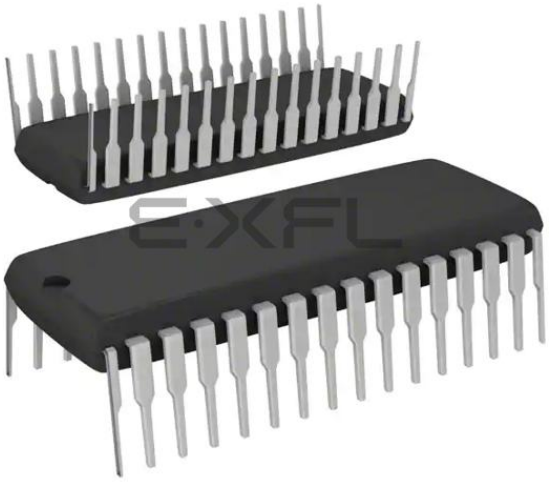


Welcome to E-XFL.COM

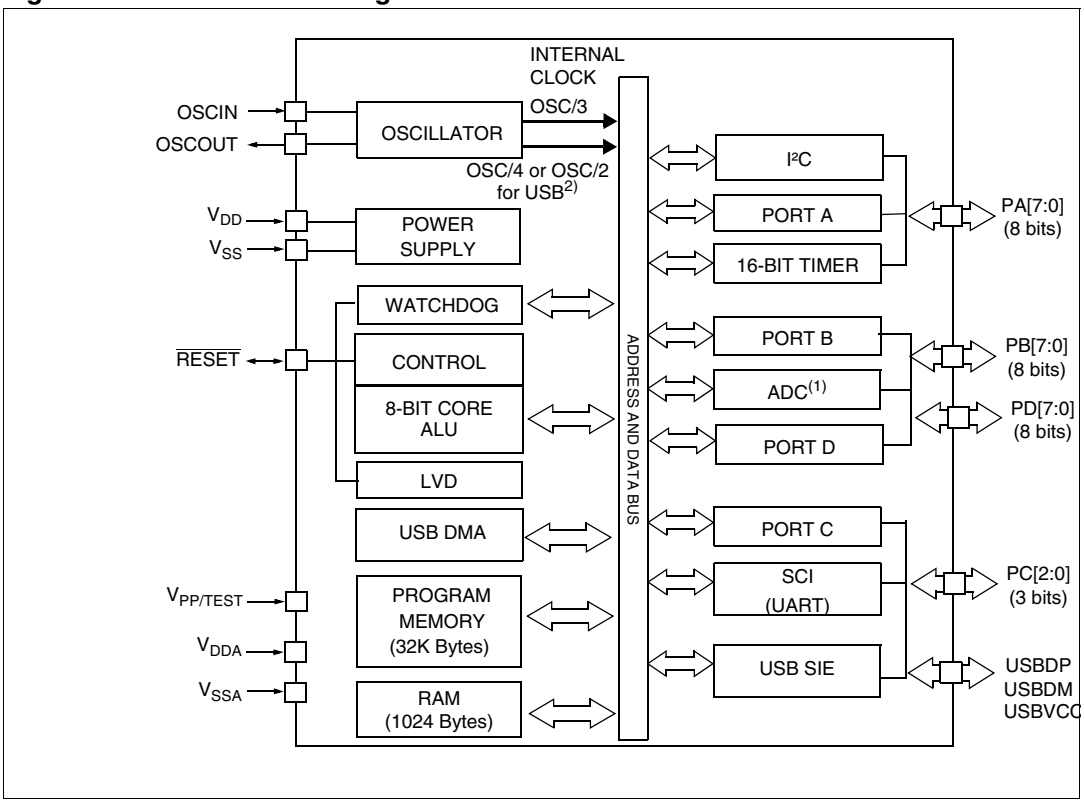
What is "Embedded - Microcontrollers"?

"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	19
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	32-SDIP (0.400", 10.16mm)
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f63bk1b1

Figure 1. General block diagram



1. ADC channels:
   12 on 48-pin devices (Port B and Port D[3:0])
   8 on 34 and 32-pin devices (Port B)
   None on 24-pin devices
2. 12 or 24 MHz OSCIN frequency required to generate 6 MHz USB clock.
3. The drive from USBVCC is sufficient to only drive an external pull-up in addition to the internal transceiver.

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.

7 Interrupts

The ST7 core may be interrupted by one of two different methods: maskable hardware interrupts as listed in *Table 9* and a non-maskable software interrupt (TRAP). The Interrupt processing flowchart is shown in *Figure 19*.

The maskable interrupts must be enabled clearing the I bit in order to be serviced. However, disabled interrupts may be latched and processed when they are enabled (see external interrupts subsection).

When an interrupt has to be serviced:

- Normal processing is suspended at the end of the current instruction execution.
- The PC, X, A and CC registers are saved onto the stack.
- The I bit of the CC register is set to prevent additional interrupts.
- The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to *Table 9* for vector addresses).

The interrupt service routine should finish with the IRET instruction which causes the contents of the saved registers to be recovered from the stack.

Note: As a consequence of the IRET instruction, the I bit will be cleared and the main program will resume.

Priority management

By default, a servicing interrupt cannot be interrupted because the I bit is set by hardware entering in interrupt routine.

In the case several interrupts are simultaneously pending, a hardware priority defines which one will be serviced first (see *Table 9*).

Non-maskable software interrupts

This interrupt is entered when the TRAP instruction is executed regardless of the state of the I bit. It will be serviced according to the flowchart on *Figure 19*.

Interrupts and low power mode

All interrupts allow the processor to leave the Wait low power mode. Only external and specific mentioned interrupts allow the processor to leave the Halt low power mode (refer to the “Exit from HALT“ column in *Table 9*).

External interrupts

The pins ITi/PAk and ITj/PBk (i=1,2; j= 5,6; k=4,5) can generate an interrupt when a rising edge occurs on this pin. Conversely, the ITl/PAn and ITm/PBn pins (l=3,4; m= 7,8; n=6,7) can generate an interrupt when a falling edge occurs on this pin.

Interrupt generation will occur if it is enabled with the ITiE bit (i=1 to 8) in the ITRFRE register and if the I bit of the CC is reset.

Table 9. Interrupt mapping

N°	Source block	Description	Register label	Priority order	Exit from Halt	Vector address
	RESET	Reset	N/A	Highest Priority ↓ Lowest Priority	yes	FFFEh-FFFFh
	TRAP	Software interrupt			no	FFFCh-FFFDh
	FLASH	Flash Start Programming interrupt			yes	FFFAh-FFFBh
	USB	End Suspend mode	ISTR		yes	FFF8h-FFF9h
1	ITi	External interrupts	ITRFRE			FFF6h-FFF7h
2	TIMER	Timer Peripheral interrupts	TIMSR		no	FFF4h-FFF5h
3	I²C	I²C Peripheral interrupts	I²CSR1			FFF2h-FFF3h
			I²CSR2			
4	SCI	SCI Peripheral interrupts	SCISR			FFF0h-FFF1h
5	USB	USB Peripheral interrupts	ISTR			FFEEh-FFEFh

7.1 Interrupt register (ITRFRE)

Address: 0008h

Reset value: 0000 0000 (00h)

7							0
IT8E	IT7E	IT6E	IT5E	IT4E	IT3E	IT2E	IT1E
Read/write							

[7:0] **TiE (i=1 to 8)**. *Interrupt Enable Control Bits*.

If an ITiE bit is set, the corresponding interrupt is generated when

- A rising edge occurs on the pin PA4/IT1 or PA5/IT2 or PB4/IT5 or PB5/IT6
- Or a falling edge occurs on the pin PA6/IT3 or PA7/IT4 or PB6/IT7 or PB7/IT8

No interrupt is generated elsewhere.

Note: Analog input must be disabled for interrupts coming from port B.

9.3 I/O port implementation

The hardware implementation on each I/O port depends on the settings in the DDR register and specific feature of the I/O port such as ADC input or true open drain.

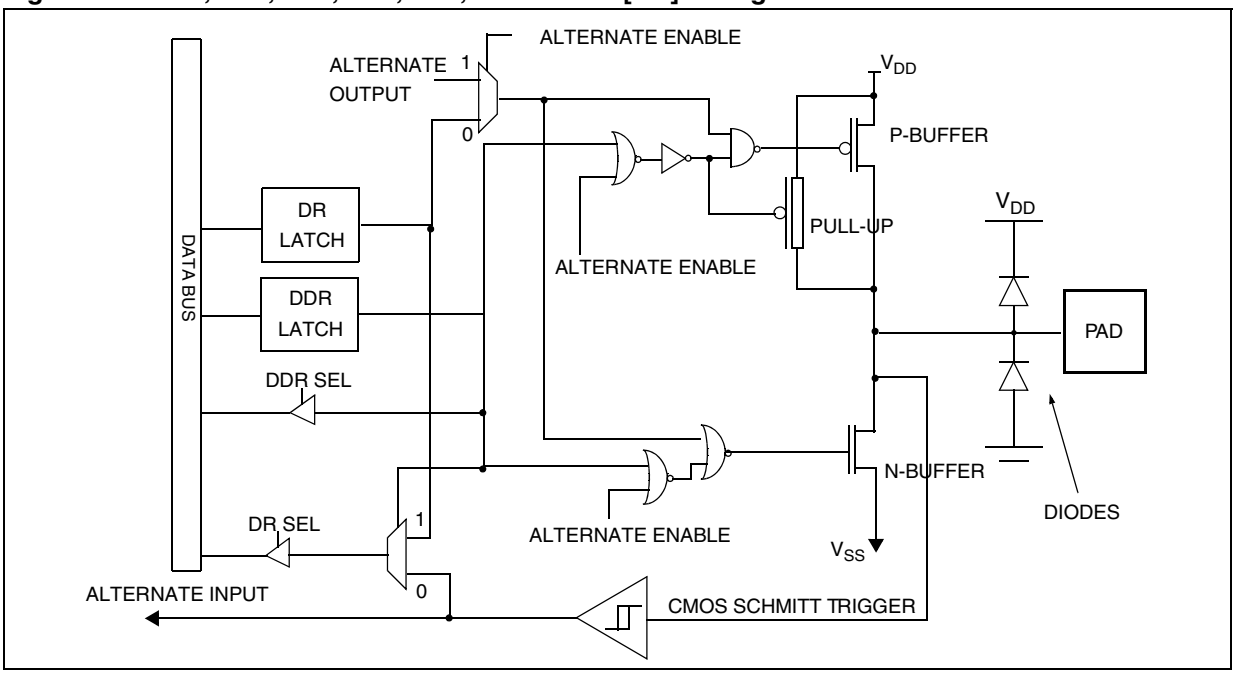
9.3.1 Port A

Table 11. Port A0, A3, A4, A5, A6, A7 description

PORT A	I/Os		Alternate function	
	Input(1)	Output	Signal	Condition
PA0	with pull-up	push-pull	MCO (Main Clock output)	MCO = 1 (MISCR)
PA3	with pull-up	push-pull	Timer EXTCLK	CC1 =1 CC0 = 1 (Timer CR2)
PA4	with pull-up	Push-pull	Timer ICAP1	
			IT1 Schmitt triggered input	IT1E = 1 (ITIFRE)
PA5	with pull-up	Push-pull	Timer ICAP2	
			IT2 Schmitt triggered input	IT2E = 1 (ITIFRE)
PA6(2)	with pull-up	Push-pull	Timer OCMP1	OC1E = 1
			IT3 Schmitt triggered input	IT3E = 1 (ITIFRE)
PA7	with pull-up	Push-pull	Timer OCMP2	OC2E = 1
			IT4 Schmitt triggered input	IT4E = 1 (ITIFRE)

1. Reset state.
2. Not available on SO24

Figure 22. PA0, PA3, PA4, PA5, PA6, PA7 and PD[7:4] configuration



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

10 Miscellaneous register

Miscellaneous register (MISCR)

Address: 0009h

Reset value: 0000 0000 (00h)

7							0
-	-	-	-	-	SMS	USBOE	MCO
Read/write							

[7:3] Reserved

2 **SMS** *Slow mode Select*.

This bit is set by software and only cleared by hardware after a reset. If this bit is set, it enables the use of an internal divide-by-2 clock divider (refer to *Figure 18 on page 36*). The SMS bit has no effect on the USB frequency.

0: Divide-by-2 disabled and CPU clock frequency is standard

1: Divide-by-2 enabled and CPU clock frequency is halved.

1 **USBOE** *USB enable.*

If this bit is set, the port PC2 (PB1 on SO24) outputs the USB output enable signal (at “1” when the ST7 USB is transmitting data).

Unused bits 7-4 are set.

0 **MCO** *Main Clock Out selection*

This bit enables the MCO alternate function on the PA0 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)

PRESC= Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits, see *Table 24*)

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

$$\Delta OCiR = \Delta t * f_{EXT}$$

Where:

Δt = Output compare period (in seconds)

f_{EXT} = External timer clock frequency (in hertz)

Clearing the output compare interrupt request (that is, clearing the OCF*i* bit) is done by:

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

The following procedure is recommended to prevent the OCF*i* bit from being set between the time it is read and the write to the OC*i*R register:

- Write to the OC*i*HR register (further compares are inhibited).
- Read the SR register (first step of the clearance of the OCF*i* bit, which may be already set).
- Write to the OC*i*LR register (enables the output compare function and clears the OCF*i* bit).

Note: 1 After a processor write cycle to the OCiHR register, the output compare function is inhibited until the OCiLR register is also written.

2 If the OCiE bit is not set, the OCMPi pin is a general I/O port and the OLVLi bit will not appear when a match is found but an interrupt could be generated if the OCIE bit is set.

3 In both internal and external clock modes, OCFi and OCMPi are set while the counter value equals the OCiR register value (see Figure 35 on page 67 for an example with $f_{CPU}/2$ and Figure 36 on page 67 for an example with $f_{CPU}/4$). This behavior is the same in OPM or PWM mode.

4 The output compare functions can be used both for generating external events on the OCMPi pins even if the input capture mode is also used.

5 The value in the 16-bit OCiR register and the OLVi bit should be changed after each successful comparison in order to control an output waveform or establish a new elapsed timeout.

Forced Compare output capability

When the FOLV*i* bit is set by software, the OLVL*i* bit is copied to the OCMP*i* pin. The OLV*i* bit has to be toggled in order to toggle the OCMP*i* pin when it is enabled (OC*i*E bit = 1). The OCF*i* bit is then not set by hardware, and thus no interrupt request is generated.

The FOLVL*i* bits have no effect in both One Pulse mode and PWM mode.

When a overrun error occurs:

- The OR bit is set.
- The RDR content will not be lost.
- The shift register will be overwritten.
- An interrupt is generated if the RIE bit is set and the I bit is cleared in the CC register.

The OR bit is reset by an access to the SCISR register followed by a SCIDR register read operation.

Noise error

Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise. Normal data bits are considered valid if three consecutive samples (8th, 9th, 10th) have the same bit value, otherwise the NF flag is set. In the case of start bit detection, the NF flag is set on the basis of an algorithm combining both valid edge detection and three samples (8th, 9th, 10th). Therefore, to prevent the NF flag getting set during start bit reception, there should be a valid edge detection as well as three valid samples.

When noise is detected in a frame:

- The NF flag is set at the rising edge of the RDRF bit.
- Data is transferred from the Shift register to the SCIDR register.
- No interrupt is generated. However this bit rises at the same time as the RDRF bit which itself generates an interrupt.

The NF flag is reset by a SCISR register read operation followed by a SCIDR register read operation.

During reception, if a false start bit is detected (e.g. 8th, 9th, 10th samples are 011,101,110), the frame is discarded and the receiving sequence is not started for this frame. There is no RDRF bit set for this frame and the NF flag is set internally (not accessible to the user). This NF flag is accessible along with the RDRF bit when a next valid frame is received.

Note: If the application Start Bit is not long enough to match the above requirements, then the NF Flag may get set due to the short Start Bit. In this case, the NF flag may be ignored by the application software when the first valid byte is received.

See also *Section* .

Framing error

A framing error is detected when:

- The stop bit is not recognized on reception at the expected time, following either a de-synchronization or excessive noise.
- A break is received.

When the framing error is detected:

- the FE bit is set by hardware
- Data is transferred from the Shift register to the SCIDR register.
- No interrupt is generated. However this bit rises at the same time as the RDRF bit which itself generates an interrupt.

The FE bit is reset by a SCISR register read operation followed by a SCIDR register read operation.

Interrupt/DMA register (IDR)

Reset value: xxxx 0000 (x0h)

7							0
DA7	DA6	EP1	EP0	CNT3	CNT2	CNT1	CNT0
Read.write							

[7:6] **DA[7:6]** *DMA address bits 7-6.*
Software must reset these bits. See the description of the DMAR register and *Figure 45*.

[5:4] **EP[1:0]** *Endpoint number* (read-only). These bits identify the endpoint which required attention.

00: Endpoint 0

01: Endpoint 1

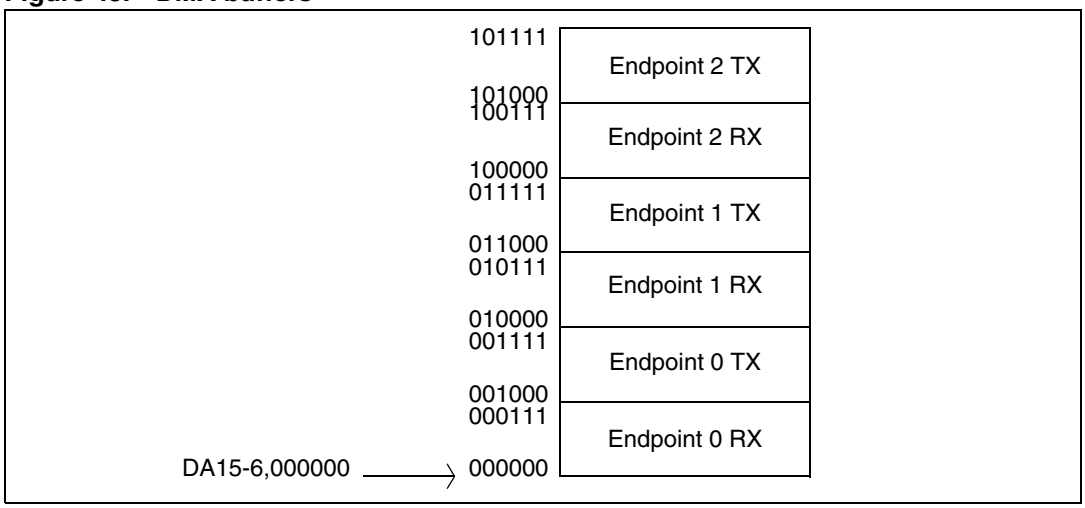
10: Endpoint 2

When a CTR interrupt occurs (see register ISTR) the software should read the EP bits to identify the endpoint which has sent or received a packet.

[3:0] **CNT[3:0]** *Byte count* (read only).
This field shows how many data bytes have been received during the last data reception.

Note: Not valid for data transmission.

Figure 45. DMA buffers



2 **ARLO** *Arbitration lost.*
This bit is set by hardware when the interface loses the arbitration of the bus to another master. 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).
After an ARLO event the interface switches back automatically to Slave mode (M/SL=0).
The SCL line is not held low while ARLO=1.
0: No arbitration lost detected
1: Arbitration lost detected
Note: In a Multimaster environment, when the interface is configured in Master Receive mode it does not perform arbitration during the reception of the Acknowledge Bit. Mishandling of the ARLO bit from the I2CSR2 register may occur when a second master simultaneously requests the same data from the same slave and the I^2C master does not acknowledge the data. The ARLO bit is then left at 0 instead of being set.

1 **BERR** *Bus error.*
This bit is set by hardware when the interface detects a misplaced Start or Stop condition. 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 BERR=1.
0: No misplaced Start or Stop condition
1: Misplaced Start or Stop condition
Note: If a Bus Error occurs, a Stop or a repeated Start condition should be generated by the Master to re-synchronize communication, get the transmission acknowledged and the bus released for further communication

0 **GCAL** *General Call (Slave mode).*
This bit is set by hardware when a general call address is detected on the bus while ENGC=1. It is cleared by hardware detecting a Stop condition (STOPF=1) or when the interface is disabled (PE=0).
0: No general call address detected on bus
1: general call address detected on bus

I²C Clock Control register (CCR)

Reset value: 0000 0000 (00h)

7							0
FM/SM	CC6	CC5	CC4	CC3	CC2	CC1	CC0
Read/write							

7 **FM/SM** *Fast/Standard I²C mode.*
This bit is set and cleared by software. It is not cleared when the interface is disabled (PE=0).
0: Standard I²C mode
1: Fast I²C mode

[6:0] **CC[6:0]** *7-bit clock divider.*
These bits select the speed of the bus (F_{SCL}) depending on the I²C mode. They are not cleared when the interface is disabled (PE=0).
Refer to the Electrical Characteristics section for the table of value.
Note: The programmed F_{SCL} assumes no load on SCL and SDA lines.

Table 51. Instructions supporting Direct, Indexed, Indirect and Indirect Indexed addressing modes (continued)

Long and Short instructions	Function
BCP	Bit Compare
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 52. Instructions supporting relative addressing mode

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

The relative addressing mode consists of two sub-modes:

Relative (Direct)

The offset follows the opcode.

Relative (Indirect)

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

13.4 Supply current characteristics

The following current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To get the total device consumption, the two current values must be added (except for Halt mode for which the clock is stopped).

Table 60. Supply current characteristics

Symbol	Parameter	Conditions		Typ	Max	Unit
$\Delta I_{DD(\Delta Ta)}$	Supply current variation vs. temperature	Constant V_{DD} and f_{CPU}		-	10(1)	%
I_{DD}	CPU Run mode	I/Os in input mode	f_{CPU} = 4 MHz	7.5	9(2)(1)	mA
			f_{CPU} = 8 MHz	10.5	13(2)	
	CPU Wait mode		f_{CPU} = 4 MHz	6	8(1)	mA
			f_{CPU} = 8 MHz	8.5	11(2)	
	CPU Halt mode(3)	LVD disabled		25	40(1)	µA
	USB Suspend mode(4)	LVD disabled		100	120	µA
		LVD enabled		230	-	

1. Not tested in production, guaranteed by characterization.
2. Oscillator and watchdog running. All others peripherals disabled.
3. USB Transceiver and ADC are powered down.
4. CPU in Halt mode. Current consumption of external pull-up (1.5Kohms to USBVCC) and pull-down (15Kohms to V_{SSA}) not included.

Figure 55. Typ. I_{DD} in Run at f_{CPU} = 4 and 8 MHz

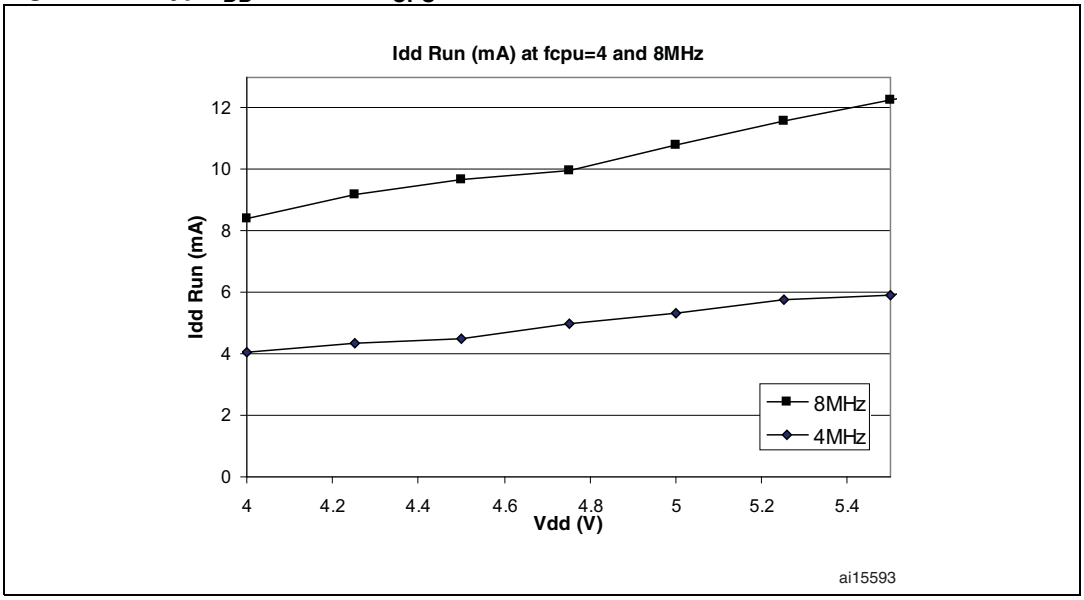


Figure 61. Typ. I_{PU} vs. V_{DD}

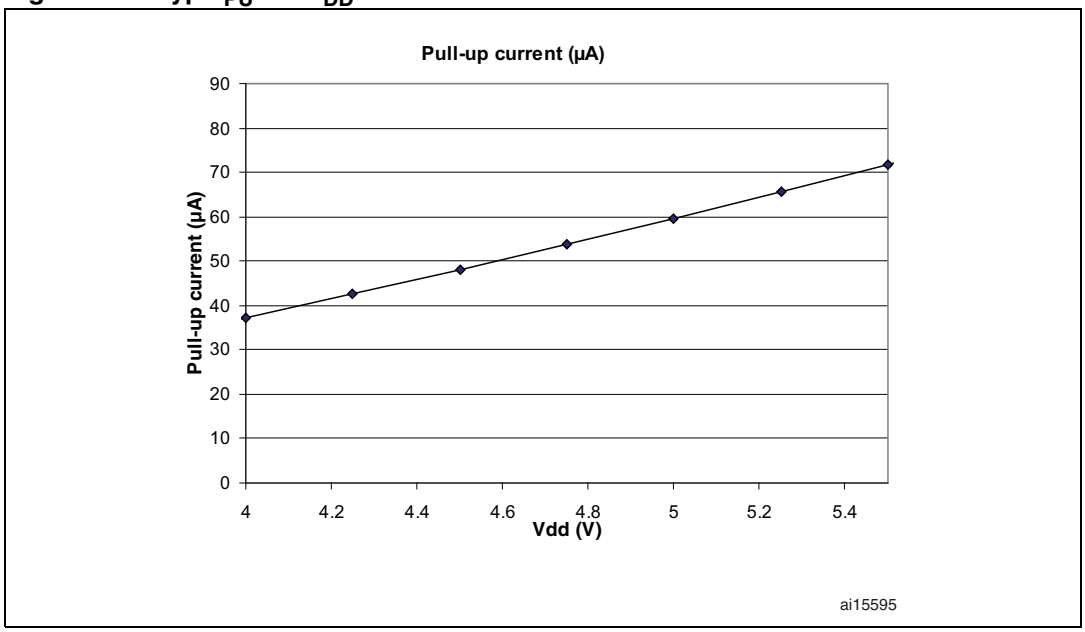


Figure 62. Typ. R_{PU} vs. V_{DD}

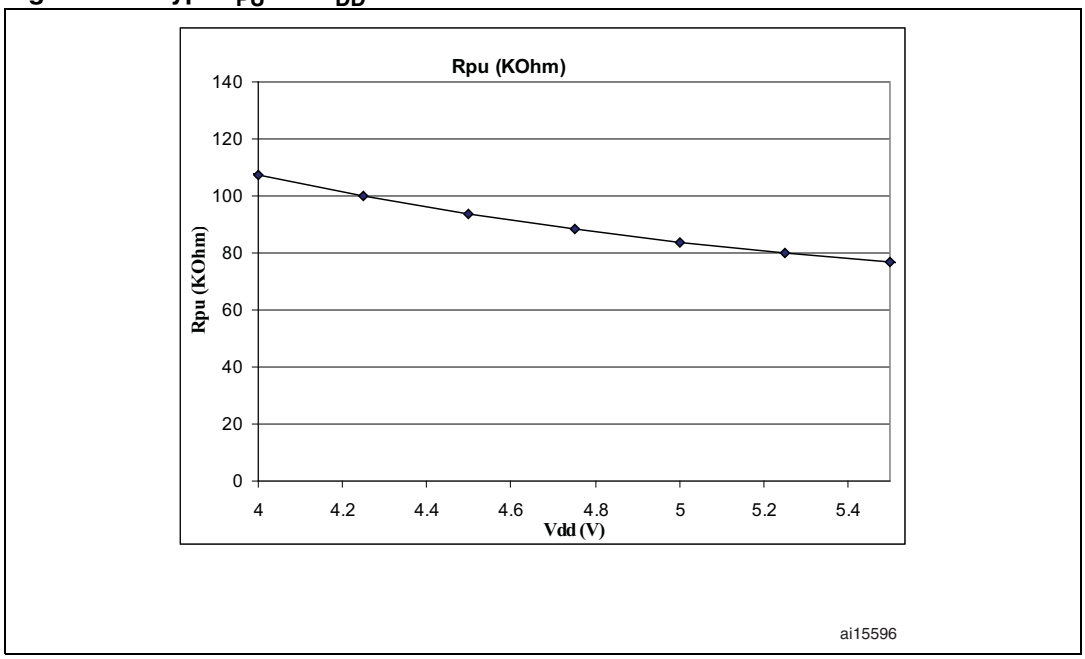


Figure 66. V_{OL} standard vs. V_{DD}

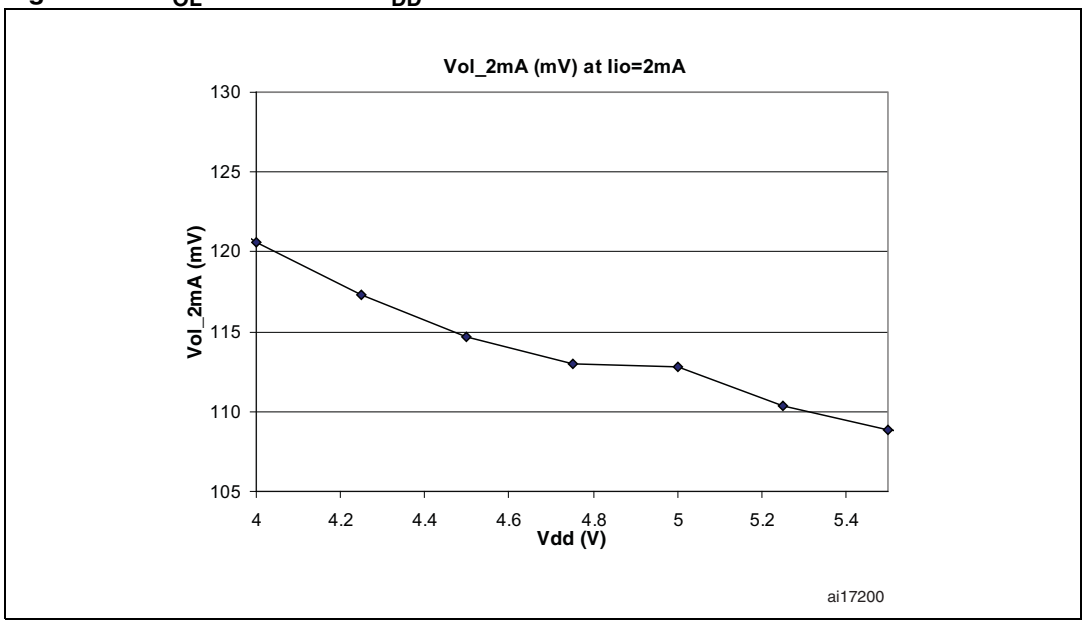


Figure 67. V_{OL} high sink vs. V_{DD}

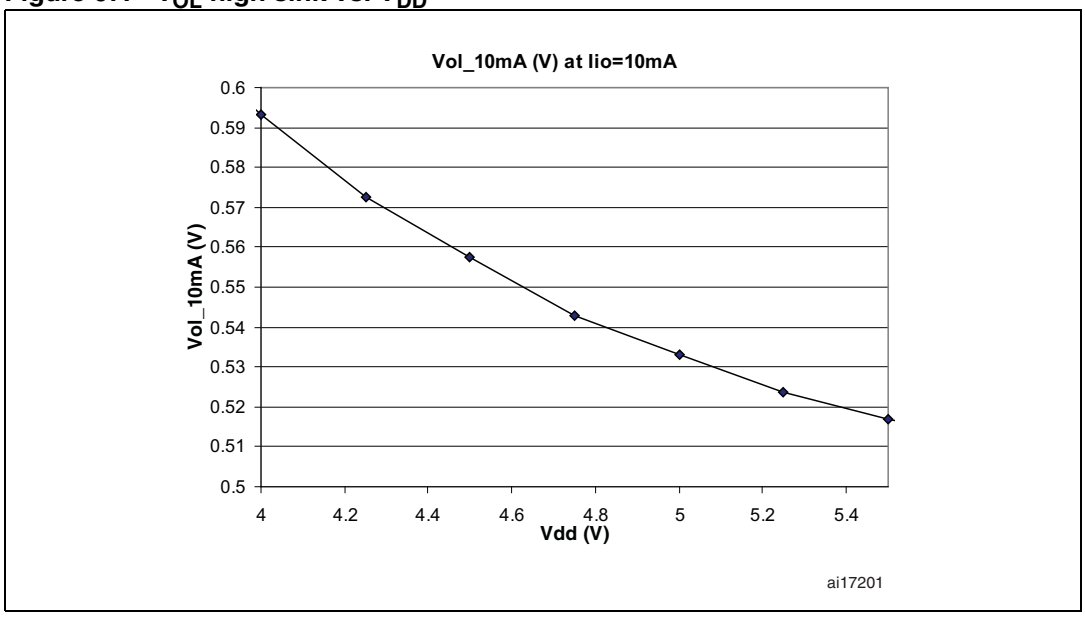
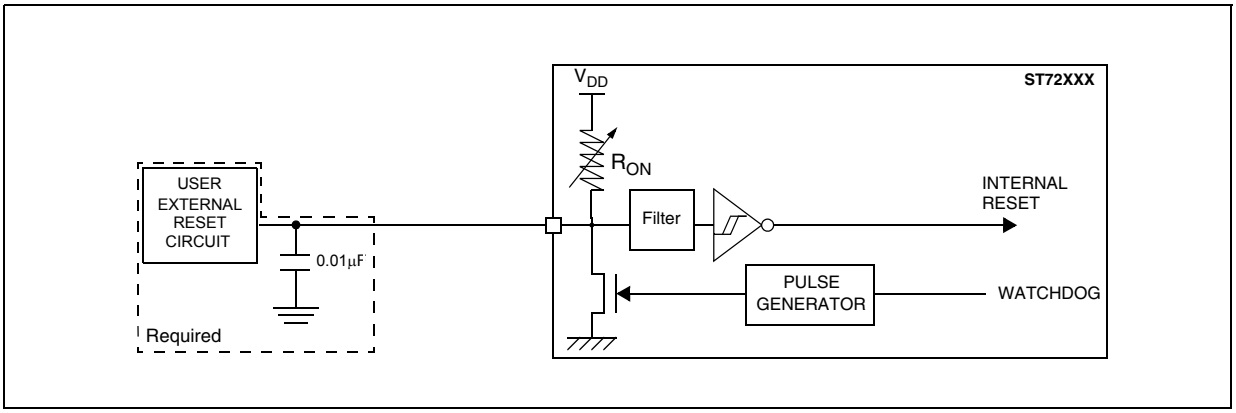


Figure 74. $\overline{\text{RESET}}$ pin protection when LVD is disabled



13.10 Communication interface characteristics

13.10.1 USB interface

Operating conditions T_A = 0 to +70 °C, V_{DD} = 4.0 to 5.25 V unless otherwise specified.

Table 73. USB DC characteristics

Symbol	Parameter	Conditions	Min.	Max.	Unit
V_{DI}	Differential input sensitivity	I(D+, D-)	0.2	-	V(1)
V_{CM}	Differential common mode range	Includes V_{DI} range	0.8	2.5	
V_{SE}	Single ended receiver threshold		0.8	2.0	
V_{OL}	Static output low	R_L(2)of 1.5 KΩ to 3.6 V	-	0.3	
V_{OH}	Static output high	R_L(2) of 15 KΩ to V_{SS}	2.8	3.6	
USBV	USBVCC: voltage level(3)	V_{DD}=5 V	3.00	3.60	

1. All the voltages are measured from the local ground potential.
2. R_L is the load connected on the USB drivers.
3. To improve EMC performance (noise immunity), it is recommended to connect a 100nF capacitor to the USBVCC pin.

Figure 75. USB data signal rise and fall time

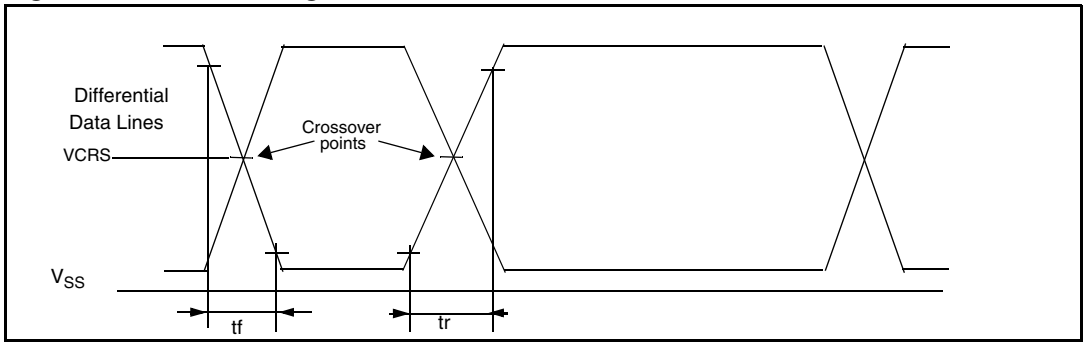


Figure 77. Typical application with ADC

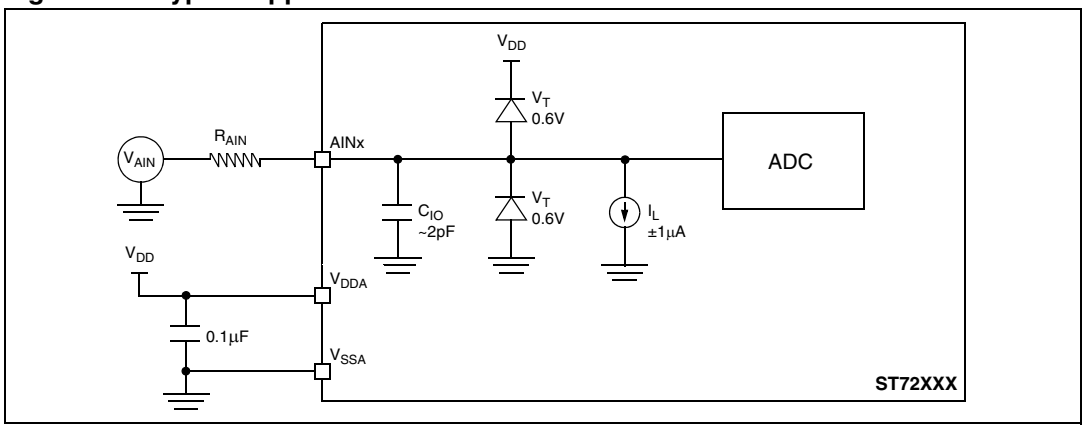


Table 79. ADC accuracy with V_{DD}=5 V, f_{CPU}= 8 MHz, f_{ADC}=4 MHz, R_{AIN}< 10 kΩ

Symbol	Parameter	Typ	Max(1)(2)
$\|E_T\|$	Total unadjusted error(3)	1.5	2
$\|E_{OI}\|$	Offset error(3)	0.5	1
$\|E_{GI}\|$	Gain Error(3)	0.5	1.5
$\|E_D\|$	Differential linearity error(3)	1	1.5
$\|E_L\|$	Integral linearity error(3)	1	1.5

1. Data based on characterization results over the whole temperature range, not tested in production.
2. Data based on characterization results, to guarantee 99.73% within ± max value from 0 to 70 °C (± 3s distribution limits).
3. ADC Accuracy vs. Negative Injection Current:
For I_{INJ-}=0.8mA, the typical leakage induced inside the die is 1.6µA and the effect on the ADC accuracy is a loss of 1 LSB for each 10KΩ increase of the external analog source impedance. This effect on the ADC accuracy has been observed under worst-case conditions for injection:
- negative injection
- injection to an input with analog capability, adjacent to the enabled Analog input
- at 5V V_{DD} supply, and worst case temperature.

Table 88. ST7 application notes (continued)

Identification	Description
AN1276	BLDC Motor Start Routine for the ST72141 Microcontroller
AN1321	Using the ST72141 Motor Control MCU in Sensor mode
AN1325	Using the ST7 USB LOW-SPEED Firmware V4.x
AN1445	Emulated 16-bit Slave SPI
AN1475	Developing an ST7265X Mass Storage Application
AN1504	Starting a PWM Signal Directly at High Level Using the ST7 16-bit Timer
AN1602	16-bit Timing Operations Using ST7262 or ST7263B ST7 USB MCUs
AN1633	Device Firmware Upgrade (DFU) Implementation in ST7 Non-USB Applications
AN1712	Generating a High Resolution Sinewave Using ST7 PWMART
AN1713	SMBus Slave Driver for ST7 I^2C Peripherals
AN1753	Software UART Using 12-bit ART
AN1947	ST7MC PMAC Sine Wave Motor Control Software Library
General purpose	
AN1476	Low Cost Power Supply for Home Appliances
AN1526	ST7FLITE0 Quick Reference Note
AN1709	EMC Design for ST Microcontrollers
AN1752	ST72324 Quick Reference Note
Product evaluation	
AN 910	Performance Benchmarking
AN 990	ST7 Benefits vs Industry Standard
AN1077	Overview of Enhanced CAN Controllers for ST7 and ST9 MCUs
AN1086	U435 Can-Do Solutions for Car Multiplexing
AN1103	Improved B-EMF detection for Low Speed, Low Voltage with ST72141
AN1150	Benchmark ST72 vs PC16
AN1151	Performance Comparison Between ST72254 & PC16F876
AN1278	LIN (Local Interconnect Network) Solutions
Product migration	
AN1131	Migrating Applications from ST72511/311/214/124 to ST72521/321/324
AN1322	Migrating an Application from ST7263 Rev.B to ST7263B
AN1365	Guidelines for Migrating ST72C254 Applications to ST72F264
AN1604	How to Use ST7MDT1-TRAIN with ST72F264
AN2200	Guidelines for Migrating ST7LITE1x Applications to ST7FLITE1xB
Product optimization	
AN 982	Using ST7 with Ceramic Resonator