Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

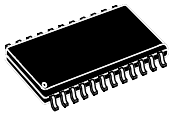
Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

ST7260xx

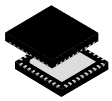
Low speed USB 8-bit MCU family with up to 8K Flash and serial communications interface

Features

- **Memories**
  - 4 or 8 Kbytes program memory: high density Flash (HDFlash), or FastROM with readout and write protection
  - In-application programming (IAP) and in-circuit programming (ICP)
  - 384 bytes RAM memory (128-byte stack)
- **Clock, reset and supply management**
  - Run, Wait, Slow and Halt CPU modes
  - 12 or 24 MHz oscillator
  - RAM Retention mode
  - Optional low voltage detector (LVD)
- **USB (Universal Serial Bus) interface**
  - DMA for low speed applications compliant with USB 1.5 Mbs (version 2.0) and HID specifications (version 1.0)
  - Integrated 3.3 V voltage regulator and transceivers
  - Supports USB DFU class specification
  - Suspend and Resume operations
  - 3 Endpoints with programmable In/Out configuration
- **Up to 19 I/O ports**
  - Up to 8 high sink I/Os (10 mA at 1.3 V)



SO24



QFN40 (6x6)

  - 2 very high sink true open drain I/Os (25 mA at 1.5 V)
  - Up to 8 lines with interrupt capability
- **2 timers**
  - Programmable Watchdog
  - 16-bit Timer with 2 Input Captures, 2 Output Compares, PWM output and clock input
- **Communications interface**
  - Asynchronous serial communications interface (SCI)
- **Instruction set**
  - 63 basic instructions
  - 17 main addressing modes
  - 8 x 8 unsigned multiply instruction
- **Development tools**
  - Versatile development tools including , software library, hardware emulator, programming boards, HID and DFU software layer

Table 1. Device summary

Features	ST7260K2	ST7260K1	ST7260E2	ST7260E1
Flash program memory - bytes	8 K	4 K	8 K	4 K
RAM (stack) - bytes	384 (128)			
Peripherals	Watchdog timer, 16-bit timer, USB, SCI			
Operating supply	4.0 V to 5.5 V			
CPU frequency	8 MHz (with 24 MHz oscillator) or 4 MHz (with 12 MHz oscillator)			
Operating temperature	0 °C to +70 °C			
Packages	QFN40 (6x6)		SO24	

Contents

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.

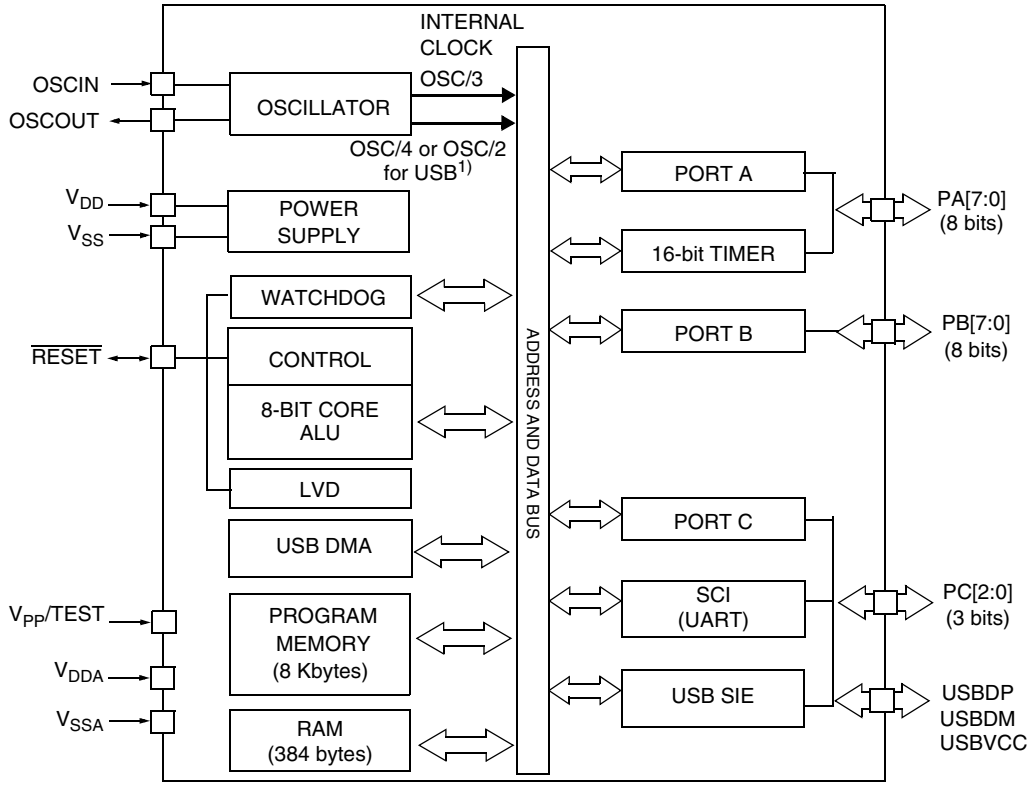
All devices are based on a common industry-standard 8-bit core, featuring an enhanced instruction set and are available with Flash program memory. The ST7 family architecture offers both power and flexibility to software developers, enabling the design of highly efficient and compact application code.

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.

2 Block diagram

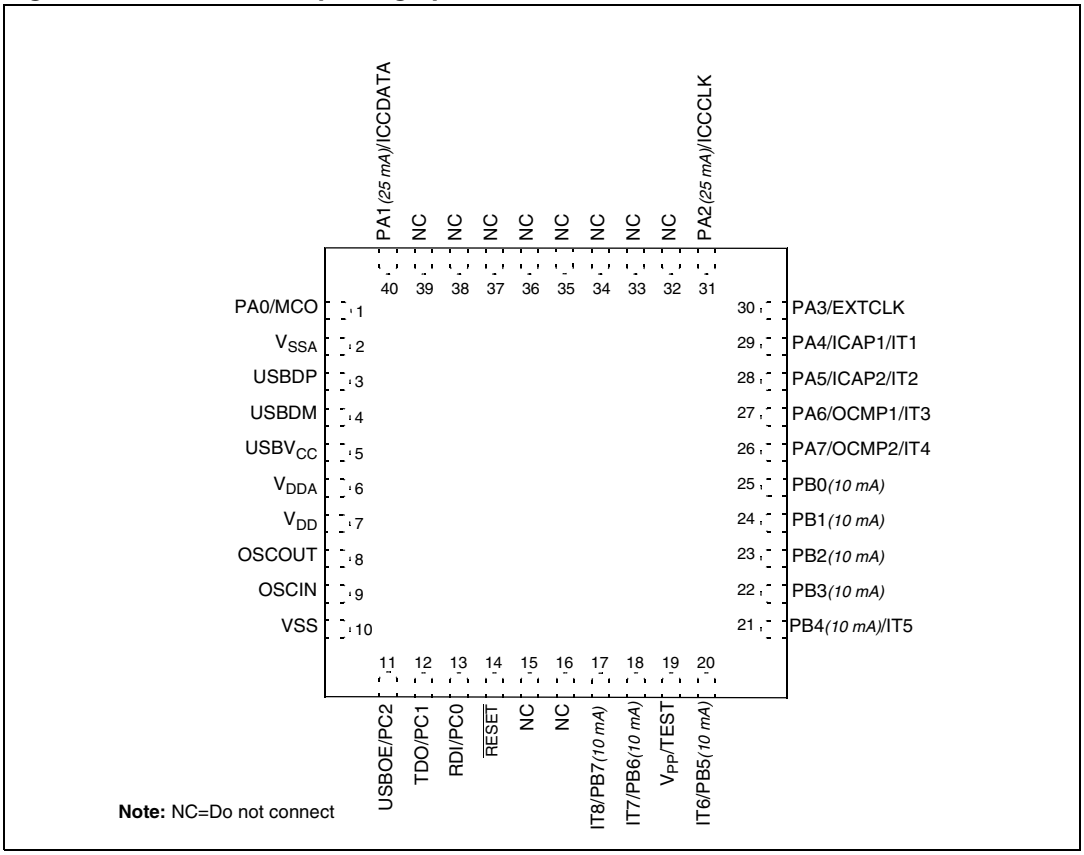
Figure 1. General block diagram



1) 12 or 24 MHz OSCIN frequency required to generate 6 MHz USB clock.

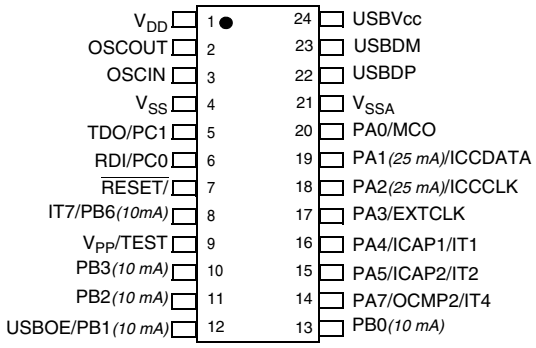
3 Pin description

Figure 2. 40-lead QFN package pinout



Note: NC=Do not connect

Figure 3. 24-pin SO package pinout



$\overline{\text{RESET}}$ (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.

Table 2. Device pin description (QFN40)

Pin n°	Pin name	Type	Level		Port / control					Main function (after reset)	Alternate function
			Input	Output	Input			Output			
					float	wpu	int	OD	PP		
1	PA0/MCO	I/O	CT				X		X	Port A0	Main Clock Output
2	V_{SSA}	S								Analog ground	
3	USBDP	I/O								USB bidirectional data (data +)	
4	USBDM	I/O								USB bidirectional data (data -)	
5	USBVCC	O								USB power supply	
6	V_{DDA}	S								Analog supply voltage	
7	V_{DD}	S								Power supply voltage (4V - 5.5V)	
8	OSCOUT	O								Oscillator output	
9	OSCIN	I								Oscillator input	
10	V_{SS}	S								Digital ground	
11	PC2/USBOE	I/O	CT			X			X	Port C2	USB Output Enable
12	PC1/TDO	I/O	CT			X			X	Port C1	SCI Transmit Data Output
13	PC0/RDI	I/O	CT			X			X	Port C0	SCI Receive Data Input
14	RESET	I/O				X		X		Reset	
15	NC	--								Not connected	
16	NC	--								Not connected	
17	PB7/IT8	I/O	CT	10 mA	X		X		X	Port B7	
18	PB6/IT7	I/O	CT	10 mA	X		X		X	Port B6	
19	V_{PP}/TEST	S								Programming supply	
20	PB5/IT6	I/O	CT	10 mA	X		X		X	Port B5	
21	PB4/IT5	I/O	CT	10 mA	X		X		X	Port B4	
22	PB3	I/O	CT	10 mA	X				X	Port B3	
23	PB2	I/O	CT	10 mA	X				X	Port B2	
24	PB1	I/O	CT	10 mA	X				X	Port B1	
25	PB0	I/O	CT	10 mA	X				X	Port B0	
26	PA7/OCMP2/IT4	I/O	CT			X	X		X	Port A7	Timer Output Compare 2
27	PA6/OCMP1/IT3	I/O	CT			X	X		X	Port A6	Timer Output Compare 1
28	PA5/ICAP2/IT2	I/O	CT			X	X		X	Port A5	Timer Input Capture 2
29	PA4/ICAP1/IT1	I/O	CT			X	X		X	Port A4	Timer Input Capture 1

Table 2. Device pin description (QFN40) (continued)

Pin n°	Pin name	Type	Level		Port / control					Main function (after reset)	Alternate function
			Input	Output	Input			Output			
					float	wpu	int	OD	PP		
30	PA3/EXTCLK	I/O	CT			X			X	Port A3	Timer External Clock
31	PA2/ICCCLK	I/O	C_T	25 mA	X			T		Port A2	ICC Clock
32	NC	--								Do not connect	
33	NC	--								Do not connect	
34	NC	--								Do not connect	
35	NC	--								Do not connect	
36	NC	--								Do not connect	
37	NC	--								Do not connect	
38	NC	--								Do not connect	
39	NC	--								Do not connect	
40	PA1/ICCDATA	I/O	CT	25 mA	X			T		Port A1	ICC Data

Table 3. Device pin description (SO24)

Pin n°	Pin name	Type	Level		Port / control					Main function (after reset)	Alternate function
			Input	Output	Input			Output			
					float	wpu	int	OD	PP		
1	V_{DD}	S								Power supply voltage (4 V - 5.5 V)	
2	OSCOUT	O								Oscillator output	
3	OSCIN	I								Oscillator input	
4	V_{SS}	S								Digital ground	
5	PC1/TDO	I/O	CT			X			X	Port C1	SCI Transmit Data Output
6	PC0/RDI	I/O	CT			X			X	Port C0	SCI Receive Data Input
7	RESET	I/O				X		X		Reset	
8	PB6/IT7	I/O	CT	10 mA	X		X		X	Port B6	
9	V_{PP}/TEST	S								Programming supply	
10	PB3	I/O	CT	10 mA	X				X	Port B3	
11	PB2	I/O	CT	10 mA	X				X	Port B2	
12	PB1/USBOE	I/O	CT	10 mA	X				X	Port B1	USB Output Enable
13	PB0	I/O	CT	10 mA	X				X	Port B0	
14	PA7/OCMP2/IT4	I/O	CT			X	X		X	Port A7	Timer Output Compare 2
15	PA5/ICAP2/IT2	I/O	CT			X	X		X	Port A5	Timer Input Capture 2
16	PA4/ICAP1/IT1	I/O	CT			X	X		X	Port A4	Timer Input Capture 1
17	PA3/EXTCLK	I/O	CT			X			X	Port A3	Timer External Clock
18	PA2/ICCCLK	I/O	C_T	25 mA	X			T		Port A2	ICC Clock
19	PA1/ICCDATA	I/O	CT	25 mA	X			T		Port A1	ICC Data
20	PA0/MCO	I/O	CT				X		X	Port A0	Main Clock Output
21	V_{SSA}	S								Analog ground	
22	USBDP	I/O								USB bidirectional data (data +)	
23	USBDM	I/O								USB bidirectional data (data -)	
24	USBVCC	O								USB power supply	

4 Register & memory map

As shown in *Figure 4*, the MCU is capable of addressing 8 Kbytes of memories and I/O registers.

The available memory locations consist of up to 384 bytes of RAM including 64 bytes of register locations, and up to 8 Kbytes of user program memory in which the upper 32 bytes are reserved for interrupt vectors. The RAM space includes up to 128 bytes for the stack from 0100h to 017Fh.

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

Note: ***Important****: memory locations noted "Reserved" must never be accessed. Accessing a reserved area can have unpredictable effects on the device.*

Figure 4. Memory map

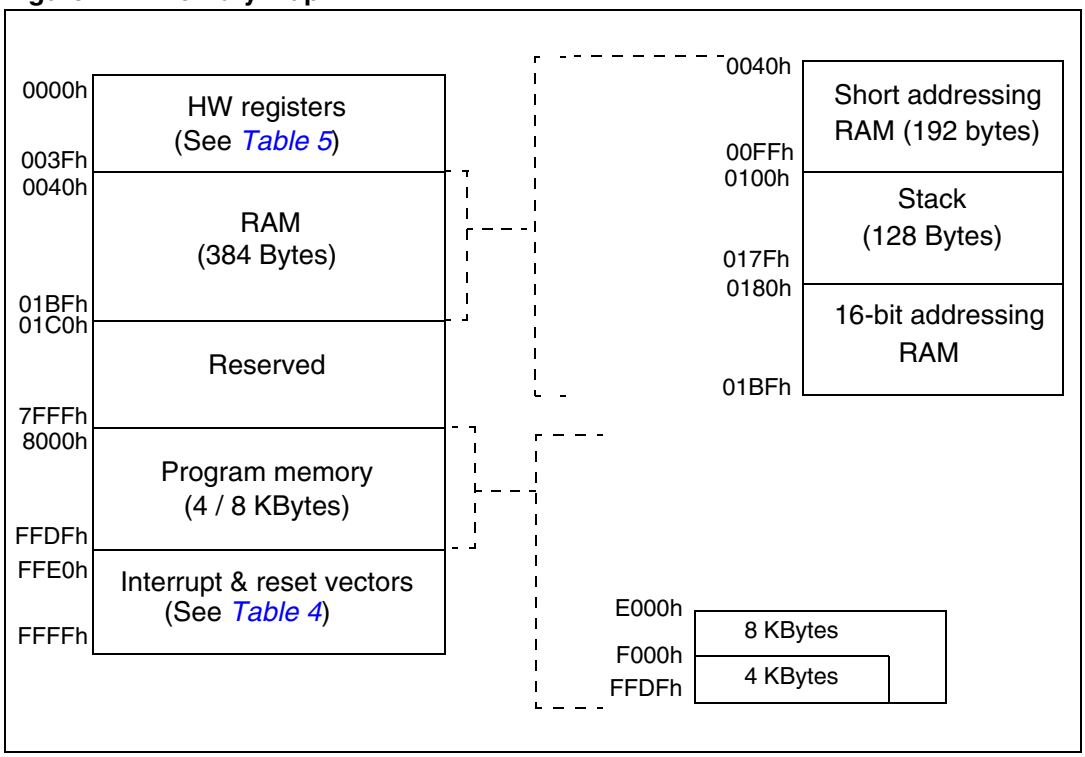


Table 4. Interrupt vector map

Vector address	Description	Masked by	Remarks	Exit from Halt mode
FFE0h-FFEDh	Reserved area			
FFEEh-FFEFh	USB interrupt vector	I- bit	Internal interrupt	No
FFF0h-FFF1h	SCI interrupt vector	I- bit	Internal interrupt	No
FFF2h-FFF3h	Reserved area			
FFF4h-FFF5h	TIMER interrupt vector	I- bit	Internal interrupt	No
FFF6h-FFF7h	IT1 to IT8 interrupt vector	I- bit	External interrupt	Yes
FFF8h-FFF9h	USB end suspend mode interrupt vector	I- bit	External interrupts	Yes
FFFAh-FFFBh	Flash start programming interrupt vector	I- bit	Internal interrupt	Yes
FFFCh-FFFDh	TRAP (software) interrupt vector	None	CPU interrupt	No
FFFEh-FFFFh	RESET vector	None		Yes

Table 5. Hardware register memory map

Address	Block	Register label	Register name	Reset status	Remarks
0000h 0001h	Port A	PADR PADDR	Port A Data Register Port A Data Direction Register	00h 00h	R/W R/W
0002h 0003h	Port B	PBDR PBDDR	Port B Data Register Port B Data Direction Register	00h 00h	R/W R/W
0004h 0005h	Port C	PCDR PCDDR	Port C Data Register Port C Data Direction Register	1111 x000b 1111 x000b	R/W R/W
0006h to 0007h	Reserved (2 bytes)				
0008h	ITC	ITIFRE	Interrupt Register	00h	R/W
0009h	MISC	MISCR	Miscellaneous Register	00h	R/W
000Ah to 000Bh	Reserved (2 bytes)				
000Ch	WDG	WDGCR	Watchdog Control Register	7Fh	R/W
000Dh to 0010h	Reserved (4 bytes)				

Table 5. Hardware register memory map (continued)

Address	Block	Register label	Register name	Reset status	Remarks
0011h	TIM	TCR2	Timer Control Register 2	00h	R/W
0012h		TCR1	Timer Control Register 1	00h	R/W
0013h		TCSR	Timer Control/Status Register	00h	R/W
0014h		TIC1HR	Timer Input Capture High Register 1	xxh	Read only
0015h		TIC1LR	Timer Input Capture Low Register 1	xxh	Read only
0016h		TOC1HR	Timer Output Compare High Register 1	80h	R/W
0017h		TOC1LR	Timer Output Compare Low Register 1	00h	R/W
0018h		TCHR	Timer Counter High Register	FFh	Read only
0019h		TCLR	Timer Counter Low Register	FCh	R/W
001Ah		TACHR	Timer Alternate Counter High Register	FFh	Read only
001Bh		TACLR	Timer Alternate Counter Low Register	FCh	R/W
001Ch		TIC2HR	Timer Input Capture High Register 2	xxh	Read only
001Dh		TIC2LR	Timer Input Capture Low Register 2	xxh	Read only
001Eh		TOC2HR	Timer Output Compare High Register 2	80h	R/W
001Fh		TOC2LR	Timer Output Compare Low Register 2	00h	R/W
0020h	SCI	SCISR	SCI Status Register	C0h	Read only
0021h		SCIDR	SCI Data Register	xxh	R/W
0022h		SCIBRR	SCI Baud Rate Register	00h	R/W
0023h		SCICR1	SCI Control Register 1	x000 0000b	R/W
0024h		SCICR2	SCI Control Register 2	00h	R/W
0025h	USB	USBPIDR	USB PID Register	x0h	Read only
0026h		USBDMAR	USB DMA address Register	xxh	R/W
0027h		USBIDR	USB Interrupt/DMA Register	x0h	R/W
0028h		USBISTR	USB Interrupt Status Register	00h	R/W
0029h		USBIMR	USB Interrupt Mask Register	00h	R/W
002Ah		USBCTLR	USB Control Register	06h	R/W
002Bh		USBDADDR	USB Device Address Register	00h	R/W
002Ch		USBEP0RA	USB Endpoint 0 Register A	0000 xxxxb	R/W
002Dh		USBEP0RB	USB Endpoint 0 Register B	80h	R/W
002Eh		USBEP1RA	USB Endpoint 1 Register A	0000 xxxxb	R/W
002Fh		USBEP1RB	USB Endpoint 1 Register B	0000 xxxxb	R/W
0030h		USBEP2RA	USB Endpoint 2 Register A	0000 xxxxb	R/W
0031h		USBEP2RB	USB Endpoint 2 Register B	0000 xxxxb	R/W
0032h to 0036h	Reserved (5 Bytes)				
0037h	Flash	FCSR	Flash Control /Status Register	00h	R/W
0038h to 003Fh	Reserved (8 bytes)				

5 Flash program memory

5.1 Introduction

The ST7 dual voltage High Density Flash (HDFlash) is a non-volatile memory that can be electrically erased as a single block or by individual sectors and programmed on a byte-by-byte basis using an external V_{PP} supply.

The HDFlash devices can be programmed and erased off-board (plugged in a programming tool) or on-board using ICP (in-circuit programming) or IAP (in-application programming).

The array matrix organization allows each sector to be erased and reprogrammed without affecting other sectors.

5.2 Main features

- 3 Flash programming modes:
  - Insertion in a programming tool. In this mode, all sectors including option bytes can be programmed or erased.
  - ICP (in-circuit programming). In this mode, all sectors including option bytes can be programmed or erased without removing the device from the application board.
  - IAP (in-application programming). In this mode, all sectors, except Sector 0, can be programmed or erased without removing the device from the application board and while the application is running.
- ICT (in-circuit testing) for downloading and executing user application test patterns in RAM
- Readout protection
- Register Access Security System (RASS) to prevent accidental programming or erasing

5.3 Structure

The Flash memory is organized in sectors and can be used for both code and data storage.

Depending on the overall Flash memory size in the microcontroller device, there are up to three user sectors (see *Table 6*). Each of these sectors can be erased independently to avoid unnecessary erasing of the whole Flash memory when only a partial erasing is required.

The first two sectors have a fixed size of 4 Kbytes (see *Figure 5*). They are mapped in the upper part of the ST7 addressing space so the reset and interrupt vectors are located in Sector 0 (F000h-FFFFh).

Table 6. Sectors available in Flash devices

Flash size (bytes)	Available sectors
4K	Sector 0
8K	Sectors 0, 1
>8K	Sectors 0, 1, 2

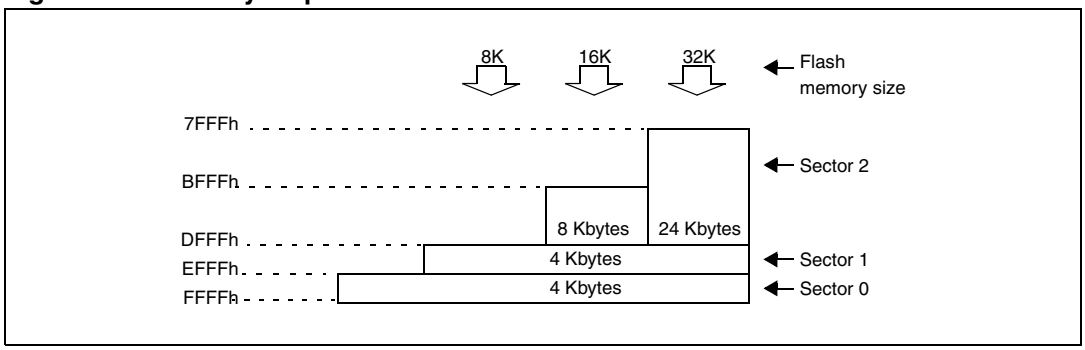
5.3.1 Readout protection

Readout protection, when selected, provides a protection against program memory content extraction and against write access to Flash memory. Even if no protection can be considered as totally unbreakable, the feature provides a very high level of protection for a general purpose microcontroller.

In Flash devices, this protection is removed by reprogramming the option. In this case, the entire program memory is first automatically erased.

Readout protection is enabled and removed through the FMP_R bit in the option byte.

Figure 5. Memory map and sector address

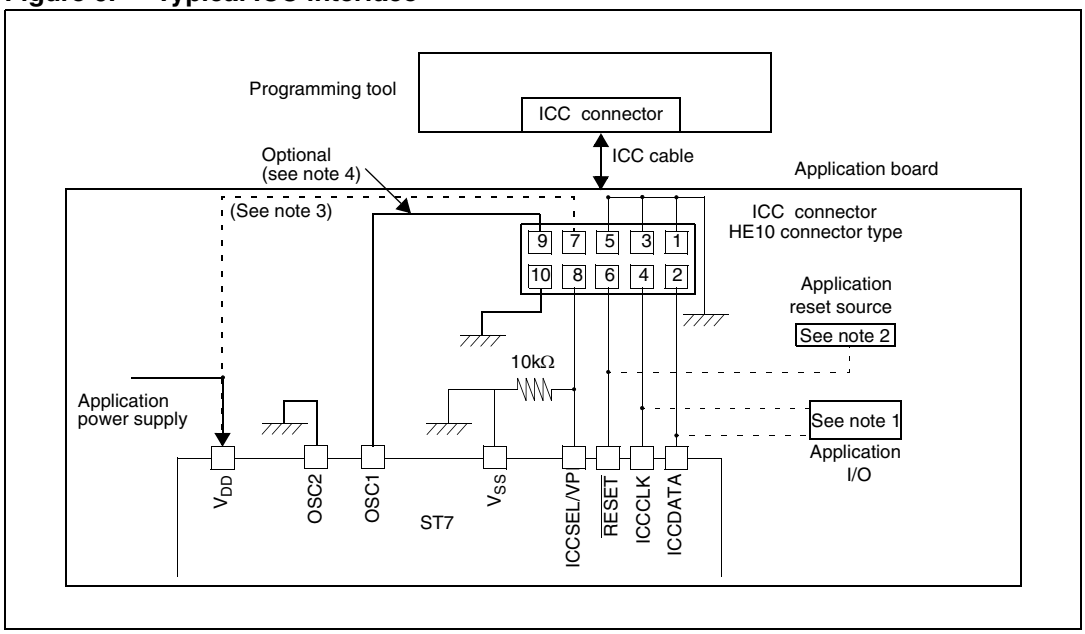


5.4 ICC interface

ICC needs a minimum of 4 and up to 6 pins to be connected to the programming tool (see *Figure 6*). These pins are:

- $\overline{\text{RESET}}$: device reset
- V_{SS}: device power supply ground
- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input/output serial data pin
- ICCSEL/V_{PP}: programming voltage
- OSC1 (or OSCIN): main clock input for external source (optional)
- V_{DD}: application board power supply (see *Figure 6*, Note 3).

Figure 6. Typical ICC interface



1. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the programming tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to be implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.
2. During the ICC session, the programming tool must control the $\overline{\text{RESET}}$ pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5mA at high level (PUSH-pull output or pull-up resistor <1K). A schottky diode can be used to isolate the application reset circuit in this case. When using a classical RC network with R>1K or a reset management IC with open drain output and pull-up resistor >1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.
3. The use of Pin 7 of the ICC connector depends on the programming tool architecture. This pin must be connected when using most ST programming tools (it is used to monitor the application power supply). Please refer to the programming tool manual.
4. Pin 9 has to be connected to the OSC1 (OSCIN) pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte. ST7 devices with multi-oscillator capability need to have OSC2 grounded in this case.

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

6 Central processing unit (CPU)

6.1 Introduction

This CPU has a full 8-bit architecture and contains six internal registers allowing efficient 8-bit data manipulation.

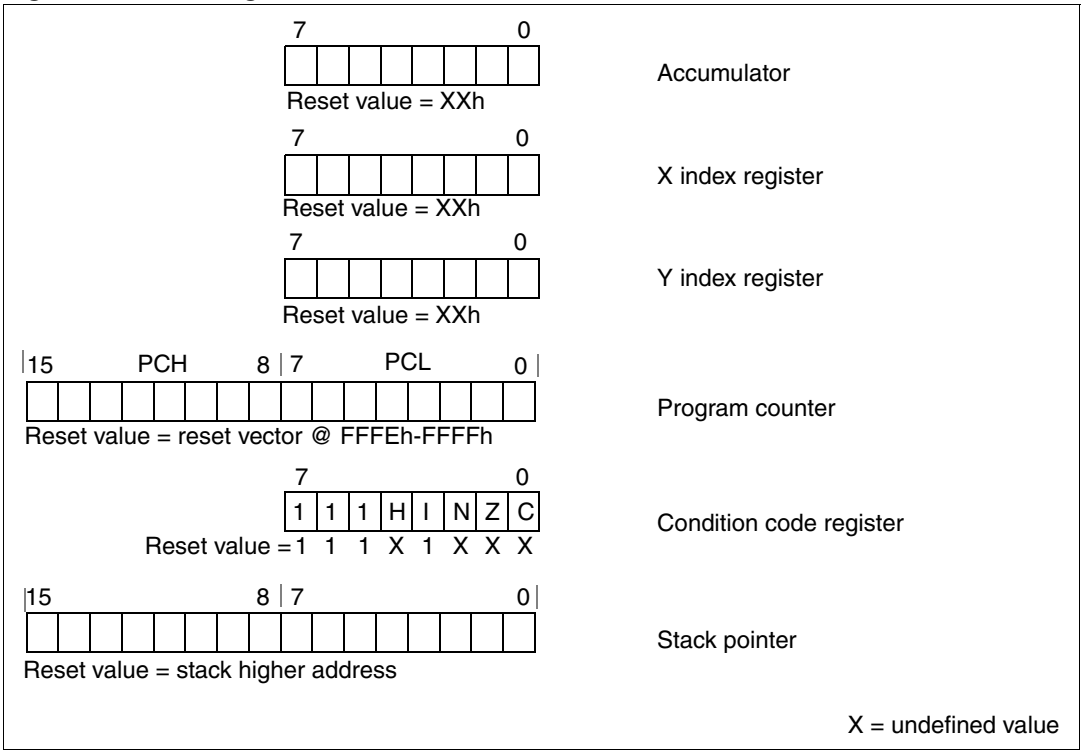
6.2 Main features

- 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes
- Two 8-bit index registers
- 16-bit stack pointer
- Low power modes
- Maskable hardware interrupts
- Non-maskable software interrupt

6.3 CPU registers

The six CPU registers shown in *Figure 7* are not present in the memory mapping and are accessed by specific instructions.

Figure 7. CPU registers



6.3.1 Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

6.3.2 Index registers (X and Y)

In indexed addressing modes, these 8-bit registers are used to create effective addresses or as temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures (not pushed to and popped from the stack).

6.3.3 Program counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

6.3.4 Condition code register (CC)

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.

CC Reset value: 111x1xxx

7	6	5	4	3	2	1	0
1	1	1	H	I	N	Z	C
			R/W	R/W	R/W	R/W	R/W

Table 8. CC register description

Bit	Name	Function
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 instructions. 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.

Table 8. CC register description

Bit	Name	Function
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
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 result 7th bit. 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.
1	Z	Zero (Arithmetic Management bit) 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.
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.

6.3.5 Stack pointer register (SP)

SP — Reset value: 01 7Fh

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
									R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 8*).

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 SP7 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 an LD instruction.

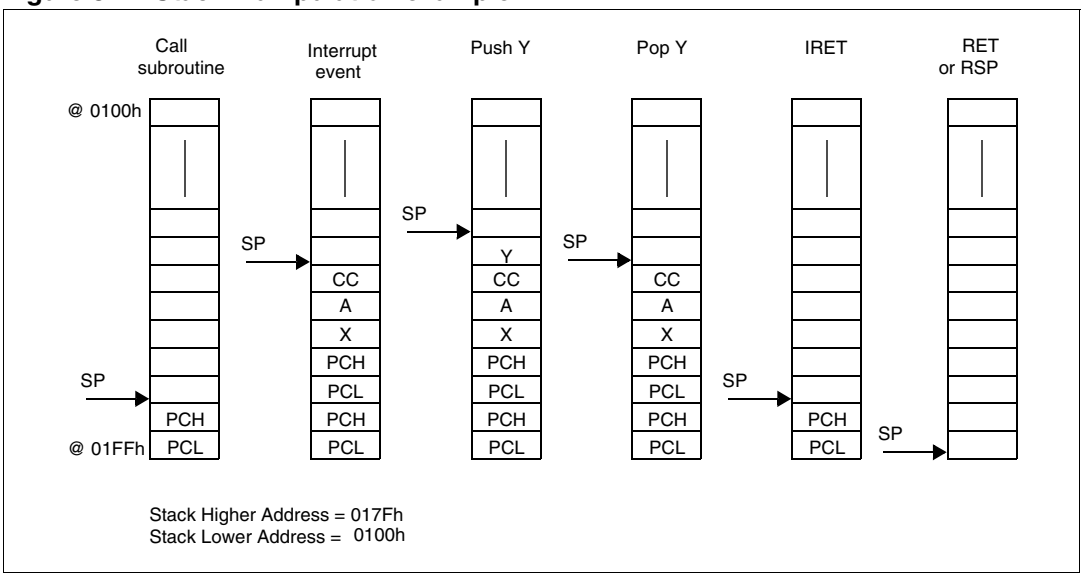
Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost. The stack also wraps in case of an underflow.

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 pointed to by the SP. Then the other registers are stored in the next locations as shown in *Figure 8*.

- When an interrupt is received, the SP is decremented and the context is pushed on the stack.
- On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.

Figure 8. Stack manipulation example



7 Reset and clock management

7.1 Reset

The Reset procedure is used to provide an orderly software start-up or to exit low power modes.

Three reset modes are provided: a low voltage (LVD) reset, a watchdog reset and an external reset at the $\overline{\text{RESET}}$ pin.

A reset causes the reset vector to be fetched from addresses FFFEh and FFFFh in order to be loaded into the PC and with program execution starting from this point.

An internal circuitry provides a 4096 CPU clock cycle delay from the time that the oscillator becomes active.

Caution: When the ST7 is unprogrammed or fully erased, the Flash is blank and the RESET vector is not programmed. For this reason, it is recommended to keep the $\overline{\text{RESET}}$ pin in low state until programming mode is entered, in order to avoid unwanted behavior

7.2 Low voltage detector (LVD)

Low voltage reset circuitry generates a reset when V_{DD} is:

- below V_{IT+} when V_{DD} is rising,
- below V_{IT-} when V_{DD} is falling.

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

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.

7.2.1 Watchdog reset

When a watchdog reset occurs, the $\overline{\text{RESET}}$ pin is pulled low permitting the MCU to reset other devices in the same way as the low voltage reset (*Figure 9*).

7.2.2 External reset

The external reset is an active low input signal applied to the $\overline{\text{RESET}}$ pin of the MCU. As shown in *Figure 12*, the $\overline{\text{RESET}}$ signal must stay low for a minimum of one and a half CPU clock cycles.

An internal Schmitt trigger at the $\overline{\text{RESET}}$ pin is provided to improve noise immunity.