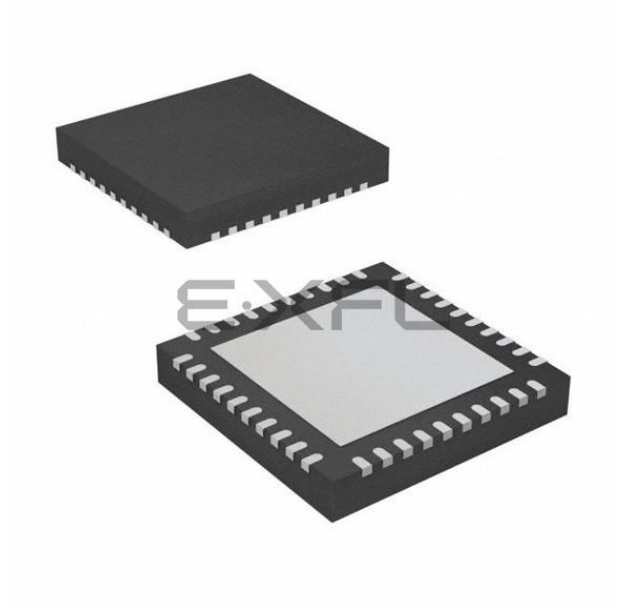


Welcome to E-XFL.COM

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	SCI, UART/USART, USB
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	19
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	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	40-VFQFN Exposed Pad
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f60k2u1tr

7.3 Clock system 27
7.3.1 General description 27
7.3.2 External clock 28

8 Interrupts 30
8.0.1 Interrupt register (ITRFRE) 32

9 Power saving modes 33
9.1 Introduction 33
9.2 Halt mode 33
9.3 Slow mode 34
9.4 Wait mode 34

10 I/O ports 36
10.1 Introduction 36
10.2 Functional description 36
10.2.1 Port A 37
10.2.2 Port B 39
10.2.3 Port C 40
10.2.4 Register description 41
10.2.5 Data register (PxDR) 41
10.2.6 Data direction register (PxDDR) 42
10.2.7 Related documentation 42

11 Miscellaneous register 43

12 Watchdog timer (WDG) 44
12.1 Introduction 44
12.2 Main features 44
12.3 Functional description 44
12.3.1 Software watchdog option 45
12.3.2 Hardware watchdog option 45
12.3.3 Low power modes 46
12.3.4 Using Halt mode with the WDG (option) 46
12.3.5 Interrupts 46
12.3.6 Control register (WDGCR) 47

12.4 16-bit timer 48
12.4.1 Introduction 48
12.4.2 Main features 48
12.4.3 Functional description 49
12.4.4 Low power modes 62
12.4.5 Interrupts 62
12.4.6 Summary of timer modes 63
12.4.7 16-bit timer registers 63

13 Serial communications interface (SCI) 71
13.1 Introduction 71
13.2 Main features 71
13.2.1 General description 72
13.2.2 Functional description 73
13.2.3 Low power modes 81
13.2.4 Interrupts 81
13.3 Register description 82
13.3.1 Status register (SCISR) 82
13.3.2 Control register 1 (SCICR1) 84
13.3.3 Control register 2 (SCICR2) 85
13.3.4 Data register (SCIDR) 86
13.3.5 Baud rate register (SCIBRR) 87

14 USB interface (USB) 89
14.1 Introduction 89
14.2 Main features 89
14.3 Functional description 89
14.4 Register description 90
14.4.1 DMA address register (DMAR) 90
14.4.2 Interrupt/DMA register (IDR) 90
14.4.3 PID register (PIDR) 91
14.4.4 Interrupt status register (ISTR) 92
14.4.5 Interrupt mask register (IMR) 93
14.4.6 Control register (CTLR) 94
14.4.7 Device address register (DADDR) 95
14.4.8 Endpoint n register A (EPnRA) 95

1 Description

The ST7260xx devices are members of the ST7 microcontroller family designed for USB applications running from 4.0 to 5.5 V. Different package options offer up to 19 I/O pins.

All devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set and are available with Flash program memory. The ST7 family architecture offers both power and flexibility to software developers, enabling the design of highly efficient and compact application code.

The on-chip peripherals include a low speed USB interface and an asynchronous SCI interface. For power economy, the microcontroller can switch dynamically into, Slow, Wait, Active Halt or Halt mode when the application is in idle or stand-by state.

Typical applications include consumer, home, office and industrial products.

$\overline{\text{RESET}}$ (see Note 1): Bidirectional. This active low signal forces the initialization of the MCU. This event is the top priority non maskable interrupt. This pin is switched low when the Watchdog is triggered or the V_{DD} is low. It can be used to reset external peripherals.

OSCIN/OSCOUT: Input/Output Oscillator pin. These pins connect a parallel-resonant crystal, or an external source, to the on-chip oscillator.

V_{DD}/V_{SS} (see Note 2): Main power supply and ground voltages.

V_{DDA}/V_{SSA} (see Note 2): Power supply and ground voltages for analog peripherals.

Alternate functions: Several pins of the I/O ports assume software programmable alternate functions as shown in the pin description.

Note: 1 Note 1: Adding two 100 nF decoupling capacitors on the Reset pin (respectively connected to VDD and VSS) will significantly improve product electromagnetic susceptibility performance.

2 To enhance the reliability of operation, it is recommended that V_{DDA} and V_{DD} be connected together on the application board. This also applies to V_{SSA} and V_{SS}.

3 The USBOE alternate function is mapped on Port C2 in QFN40 devices. In SO24 devices it is mapped on Port B1.

4 The timer OCMP1 alternate function is mapped on Port A6 in QFN40 pin devices. In SO24 devices it is not available.

Legend / abbreviations for *Figure 2*, *Figure 3* and *Table 2*, *Table 3*:

Type: I = input, O = output, S = supply

In/Output level: CT = CMOS 0.3 V_{DD} / 0.7 V_{DD} with input trigger

Output level: 10 mA = 10 mA high sink (Fn N-buffer only)

25 mA = 25 mA very high sink (on N-buffer only)

Port and control configuration:

- Input: float = floating, wpu = weak pull-up, int = interrupt
- Output: OD = open drain, PP = push-pull, T = True open drain

The RESET configuration of each pin is shown in bold. This configuration is kept as long as the device is under reset state.

5.5 ICP (in-circuit programming)

To perform ICP the microcontroller must be switched to ICC (in-circuit communication) mode by an external controller or programming tool.

Depending on the ICP code downloaded in RAM, Flash memory programming can be fully customized (number of bytes to program, program locations, or selection serial communication interface for downloading).

When using an STMicroelectronics or third-party programming tool that supports ICP and the specific microcontroller device, the user needs only to implement the ICP hardware interface on the application board (see *Figure 6*). For more details on the pin locations, refer to the device pinout description.

5.6 IAP (in-application programming)

This mode uses a BootLoader program previously stored in Sector 0 by the user (in ICP mode or by plugging the device in a programming tool).

This mode is fully controlled by user software. This allows it to be adapted to the user application, (such as user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored). For example, it is possible to download code from the SCI, or USB interface and program it in the Flash. IAP mode can be used to program any of the Flash sectors except Sector 0, which is write/erase protected to allow recovery in case errors occur during the programming operation.

5.7 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*.

5.7.1 Flash control/status register (FCSR)

This register is reserved for use by programming tool software. It controls the Flash programming and erasing operations.

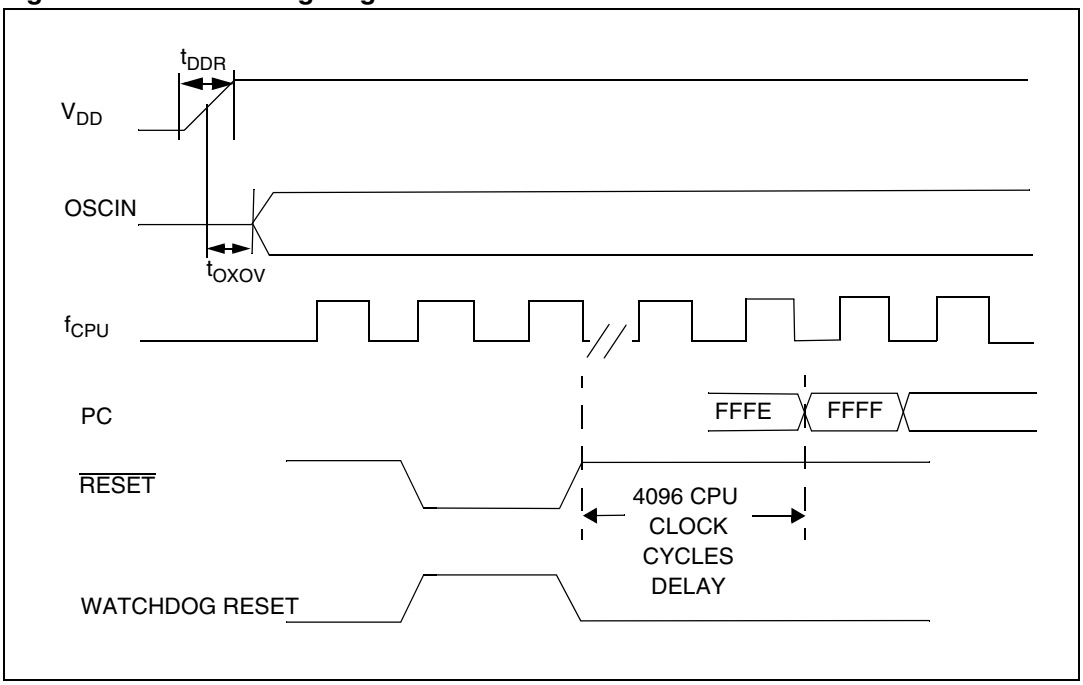
FCSR Reset value:0000 0000 (00h)

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7. Flash control/status register address and reset value

Address (Hex)	Register label	7	6	5	4	3	2	1	0
0037h	FCSR reset value	0	0	0	0	0	0	0	0

Figure 12. Reset timing diagram



Note: Refer to Electrical Characteristics for values of t_{DDR}, t_{oxov}, V_{IT+}, V_{IT-} and V_{hys}

7.3 Clock system

7.3.1 General description

The MCU accepts either a crystal or ceramic resonator, or an external clock signal to drive the internal oscillator. The internal clock (f_{CPU}) is derived from the external oscillator frequency (f_{OSC}), which is divided by 3 (and by 2 or 4 for USB, depending on the external clock used). The internal clock is further divided by 2 by setting the SMS bit in the Miscellaneous Register.
Using the OSC24/12 bit in the option byte, a 12 MHz or a 24 MHz external clock can be used to provide an internal frequency of either 2, 4 or 8 MHz while maintaining a 6 MHz for the USB (refer to *Figure 15*).

The internal clock signal (f_{CPU}) is also routed to the on-chip peripherals. The CPU clock signal consists of a square wave with a duty cycle of 50%.

The internal oscillator is designed to operate with an AT-cut parallel resonant quartz or ceramic resonator in the frequency range specified for f_{osc}. The circuit shown in *Figure 14* is recommended when using a crystal, and *Table 9* lists the recommended capacitance. The crystal and associated components should be mounted as close as possible to the input pins in order to minimize output distortion and start-up stabilisation time.

In this mode, writing “0” or “1” to the DR register applies this digital value to the I/O pin through the latch. Therefore, the previously saved value is restored when the DR register is read.

Note: The interrupt function is disabled in this mode.

Alternate function

When an on-chip peripheral is configured to use a pin, the alternate function is automatically selected. This alternate function takes priority over standard I/O programming. When the signal is coming from an on-chip peripheral, the I/O pin is automatically configured in output mode (push-pull or open drain according to the peripheral).

When the signal is going to an on-chip peripheral, the I/O pin has to be configured in input mode. In this case, the pin’s state is also digitally readable by addressing the DR register.

Note: 1 Input pull-up configuration can cause an unexpected value at the input of the alternate peripheral input.

2 When the on-chip peripheral uses a pin as input and output, this pin must be configured as an input (DDR = 0).

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

10.2.1 Port A

Table 14. Port A0, A3, A4, A5, A6, A7 description

Port A	I/O		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

12.3.3 Low power modes

WAIT Instruction

No effect on Watchdog.

HALT Instruction

If the Watchdog reset on HALT option is selected by option byte, a HALT instruction causes an immediate reset generation if the Watchdog is activated (WDGA bit is set).

12.3.4 Using Halt mode with the WDG (option)

If the Watchdog reset on HALT option is not selected by option byte, the Halt mode can be used when the watchdog is enabled.

In this case, the HALT instruction stops the oscillator. When the oscillator is stopped, the WDG stops counting and is no longer able to generate a reset until the microcontroller receives an external interrupt or a reset.

If an external interrupt is received, the WDG restarts counting after 4096 CPU clocks. If a reset is generated, the WDG is disabled (reset state).

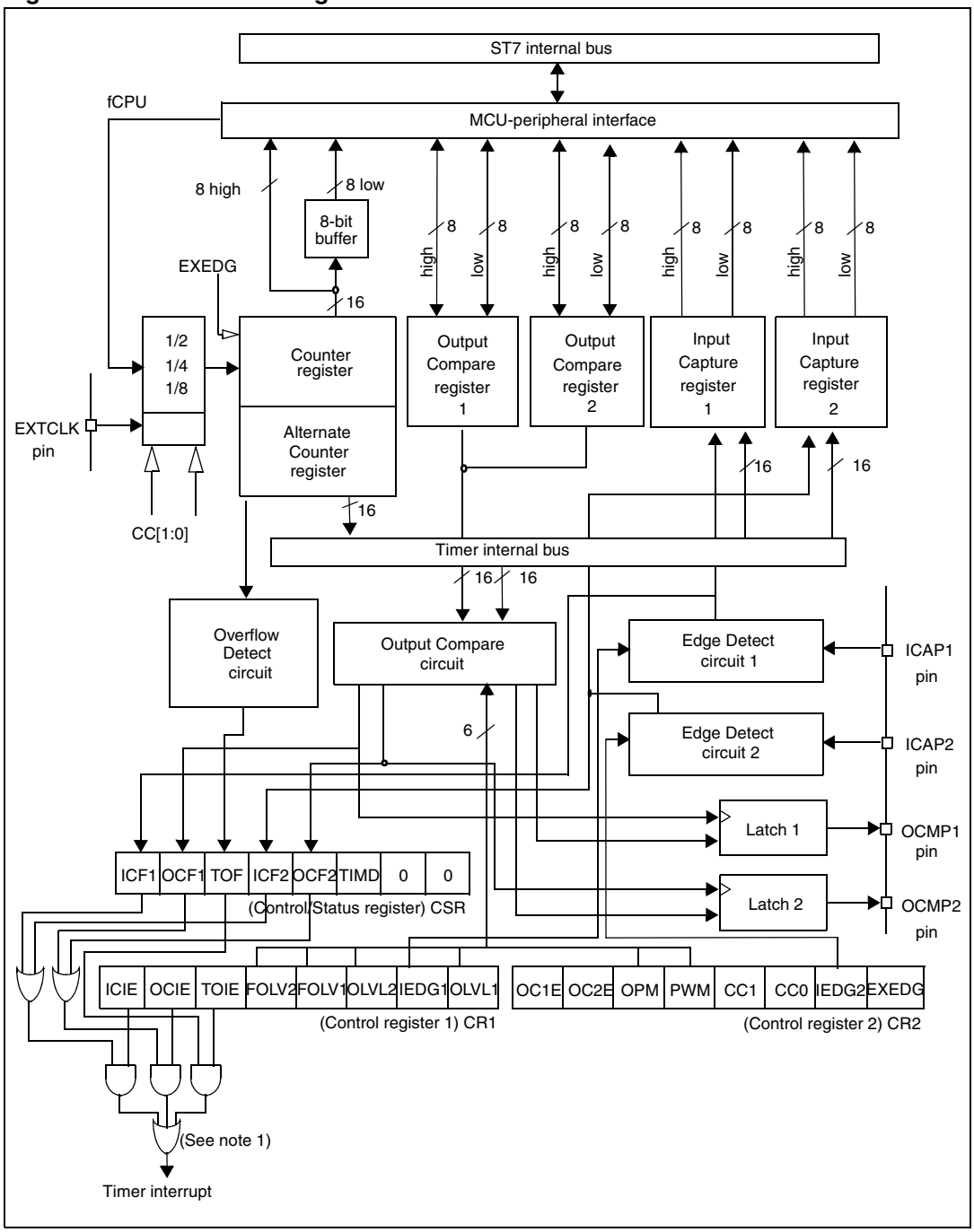
Recommendations

- Make sure that an external event is available to wake up the microcontroller from Halt mode.
- Before executing the HALT instruction, refresh the WDG counter, to avoid an unexpected WDG reset immediately after waking up the microcontroller.
- When using an external interrupt to wake up the microcontroller, reinitialize the corresponding I/O as “Input Pull-up with Interrupt” before executing the HALT instruction. The main reason for this is that the I/O may be wrongly configured due to external interference or by an unforeseen logical condition.
- For the same reason, reinitialize the level sensitiveness of each external interrupt as a precautionary measure.
- The opcode for the HALT instruction is 0x8E. To avoid an unexpected HALT instruction due to a program counter failure, it is advised to clear all occurrences of the data value 0x8E from memory. For example, avoid defining a constant with the value 0x8E.
- As the HALT instruction clears the I bit in the CC register to allow interrupts, the user may choose to clear all pending interrupt bits before executing the HALT instruction. This avoids entering other peripheral interrupt routines after executing the external interrupt routine corresponding to the wake-up event (reset or external interrupt).

12.3.5 Interrupts

None.

Figure 24. Timer block diagram



1. If IC, OC and TO interrupt requests have separate vectors then the last OR is not present (see *Table 10: Interrupt mapping on page 32*).

12.4.6 Summary of timer modes

Table 30. Summary of timer modes

Mode	Timer resources			
	Input capture 1	Input capture 2	Output compare 1	Output compare 2
Input Capture (1 and/or 2)	Yes	Yes	Yes	Yes
Output Compare (1 and/or 2)				
One Pulse mode	No	Not recommended(1)	No	Partially(2)
PWM mode		Not recommended(3)		No

1. See note 4 in *One pulse mode on page 58*.
2. See note 5 in *One pulse mode on page 58*.
3. See note 4 in *Pulse width modulation mode on page 60*.

12.4.7 16-bit timer registers

Each timer is associated with three control and status registers, and with six pairs of data registers (16-bit values) relating to the two input captures, the two output compares, the counter and the alternate counter.

Control Register 1 (CR1)

CR1 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31. CR1 register description

Bit	Name	Function
7	ICIE	Input Capture Interrupt Enable 0: Interrupt is inhibited. 1: A timer interrupt is generated whenever the ICF1 or ICF2 bit of the SR register is set.
6	OCIE	Output Compare Interrupt Enable 0: Interrupt is inhibited. 1: A timer interrupt is generated whenever the OCF1 or OCF2 bit of the SR register is set.
5	TOIE	Timer Overflow Interrupt Enable 0: Interrupt is inhibited. 1: A timer interrupt is enabled whenever the TOF bit of the SR register is set.

Table 34. 16-bit timer register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
11	CR1 Reset value	ICIE 0	OCIE 0	TOIE 0	FOLV2 0	FOLV1 0	OLVL2 0	IEDG1 0	OLVL1 0
12	CR2 Reset value	OC1E 0	OC2E 0	OPM 0	PWM 0	CC1 0	CC0 0	IEDG2 0	EXEDG 0
13	CSR Reset value	ICF1 x	OCF1 x	TOF x	ICF2 x	OCF2 x	TIMD 0	- x	- x
14	IC1HR Reset value	MSB x	x	x	x	x	x	x	LSB x
15	IC1LR Reset value	MSB x	x	x	x	x	x	x	LSB x
16	OC1HR Reset value	MSB 1	0	0	0	0	0	0	LSB 0
17	OC1LR Reset value	MSB 0	0	0	0	0	0	0	LSB 0
18	CHR Reset value	MSB 1	1	1	1	1	1	1	LSB 1
19	CLR Reset value	MSB 1	1	1	1	1	1	0	LSB 0
1A	ACHR Reset value	MSB 1	1	1	1	1	1	1	LSB 1
1B	ACLR Reset value	MSB 1	1	1	1	1	1	0	LSB 0
1C	IC2HR Reset value	MSB x	x	x	x	x	x	x	LSB x
1D	IC2LR Reset value	MSB x	x	x	x	x	x	x	LSB x
1E	OC2HR Reset value	MSB 1	0	0	0	0	0	0	LSB 0
1F	OC2LR Reset value	MSB 0	0	0	0	0	0	0	LSB 0

14 USB interface (USB)

14.1 Introduction

The USB Interface implements a low-speed function interface between the USB and the ST7 microcontroller. It is a highly integrated circuit which includes the transceiver, 3.3 voltage regulator, SIE and DMA. No external components are needed apart from the external pull-up on USBDM for low speed recognition by the USB host. The use of DMA architecture allows the endpoint definition to be completely flexible. Endpoints can be configured by software as in or out.

14.2 Main features

- USB specification version 1.1 compliant
- Supports Low-Speed USB protocol
- Two or three Endpoints (including default one) depending on the device (see device feature list and register map)
- CRC generation/checking, NRZI encoding/decoding and bit-stuffing
- USB Suspend/Resume operations
- DMA data transfers
- On-chip 3.3V regulator
- On-chip USB transceiver

14.3 Functional description

The block diagram in *Figure 41*, gives an overview of the USB interface hardware.

For general information on the USB, refer to the “Universal Serial Bus Specifications” document available at http//:www.usb.org.

Serial interface engine

The SIE (Serial Interface Engine) interfaces with the USB, via the transceiver.

The SIE processes tokens, handles data transmission/reception, and handshaking as required by the USB standard. It also performs frame formatting, including CRC generation and checking.

Endpoints

The Endpoint registers indicate if the microcontroller is ready to transmit/receive, and how many bytes need to be transmitted.

DMA

When a token for a valid Endpoint is recognized by the USB interface, the related data transfer takes place, using DMA. At the end of the transaction, an interrupt is generated.

Interrupts

By reading the Interrupt Status register, application software can know which USB event has occurred.

14.4.7 Device address register (DADDR)

DADDR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
0	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7 = Reserved. Forced by hardware to 0.

Bits 6:0 = **ADD[6:0]** *Device address, 7 bits.*

Software must write into this register the address sent by the host during enumeration.

Note: This register is also reset when a USB reset is received from the USB bus or forced through bit FRES in the CTLR register.

14.4.8 Endpoint n register A (EPnRA)

EPnRA Reset value: 0000 xxxx (0xh)

7	6	5	4	3	2	1	0
ST_ OUT	DTOG _TX	STAT _TX1	STAT _TX0	TBC3	TBC2	TBC1	TBC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

These registers (**EP0RA**, **EP1RA** and **EP2RA**) are used for controlling data transmission. They are also reset by the USB bus reset.

Note: Endpoint 2 and the EP2RA register are not available on some devices (see device feature list and register map).

Bit 7 = **ST_OUT** *Status out.*
This bit is set by software to indicate that a status out packet is expected: in this case, all nonzero OUT data transfers on the endpoint are STALLed instead of being ACKed. When ST_OUT is reset, OUT transactions can have any number of bytes, as needed.

Bit 6 = **DTOG_TX** *Data Toggle, for transmission transfers.*
It contains the required value of the toggle bit (0=DATA0, 1=DATA1) for the next transmitted data packet. This bit is set by hardware at the reception of a SETUP PID. DTOG_TX toggles only when the transmitter has received the ACK signal from the USB host. DTOG_TX and also DTOG_RX (see EPnRB) are normally updated by hardware, at the receipt of a relevant PID. They can be also written by software.

Bits 5:4 = **STAT_TX[1:0]** *Status bits, for transmission transfers.*
These bits contain the information about the endpoint status, which are listed below:

End suspend (ESUSP)

The CPU is alerted by activity on the USB, which causes an ESUSP interrupt. The ST7 automatically terminates HALT mode.

Correct transfer (CTR)

1. When this event occurs, the hardware automatically sets the STAT_TX or STAT_RX to NAK.

Note: Every valid endpoint is NAKed until software clears the CTR bit in the ISTR register, independently of the endpoint number addressed by the transfer which generated the CTR interrupt.

Note: If the event triggering the CTR interrupt is a SETUP transaction, both STAT_TX and STAT_RX are set to NAK.

2. Read the PIDR to obtain the token and the IDR to get the endpoint number related to the last transfer.

Note: When a CTR interrupt occurs, the TP3-TP2 bits in the PIDR register and EP1-EP0 bits in the IDR register stay unchanged until the CTR bit in the ISTR register is cleared.

3. Clear the CTR bit in the ISTR register.

Table 46. USB register map and reset values

Address (Hex.)	Register name	7	6	5	4	3	2	1	0
25	PIDR Reset Value	TP3 x	TP2 x	0 0	0 0	0 0	RX_SEZ 0	RXD 0	0 0
26	DMAR Reset Value	DA15 x	DA14 x	DA13 x	DA12 x	DA11 x	DA10 x	DA9 x	DA8 x
27	IDR Reset Value	DA7 x	DA6 x	EP1 x	EP0 x	CNT3 0	CNT2 0	CNT1 0	CNT0 0
28	ISTR Reset Value	SUSP 0	DOVR 0	CTR 0	ERR 0	IOVR 0	ESUSP 0	RESET 0	SOF 0
29	IMR Reset Value	SUSPM 0	DOVRM 0	CTRM 0	ERRM 0	IOVRM 0	ESUSPM 0	RESETM 0	SOFM 0
2A	CTLR Reset Value	0 0	0 0	0 0	0 0	RESUME 0	PDWN 1	FSUSP 1	FRES 0
2B	DADDR Reset Value	0 0	ADD6 0	ADD5 0	ADD4 0	ADD3 0	ADD2 0	ADD1 0	ADD0 0
2C	EP0RA Reset Value	ST_OUT 0	DTOG_TX 0	STAT_TX1 0	STAT_TX0 0	TBC3 x	TBC2 x	TBC1 x	TBC0 x
2D	EP0RB Reset Value	1 1	DTOG_RX 0	STAT_RX 1 0	STAT_RX 0 0	0 0	0 0	0 0	0 0

Indexed (long)

The offset is a word, thus allowing 64 Kbyte addressing space and requires 2 bytes after the opcode.

15.1.5 Indirect (short, long)

The required data byte to do the operation is found by its memory address, located in memory (pointer).

The pointer address follows the opcode. The indirect addressing mode consists of two sub-modes:

Indirect (short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - FF addressing space, and requires 1 byte after the opcode.

Indirect (long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

15.1.6 Indirect indexed (short, long)

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

The indirect indexed addressing mode consists of two sub-modes:

Indirect indexed (short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - 1FE addressing space, and requires 1 byte after the opcode.

Indirect indexed (long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

Table 51. Instructions supporting direct, indexed, indirect and indirect indexed addressing modes

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

15.1.7 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. Available relative direct/indirect instructions

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.

15.2 Instruction groups

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may be subdivided into 13 main groups as illustrated in the following table:

Table 62. Control timings (continued)

Symbol	Parameter	Conditions	Value			Unit
			Min	Typ	Max	
t_{DOG}	Watchdog time-out	f_{CPU} = 8 MHz	49152 6.144		3145728 393.216	t_{CPU} ms
t_{OXOV}	Crystal oscillator start-up time		20(1)	30	40(1)	ms
t_{DDR}	Power up rise time	from V_{DD} = 0 to 4 V			100(1)	ms

1. Not tested in production, guaranteed by characterization.

16.8.2 Output driving current

Subject to general operating condition for V_{DD}, f_{CPU}, and T_A unless otherwise specified.

Table 71. Output current characteristics

Symbol	Parameter	Conditions		Min	Max	Unit
V_{OL} (1)	Output low level voltage for a standard I/O pin when up to 8 pins are sunk at the same time, Port A0, Port A(3:7), Port C(0:2)	V_{DD}=5 V	I_{IO}=+1.6 mA		0.4	V
	Output low level voltage for a high sink I/O pin when up to 4 pins are sunk at the same time, Port B(0:7)		I_{IO}=+10 mA		1.3	
	Output low level voltage for a very high sink I/O pin when up to 2 pins are sunk at the same time, Port A1, Port A2		I_{IO}=+25 mA		1.5	
V_{OH} (2)	Output high level voltage for an I/O pin when up to 8 pins are sourced at same time		I_{IO}=-10 mA	V_{DD}-1.3		
			I_{IO}=-1.6 mA	V_{DD}-0.8		

1. The I_{IO} current sunk must always respect the absolute maximum rating specified in *Section 16.2* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VSS}.
2. The I_{IO} current sourced must always respect the absolute maximum rating specified in *Section 16.2* and the sum of I_{IO} (I/O ports and control pins) must not exceed I_{VDD}. True open drain I/O pins does not have V_{OH}.

Figure 54. V_{OL} standard V_{DD}=5 V

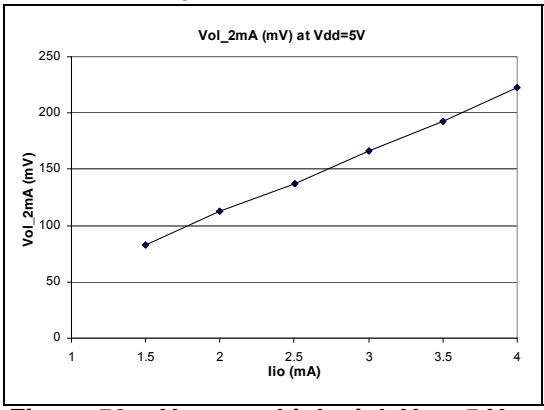


Figure 55. V_{OL} high sink V_{DD}=5 V

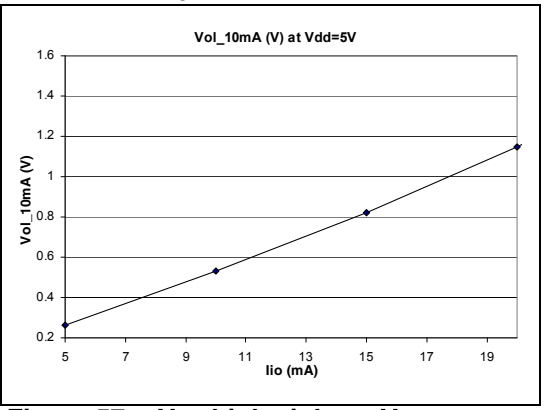


Figure 56. V_{OL} very high sink V_{DD}=5 V

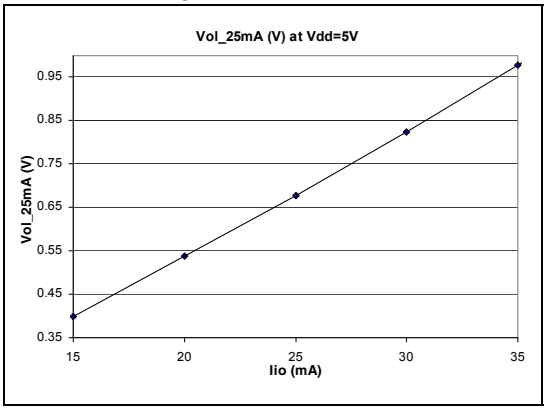
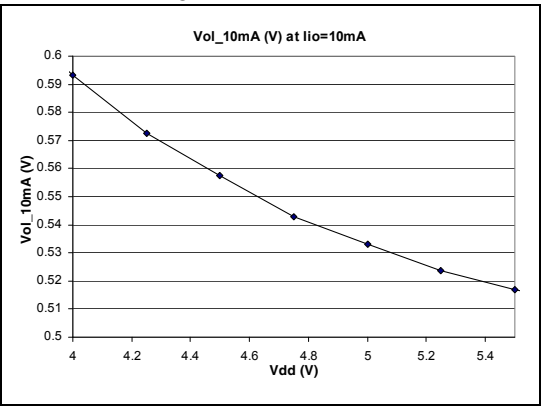


Figure 57. V_{OL} high sink vs. V_{DD}



19 Known limitations

19.1 PA2 limitation with OCMP1 enabled

Description

This limitation affects only Rev B Flash devices (with Internal Sales Type 72F60xxxxx$x7); it has been corrected in Rev W Flash devices (with Internal Sales Type 72F60xxxxx$x9).

Refer to *Figure 69 on page 137*.

When Output Compare 1 function (OCMP1) on pin PA6 is enabled by setting the OC1E bit in the TCR2 register, pin PA2 is also affected.

In particular, the PA2 pin is forced to be floating even if port configuration (PADDR+PADR) has set it as output low. However, it can be still used as an input.

19.2 Unexpected reset fetch

If an interrupt request occurs while a "POP CC" instruction is executed, the interrupt controller does not recognise the source of the interrupt and, by default, passes the RESET vector address to the CPU.

Workaround

To solve this issue, a "POP CC" instruction must always be preceded by a "SIM" instruction.

19.3 SCI wrong break duration

Description

A single break character is sent by setting and resetting the SBK bit in the SCICR2 register. In some cases, the break character may have a longer duration than expected:

- 20 bits instead of 10 bits if M=0
- 22 bits instead of 11 bits if M=1

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 baudrate. With a transmit frequency of 19200 baud (f_{CPU}=8 MHz 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.

21 Revision history

Table 81. Document revision history

Date	Revision	Changes
13-Feb-2006	1	Initial release.
18-Oct-2006	2	Added known limitations section
05-Feb-2009	3	Added caution in *Section 7.1 on page 25* Added reference to watchdog reset pulse t_{DOG} in *Section 12.3 on page 44* Removed EMC protective circuitry in *Figure 65 on page 127* (device works correctly without these components) Modified notes below *Table 76: Package thermal characteristics on page 130* Replaced soldering information with ECOPACK reference in *Section 17 on page 129*