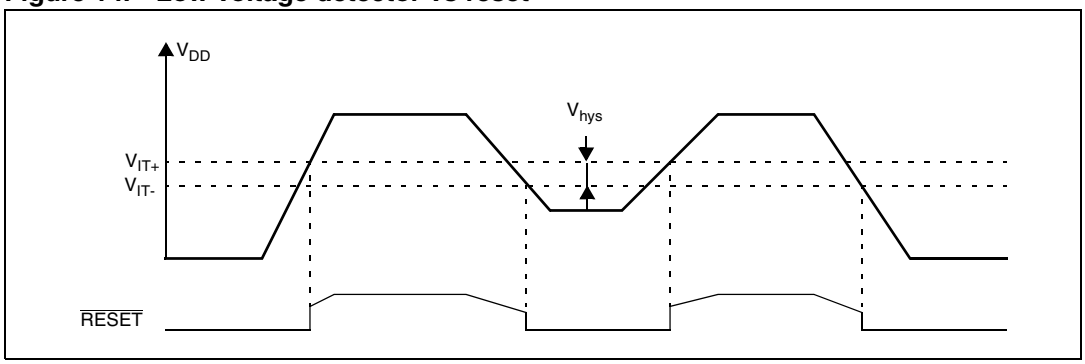
Note: 1 The LVD allows the device to be used without any external reset circuitry.

2 If the medium or low thresholds are selected, the detection may occur outside the specified operating voltage range. Below 3.8V, device operation is not guaranteed.

3 The LVD is an optional function which can be selected by option byte.

4 It is recommended to make sure that the V_{DD} supply voltage rises monotonously when the device is exiting from reset, to ensure the application functions properly.

Figure 14. Low voltage detector vs reset



6.5.2 AVD (auxiliary voltage detector)

The AVD is based on an analog comparison between a $V_{IT-(AVD)}$ and $V_{IT+(AVD)}$ reference value and the V_{DD} main supply. The V_{IT-} reference value for falling voltage is lower than the V_{IT+} reference value for rising voltage in order to avoid parasitic detection (hysteresis).

The output of the AVD comparator is directly readable by the application software through a real-time status bit (AVDF) in the SICSR register. This bit is read only.

Caution: The AVD function is active only if the LVD is enabled through the option byte (see *Section 14.1 on page 179*).

Monitoring the V_{DD} main supply

The AVD voltage threshold value is relative to the selected LVD threshold configured by option byte (see *Section 14.1 on page 179*).

If the AVD interrupt is enabled, an interrupt is generated when the voltage crosses the $V_{IT+(AVD)}$ or $V_{IT-(AVD)}$ threshold (AVDF bit toggles).

In the case of a drop in voltage, the AVD interrupt acts as an early warning, allowing software to shut down safely before the LVD resets the microcontroller. See *Figure 15*.

9 I/O ports

9.1 Introduction

The I/O ports offer different functional modes:

- transfer of data through digital inputs and outputs,

and for specific pins:

- external interrupt generation,
- alternate signal input/output for the on-chip peripherals.

An I/O port contains up to 8 pins. Each pin can be programmed independently as digital input (with or without interrupt generation) or digital output.

9.2 Functional description

Each port has two main registers:

- Data Register (DR)
- Data Direction Register (DDR)

and one optional register:

- Option Register (OR)

Each I/O pin may be programmed using the corresponding register bits in the DDR and OR registers: bit X corresponding to pin X of the port. The same correspondence is used for the DR register.

The following description takes into account the OR register, (for specific ports which do not provide this register refer to *Section 9.3: I/O port implementation on page 62*). The generic I/O block diagram is shown in *Figure 28.*

9.2.1 Input modes

The input configuration is selected by clearing the corresponding DDR register bit.

In this case, reading the DR register returns the digital value applied to the external I/O pin.

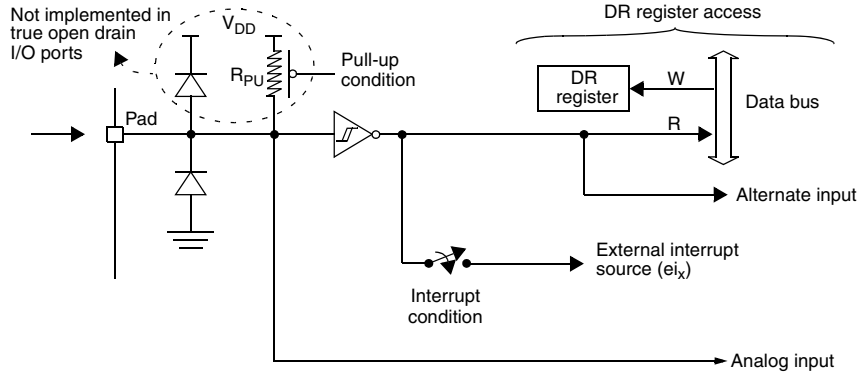
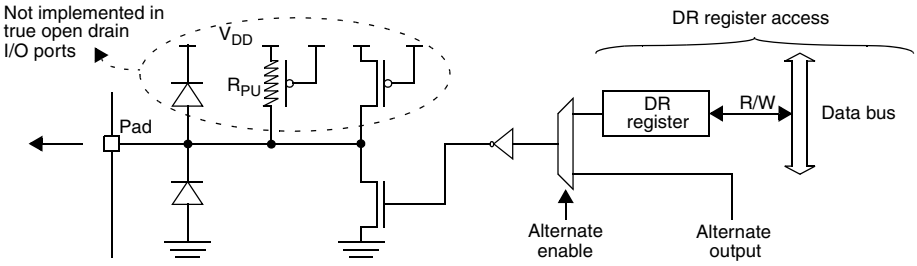
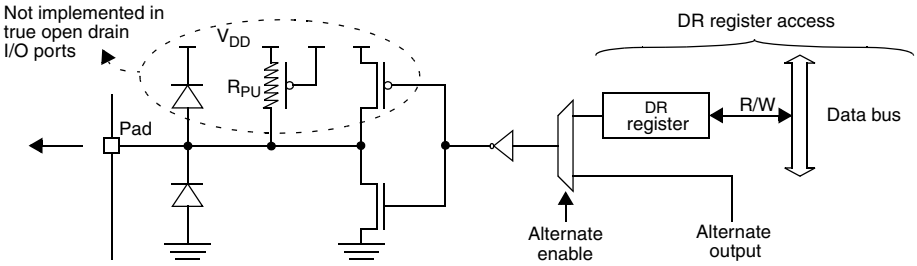
Different input modes can be selected by software through the OR register.

Note: 1 Writing the DR register modifies the latch value but does not affect the pin status.

2 When switching from input to output mode, the DR register has to be written first to drive the correct level on the pin as soon as the port is configured as an output.

3 Do not use read/modify/write instructions (BSET or BRES) to modify the DR register as this might corrupt the DR content for I/Os configured as input.

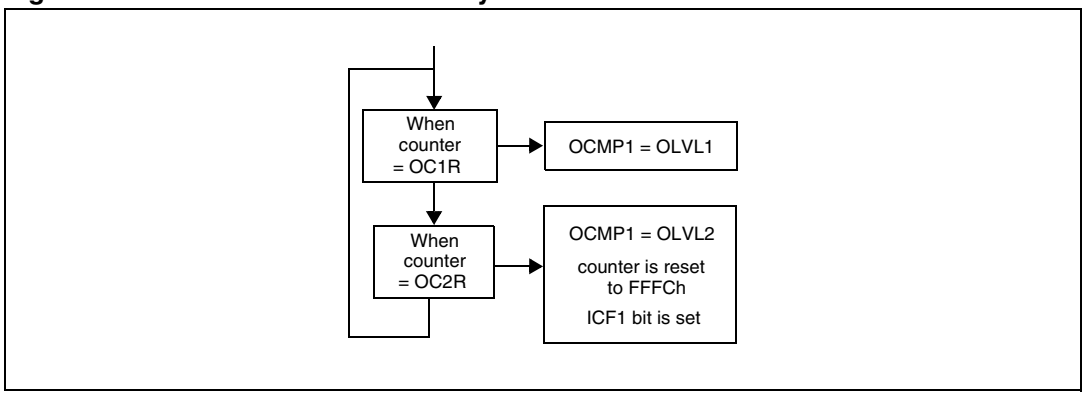
Table 29. I/O port configurations

	Hardware configuration
Input(1)	
Open-drain output(2)	
PUSH-pull output(2)	

1. When the I/O port is in input configuration and the associated alternate function is enabled as an output, reading the DR register will read the alternate function output status.
2. When the I/O port is in output configuration and the associated alternate function is enabled as an input, the alternate function reads the pin status given by the DR register content.

Caution: The alternate function must not be activated as long as the pin is configured as input with interrupt, in order to avoid generating spurious interrupts.

Figure 47. 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 OC1R register value required for a specific timing application can be calculated using the following formula:

$$OC\mathit{i}R\ value = \frac{t * f_{CPU}}{PRESC} - 5$$

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 the CC[1:0] bits; see *Table 50*)

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

$$OC\mathit{i}R = t * f_{EXT} - 5$$

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 *Table 46*).

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 can not 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 52. 16-bit timer register map and reset values (continued)

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
Timer A: 35 Timer B: 45	IC1LR Reset value	MSB x	x	x	x	x	x	x	LSB x
Timer A: 36 Timer B: 46	OC1HR Reset value	MSB 1	0	0	0	0	0	0	LSB 0
Timer A: 37 Timer B: 47	OC1LR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
Timer A: 3E Timer B: 4E	OC2HR Reset value	MSB 1	0	0	0	0	0	0	LSB 0
Timer A: 3F Timer B: 4F	OC2LR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
Timer A: 38 Timer B: 48	CHR Reset value	MSB 1	1	1	1	1	1	1	LSB 1
Timer A: 39 Timer B: 49	CLR Reset value	MSB 1	1	1	1	1	1	0	LSB 0
Timer A: 3A Timer B: 4A	ACHR Reset value	MSB 1	1	1	1	1	1	1	LSB 1
Timer A: 3B Timer B: 4B	ACLR Reset value	MSB 1	1	1	1	1	1	0	LSB 0
Timer A: 3C Timer B: 4C	IC2HR Reset value	MSB x	x	x	x	x	x	x	LSB x
Timer A: 3D Timer B: 4D	IC2LR Reset value	MSB x	x	x	x	x	x	x	LSB x

10.4 Serial peripheral interface (SPI)

10.4.1 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. However, the SPI interface can not be a master in a multi-master system.

10.4.2 Main features

- Full duplex synchronous transfers (on 3 lines)
- Simplex synchronous transfers (on 2 lines)
- Master or slave operation
- 6 master mode frequencies (f_{CPU}/4 max.)
- f_{CPU}/2 max. slave mode frequency (see 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

10.4.6 Low power modes

Table 53. Effect of low power modes on SPI

Mode	Description
Wait	No effect on SPI. SPI interrupt events cause the device to exit from Wait mode.
Halt	SPI registers are frozen. In Halt mode, the SPI is inactive. SPI operation resumes when the MCU is woken up by an interrupt with Exit from Halt mode capability. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetching). If several data are received before the wake-up event, then an overrun error is generated. This error can be detected after the fetch of the interrupt routine that woke up the device.

Using the SPI to wake up the MCU from Halt mode

In slave configuration, the SPI is able to wake up the ST7 device from Halt mode through a SPIF interrupt. The data received is subsequently read from the SPIDR register when the software is running (interrupt vector fetch). If multiple data transfers have been performed before software clears the SPIF bit, then the OVR bit is set by hardware.

Note: When waking up from Halt mode, if the SPI remains in Slave mode, it is recommended to perform an extra communications cycle to bring the SPI from Halt mode state to normal state. If the SPI exits from Slave mode, it returns to normal state immediately.

Caution: The SPI can wake up the ST7 from Halt mode only if the Slave Select signal (external $\overline{SS}$ pin or the SSI bit in the SPICSR register) is low when the ST7 enters Halt mode. Therefore, if Slave selection is configured as external (see *Slave Select management on page 99*), make sure the master drives a low level on the $\overline{SS}$ pin when the slave enters Halt mode.

10.4.7 Interrupts

Table 54. SPI interrupt control/wake-up capability(1)

Interrupt event	Event flag	Enable control bit	Exit from WAIT	Exit from HALT
SPI end of transfer event	SPIF	SPIE	Yes	Yes
Master mode fault event	MODF			No
Overrun error	OVR			

1. The SPI interrupt events are connected to the same interrupt vector (see *Section 7: Interrupts*). They generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

10.5.4 Functional description

The block diagram of the serial control interface is shown in *Figure 55*. It contains six dedicated registers:

- 2 control registers (SCICR1 and SCICR2)
- a status register (SCISR)
- a baud rate register (SCIBRR)
- an extended prescaler receiver register (SCIERPR)
- an extended prescaler transmitter register (SCIETPR)

Refer to the register descriptions in *Section 10.5.7* for the definitions of each bit.

Serial data format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the SCICR1 register (see *Figure 55*).

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

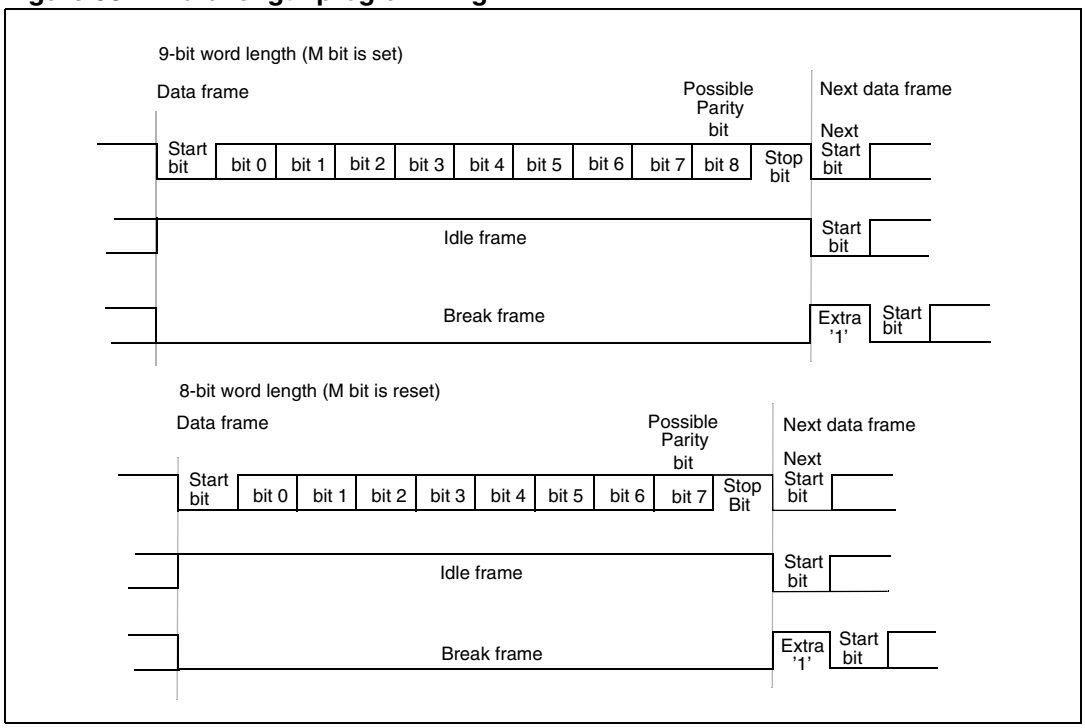
The TDO pin is in high state during the stop bit.

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

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

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

Figure 56. Word length programming



Receiver muting and wake-up feature

In multiprocessor configurations it is often desirable that only the intended message recipient should actively receive the full message contents, thus reducing redundant SCI service overhead for all non-addressed receivers.

The non-addressed devices may be placed in sleep mode by means of the muting function.

Setting the RWU bit by software puts the SCI in sleep mode:

All the reception status bits cannot be set.

All the receive interrupts are inhibited.

A muted receiver may be awakened by one of the following two ways:

- by Idle Line detection if the Wake bit is reset,
- by Address Mark detection if the Wake bit is set.

A receiver wakes up by Idle Line detection when the Receive line has recognized an Idle Frame. Then the RWU bit is reset by hardware but the Idle bit is not set.

A receiver wakes up by Address Mark detection when it received a ‘1’ as the most significant bit of a word, thus indicating that the message is an address. The reception of this particular word wakes up the receiver, resets the RWU bit and sets the RDRF bit, which allows the receiver to receive this word normally and to use it as an address word.

Caution: In Mute mode, do not write to the SCICR2 register. If the SCI is in Mute mode during the read operation (RWU = 1) and an address mark wake-up event occurs (RWU is reset) before the write operation, the RWU bit will be set again by this write operation. Consequently the address byte is lost and the SCI is not woken up from Mute mode.

Parity control

Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the PCE bit in the SCICR1 register. Depending on the frame length defined by the M bit, the possible SCI frame formats are as listed in *Table 59*.

Table 59. Frame formats(1)(2)

M bit	PCE bit	SCI frame
0	0	| SB | 8 bit data | STB |
0	1	| SB | 7-bit data | PB | STB |
1	0	| SB | 9-bit data | STB |
1	1	| SB | 8-bit data PB | STB |

1. SB = Start bit, STB = Stop bit, and PB = Parity bit.
2. In case of wake-up by an address mark, the MSB bit of the data is taken into account and not the Parity bit.

10.6 10-bit A/D converter (ADC)

10.6.1 Introduction

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

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

10.6.2 Main features

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

The block diagram is shown in *Figure 59*.

Figure 59. ADC block diagram

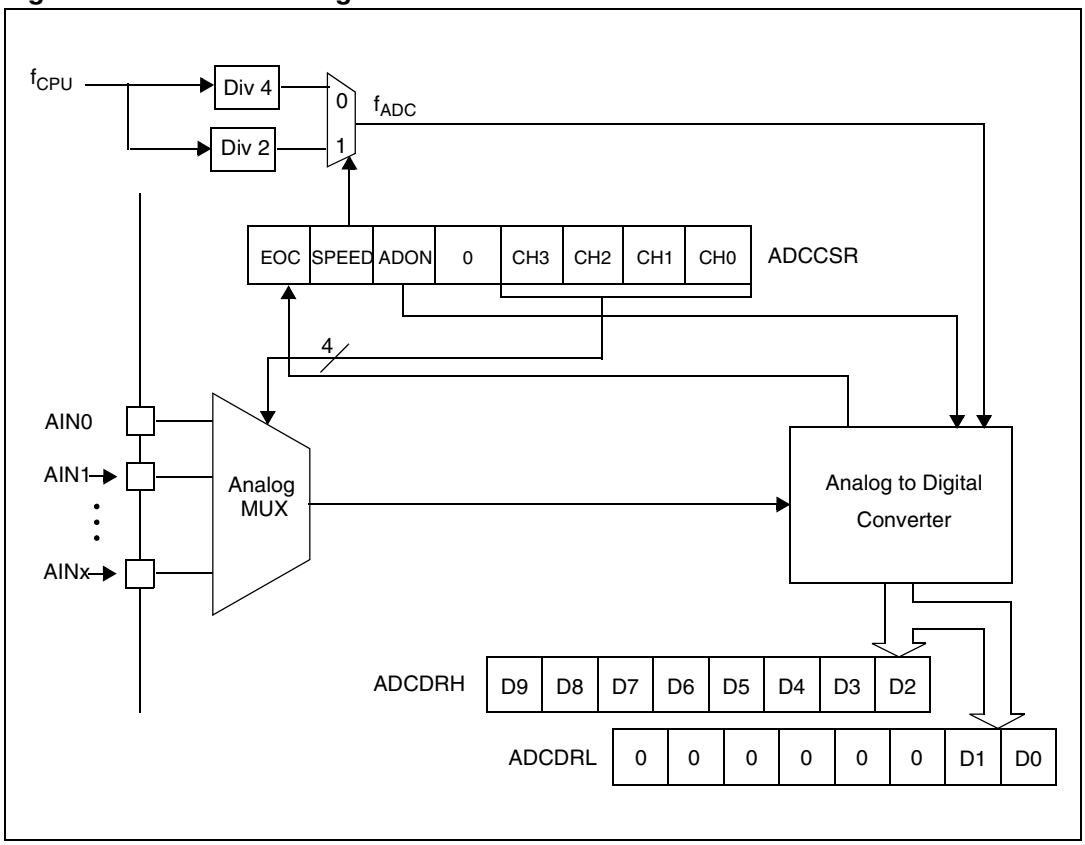


Table 76. CPU addressing mode overview (continued)

Long	Indirect	Indexed	ld A,([$10.w],X)	0000..FFFF	00..FF	word	+ 2
Relative	Direct		jrne loop	PC+/-127			+ 1
Relative	Indirect		jrne [$10]	PC+/-127	00..FF	byte	+ 2
Bit	Direct		bset $10,#7	00..FF			+ 1
Bit	Indirect		bset [$10],#7	00..FF	00..FF	byte	+ 2
Bit	Direct	Relative	btjt $10,#7,skip	00..FF			+ 2
Bit	Indirect	Relative	btjt [$10],#7,skip	00..FF	00..FF	byte	+ 3

11.1.1 Inherent

All Inherent instructions consist of a single byte. The opcode fully specifies all the required information for the CPU to process the operation.

Table 77. Inherent instructions

Instruction	Function
NOP	No Operation
TRAP	S/W Interrupt
WFI	Wait for Interrupt (low power mode)
HALT	Halt oscillator (lowest power mode)
RET	Sub-routine Return
IRET	Interrupt sub-routine Return
SIM	Set Interrupt Mask (level 3)
RIM	Reset Interrupt Mask (level 0)
SCF	Set Carry Flag
RCF	Reset Carry Flag
RSP	Reset Stack Pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
MUL	Byte Multiplication
SLL, SRL, SRA, RLC, RRC	Shift and Rotate operations
SWAP	Swap nibbles

Table 82. Instruction set overview (continued)

Mnemo	Description	Function/example	Dst	Src	I1	H	I0	N	Z	C
JRUGT	Jump if (C + Z = 0)	Unsigned >								
JRULE	Jump if (C + Z = 1)	Unsigned <=								
LD	Load	dst <= src	reg, M	M, reg				N	Z	
MUL	Multiply	X,A = X * A	A, X, Y	X, Y, A		0				0
NEG	Negate (2's compl)	neg $10	reg, M					N	Z	C
NOP	No Operation									
OR	OR operation	A = A + M	A	M				N	Z	
POP	Pop from the Stack	pop reg	reg	M						
		pop CC	CC	M	I1	H	I0	N	Z	C
PUSH	Push onto the Stack	push Y	M	reg, CC						
RCF	Reset carry flag	C = 0								0
RIM	Enable Interrupts	I1:0 = 10 (level 0)			1		0			
RLC	Rotate Left true C	C <= A <= C	reg, M					N	Z	C
RRC	Rotate Right true C	C => A => C	reg, M					N	Z	C
RSP	Reset Stack Pointer	S = Max allowed								
SBC	Subtract with Carry	A = A - M - C	A	M				N	Z	C
SCF	Set CARRY FLAG	C = 1								1
SIM	Disable Interrupts	I1:0 = 11 (level 3)			1		1			
SLA	Shift Left Arithmetic	C <= A <= 0	reg, M					N	Z	C
SLL	Shift Left Logic	C <= A <= 0	reg, M					N	Z	C
SRL	Shift Right Logic	0 => A => C	reg, M					0	Z	C
SRA	Shift Right Arithmetic	A7 => A => C	reg, M					N	Z	C
SUB	Subtraction	A = A - M	A	M				N	Z	C
SWAP	SWAP nibbles	A7-A4 <=> A3-A0	reg, M					N	Z	
TNZ	Test for Neg and Zero	tnz lbl1						N	Z	
TRAP	S/W TRAP	S/W interrupt			1		1			
WFI	WAIT for Interrupt				1		0			
XOR	Exclusive OR	A = A XOR M	A	M				N	Z	

12.4 LVD/AVD characteristics

12.4.1 Operating conditions with LVD

Subject to general operating conditions for T_A.

Table 87. Operating conditions with LVD

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(LVD)}$	Reset release threshold (V_{DD} rise)	VD level = high in option byte	4.0(1)	4.2	4.5	V
		VD level = med. in option byte(2)	3.55(1)	3.75	4.0(1)	
		VD level = low in option byte(2)	2.95(1)	3.15	3.35(1)	
$V_{IT-(LVD)}$	Reset generation threshold (V_{DD} fall)	VD level = high in option byte	3.8	4.0	4.25(1)	
		VD level = med. in option byte(2)	3.35(1)	3.55	3.75(1)	
		VD level = low in option byte(2)	2.8(1)	3.0	3.15(1)	
$V_{hys(LVD)}$	LVD voltage threshold hysteresis(1)	$V_{IT+(LVD)}$-$V_{IT-(LVD)}$	150	200	250	mV
Vt_{POR}	V_{DD} rise time(1)	Flash devices	6µs/V		100ms/V	
		8/16K ROM devices			20ms/V	
		32K ROM devices			∝ ms/V	
$t_{g(VDD)}$	Filtered glitch delay on V_{DD}(1)	Not detected by the LVD			40	ns

1. Data based on characterization results, tested in production for ROM devices only.
2. If the medium or low thresholds are selected, the detection may occur outside the specified operating voltage range.

12.4.2 Auxiliary voltage detector (AVD) thresholds

Subject to general operating conditions for T_A.

Table 88. AVD thresholds

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IT+(AVD)}$	1 ⇒ 0 AVDF flag toggle threshold (V_{DD} rise)	VD level = high in option byte	4.4(1)	4.6	4.9	V
		VD level = med. in option byte	3.95(1)	4.15	4.4(1)	
		VD level = low in option byte	3.4(1)	3.6	3.8(1)	
$V_{IT-(AVD)}$	0 ⇒ 1 AVDF flag toggle threshold (V_{DD} fall)	VD level = high in option byte	4.2	4.4	4.65(1)	
		VD level = med. in option byte	3.75(1)	4.0	4.2(1)	
		VD level = low in option byte	3.2(1)	3.4	3.6(1)	
$V_{hys(AVD)}$	AVD voltage threshold hysteresis	$V_{IT+(AVD)}$-$V_{IT-(AVD)}$		200		mV
ΔV_{IT-}	Voltage drop between AVD flag set and LVD reset activated	$V_{IT-(AVD)}$-$V_{IT-(LVD)}$		450		

1. Data based on characterization results, tested in production for ROM devices only.

Table 101. EMS test results

Symbol	Parameter	Conditions	Level/class
V_{FESD}	Voltage limits to be applied on any I/O pin to induce a functional disturbance	32 Kbyte Flash or ROM device: V_{DD} = 5V, T_A = +25°C, f_{OSC} = 8 MHz conforms to IEC 1000-4-2	3B
		8 or 16 Kbyte ROM device: V_{DD} = 5V, T_A = +25°C, f_{OSC} = 8 MHz conforms to IEC 1000-4-2	4A
		8 or 16 Kbyte Flash device: V_{DD} = 5V, T_A = +25°C, f_{OSC} = 8 MHz conforms to IEC 1000-4-2	4B
V_{FFTB}	Fast transient voltage burst limits to be applied through 100pF on V_{DD} and V_{DD} pins to induce a functional disturbance	V_{DD} = 5V, T_A = +25°C, f_{OSC} = 8 MHz conforms to IEC 1000-4-4	4A

12.8.2 Electromagnetic interference (EMI)

Based on a simple application running on the product (toggling two LEDs through the I/O ports), the product is monitored in terms of emission. This emission test is in line with the norm SAE J 1752/3 which specifies the board and the loading of each pin.

Table 102. EMI emissions

Symbol	Parameter	Conditions	Device/package(1)	Monitored frequency band	Max vs [f_{OSC}/f_{CPU}]		Unit
					8/4 MHz	16/8 MHz	
S_{EMI}	Peak level(2)	V_{DD} = 5V T_A = +25°C conforming to SAE J 1752/3	8/16 Kbyte Flash LQFP32 and LQFP44	0.1 MHz to 30 MHz	12	18	dBµV
				30 MHz to 130 MHz	19	25	
				130 MHz to 1 GHz	15	22	
				SAE EMI Level	3	3.5	-
			32 Kbyte Flash LQFP32 and LQFP44	0.1 MHz to 30 MHz	13	14	dBµV
				30 MHz to 130 MHz	20	25	
				130 MHz to 1 GHz	16	21	
				SAE EMI Level	3.0	3.5	-
			8/16 Kbyte ROM LQFP32 and LQFP44	0.1 MHz to 30 MHz	12	15	dBµV
				30 MHz to 130 MHz	23	26	
				130 MHz to 1 GHz	15	20	
				SAE EMI Level	3.0	3.5	-
			32 Kbyte ROM LQFP32 and LQFP44	0.1 MHz to 30 MHz	17	21	dBµV
				30 MHz to 130 MHz	24	30	
				130 MHz to 1 GHz	18	23	
				SAE EMI Level	3.0	3.5	-

1. Refer to application note AN1709 for data on other package types.

12.12 Communication interface characteristics

12.12.1 Serial peripheral interface (SPI)

The following characteristics are ubject to general operating conditions for V_{DD}, f_{CPU}, and T_A unless otherwise specified. The data is based on design simulation and/or characterization results, not tested in production.

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

Table 110. SPI characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master f_{CPU} = 8 MHz	$f_{CPU}/128 = 0.0625$	$f_{CPU}/4 = 2$	MHz
		Slave f_{CPU} = 8 MHz	0	$f_{CPU}/2 = 4$	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time		see I/O port pin description		
$t_{su(\overline{SS})}$ (1)	$\overline{SS}$ setup time(2)	Slave	t_{CPU} + 50		ns
$t_{h(\overline{SS})}$ (1)	$\overline{SS}$ hold time	Slave	120		
$t_{w(SCKH)}$ (1) $t_{w(SCKL)}$ (1)	SCK high and low time	Master Slave	100 90		
$t_{su(MI)}$ (1) $t_{su(SI)}$ (1)	Data input setup time	Master Slave	100 100		
$t_{h(MI)}$ (1) $t_{h(SI)}$ (1)	Data input hold time	Master Slave	100 100		
$t_{a(SO)}$ (1)	Data output access time	Slave	0	120	
$t_{dis(SO)}$ (1)	Data output disable time	Slave		240	
$t_{v(SO)}$ (1)	Data output valid time	Slave (after enable edge)		120	
$t_{h(SO)}$ (1)	Data output hold time		0		
$t_{v(MO)}$ (1)	Data output valid time	Master (after enable edge)		120	
$t_{h(MO)}$ (1)	Data output hold time		0		

1. Data based on design simulation and/or characterization results, not tested in production.
2. Depends on f_{CPU}. For example, if f_{CPU} = 8 MHz, then $t_{CPU} = 1 / f_{CPU}$ = 125ns and $t_{su(SS)}$ = 175ns.

Case 1: Writing to PxOR or PxDDR with global interrupts enabled:

```
LD A,#01
LD sema,A; set the semaphore to '1'
LD A,PFDR
AND A,#02
LD  X,A; store the level before writing to PxOR/PxDDR
LD A,#$90
LD PFDDR,A ; Write to PFDDR
LD A,#$ff
LD  PFOR,A ; Write to PFOR
LD A,PFDR
AND A,#02
LD Y,A; store the level after writing to PxOR/PxDDR
LD A,X; check for falling edge
cp A,#02
jrne OUT
TNZ Y
jrne OUT
LD A,sema ; check the semaphore status if edge is detected
CP A,#01
jrne OUT
call call_routine ; call the interrupt routine
OUT:LD A,#00
LD sema,A
.call_routine ; entry to call_routine
PUSH A
PUSH X
PUSH CC
.ext1_rt ; entry to interrupt routine
LD A,#00
LD sema,A
IRET
```

Case 2: Writing to PxOR or PxDDR with global interrupts disabled:

```
SIM ; set the interrupt mask
LD A,PFDR
AND A,#$02
LD  X,A ; store the level before writing to PxOR/PxDDR
LD A,#$90
LD PFDDR,A ; Write into PFDDR
LD A,#$ff
LD PFOR,A ; Write to PFOR
LD A,PFDR
AND A,#$02
LD Y,A ; store the level after writing to PxOR/PxDDR
LD A,X ; check for falling edge
cp A,#$02
jrne OUT
TNZ Y
jrne OUT
LD A,#$01
LD sema,A ; set the semaphore to '1' if edge is detected
```

In the same way, as long as the SBK bit is set, break characters are sent to the TDO pin. This may lead to generate one break more than expected.

Occurrence

The occurrence of the problem is random and proportional to the baud rate. With a transmit frequency of 19200 baud (f_{CPU} = 8MHz and SCIBRR = 0xC9), the wrong break duration occurrence is around 1%.

Workaround

If this wrong duration is not compliant with the communication protocol in the application, software can request that an Idle line be generated before the break character. In this case, the break duration is always correct assuming the application is not doing anything between the idle and the break. This can be ensured by temporarily disabling interrupts.

The exact sequence is:

1. Disable interrupts
2. Reset and set TE (IDLE request)
3. Set and reset SBK (break request)
4. Re-enable interrupts

15.2 8/16 Kbyte Flash devices only

15.2.1 39-pulse ICC entry mode

ICC mode entry using ST7 application clock (39 pulses) is not supported. External clock mode must be used (36 pulses). Refer to the *ST7 Flash Programming Reference Manual*.

15.2.2 Negative current injection on pin PB0

Negative current injection on pin PB0 degrades the performance of the device and is not allowed on this pin.

15.3 8/16 Kbyte ROM devices only

15.3.1 Readout protection with LVD

Readout protection is not supported if the LVD is enabled.

15.3.2 I/O Port A and F configuration

When using an external quartz crystal or ceramic resonator, a few f_{OSC2} clock periods may be lost when the signal pattern in *Table 122* occurs. This is because this pattern causes the device to enter test mode and return to user mode after a few clock periods. User program execution and I/O status are not changed, only a few clock cycles are lost.

This happens with either one of the following configurations

- PA3 = 0, PF4 = 1, PF1 = 0 while PLL option is disabled and PF0 is toggling
- PA3 = 0, PF4 = 1, PF1 = 0, PF0 = 1 while PLL option is enabled

This is detailed in *Table 122*

Table 123. Document revision history (continued)

Date	Revision	Changes
12-Jul-2010	4	Added *Section 13.5 on page 178* Updated *Figure 90: ST72F324Bxx-Auto Flash commercial product structure on page 182*, *Figure 91: ST72P324Bxx-Auto FastROM commercial product structure on page 184* and *Figure 92: ST72324Bxx-Auto ROM commercial product structure on page 185*