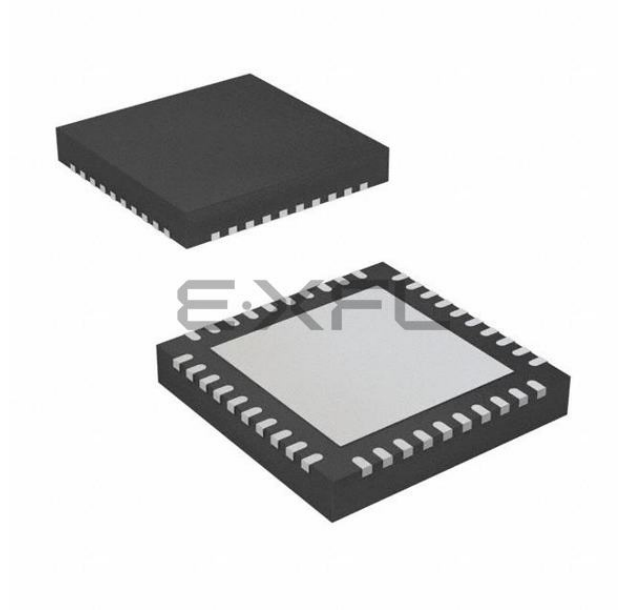


Welcome to E-XFL.COM

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	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 12x8b
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/st72f63bd6u1tr

List of figures

Figure 1. General block diagram . . . 12
Figure 2. 48-pin LQFP pinout . . . 14
Figure 3. 40-lead QFN package pinout . . . 15
Figure 4. 34-pin SO package pinout . . . 16
Figure 5. 32-pin SDIP package pinout . . . 16
Figure 6. 24-pin SO package pinout . . . 16
Figure 7. Memory map . . . 21
Figure 8. Memory map and sector address . . . 25
Figure 9. Typical ICC interface . . . 26
Figure 10. Stack manipulation example . . . 31
Figure 11. CPU registers . . . 31
Figure 12. Low voltage detector functional diagram . . . 33
Figure 13. Low Voltage Reset signal output . . . 33
Figure 14. Temporization timing diagram after an internal Reset . . . 33
Figure 15. Reset timing diagram . . . 34
Figure 16. External clock source connections . . . 35
Figure 17. Crystal/ceramic resonator . . . 36
Figure 18. Clock block diagram . . . 36
Figure 19. Interrupt processing flowchart . . . 38
Figure 20. Halt mode flowchart . . . 41
Figure 21. Wait mode flowchart . . . 42
Figure 22. PA0, PA3, PA4, PA5, PA6, PA7 and PD[7:4] configuration . . . 45
Figure 23. PA1, PA2 configuration . . . 46
Figure 24. Port B and D[3:0] configuration . . . 48
Figure 25. Port C configuration . . . 49
Figure 26. Watchdog block diagram . . . 55
Figure 27. Timer block diagram . . . 60
Figure 28. 16-bit read sequence (from either the Counter register or the Alternate Counter register) 61
Figure 29. Counter timing diagram, internal clock divided by 2 . . . 62
Figure 30. Counter timing diagram, internal clock divided by 4 . . . 62
Figure 31. Counter timing diagram, internal clock divided by 8 . . . 62
Figure 32. Input Capture block diagram . . . 64
Figure 33. Input Capture timing diagram . . . 64
Figure 34. Output Compare block diagram . . . 67
Figure 35. Output Compare timing diagram, ftimer = fcpu/2 . . . 67
Figure 36. Output Compare timing diagram, ftimeR = fCPU/4 . . . 67
Figure 37. One Pulse mode cycle . . . 68
Figure 38. One Pulse mode timing example . . . 69
Figure 39. Pulse Width modulation mode timing with 2 output Compare functions . . . 70
Figure 40. Pulse width modulation cycle . . . 71
Figure 41. SCI block diagram . . . 81
Figure 42. Word length programming . . . 82
Figure 43. Bit sampling in reception mode . . . 89
Figure 44. USB block diagram . . . 98
Figure 45. DMA buffers . . . 99
Figure 46. I²C bus protocol . . . 110
Figure 47. I²C interface block diagram . . . 111
Figure 48. Transfer sequencing . . . 115

Table 3. Device pin description (QFN40, LQFP48, SO34 and SDIP32) (continued)

Pin n°				Pin name	Type	Level		Port /control						Main function (after reset)	Alternate function
								Input				Output			
SDIP32	SO34	QFN40	LQFP48			Input	Output	float	wpu	int	ana	OD	PP		
15	16	23	29	PB2/AIN2	I/O	CT	10mA	X			X		X	Port B2	ADC analog input 2
16	17	24	30	PB1/AIN1	I/O	CT	10mA	X			X		X	Port B1	ADC analog input 1
17	18	25	31	PB0/AIN0	I/O	CT	10mA	X			X		X	Port B0	ADC analog input 0
18	19	26	32	PA7/OCMP2/IT4	I/O	CT			X	X			X	Port A7	Timer output Compare 2
19	20	27	33	PA6/OCMP1/IT3	I/O	CT			X	X			X	Port A6	Timer output Compare 1
20	21	28	34	PA5/ICAP2/IT2	I/O	CT			X	X			X	Port A5	Timer input Capture 2
21	22	29	35	PA4/ICAP1/IT1	I/O	CT			X	X			X	Port A4	Timer input Capture 1
22	23	30	36	PA3/EXTCLK	I/O	CT			X				X	Port A3	Timer External clock
23	24	31	38	PA2/SCL/ICCCLK	I/O	C_T	25mA	X				T		Port A2	I²C serial clock, ICC clock
-	-	32	39	PD0(1)/AIN8	I/O	C_T		X			X		X	Port D0	ADC analog input 8
-	-	33	40	PD1(1)/AIN9	I/O	C_T		X			X		X	Port D1	ADC analog input 9
-	-	34	41	PD2(1)/AIN10	I/O	C_T		X			X		X	Port D2	ADC analog input 10
-	-	35	42	PD3(1)/AIN11	I/O	C_T		X			X		X	Port D3	ADC analog input 11
-	-	36	43	PD4(1)	I/O	C_T			X				X	Port D4	
-	-	37	44	PD5(1)	I/O	C_T			X				X	Port D5	
-	-	38	45	PD6(1)	I/O	C_T			X				X	Port D6	
-	-	39	46	PD7(1)	I/O	C_T			X				X	Port D7	
-	25	-	-	NC	--									Not connected	
24	26	-	-	NC	--									Not connected	
25	27	-	-	NC	--									Not connected	
26	28	40	47	PA1/SDA/ICCDATA	I/O	CT	25mA	X				T		Port A1	I²C serial data, ICC data
27	29	1	48	PA0/MCO	I/O	CT				X			X	Port A0	Main clock output
28	30	2	1	V_{SSA}	S									Analog ground	
29	31	3	2	USBDP	I/O									USB bidirectional data (data +)	
30	32	4	3	USBDM	I/O									USB bidirectional data (data -)	

4.6 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, (user-defined strategy for entering programming mode, choice of communications protocol used to fetch the data to be stored, etc.). For example, it is possible to download code from the SCI or other type of serial 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.

4.7 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*.

4.8 Register description

Flash Control/status register (FCSR)

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

Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
Read/write							

pointed to by the SP. Then the other registers are stored in the next locations as shown in *Figure 10*.

- 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 10. Stack manipulation example

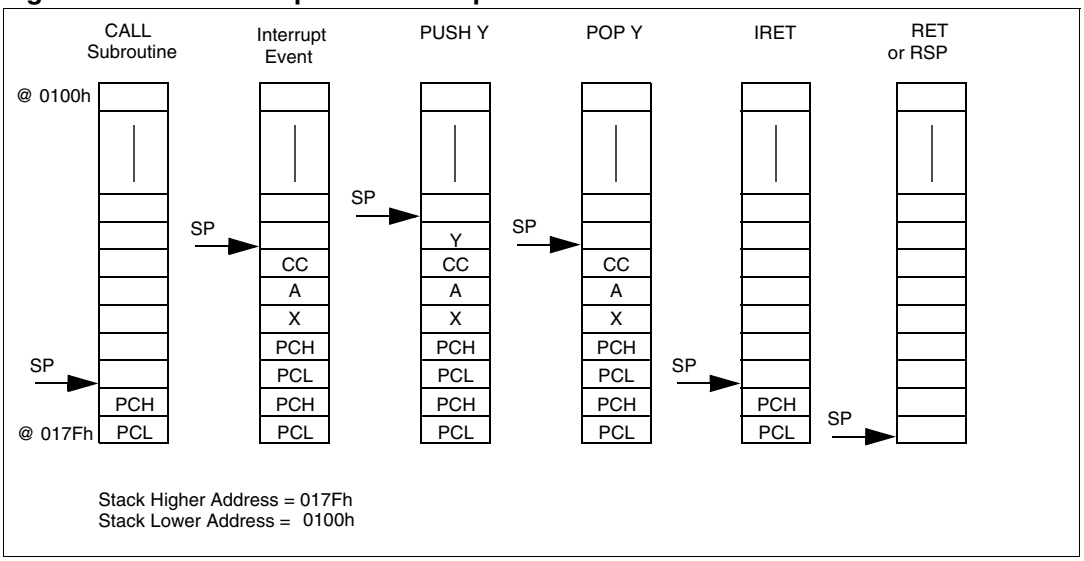
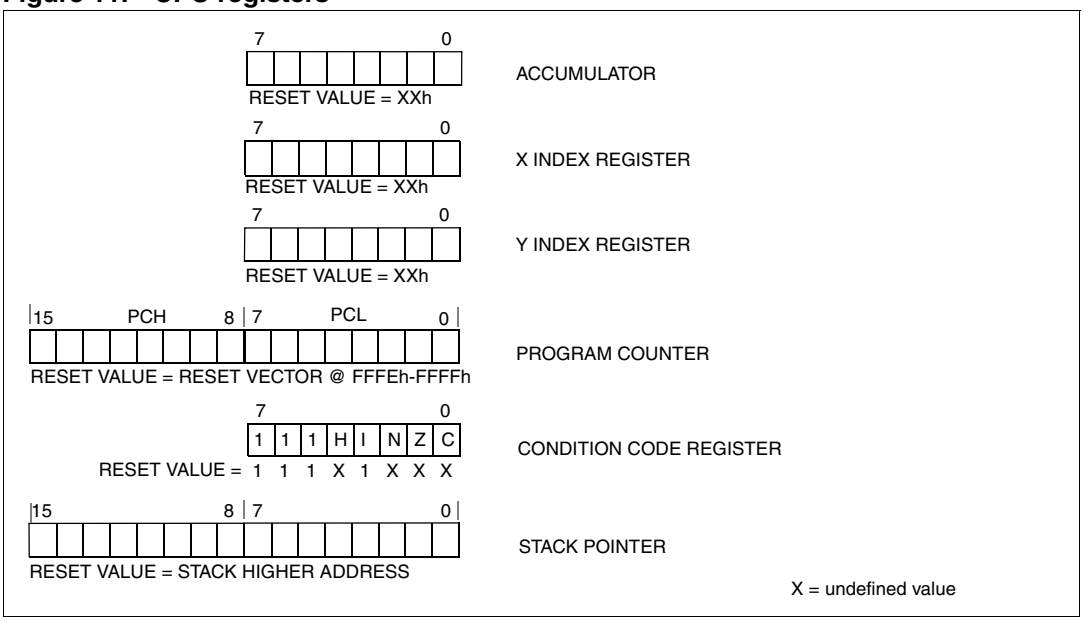


Figure 11. CPU registers



8 Power saving modes

8.1 Introduction

To give a large measure of flexibility to the application in terms of power consumption, two main power saving modes are implemented in the ST7.

After a Reset, the normal operating mode is selected by default (Run mode). This mode drives the device (CPU and embedded peripherals) by means of a master clock which is based on the main oscillator frequency divided by 3 (f_{CPU}).

From Run mode, the different power saving modes may be selected by setting the relevant register bits or by calling the specific ST7 software instruction whose action depends on the oscillator status.

8.2 Halt mode

The MCU consumes the least amount of power in Halt mode. The Halt mode is entered by executing the Halt instruction. The internal oscillator is then turned off, causing all internal processing to be stopped, including the operation of the on-chip peripherals.

When entering Halt mode, the I bit in the Condition Code register is cleared. Thus, all external interrupts (ITi or USB end suspend mode) are allowed and if an interrupt occurs, the CPU clock becomes active.

The MCU can exit Halt mode on reception of either an external interrupt on ITi, an end suspend mode interrupt coming from USB peripheral, or a reset. The oscillator is then turned on and a stabilization time is provided before releasing CPU operation. The stabilization time is 4096 CPU clock cycles.

After the start up delay, the CPU continues operation by servicing the interrupt which wakes it up or by fetching the reset vector if a reset wakes it up.

Figure 34. Output Compare block diagram

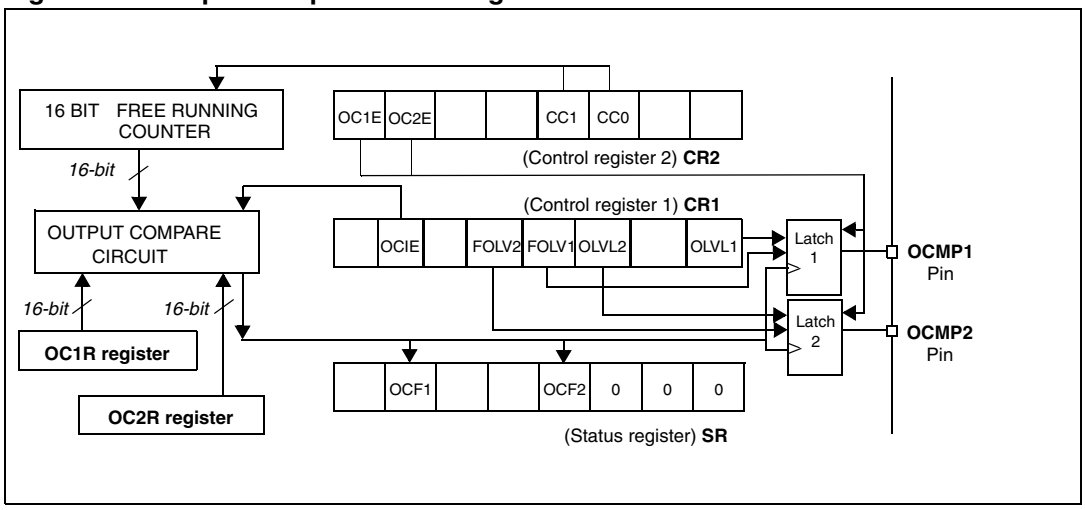


Figure 35. Output Compare timing diagram, $f_{timer} = f_{cpu}/2$

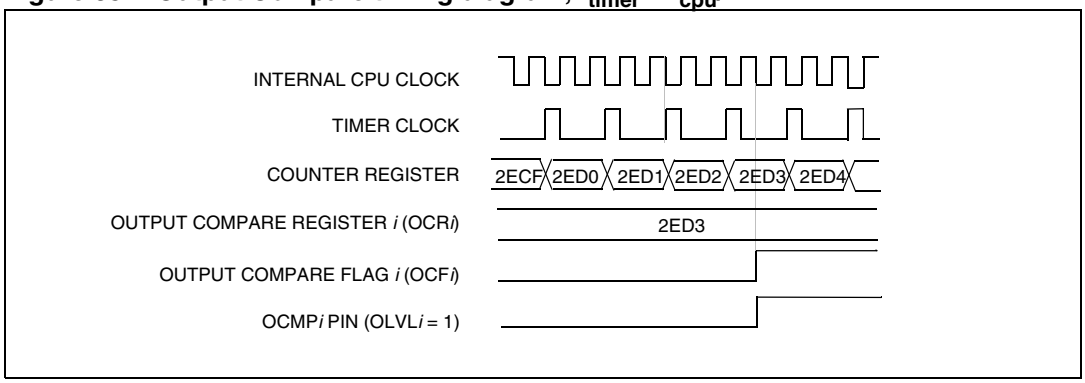


Figure 36. Output Compare timing diagram, $f_{timeR} = f_{CPU}/4$

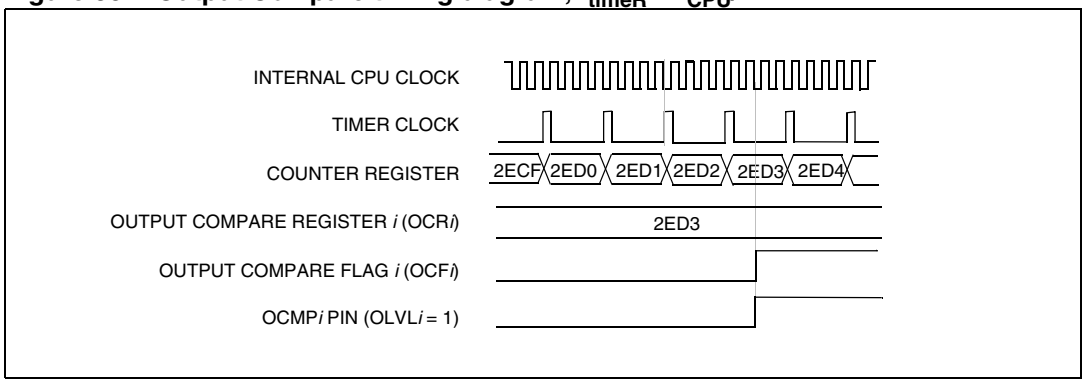
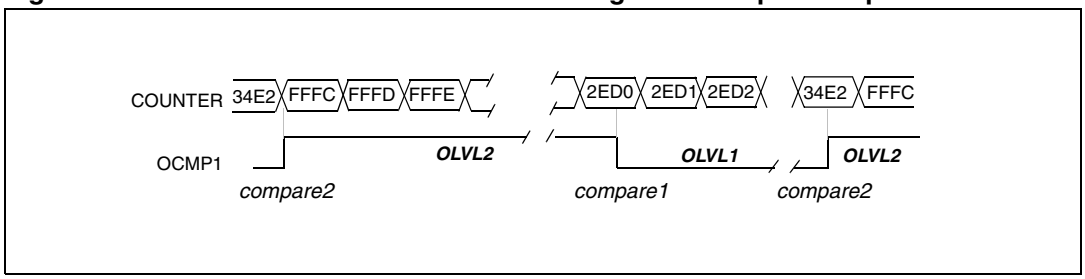


Figure 39. Pulse Width modulation mode timing with 2 output Compare functions



1. OC1R = 2ED0h, OC2R = 34E2, OLVL1 = 0, OLVL2 = 1

On timers with only one output Compare register, a fixed frequency PWM signal can be generated using the output compare and the counter overflow to define the pulse length.

Pulse width modulation mode

Pulse width modulation (PWM) mode enables the generation of a signal with a frequency and pulse length determined by the value of the OC1R and OC2R registers.

Pulse Width Modulation mode uses the complete output Compare 1 function plus the OC2R register, and so this functionality can not be used when PWM mode is activated.

In PWM mode, double buffering is implemented on the output compare registers. Any new values written in the OC1R and OC2R registers are taken into account only at the end of the PWM period (OC2) to avoid spikes on the PWM output pin (OCMP1).

Procedure

To use Pulse Width Modulation mode:

1. Load the OC2R register with the value corresponding to the period of the signal using the formula in the opposite column.
2. Load the OC1R register with the value corresponding to the period of the pulse if (OLVL1 = 0 and OLVL2 = 1) using the formula in the opposite column.
3. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after a successful comparison with the OC1R register.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with the OC2R register.
4. Select the following in the CR2 register:
   - Set OC1E bit: the OCMP1 pin is then dedicated to the output compare 1 function.
   - Set the PWM bit.
   - Select the timer clock (CC[1:0]) (see *Table 24*).

Control register 2 (CR2)

Reset value: 0000 0000 (00h)

7							0
OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG
Read/write							

7 **OC1E** output *Compare 1 Pin Enable.*
This bit is used only to output the signal from the timer on the OCMP1 pin (OLV1 in output Compare mode, both OLV1 and OLV2 in PWM and one-pulse mode). Whatever the value of the OC1E bit, the output Compare 1 function of the timer remains active.

0: OCMP1 pin alternate function disabled (I/O pin free for general-purpose I/O).

1: OCMP1 pin alternate function enabled.

6 **OC2E** output *Compare 2 Pin Enable.*
This bit is used only to output the signal from the timer on the OCMP2 pin (OLV2 in output Compare mode). Whatever the value of the OC2E bit, the output Compare 2 function of the timer remains active.

0: OCMP2 pin alternate function disabled (I/O pin free for general-purpose I/O).

1: OCMP2 pin alternate function enabled.

5 **OPM** *One Pulse mode.*

0: One Pulse mode is not active.

1: One Pulse mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.

4 **PWM** *Pulse Width Modulation.*

0: PWM mode is not active.

1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.

[3:2] **CC[1:0]** *Clock Control.*

The timer clock mode depends on these bits (see *Table 24*).

If the external clock pin is not available, programming the external clock configuration stops the counter.

1 **IEDG2** *input Edge 2.*
This bit determines which type of level transition on the ICAP2 pin will trigger the capture.

0: A falling edge triggers the capture.

1: A rising edge triggers the capture.

0 **EXEDG** *External Clock Edge.*
This bit determines which type of level transition on the external clock pin EXTCLK will trigger the counter register.

0: A falling edge triggers the counter register.

1: A rising edge triggers the counter register.

11.3.4 Functional description

The block diagram of the Serial Control Interface, is shown in *Figure 41* It contains 6 dedicated registers:

- Two control registers (SCICR1 & SCICR2)
- A status register (SCISR)
- A baud rate register (SCIBRR)

Refer to the register descriptions in *Section 11.3.7* for the definitions of each bit.

Serial data format

Word length may be selected as being either 8 or 9 bits by programming the M bit in the SCICR1 register (see *Figure 41*).

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

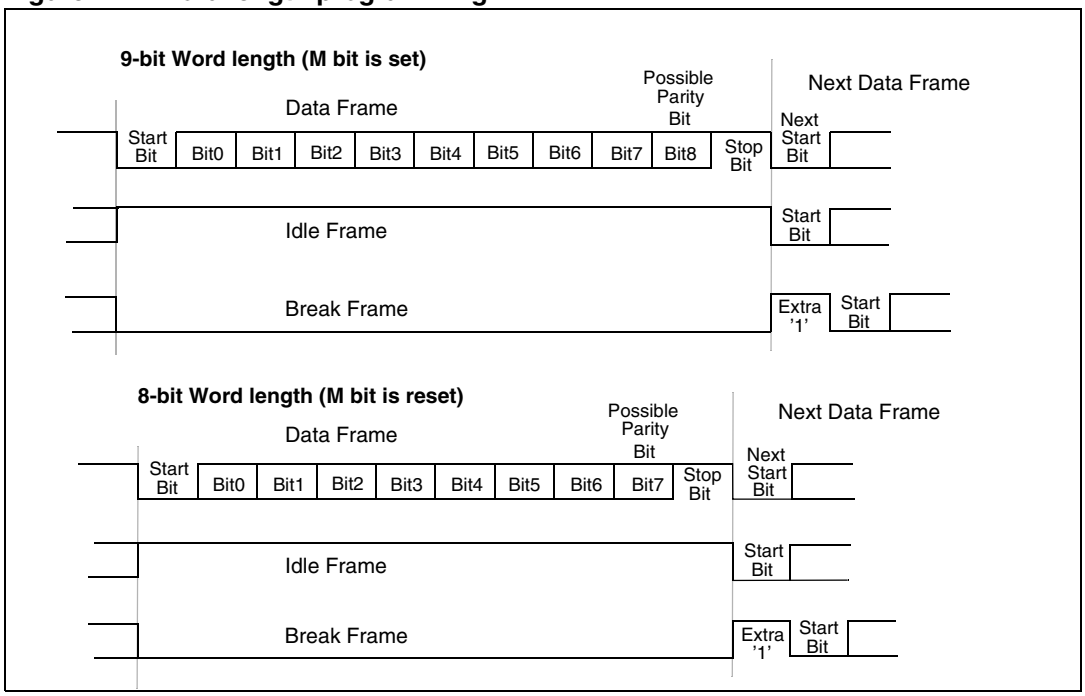
The TDO pin is in high state during the stop bit.

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

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

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

Figure 42. Word length programming



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).

Figure 48. Transfer sequencing

Table 37. Slave receiver

S	Address	A		Data1	A		Data2	A		DataN	A		P	
			EV 1			EV 2			EV 2			EV 2		EV 4

.....

Table 38. Slave Transmitter

S	Address	A			Data1	A		Data2	A		DataN	NA		P	
			EV 1	EV 3			EV 3			EV 3			EV3 -1		EV 4

.....

Table 39. Master receiver

S		Address	A		Data1	A		Data2	A		DataN	NA		P
	EV 5			EV 6			EV 7			EV 7			EV 7	

.....

Table 40. Master Transmitter

S		Address	A			Data1	A		Data2	A		DataN	A		P
	EV 5			EV 6	EV 8			EV 8			EV 8			EV 8	

.....

1. Legend:
   S=Start, P=Stop, A=Acknowledge, NA=Non-acknowledge
   EVx=Event (with interrupt if ITE=1)**EV1:** EVF=1, ADSL=1, cleared by reading the SR1 register.

EV2: EVF=1, BTF=1, cleared by reading the SR1 register followed by reading the DR register.

EV3: EVF=1, BTF=1, cleared by reading the SR1 register followed by writing the DR register.

EV3-1: EVF=1, AF=1, BTF=1; AF is cleared by reading the SR1 register. The BTF is cleared by releasing the lines (STOP=1, STOP=0) or by writing the DR register (DR=FFh).
Note: If lines are released by STOP=1, STOP=0, the subsequent EV4 is not seen.

EV4: EVF=1, STOPF=1, cleared by reading the SR2 register.

EV5: EVF=1, SB=1, cleared by reading the SR1 register followed by writing the DR register.

EV6: EVF=1, cleared by reading the SR1 register followed by writing the CR register (for example PE=1).

EV7: EVF=1, BTF=1, cleared by reading the SR1 register followed by reading the DR register.

EV8: EVF=1, BTF=1, cleared by reading the SR1 register followed by writing the DR register.

11.6.5 Interrupts

None

11.6.6 Register description

Control/Status register (CSR)

Reset value: 0000 0000 (00h)

7							0
COCO	0	ADON	0	CH3	CH2	CH1	CH0
Read/write							

7 **COCO** *Conversion Complete*
This bit is set by hardware. It is cleared by software reading the result in the DR register or writing to the CSR register.
0: Conversion is not complete
1: Conversion can be read from the DR register

6 Reserved. must always be cleared.

5 **ADON** *A/D Converter On*
This bit is set and cleared by software.
0: A/D converter is switched off
1: A/D converter is switched on

4 Reserved. must always be cleared.

[3:0] **CH[3:0]** *Channel Selection*
These bits are set and cleared by software. They select the analog input to convert (see *Table 45*).

Table 45. Channel selection

Channel pin(1)	CH3(2)	CH2	CH1	CH0
AIN0	0	0	0	0
AIN1	0	0	0	1
AIN2	0	0	1	0
AIN3	0	0	1	1
AIN4	0	1	0	0
AIN5	0	1	0	1
AIN6	0	1	1	0
AIN7	0	1	1	1
AIN8	1	0	0	0
AIN9	1	0	0	1
AIN10	1	0	1	0
AIN11	1	0	1	1

1. The number of pins AND the channel selection varies according to the device. Refer to the device pinout.

Indexed (Short)

The offset is a byte, thus requires only one byte after the opcode and allows 00 - 1FE addressing space.

Indexed (long)

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

12.1.5 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.

12.1.6 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.

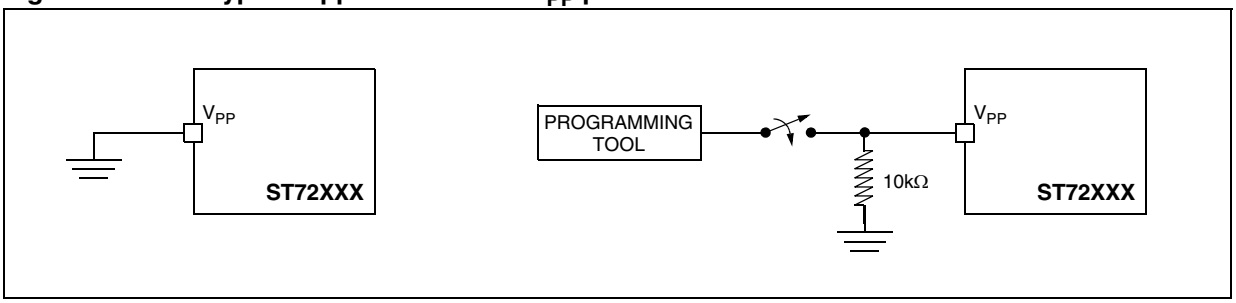
Indirect Indexed (Long)

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

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

Figure 59. Two typical applications with V_{PP} pin



1. When the ICP mode is not required by the application, V_{PP} pin must be tied to V_{SS}.

13.7 EMC characteristics

Susceptibility tests are performed on a sample basis during product characterization.

13.7.1 Functional EMS (electromagnetic susceptibility)

Based on a simple running application on the product (toggling 2 LEDs through I/O ports), the product is stressed by two electromagnetic events until a failure occurs (indicated by the LEDs).

- **ESD**: Electrostatic discharge (positive and negative) is applied on all pins of the device until a functional disturbance occurs. This test conforms with the IEC 1000-4-2 standard.
- **FTB**: A burst of fast transient voltage (positive and negative) is applied to V_{DD} and V_{SS} through a 100pF capacitor, until a functional disturbance occurs. This test conforms with the IEC 1000-4-4 standard.

A device reset allows normal operations to be resumed. The test results are given in the table below based on the EMS levels and classes defined in application note AN1709.

Designing hardened software to avoid noise problems

EMC characterization and optimization are performed at component level with a typical application environment and simplified MCU software. It should be noted that good EMC performance is highly dependent on the user application and the software in particular.

Therefore it is recommended that the user applies EMC software optimization and prequalification tests in relation with the EMC level requested for his application.

Software recommendations:

The software flowchart must include the management of runaway conditions such as:

- Corrupted program counter
- Unexpected reset
- Critical data corruption (control registers...)

Prequalification trials

Most of the common failures (unexpected reset and program counter corruption) can be reproduced by manually forcing a low state on the $\overline{\text{RESET}}$ pin or the Oscillator pins for 1 second.

To complete these trials, ESD stress can be applied directly on the device, over the range of specification values. When unexpected behavior is detected, the software can be hardened to prevent unrecoverable errors occurring (see application note AN1015).

Table 66. EMC characteristics

Symbol	Parameter	Conditions	Level/ Class
V_{FESD}	Voltage limits to be applied on any I/O pin to induce a functional disturbance	V_{DD}=5 V, T_A=+25 °C, f_{OSC}=8 MHz, SDIP32 conforms to IEC 1000-4-2	4B
V_{FFTB}	Fast transient voltage burst limits to be applied through 100 pF on V_{DD} and V_{DD} pins to induce a functional disturbance	V_{DD}=5 V, T_A=+25 °C, f_{OSC}=8 MHz, SDIP32 conforms to IEC 1000-4-4	4A

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.

Table 71. Output driving current

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), Port D(0:7)	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(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 13.2* 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 13.2* 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}.
3. The minimum V_{OH} value (with I_{IO}=-10mA) depends on the chosen device type. For Flash devices, min = V_{DD} - 1.3 V and for ROM devices, min = V_{DD} - 1.7 V

Figure 63. V_{OL} standard V_{DD}=5 V

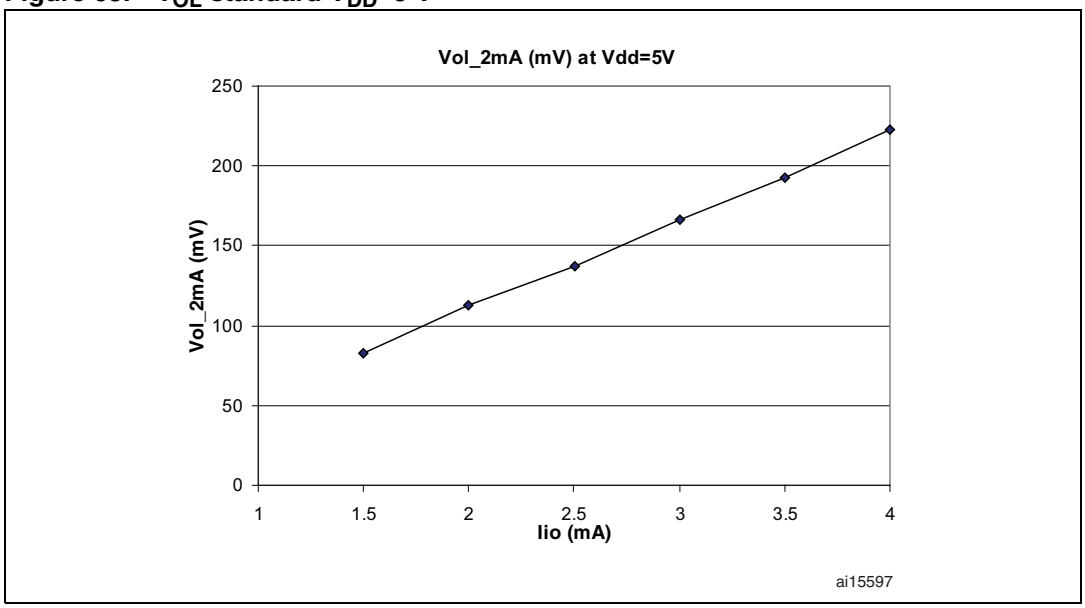


Figure 66. V_{OL} standard vs. V_{DD}

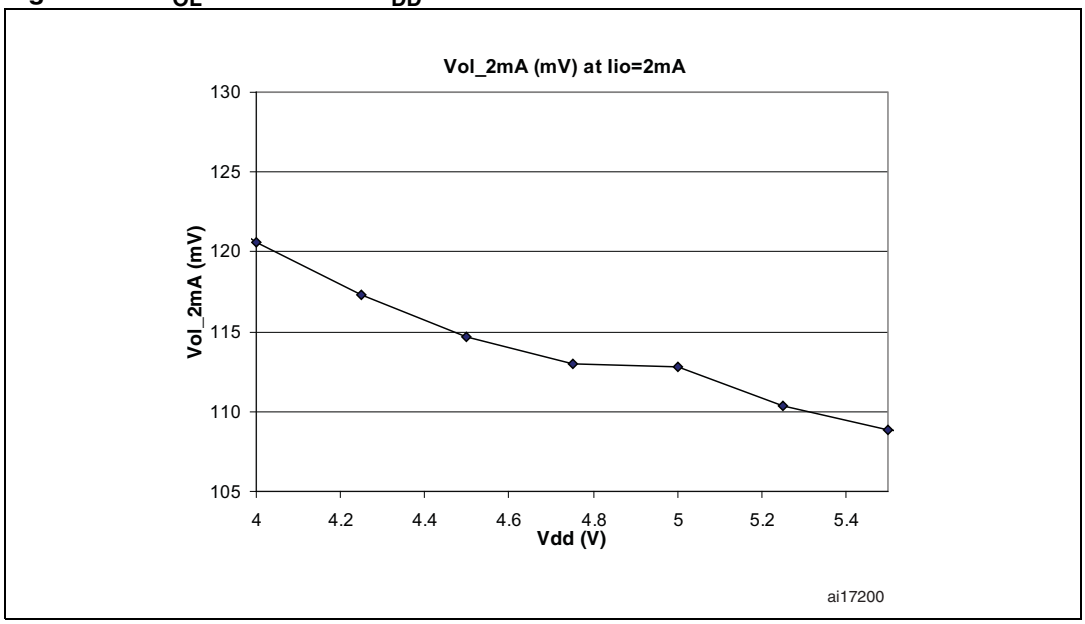


Figure 67. V_{OL} high sink vs. V_{DD}

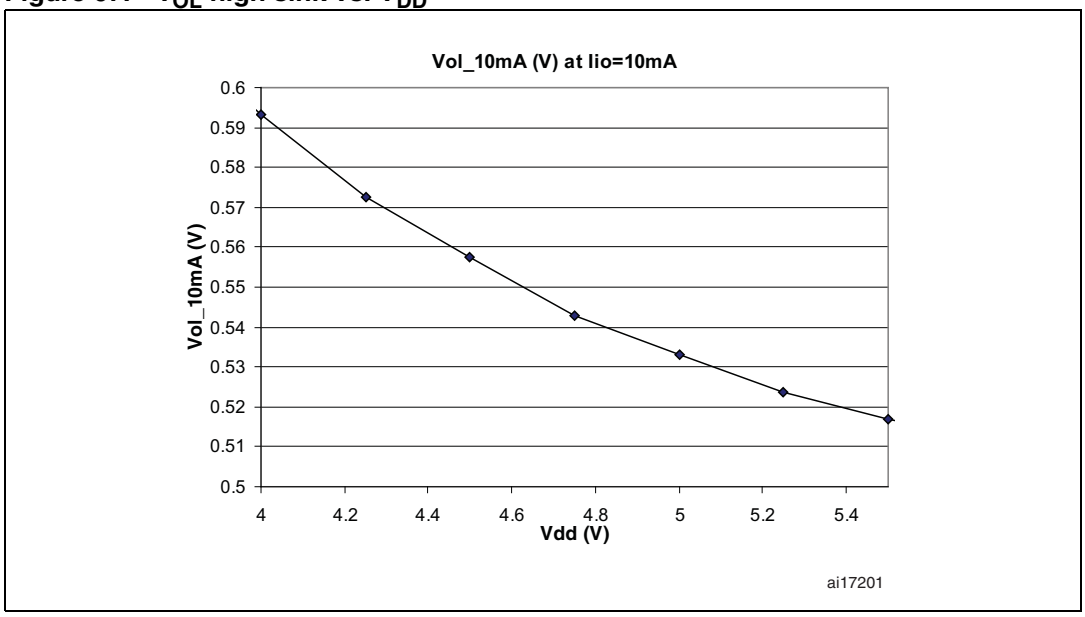


Table 86. Supported order codes (continued)

Sales type[1][2]	Program memory (bytes)	RAM (bytes)	Package
ST72F63BH2T1	8K Flash	384	LQFP48
ST72F63BK2U1			QFN40
ST72F63BK2M1			SO34
ST72F63BK2B1			SDIP32
ST72F63BE2M1			SO24
ST72F63BK1M1	4K Flash	384	SO34
ST72F63BK1B1			SDIP32
ST72F63BE1M1			SO24
ST7263BK2M1/xxx	8K ROM	384	SO34
ST7263BK2B1/xxx			SDIP32
ST7263BK1M1/xxx	4K ROM	384	SO34
ST7263BK1B1/xxx			SDIP32

1. /xxx stands for the ROM code name assigned by STMicroelectronics.
2. Contact ST sales office for FASTROM product availability.

15.3 Development tools

Development tools for the ST7 microcontrollers include a complete range of hardware systems and software tools from STMicroelectronics and third-party tool suppliers. The range of tools includes solutions to help you evaluate microcontroller peripherals, develop and debug your application, and program your microcontrollers.

15.3.1 Evaluation tools and starter kits

ST offers complete, affordable **starter kits** and full-featured **evaluation boards** that allow you to evaluate microcontroller features and quickly start developing ST7 applications. Starter kits are complete, affordable hardware/software tool packages that include features and samples to to help you quickly start developing your application. ST evaluation boards are open-design, embedded systems, which are developed and documented to serve as references for your application design. They include sample application software to help you demonstrate, learn about and implement your ST7's features.

15.3.2 Development and debugging tools

Application development for ST7 is supported by fully optimizing **C Compilers** and the **ST7 Assembler-Linker** toolchain, which are all seamlessly integrated in the ST7 integrated development environments in order to facilitate the debugging and fine-tuning of your application. The Cosmic C Compiler is available in a free version that outputs up to 16K of code.

The range of hardware tools includes full-featured **ST7-EMU3 series emulators** and the low-cost **RLink** in-circuit debugger/programmer. These tools are supported by the **ST7 Toolset** from STMicroelectronics, which includes the STVD7 integrated development

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