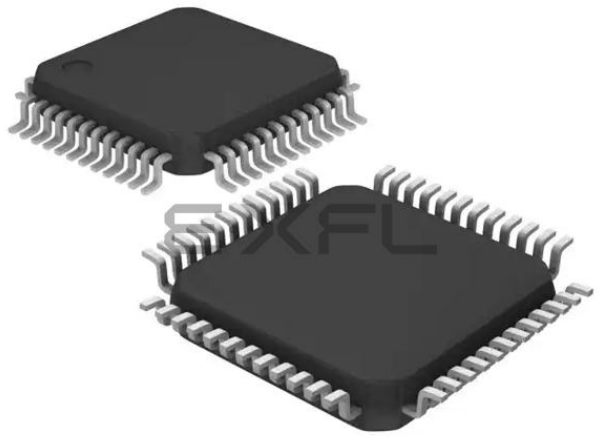


Welcome to E-XFL.COM

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, SCI, USB
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	27
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 12x8b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f63bh2t1

Condition Code register (CC)

Reset value: 111x1xxx

7	6	5	4	3	2	1	0
1	1	1	H	I	N	Z	C
Read/write							

The 8-bit Condition Code register contains the interrupt mask 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.

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 instruction. 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 3 **I** *Interrupt mask*

This bit is set by hardware when entering in interrupt or by software to disable all interrupts except the TRAP software interrupt. This bit is cleared by software.

0: Interrupts are enabled.

1: Interrupts are disabled.

This bit is controlled by the RIM, SIM and IRET instructions and is tested by the JRM and JRNM instructions.

Note: Interrupts requested while I is set are latched and can be processed when I is cleared. By default an interrupt routine is not interruptible because the I bit is set by hardware at the start of the routine and reset by the IRET instruction at the end of the routine. If the I bit is cleared by software in the interrupt routine, pending interrupts are serviced regardless of the priority level of the current interrupt routine.

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 is a copy of the 7th bit of the result.

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.

Stack Pointer (SP)

Reset value: 017Fh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	0	SP6	SP5	SP4	SP3	SP2	SP1	SP0
Read/write															

The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see *Figure 10*).

Since the stack is 128 bytes deep, the 9 most significant bits are forced by hardware. Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP6 to SP0 bits are set) which is the stack higher address.

The least significant byte of the Stack Pointer (called S) can be directly accessed by a LD instruction.

Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location

7							0
D7	D6	D5	D4	D3	D2	D1	D0
Read/write							

[7:0] **D[7:0]** *Data register 8 bits.*

Data Direction register (PxDDR)

Address

Port A Data Direction register (PADDR): 0001h

Port B Data Direction register (PBDDR): 0003h

Port C Data Direction register (PCDDR): 0005h

Port D Data Direction register (PDDDR): 0007h

Reset value

Port A: 0000 0000 (00h)

Port B: 0000 0000 (00h)

Port C: 1111 x000 (FXh)

Port D: 0000 0000 (00h)

Note: For Port C, unused bits (7-3) are not accessible

7							0
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
Read/write							

[7:0]D **D[7:0]** *Data Direction register 8 bits.*

The DDR register gives the input/output direction configuration of the pins. Each bit is set and cleared by software.

0: input mode

1: output mode

Table 16. I/O ports register map

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
00	PADR	MSB							LSB
01	PADDR	MSB							LSB
02	PBDR	MSB							LSB
03	PBDDR	MSB							LSB
04	PCDR	MSB							LSB
05	PCDDR	MSB							LSB
06	PDDR	MSB							LSB
07	PDDDR	MSB							LSB

11 On-chip peripherals

11.1 Watchdog timer (WDG)

11.1.1 Introduction

The Watchdog timer is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to abandon its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before the T6 bit becomes cleared.

11.1.2 Main features

- Programmable free-running counter (64 increments of 49,152 CPU cycles)
- Programmable reset
- Reset (if watchdog activated) when the T6 bit reaches zero
- Optional reset on HALT instruction (configurable by option byte)
- Hardware Watchdog selectable by option byte.

11.1.3 Functional description

The counter value stored in the CR register (bits T6:T0), is decremented every 49,152 machine cycles, and the length of the timeout period can be programmed by the user in 64 increments.

If the watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T6:T0) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle by driving low the reset pin for $t_{W(RSTL)out}$ (see *Table 72*).

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. This down counter is free-running: 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 17*):

- The WDGA bit is set (watchdog enabled)
- The T6 bit is set to prevent generating an immediate reset
- The T5:T0 bits contain the number of increments which represents the time delay before the watchdog produces a reset.

Input Capture

In this section, the index, *i*, may be 1 or 2 because there are two input capture functions in the 16-bit timer.

The two 16-bit input capture registers (IC1R and IC2R) are used to latch the value of the free running counter after a transition is detected on the ICAP*i* pin (see *Figure 32*).

Table 19. IC/R register

	MS Byte	LS Byte
ICiR	IC*i*HR	IC*i*LR

IC*i*R register is a read-only register.

The active transition is software programmable through the IEDG*i* bit of Control registers (CR*i*).

Timing resolution is one count of the free running counter: (f_{CPU}/CC[1:0]).

Procedure

To use the input capture function select the following in the CR2 register:

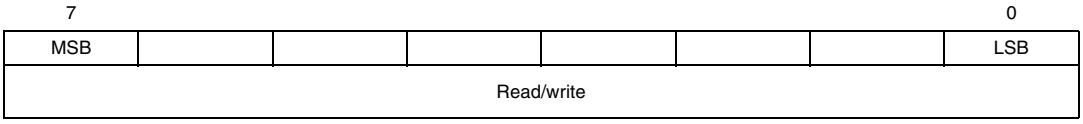
1. Select the timer clock (CC[1:0]) (see *Table 24*).
2. Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit (the ICAP2 pin must be configured as floating input or input with pull-up without interrupt if this configuration is available).
3. Select the following in the CR1 register:
   a) Set the ICIE bit to generate an interrupt after an input capture coming from either the ICAP1 pin or the ICAP2 pin
   b) Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1pin must be configured as floating input or input with pull-up without interrupt if this configuration is available).

When an input capture occurs:

- ICF*i* bit is set.
- The IC*i*R register contains the value of the free running counter on the active transition on the ICAP*i* pin (see *Figure 33*).
- A timer interrupt is generated if the ICIE bit is set and the I bit is cleared in the CC register. Otherwise, the interrupt remains pending until both conditions become true.

Clearing the input Capture interrupt request (that is, clearing the ICF*i* bit) is done in two steps:

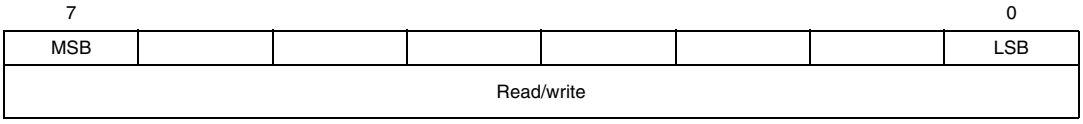
1. Reading the SR register while the ICF*i* bit is set.
2. An access (read or write) to the IC*i*LR register.



Output Compare 2 High register (OC2HR)

Reset value: 1000 0000 (80h)

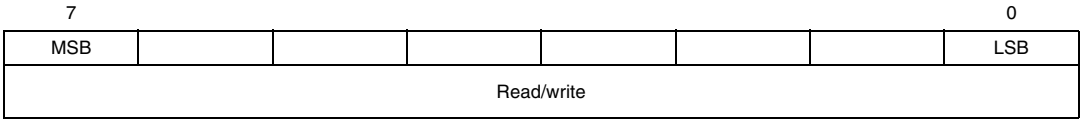
This is an 8-bit register that contains the high part of the value to be compared to the CHR register.



Output Compare 2 Low register (OC2LR)

Reset value: 0000 0000 (00h)

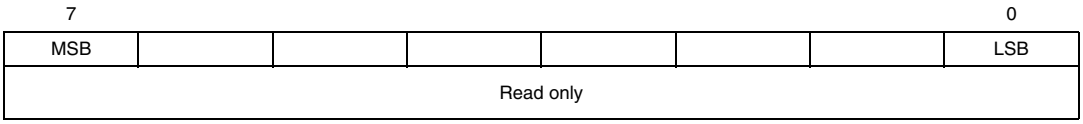
This is an 8-bit register that contains the low part of the value to be compared to the CLR register.



Counter High register (CHR)

Reset value: 1111 1111 (FFh)

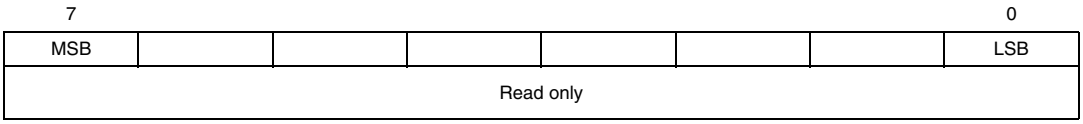
This is an 8-bit register that contains the high part of the counter value.



Counter Low register (CLR)

Reset value: 1111 1100 (FCh)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the CSR register clears the TOF bit.



Alternate Counter High register (ACHR)

Reset value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

Control register 2 (SCICR2)

Reset value: 0000 0000 (00h)

7							0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Read/write							

7 TIE *Transmitter interrupt enable*.
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SCI interrupt is generated whenever TDRE=1 in the SCISR register

6 TCIE *Transmission complete interrupt enable*
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SCI interrupt is generated whenever TC=1 in the SCISR register

5 **RIE** *Receiver interrupt enable*.
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SCI interrupt is generated whenever OR=1 or RDRF=1 in the SCISR register

4 ILIE *Idle line interrupt enable.*
This bit is set and cleared by software.
0: Interrupt is inhibited
1: An SCI interrupt is generated whenever IDLE=1 in the SCISR register.

3 TE *Transmitter enable.*
This bit enables the transmitter. It is set and cleared by software.
0: Transmitter is disabled
1: Transmitter is enabled

Note: During transmission, a "0" pulse on the TE bit ("0" followed by "1") sends a preamble (idle line) after the current word.
When TE is set there is a 1 bit-time delay before the transmission starts.

Caution: The TDO pin is free for general purpose I/O only when the TE and RE bits are both cleared (or if TE is never set).

2 **ESUSP** *End suspend mode.*
This bit is set by hardware when, during suspend mode, activity is detected that wakes the USB interface up from suspend mode.
This interrupt is serviced by a specific vector, in order to wake up the ST7 from Halt mode.
0: No End Suspend detected
1: End Suspend detected

1 **RESET** *USB reset.*
This bit is set by hardware when the USB reset sequence is detected on the bus.
0: No USB reset signal detected
1: USB reset signal detected
Note: The DADDR, EP0RA, EP0RB, EP1RA, EP1RB, EP2RA and EP2RB registers are reset by a USB reset.

0 **SOF** *Start of frame.*
This bit is set by hardware when a low-speed SOF indication (keep-alive strobe) is seen on the USB bus. It is also issued at the end of a resume sequence.
0: No SOF signal detected
1: SOF signal detected

Note: To avoid spurious clearing of some bits, it is recommended to clear them using a load instruction where all bits which must not be altered are set, and all bits to be cleared are reset. Avoid read-modify-write instructions like AND, XOR.

Interrupt Mask register (IMR)

These bits are mask bits for all interrupt condition bits included in the ISTR. Whenever one of the IMR bits is set, if the corresponding ISTR bit is set, and the I bit in the CC register is cleared, an interrupt request is generated. For an explanation of each bit, please refer to the corresponding bit description in ISTR.

Reset value: 0000 0000 (00h)

7							0
SUSPM	DOVRM	CTRM	ERRM	IOVRM	ESUSPM	RESETM	SOFM
Read.write							

Endpoint 0 register B (EP0RB)

This register is used for controlling data reception on Endpoint 0. It is also reset by the USB bus reset.

Reset value: 1000 0000 (80h)

7							0
1	DTOG RX	STAT RX1	STAT RX0	0	0	0	0
Read.write							

7 Forced by hardware to 1.

[6:4] Refer to the EPnRB register for a description of these bits.

[3:0] Forced by hardware to 0.

11.4.5 Programming considerations

The interaction between the USB interface and the application program is described below. Apart from system reset, action is always initiated by the USB interface, driven by one of the USB events associated with the Interrupt Status register (ISTR) bits.

Initializing the registers

At system reset, the software must initialize all registers to enable the USB interface to properly generate interrupts and DMA requests.

1. Initialize the DMAR, IDR, and IMR registers (choice of enabled interrupts, address of DMA buffers). Refer the paragraph titled initializing the DMA Buffers.
2. Initialize the EP0RA and EP0RB registers to enable accesses to address 0 and endpoint 0 to support USB enumeration. Refer to the paragraph titled Endpoint Initialization.
3. When addresses are received through this channel, update the content of the DADDR.
4. If needed, write the endpoint numbers in the EA fields in the EP1RB and EP2RB register.

Initializing DMA buffers

The DMA buffers are a contiguous zone of memory whose maximum size is 48 bytes. They can be placed anywhere in the memory space to enable the reception of messages. The 10 most significant bits of the start of this memory area are specified by bits DA15-DA6 in registers DMAR and IDR, the remaining bits are 0. The memory map is shown in *Figure 45*.

Each buffer is filled starting from the bottom (last 3 address bits=000) up.

Endpoint Initialization

To be ready to receive, set STAT_RX to VALID (11b) in EP0RB to enable reception.

To be ready to transmit:

1. Write the data in the DMA transmit buffer.
2. In register EPnRA, specify the number of bytes to be transmitted in the TBC field
3. Enable the endpoint by setting the STAT_TX bits to VALID (11b) in EPnRA.

11.5 I²C bus interface

11.5.1 Introduction

The I²C bus interface serves as an interface between the microcontroller and the serial I²C bus. It provides both multimaster and slave functions, and controls all I²C bus-specific sequencing, protocol, arbitration and timing. It supports fast I²C mode (400 kHz).

11.5.2 Main features

- Parallel-bus/I²C protocol converter
- Multimaster capability
- 7-bit addressing
- Transmitter/receiver flag
- End-of-byte transmission flag
- Transfer problem detection

I²C master features

- Clock generation
- I²C bus busy flag
- Arbitration Lost Flag
- End of byte transmission flag
- Transmitter/Receiver Flag
- Start bit detection flag
- Start and Stop generation

I²C slave features

- Stop bit detection
- I²C bus busy flag
- Detection of misplaced start or stop condition
- Programmable I²C Address detection
- Transfer problem detection
- End-of-byte transmission flag
- Transmitter/Receiver flag

11.5.3 General description

In addition to receiving and transmitting data, this interface converts it from serial to parallel format and vice versa, using either an interrupt or polled handshake. The interrupts are enabled or disabled by software. The interface is connected to the I²C bus by a data pin (SDAI) and by a clock pin (SCLI). It can be connected both with a standard I²C bus and a Fast I²C bus. This selection is made by software.

3 **BTF** *Byte transfer finished.*
This bit is set by hardware as soon as a byte is correctly received or transmitted with interrupt generation if ITE=1. It is cleared by software reading SR1 register followed by a read or write of DR register. It is also cleared by hardware when the interface is disabled (PE=0).

Following a byte transmission, this bit is set after reception of the acknowledge clock pulse. In case an address byte is sent, this bit is set only after the EV6 event (See *Figure 48*). BTF is cleared by reading SR1 register followed by writing the next byte in DR register.

Following a byte reception, this bit is set after transmission of the acknowledge clock pulse if ACK=1. BTF is cleared by reading SR1 register followed by reading the byte from DR register.

The SCL line is held low while BTF=1.

0: Byte transfer not done

1: Byte transfer succeeded

2 **ADSL** *Address matched (Slave mode).* This bit is set by hardware as soon as the received slave address matched with the OAR register content or a general call is recognized. An interrupt is generated if ITE=1. It is cleared by software reading SR1 register or by hardware when the interface is disabled (PE=0).

The SCL line is held low while ADSL=1.

0: Address mismatched or not received

1: Received address matched

1 **M/SL** *Master/Slave.*
This bit is set by hardware as soon as the interface is in Master mode (writing START=1). It is cleared by hardware after detecting a Stop condition on the bus or a loss of arbitration (ARLO=1). It is also cleared when the interface is disabled (PE=0).

0: Slave mode

1: Master mode

0 **SB** *Start bit (Master mode).*
This bit is set by hardware as soon as the Start condition is generated (following a write START=1). An interrupt is generated if ITE=1. It is cleared by software reading SR1 register followed by writing the address byte in DR register. It is also cleared by hardware when the interface is disabled (PE=0).

0: No Start condition

1: Start condition generated

I²C Status register 2 (SR2)

Reset value: 0000 0000 (00h)

7							0
0	0	0	AF	STOPF	ARLO	BERR	GCAL
Read only							

[7:5] Reserved. Forced to 0 by hardware.

4 **AF** *Acknowledge failure*.
This bit is set by hardware when no acknowledge is returned. An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

0: No acknowledge failure

1: Acknowledge failure

Note: While AF=1, the SCL line may be held low due to SB or BTF flags that are set at the same time. It is then necessary to release both lines by software.

3 **STOPF** *Stop detection (Slave mode).*
This bit is set by hardware when a Stop condition is detected on the bus after an acknowledge (if ACK=1). An interrupt is generated if ITE=1. It is cleared by software reading SR2 register or by hardware when the interface is disabled (PE=0).

The SCL line is not held low while STOPF=1.

0: No Stop condition detected

1: Stop condition detected

I²C Data register (DR)

These bits contain the byte to be received or transmitted on the bus.

- Transmitter mode: byte transmission start automatically when the software writes in the DR register.
- Receiver mode: the first data byte is received automatically in the DR register using the least significant bit of the address. The following data bytes are then received one by one after reading the DR register.

Reset value: 0000 0000 (00h)

7							0
D7	D6	D5	D4	D3	D2	D1	D0
Read/write							

I²C Own Address register (OAR)

Reset value: 0000 0000 (00h)

7							0
ADD7	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
Read/write							

[7:1] **ADD[7:1]** *Interface address.*
These bits define the I²C bus address of the interface. They are not cleared when the interface is disabled (PE=0).

0 **ADD0** *Address direction bit.*
This bit is don't care, the interface acknowledges either 0 or 1. It is not cleared when the interface is disabled (PE=0).

Note: Address 01h is always ignored.

2. For SDIP/SO34 devices, the CH3 bit is always at '0'. If, however, set to '1' on error, channel (11:8) becomes enabled which may result in a higher and unnecessary level of consumption.

Data register (DR)

This register contains the converted analog value in the range 00h to FFh.

Reset value: 0000 0000 (00h)

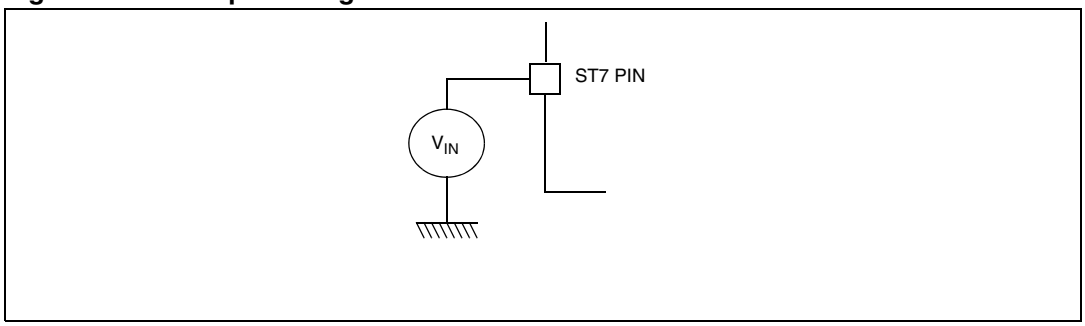
7							0
D7	D6	D5	D4	D3	D2	D1	D0
Read only							

Note: Reading this register reset the COCO flag.

Table 46. ADC register map

Address (Hex.)	Register name	7	6	5	4	3	2	1	0
0Ah	DR	AD7 .. AD0							
0Bh	CSR	COCO	0	ADON	0	CH3	CH2	CH1	CH0

Figure 53. Pin input voltage



13.2 Absolute maximum ratings

Stresses above those listed as “absolute maximum ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device under these conditions is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

Note: Directly connecting the $\overline{RESET}$ and I/O pins to V_{DD} or V_{SS} could damage the device if an unintentional internal reset is generated or an unexpected change of the I/O configuration occurs (for example, due to a corrupted program counter). To guarantee safe operation, this connection has to be done through a pull-up or pull-down resistor (typical: 4.7 kΩ for $\overline{RESET}$, 10 kΩ for I/Os). Unused I/O pins must be tied in the same way to V_{DD} or V_{SS} according to their reset configuration.

Table 55. Voltage characteristics

Symbol	Ratings	Maximum value	Unit
V_{DD} - V_{SS}	Supply voltage	6.0	V
V_{IN} (1)(2)	Input voltage on true open drain pins	V_{SS}-0.3 to 6.0	
	Input voltage on any other pin	V_{SS}-0.3 to V_{DD}+0.3	
$V_{ESD(HBM)}$	Electrostatic discharge voltage (Human Body model)	*Section 13.7.3*	

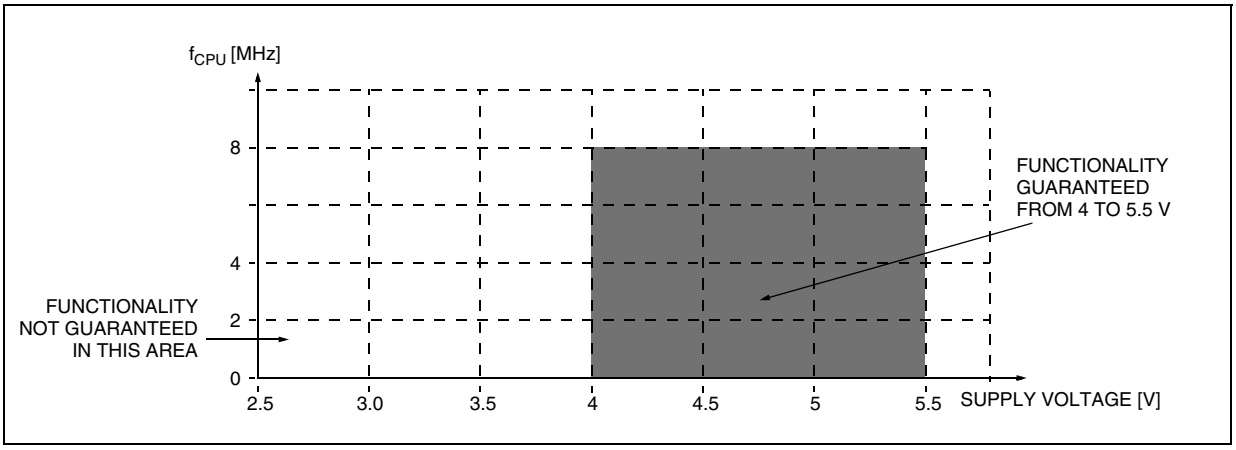
1. Directly connecting the RESET and I/O pins to V_{DD} or V_{SS} could damage the device if an unintentional internal reset is generated or an unexpected change of the I/O configuration occurs (for example, due to a corrupted program counter). To guarantee safe operation, this connection has to be done through a pull-up or pull-down resistor (typical: 4.7 kΩ for RESET, 10 kΩ for I/Os). Unused I/O pins must be tied in the same way to VDD or VSS according to their reset configuration.
2. $I_{INJ(PIN)}$ must never be exceeded. This is implicitly insured if V_{IN} maximum is respected. If V_{IN} maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by $V_{IN}>V_{DD}$ while a negative injection is induced by $V_{IN}<V_{SS}$. For true open-drain pads, there is no positive injection current, and the corresponding V_{IN} maximum must always be respected.

13.3 Operating conditions

Table 58. General operating conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD}	Operating Supply Voltage	f_{CPU} = 8 MHz	4	5	5.5	V
V_{DDA}	Analog reference voltage		V_{DD}	-	V_{DD}	
V_{SSA}	Analog reference voltage		V_{SS}	-	V_{SS}	
f_{CPU}	Operating frequency	f_{OSC} = 24 MHz	-	-	8	MHz
		f_{OSC} = 12 MHz	-	-	4	
T_A	Ambient temperature range		0	-	70	°C

Figure 54. f_{CPU} maximum operating frequency versus V_{DD} supply voltage



13.3.1 Operating conditions with low voltage detector (LVD)

Subject to general operating conditions for V_{DD}, f_{CPU}, and T_A. Refer to *Figure 12 on page 33*.

Table 59. Operating conditions with LVD

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IT+}	Low voltage reset threshold (V_{DD} rising)	V_{DD} max. variation 50 V/ms	3.4	3.7	4.0	V
V_{IT-}	Low voltage reset threshold (V_{DD} falling)	V_{DD} max. variation 50V/ms	3.2	3.5	3.8	V
V_{hyst}	Hysteresis (V_{IT+} - V_{IT-})(1)		100	175	220	mV
Vt_{POR}	V_{DD} rise time rate(2)		0.5	-	50	V/ms

1. Guaranteed by characterization - not tested in production.
2. The V_{DD} rise time rate condition is needed to insure a correct device power-on and LVD reset. Not tested in production.

13.7.2 Electromagnetic Interference (EMI)

Based on a simple application running on the product (toggling 2 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 67. EMI characteristics

Symbol	Parameter	Conditions	Monitored frequency band	Max vs. [f_{OSC}/f_{CPU}]	Unit
				16/8 MHz	
S_{EMI}	Peak level(1)	V_{DD}=5 V, T_A=+25 °C, SDIP32 package conforming to SAE J 1752/3(2)	0.1 MHz to 30 MHz	36	dBµV
			30 MHz to 130 MHz	39	
			130 MHz to 1 GHz	26	
			SAE EMI Level	3.5	-

1. Data based on characterization results, not tested in production.
2. Refer to application note AN1709 for data on other package types.

13.7.3 Absolute maximum ratings (electrical sensitivity)

Based on three different tests (ESD, and LU) using specific measurement methods, the product is stressed in order to determine its performance in terms of electrical sensitivity. For more details, refer to the application note AN1181.

Electrostatic discharge (ESD)

Electrostatic discharges (a positive then a negative pulse separated by 1 second) are applied to the pins of each sample according to each pin combination. The sample size depends on the number of supply pins in the device (3 parts*(n+1) supply pin). This test conforms to the JESD22-A114A/A115A standard.

Table 68. Absolute maximum ratings

Symbol	Ratings	Conditions	Maximum value(1)	Unit
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	T_A=+25 °C	2000	V

1. Data based on characterization results, not tested in production.

Static latchup (LU)

3 complementary static tests are required on 10 parts to assess the latchup performance. A supply overvoltage (applied to each power supply pin) and a current injection (applied to each input, output and configurable I/O pin) are performed on each sample. This test conforms to the EIA/JESD 78 IC latch-up standard. For more details, refer to the application note AN1181.

Figure 64. V_{OL} high sink V_{DD}=5 V

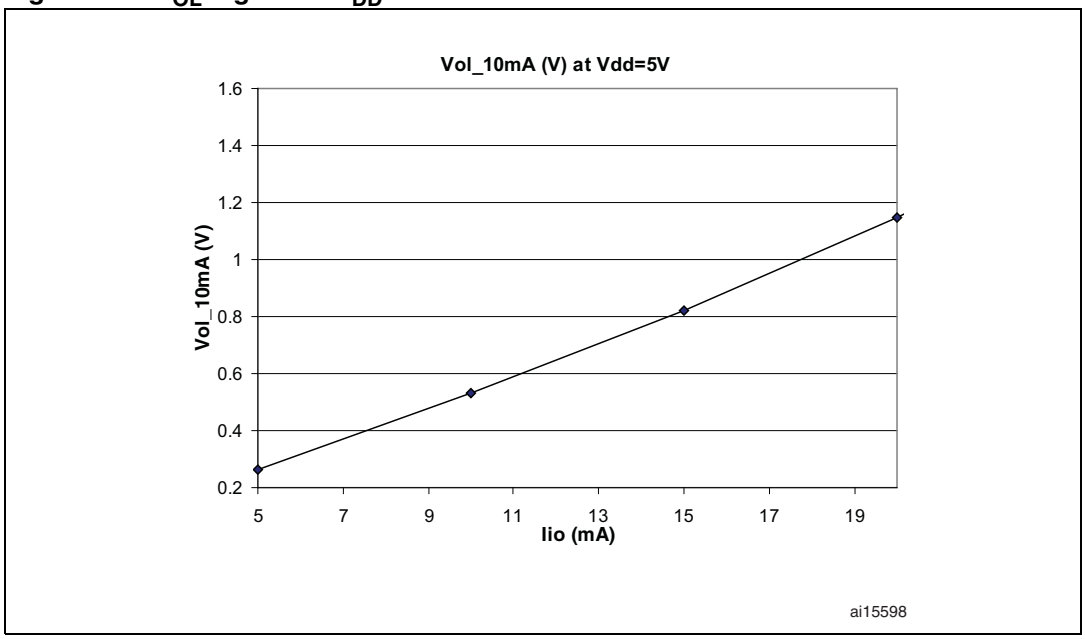
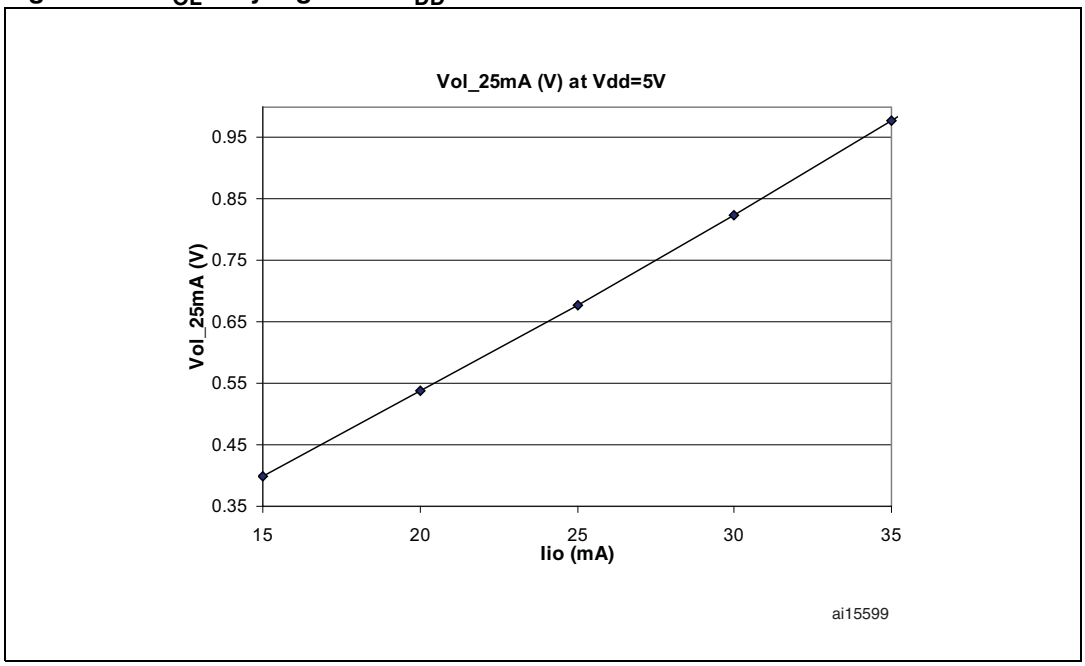


Figure 65. V_{OL} very high sink V_{DD}=5 V



OPT 2 Reserved.

OPT 1 **OSC24/12** *Oscillator Selection*
This option bit selects the clock divider used to drive the USB interface at 6MHz.
0: 24 MHz oscillator
1: 12 Mhz oscillator

OPT 0 **FMP_R** Flash memory readout protection
This option indicates if the user flash memory is protected against readout. Readout protection, when selected, provides a protection against program memory content extraction and against write access to Flash memory. Erasing the option bytes when the FMP_R option is selected, causes the whole user memory to be erased first and the device can be reprogrammed. Refer to the ST7 Flash Programming Reference Manual and *Section 4.3.1: Readout protection* for more details.
0: Readout protection enabled
1: Readout protection disabled

15.2 Device ordering information and transfer of customer code

Customer code is made up of the ROM contents and the list of the selected options (if any). The ROM contents are to be sent on diskette, or by electronic means, with the hexadecimal file in .S19 format generated by the development tool. All unused bytes must be set to FFh.

The selected options are communicated to STMicroelectronics using the correctly completed option list appended (see *Section 15.2*).

Refer to application note AN1635 for information on the counter listing returned by ST after code has been transferred.

The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.

Table 86. Supported order codes

Sales type[1][2]	Program memory (bytes)	RAM (bytes)	Package
ST72F63BH6T1	32K Flash	1024	LQFP48
ST72F63BD6U1			QFN40
ST72F63BK6M1			SO34
ST72F63BK6B1			SDIP32
ST72F63BE6M1			SO24
ST72F63BH4T1	16K Flash	512	LQFP48
ST72F63BK4M1			SO34
ST72F63BK4B1			SDIP32
ST72F63BE4M1			SO24

15.4 ST7 application notes

Table 88. ST7 application notes

Identification	Description
Application examples	
AN1658	Serial Numbering Implementation
AN1720	Managing the Readout Protection in Flash Microcontrollers
AN1755	A High Resolution/precision Thermometer Using ST7 and NE555
AN1756	Choosing a DALI Implementation Strategy with ST7DALI
AN1812	A High Precision, Low Cost, Single Supply ADC for Positive and Negative input Voltages
Example drivers	
AN 969	SCI Communication Between ST7 and PC
AN 971	I²C Communication Between ST7 and M24Cxx EEPROM
AN 973	SCI Software Communication with a PC Using ST72251 16-Bit Timer
AN 974	Real Time Clock with ST7 Timer output Compare
AN 976	Driving a Buzzer Through ST7 Timer PWM Function
AN 979	Driving an Analog Keyboard with the ST7 ADC
AN 980	ST7 Keypad Decoding Techniques, Implementing wakeup on Keystroke
AN1017	Using the ST7 Universal Serial Bus Microcontroller
AN1041	Using ST7 PWM Signal to Generate Analog output (Sinusoïd)
AN1042	ST7 Routine for I²C Slave mode Management
AN1044	Multiple Interrupt Sources Management for ST7 MCUs
AN1045	ST7 S/W Implementation of I²C Bus Master
AN1046	UART Emulation Software
AN1047	Managing Reception Errors with the ST7 SCI Peripherals
AN1048	ST7 Software LCD Driver
AN1078	PWM Duty Cycle Switch Implementing True 0% & 100% Duty Cycle
AN1082	Description of the ST72141 Motor Control Peripherals registers
AN1083	ST72141 BLDC Motor Control Software and Flowchart Example
AN1105	ST7 pCAN Peripheral Driver
AN1129	PWM Management for BLDC Motor Drives Using the ST72141
AN1130	An Introduction to Sensorless Brushless DC Motor Drive Applications with the ST72141
AN1148	Using the ST7263 for Designing a USB Mouse
AN1149	Handling Suspend mode on a USB Mouse
AN1180	Using the ST7263 Kit to Implement a USB Game Pad