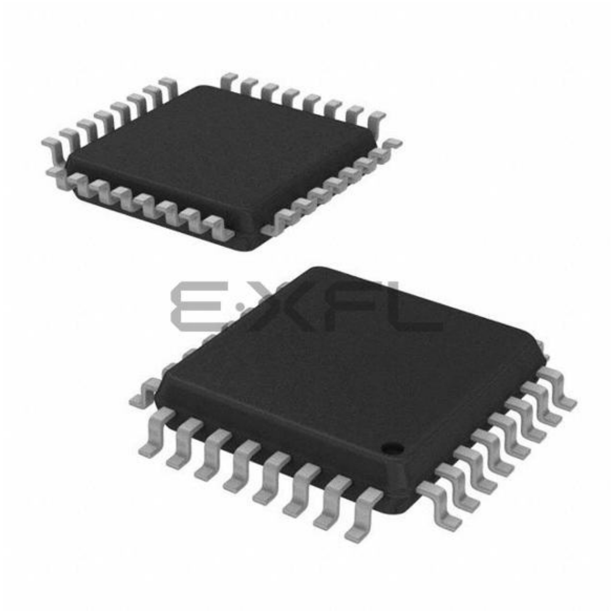
STMicroelectronics - ST7FLI49MK1T6 Datasheet



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
Peripherals	LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st7fli49mk1t6

4 Flash programmable memory

4.1 Introduction

The ST7 single voltage extended Flash (XFlash) is a non-volatile memory that can be electrically erased and programmed either on a byte-by-byte basis or up to 32 bytes in parallel.

The XFlash devices can be programmed off-board (plugged in a programming tool) or on-board using in-circuit programming or in-application programming.

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

4.2 Main features

- ICP (in-circuit programming)
- IAP (in-application programming)
- ICt (in-circuit testing) for downloading and executing user application test patterns in RAM
- Sector 0 size configurable by option byte
- Read-out and write protection

4.3 Programming modes

The ST7 can be programmed in three different ways:

- Insertion in a programming tool. In this mode, Flash sectors 0 and 1, option byte row and data EEPROM (if present) can be programmed or erased.
- In-circuit programming. In this mode, Flash sectors 0 and 1, option byte row and data EEPROM (if present) can be programmed or erased without removing the device from the application board.
- In-application programming. In this mode, sector 1 and data EEPROM (if present) can be programmed or erased without removing the device from the application board and while the application is running.

4.3.1 In-circuit programming (ICP)

ICP uses a protocol called ICC (in-circuit communication) which allows an ST7 plugged on a printed circuit board (PCB) to communicate with an external programming device connected via cable. ICP is performed in three steps:

Switch the ST7 to ICC mode (in-circuit communications). This is done by driving a specific signal sequence on the ICCCLK/DATA pins while the $\overline{\text{RESET}}$ pin is pulled low. When the ST7 enters ICC mode, it fetches a specific reset vector which points to the ST7 System Memory containing the ICC protocol routine. This routine enables the ST7 to receive bytes from the ICC interface.

- Download ICP Driver code in RAM from the ICCDATA pin
- Execute ICP Driver code in RAM to program the Flash memory

6 Central processing unit

6.1 Introduction

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

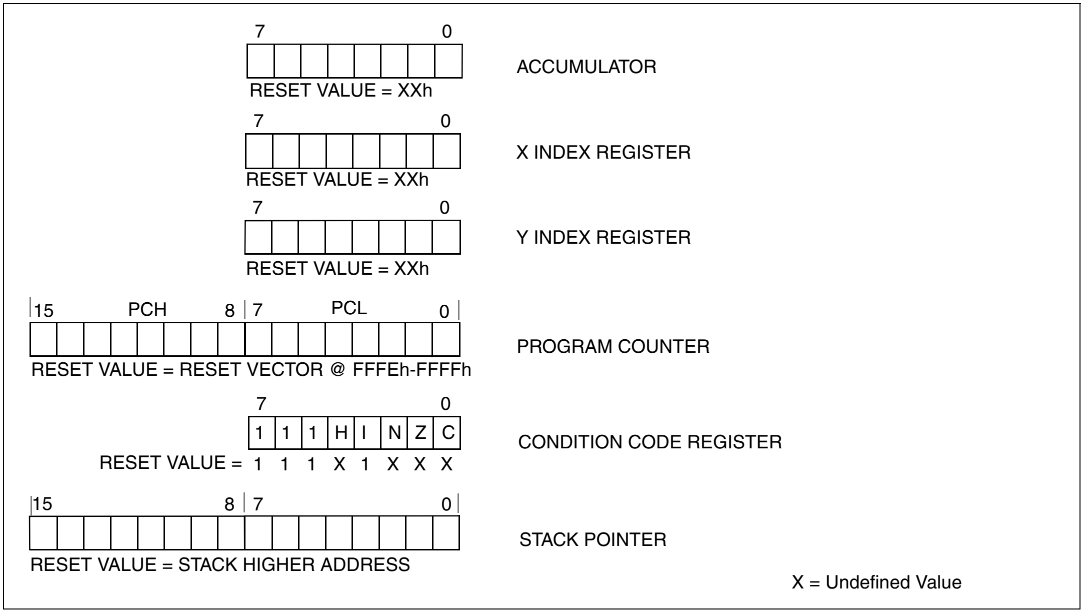
6.2 Main features

- 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 10*. They are not present in the memory mapping and are accessed by specific instructions.

Figure 10. CPU registers



In 38-pulse ICC mode, the internal RC oscillator is forced as a clock source, regardless of the selection in the option byte.

Section 13: Electrical characteristics on page 139 for more information on the frequency and accuracy of the RC oscillator.

To improve clock stability and frequency accuracy, it is recommended to place a decoupling capacitor, typically 100 nF, between the V_{DD} and V_{SS} pins and also between the V_{DDA} and V_{SSA} pins as close as possible to the ST7 device.

These bytes are systematically programmed by ST, including on FASTROM devices.

Caution: If the voltage or temperature conditions change in the application, the frequency may need to be recalibrated. Refer to application note AN1324 for information on how to calibrate the RC frequency using an external reference signal.

7.1.2 Auto-wakeup RC oscillator

The ST7LITE49M also contains an Auto-wakeup RC oscillator. This RC oscillator should be enabled to enter auto-wakeup from halt mode.

The auto-wakeup (AWU) RC oscillator can also be configured as the startup clock through the CKSEL[1:0] option bits (see *Section 14.1: Option bytes on page 173*).

This is recommended for applications where very low power consumption is required.

Switching from one startup clock to another can be done in run mode as follows (see *Figure 12*):

Case 1 Switching from internal RC to AWU

1. Set the RC/AWU bit in the CKCNTCSR register to enable the AWU RC oscillator
2. The RC_FLAG is cleared and the clock output is at 1.
3. Wait 3 AWU RC cycles till the AWU_FLAG is set
4. The switch to the AWU clock is made at the positive edge of the AWU clock signal
5. Once the switch is made, the internal RC is stopped

Case 2 Switching from AWU RC to internal RC

1. Reset the RC/AWU bit to enable the internal RC oscillator
2. Using a 4-bit counter, wait until 8 internal RC cycles have elapsed. The counter is running on internal RC clock.
3. Wait till the AWU_FLAG is cleared (1AWU RC cycle) and the RC_FLAG is set (2 RC cycles)
4. The switch to the internal RC clock is made at the positive edge of the internal RC clock signal
5. Once the switch is made, the AWU RC is stopped

Note: 1 *When the internal RC is not selected, it is stopped so as to save power consumption.*

2 *When the internal RC is selected, the AWU RC is turned on by hardware when entering Auto-wakeup from Halt mode.*

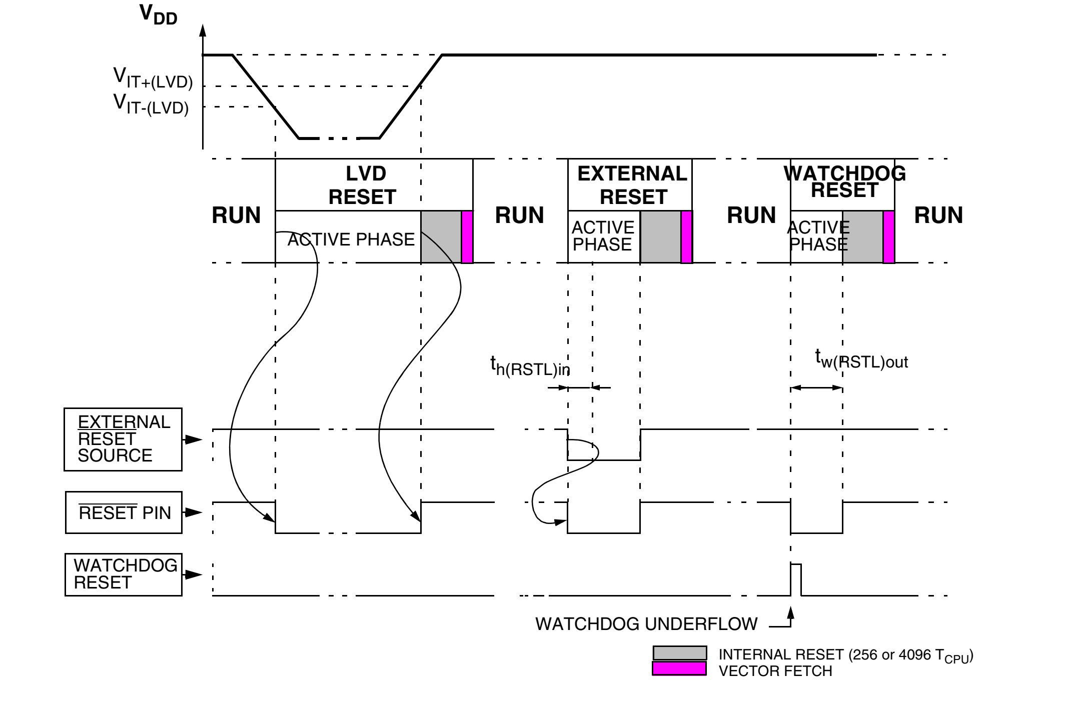
3 *When the external clock is selected, the AWU RC oscillator is always on.*

7.3.5 Internal watchdog reset

The reset sequence generated by an internal watchdog counter overflow is shown in *Figure 16: Reset sequences*

Starting from the watchdog counter underflow, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.

Figure 16. Reset sequences



7.4 System integrity management (SI)

The system integrity management block contains the low voltage detector (LVD) and auxiliary voltage detector (AVD) functions. It is managed by the SICSR register.

Note: A reset can also be triggered following the detection of an illegal opcode or prebyte code. Refer to Section 12.2.1 on page 136 for further details.

7.4.1 Low voltage detector (LVD)

The low voltage detector function (LVD) generates a static reset when the V_{DD} supply voltage is below a $V_{IT-(LVD)}$ reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The $V_{IT-(LVD)}$ reference value for a voltage drop is lower than the $V_{IT+(LVD)}$ reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

The LVD reset circuitry generates a reset when V_{DD} is below:

- $V_{IT+(LVD)}$ when V_{DD} is rising
- $V_{IT-(LVD)}$ when V_{DD} is falling

The LVD function is illustrated in *Figure 17*.

The voltage threshold can be configured by option byte to be low, medium or high. See *Section 14.1 on page 173*.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is above $V_{IT-(LVD)}$, the MCU can only be in two modes:

- Under full software control
- In static safe reset

In these conditions, secure operation is always ensured for the application without the need for external reset hardware.

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

Note: *Use of LVD with capacitive power supply: with this type of power supply, if power cuts occur in the application, it is recommended to pull V_{DD} down to 0 V to ensure optimum restart conditions. Refer to circuit example in Figure 96 on page 169 and note 4.*

The LVD is an optional function which can be selected by option byte. See Section 14.1 on page 173.

It allows the device to be used without any external RESET circuitry.

If the LVD is disabled, an external circuitry must be used to ensure a proper power-on reset.

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.

Make sure that the right combination of LVD and AVD thresholds is used as LVD and AVD levels are not correlated. Refer to section Section 13.3.2 on page 142 and Section 13.3.3 on page 143 for more details.

Caution: If an LVD reset occurs after a watchdog reset has occurred, the LVD will take priority and will clear the watchdog flag.

8 Interrupts

8.1 Introduction

The ST7 enhanced interrupt management provides the following features:

- Hardware interrupts
- Software interrupt (TRAP)
- Nested or concurrent interrupt management with flexible interrupt priority and level management:
  - Up to 4 software programmable nesting levels
  - 13 interrupt vectors fixed by hardware
  - 2 non maskable events: RESET, TRAP

This interrupt management is based on:

- Bit 5 and bit 3 of the CPU CC register (I1:0),
- Interrupt software priority registers (ISPRx),
- Fixed interrupt vector addresses located at the high addresses of the memory mapping (FFE0h to FFFFh) sorted by hardware priority order.

This enhanced interrupt controller guarantees full upward compatibility with the standard (not nested) ST7 interrupt controller.

8.2 Masking and processing flow

The interrupt masking is managed by the I1 and I0 bits of the CC register and the ISPRx registers which give the interrupt software priority level of each interrupt vector (see *Table 14*). The processing flow is shown in *Figure 20*.

When an interrupt request has to be serviced:

- Normal processing is suspended at the end of the current instruction execution.
- The PC, X, A and CC registers are saved onto the stack.
- I1 and I0 bits of CC register are set according to the corresponding values in the ISPRx registers of the serviced interrupt vector.
- The PC is then loaded with the interrupt vector of the interrupt to service and the first instruction of the interrupt service routine is fetched (refer to interrupt mapping table for vector addresses).

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

Note: *As a consequence of the IRET instruction, the I1 and I0 bits will be restored from the stack and the program in the previous level will resume.*

9.2 Slow mode

This mode has two targets:

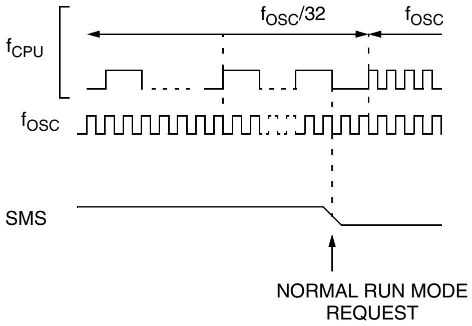
- To reduce power consumption by decreasing the internal clock in the device,
- To adapt the internal clock frequency (f_{CPU}) to the available supply voltage.

Slow mode is controlled by the SMS bit in the MCCSR register which enables or disables Slow mode.

In this mode, the oscillator frequency is divided by 32. The CPU and peripherals are clocked at this lower frequency.

Note: Slow-wait mode is activated when entering Wait mode while the device is already in Slow mode.

Figure 25. Slow mode clock transition



9.3 Wait mode

Wait mode places the MCU in a low power consumption mode by stopping the CPU.

This power saving mode is selected by calling the ‘WFI’ instruction.

All peripherals remain active. During Wait mode, the I bit of the CC register is cleared, to enable all interrupts. All other registers and memory remain unchanged. The MCU remains in Wait mode until an interrupt or reset occurs, whereupon the program counter branches to the starting address of the interrupt or reset service routine.

The MCU will remain in Wait mode until a reset or an interrupt occurs, causing it to wake up.

Refer to *Figure 26* for a description of the Wait mode flowchart.

9.5.3 AWUFH prescaler register (AWUPR)

Reset value: 1111 1111 (FFh)

7							0
AWUPR7	AWUPR6	AWUPR5	AWUPR4	AWUPR3	AWUPR2	AWUPR1	AWUPR0
Read/Write							

Bits 7:0= **AWUPR[7:0]** *Auto-wakeup prescaler*

These 8 bits define the AWUPR dividing factor (see *Table 21*).

Table 21. Configuring the dividing factor

AWUPR[7:0]	Dividing factor
00h	Forbidden
01h	1
...	...
FEh	254
FFh	255

In AWU mode, the time during which the MCU stays in Halt mode, t_{AWU}, is given by the equation below. See also *Figure 32 on page 64*.

$$t_{AWU} = 64 \times AWUPR \times \frac{1}{f_{AWURC}} + t_{RCSTRT}$$

The AWUPR prescaler register can be programmed to modify the time during which the MCU stays in Halt mode before waking up automatically.

Note: If 00h is written to AWUPR, the AWUPR remains unchanged.

Table 22. AWU register mapping and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0048h	**AWUCSR** Reset value	0	0	0	0	0	AWUF	AWUM	AWUEN
0049h	**AWUPR** Reset value	AWUPR7 1	AWUPR6 1	AWUPR5 1	AWUPR4 1	AWUPR3 1	AWUPR2 1	AWUPR1 1	AWUPR0 1

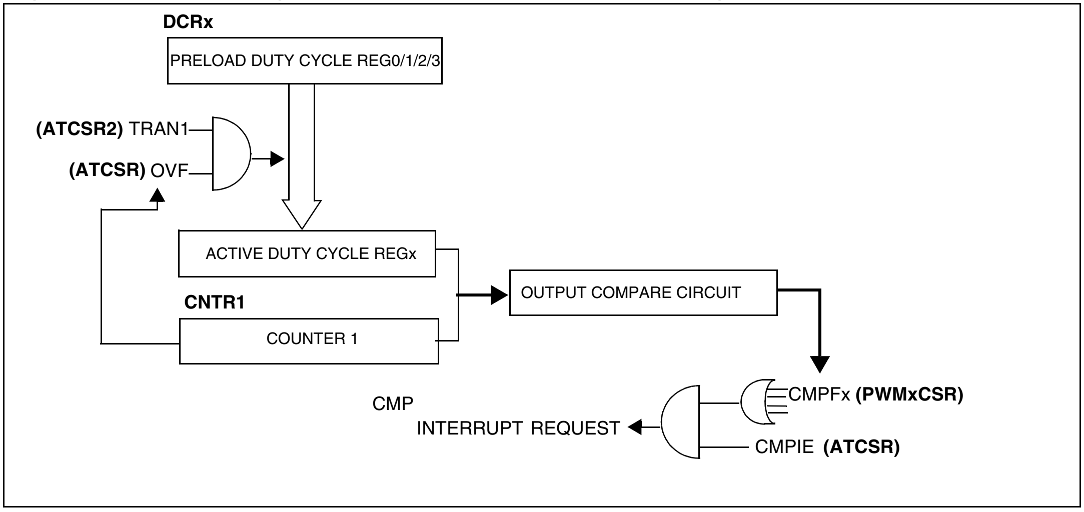
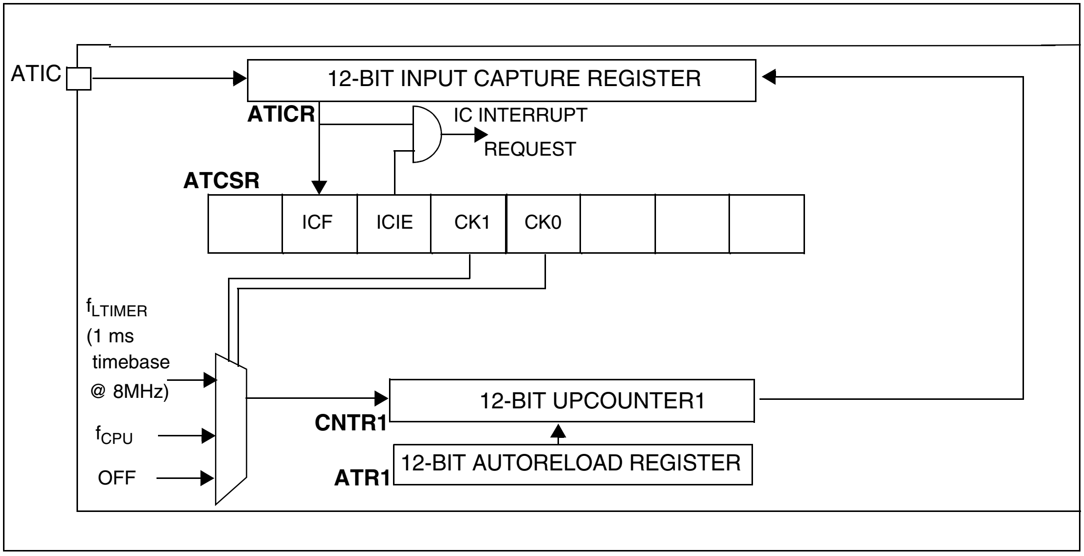


Figure 44. Block diagram of output compare mode (single timer)

Input capture mode

The 12-bit ATICR register is used to latch the value of the 12-bit free running upcounter CNTR1 after a rising or falling edge is detected on the ATIC pin. When an Input Capture occurs, the ICF bit is set and the ATICR register contains the value of the free running upcounter. An IC interrupt is generated if the ICIE bit is set. The ICF bit is reset by reading the ATICRH/ATICRL register when the ICF bit is set. The ATICR is a read only register and always contains the free running upcounter value which corresponds to the most recent input capture. Any further input capture is inhibited while the ICF bit is set.

Figure 45. Block diagram of input capture mode



One-pulse mode

One-pulse mode can be used to control PWM2/3 signal with an external LTIC pin. This mode is available only in Dual Timer mode i.e. only for CNTR2, when the OP_EN bit in PWM3CSR register is set.

One-pulse mode is activated by the external LTIC input. The active edge of the LTIC pin is selected by the OPEDGE bit in the PWM3CSR register.

After getting the active edge of the LTIC pin, CNTR2 is reset (000h) and PWM3 is set to high. CNTR2 starts counting from 000h, when it reaches the active DCR3 value then PWM3 goes low. Till this time, any further transitions on the LTIC signal will have no effect. If there are LTIC transitions after CNTR2 reaches DCR3 value, CNTR2 is reset again and PWM3 goes high.

If there is no LTIC active edge, CNTR2 counts until it reaches the ATR2 value, then it is reset again and PWM3 is set to high. The counter again starts counting from 000h, when it reaches the active DCR3 value PWM3 goes low, the counter counts until it reaches ATR2, it resets and PWM3 is set to high and so on.

The same operation applies for PWM2, but in this case the comparison is done on DCR2. OP_EN and OPEDGE bits take effect on the fly and are not synchronized with Counter 2 overflow. The output bit OP2/3 can be used to inverse the polarity of PWM2/3 in one-pulse mode. The update of these bits (OP2/3) is synchronized with the counter 2 overflow, they will be updated if the TRAN2 bit is set.

The time taken from activation of LTIC input and CNTR2 reset is between 1 and 2 t_{CPU} cycles, that is, 125 ns to 250 ns (with 8-MHz f_{CPU}).

Lite timer Input Capture interrupt should be disabled while 12-bit ARtimer is in One-pulse mode. This is to avoid spurious interrupts.

The priority of the various conditions for PWM3 is the following: Break > one-pulse mode with active LTIC edge > Forced overflow by s/w > one-pulse mode without active LTIC edge > normal PWM operation.

It is possible to update DCR2/3 and OP2/3 at the counter 2 reset, the update is synchronized with the counter reset. This is managed by the overflow interrupt which is generated if counter is reset either due to ATR match or active pulse at LTIC pin. DCR2/3 and OP2/3 update in one-pulse mode is performed dynamically using a software force update. DCR3 update in this mode is not synchronized with any event. That may lead to a longer next PWM3 cycle duration than expected just after the change.

In One-pulse mode ATR2 value must be greater than DCR2/3 value for PWM2/3. (opposite to normal PWM mode).

If there is an active edge on the LTIC pin after the counter has reset due to an ATR2 match, then the timer again gets reset and appears as modified Duty cycle depending on whether the new DCR value is less than or more than the previous value.

The TRAN2 bit should be set along with the FORCE2 bit with the same instruction after a write to the DCR register.

ATR2 value should be changed after an overflow in one-pulse mode to avoid any irregular PWM cycle.

When exiting from one-pulse mode, the OP_EN bit in the PWM3CSR register should be reset first and then the ENCNTR2 bit (if counter 2 must be stopped).

Bits 11:0 = **CNTR1[11:0]** *Counter value*

This 12-bit register is read by software and cleared by hardware after a reset. The counter CNTR1 increments continuously as soon as a counter clock is selected. To obtain the 12-bit value, software should read the counter value in two consecutive read operations. As there is no latch, it is recommended to read LSB first. In this case, CNTR1H can be incremented between the two read operations and to have an accurate result when f_{timer}=f_{CPU}, special care must be taken when CNTR1L values close to FFh are read.

When a counter overflow occurs, the counter restarts from the value specified in the ATR1 register.

Autoreload register (ATR1H)

Reset value: 0000 0000 (00h)

15							8
0	0	0	0	ATR11	ATR10	ATR9	ATR8
Read/write							

Autoreload register (ATR1L)

Reset value: 0000 0000 (00h)

7							0
ATR7	ATR6	ATR5	ATR4	ATR3	ATR2	ATR1	ATR0
Read/write							

Bits 11:0 = **ATR1[11:0]** *Autoreload register 1:*
This is a 12-bit register which is written by software. The ATR1 register value is automatically loaded into the upcounter CNTR1 when an overflow occurs. The register value is used to set the PWM frequency.

PWM output control register (PWMCR)

Reset value: 0000 0000 (00h)

7							0
0	OE3	0	OE2	0	OE1	0	OE0
Read/write							

Bits 7:0 = **OE[3:0]** *PWMx output enable* bits

These bits are set and cleared by software and cleared by hardware after a reset.

0: PWM mode disabled. PWMx output alternate function disabled (I/O pin free for general purpose I/O)

1: PWM mode enabled

Bit 0 = **TB2F** *Timebase 2 Interrupt flag*

This bit is set by hardware and cleared by software reading the LTCSR register. Writing to this bit has no effect.

0: No Counter 2 overflow

1: A Counter 2 overflow has occurred

Lite timer autoreload register (LTARR)

Reset value: 0000 0000 (00h)

7							0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
Read / Write							

Bits 7:0 = **AR[7:0]** *Counter 2 reload value*

These bits register is read/write by software. The LTARR value is automatically loaded into Counter 2 (LTCNTR) when an overflow occurs.

Lite timer counter 2 (LTCNTR)

Reset value: 0000 0000 (00h)

7							0
CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
Read only							

Bits 7:0 = **CNT[7:0]** *Counter 2 Reload value*

This register is read by software. The LTARR value is automatically loaded into Counter 2 (LTCNTR) when an overflow occurs.

Lite timer control/status register (LTCSR1)

Reset value: 0x00 0000 (x0h)

7							0
ICIE	ICF	TB	TB1IE	TB1F			
Read / Write							

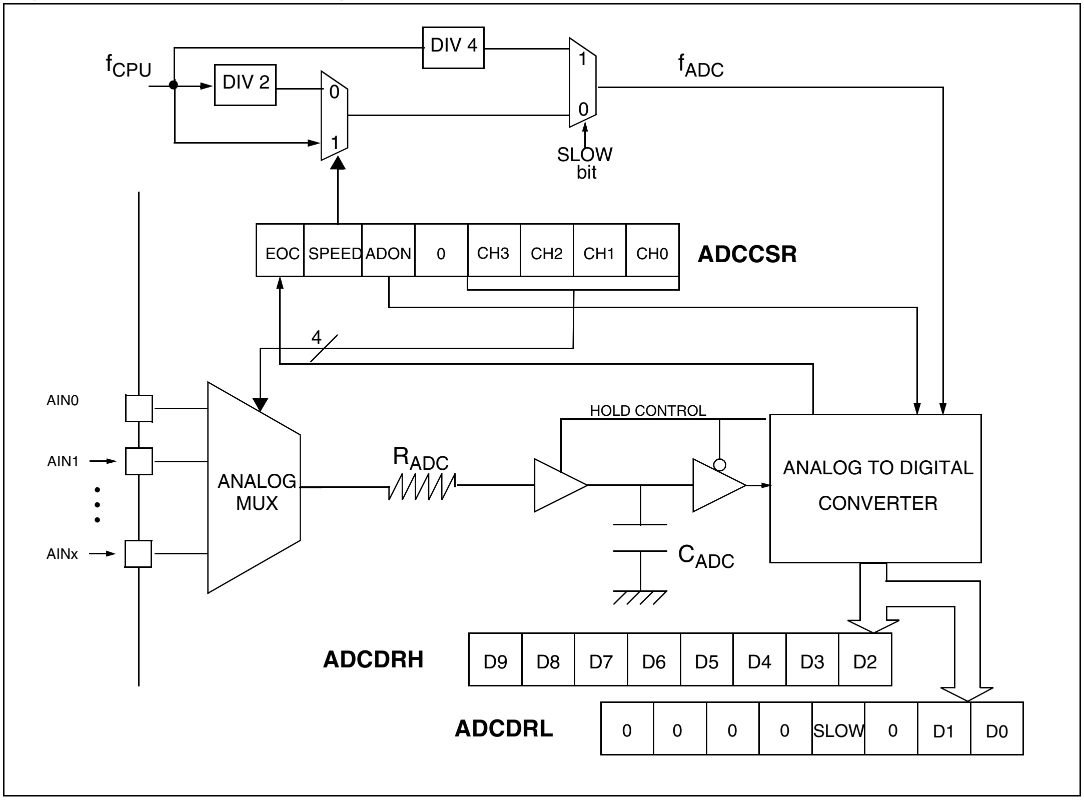
Bit 7 = **ICIE** *Interrupt enable bit*

This bit is set and cleared by software.

0: Input Capture (IC) interrupt disabled

1: Input Capture (IC) interrupt enabled

Figure 59. ADC block diagram



Digital A/D conversion result

The conversion is monotonic, meaning that the result never decreases if the analog input does not and never increases if the analog input does not.

If the input voltage (V_{AIN}) is greater than V_{DDA} (high-level voltage reference) then the conversion result is FFh in the ADCDRH register and 03h in the ADCDRL register (without overflow indication).

If the input voltage (V_{AIN}) is lower than V_{SSA} (low-level voltage reference) then the conversion result in the ADCDRH and ADCDRL registers is 00 00h.

The A/D converter is linear and the digital result of the conversion is stored in the ADCDRH and ADCDRL registers. The accuracy of the conversion is described in the Electrical Characteristics section.

R_{AIN} is the maximum recommended impedance for an analog input signal. If the impedance is too high, this will result in a loss of accuracy due to leakage and sampling not being completed in the alloted time.

11.5.6 Register description

Control/status register (ADCCSR)

Reset value: 0000 0000 (00h)

7							0
EOC	SPEED	ADON	0	CH3	CH2	CH1	CH0
Read only	Read/write						

Bit 7 = **EOC** *End of conversion bit*

This bit is set by hardware. It is cleared by hardware when software reads the ADCDRH register or writes to any bit of the ADCCSR register.

0: Conversion is not complete

1: Conversion complete

Bit 6 = **SPEED** *ADC clock selection bit*

This bit is set and cleared by software. It is used together with the SLOW bit to configure the ADC clock speed. Refer to the table in the SLOW bit description (ADCDRL register).

Bit 5 = **ADON** *A/D converter ON bit*

This bit is set and cleared by software.

0: A/D converter is switched off

1: A/D converter is switched on

Bit 4 = Reserved**,** must be kept cleared.

Bits 3:0 = **CH[3:0]** *Channel selection*

These bits select the analog input to convert. They are set and cleared by software.

Table 47. Channel selection using CH[3:0]

Channel pin[1]	CH3	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

[1] The number of channels is device dependent. Refer to the device pinout description.

12 Instruction set

12.1 ST7 addressing modes

The ST7 core features 17 different addressing modes which can be classified in seven main groups:

Table 50. Description of addressing modes

Addressing mode	Example
Inherent	nop
Immediate	ld A,#$55
Direct	ld A,$55
Indexed	ld A,($55,X)
Indirect	ld A,([$55],X)
Relative	jrne loop
Bit operation	bset byte,#5

The ST7 instruction set is designed to minimize the number of bytes required per instruction: To do so, most of the addressing modes may be subdivided in two submodes called long and short:

- Long addressing mode is more powerful because it can use the full 64 Kbyte address space, however it uses more bytes and more CPU cycles.
- Short addressing mode is less powerful because it can generally only access page zero (0000h - 00FFh range), but the instruction size is more compact, and faster. All memory to memory instructions use short addressing modes only (CLR, CPL, NEG, BSET, BRES, BTJT, BTJF, INC, DEC, RLC, RRC, SLL, SRL, SRA, SWAP)

The ST7 Assembler optimizes the use of long and short addressing modes.

Table 51. ST7 addressing mode overview

Mode			Syntax	Destination/ source	Pointer address	Pointer size	Length (bytes)
Inherent			nop				+ 0
Immediate			ld A,#$55				+ 1
Short	Direct		ld A,$10	00..FF			+ 1
Long	Direct		ld A,$1000	0000..FFFF			+ 2
No Offset	Direct	Indexed	ld A,(X)	00..FF			+ 0 (with X register) + 1 (with Y register)
Short	Direct	Indexed	ld A,($10,X)	00..1FE			+ 1
Long	Direct	Indexed	ld A,($1000,X)	0000..FFFF			+ 2
Short	Indirect		ld A,[$10]	00..FF	00..FF	byte	+ 2
Long	Indirect		ld A,[$10.w]	0000..FFFF	00..FF	word	+ 2
Short	Indirect	Indexed	ld A,([$10],X)	00..1FE	00..FF	byte	+ 2

Table 59. Current characteristics

Symbol	Ratings	Maximum value	Unit
I_{VDD}	Total current into V_{DD} power lines (source)(1)	75	mA
I_{VSS}	Total current out of V_{SS} ground lines (sink)(1)	150	
I_{IO}	Output current sunk by any standard I/O and control pin	20	
	Output current sunk by any high sink I/O pin	40	
	Output current source by any I/Os and control pin	- 25	
$I_{INJ(PIN)}$(2)(3)	Injected current on $\overline{\text{RESET}}$ pin	± 5	
	Injected current on OSC1/CLKIN and OSC2 pins	± 5	
	Injected current on any other pin(4)	± 5	
$\Sigma I_{INJ(PIN)}$(2)	Total injected current (sum of all I/O and control pins)(4)	± 20	

1. All power (V_{DD}) and ground (V_{SS}) lines must always be connected to the external supply.
2. $I_{INJ(PIN)}$ must never be exceeded. This is implicitly insured if V_{IN} maximum is respected. If V_{IN} maximum cannot be respected, the injection current must be limited externally to the $I_{INJ(PIN)}$ value. A positive injection is induced by $V_{IN}>V_{DD}$ while a negative injection is induced by $V_{IN}<V_{SS}$. For true open-drain pads, there is no positive injection current, and the corresponding V_{IN} maximum must always be respected
3. Negative injection disturbs the analog performance of the device. In particular, it induces leakage currents throughout the device including the analog inputs. To avoid undesirable effects on the analog functions, care must be taken:
   - Analog input pins must have a negative injection less than 0.8 mA (assuming that the impedance of the analog voltage is lower than the specified limits)
   - Pure digital pins must have a negative injection less than 1.6 mA. In addition, it is recommended to inject the current as far as possible from the analog input pins.
4. When several inputs are submitted to a current injection, the maximum $\Sigma I_{INJ(PIN)}$ is the absolute sum of the positive and negative injected currents (instantaneous values). These results are based on characterization with $\Sigma I_{INJ(PIN)}$ maximum current injection on four I/O port pins of the device.

Table 60. Thermal characteristics

Symbol	Ratings	Value	Unit
T_{STG}	Storage temperature range	-65 to +150	°C
T_J	Maximum junction temperature (see *Table 98: Thermal characteristics on page 184*)		

13.4 Supply current characteristics

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

13.4.1 Supply current

T_A = -40 to +125 °C unless otherwise specified.

Table 67. Supply current characteristics

Symbol	Parameter	Conditions		Typ	Max	Unit
I_{DD}	Supply current in Run mode[1]	V_{DD}=5 V	f_{CPU} = 4 MHz	2.5	4.5[2]	mA
			f_{CPU} = 8 MHz	5.0	9.5	
	Supply current in Wait mode[3]		f_{CPU} = 4 MHz	1.1	2[2]	
			f_{CPU} = 8 MHz	2	3.5	
	Supply current in Slow mode[4]		f_{CPU}/32 = 250 kHz	550	900	µA
	Supply current in Slow-wait mode[5]		f_{CPU}/32 = 250 kHz	450	750	
	Supply current in AWUFH mode[6][7]			50	90[2]	
	Supply current in Active-halt mode			120	200	
	Supply current in Halt mode[8]		T_A = 85 °C	0.5	5	
			T_A = 125 °C	0.5	5	
I_{DD}	Supply current in Run mode[1]	V_{DD}=3 V	f_{CPU} = 4 MHz	1.4	2.5[2]	mA
	Supply current in Wait mode[3]		f_{CPU} = 4 MHz	600	900[2]	
	Supply current in Slow mode[4]		f_{CPU}/32 = 250 kHz	300	500[2]	µA
	Supply current in Slow-wait mode[5]		f_{CPU}/32 = 250 kHz	250	450[2]	
	Supply current in AWUFH mode[6][7]			20	40[2]	
	Supply current in Active-halt mode			80	120[2]	
	Supply current in Halt mode[8]		T_A = 85 °C	0.5	5[2]	
			T_A = 125 °C	0.5	5	

1. CPU running with memory access, all I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
2. Data based on characterization, not tested in production.
3. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
4. Slow mode selected with f_{CPU} based on f_{OSC} divided by 32. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
5. Slow-wait mode selected with f_{CPU} based on f_{OSC} divided by 32. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load), all peripherals in reset state; clock input (CLKIN) driven by external square wave, LVD disabled.
6. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load). Data tested in production at V_{DD} max. and f_{CPU} max.
7. This consumption refers to the Halt period only and not the associated run period which is software dependent.
8. All I/O pins in output mode with a static value at V_{SS} (no load), LVD disabled. Data based on characterization results, tested in production at V_{DD} max and f_{CPU} max.

Figure 70. Typical I_{DD} in Slow-wait mode vs. f_{CPU}

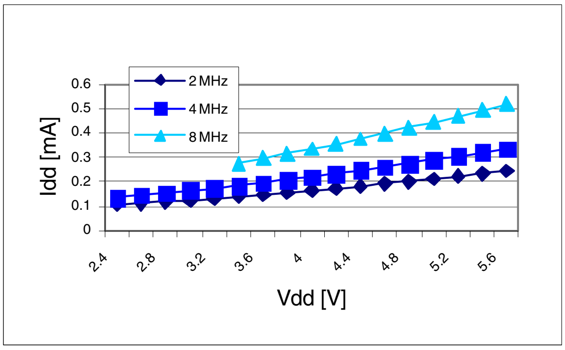


Figure 71. Typical I_{DD} vs. temperature at V_{DD} = 5 V and f_{CPU} = 8 MHz

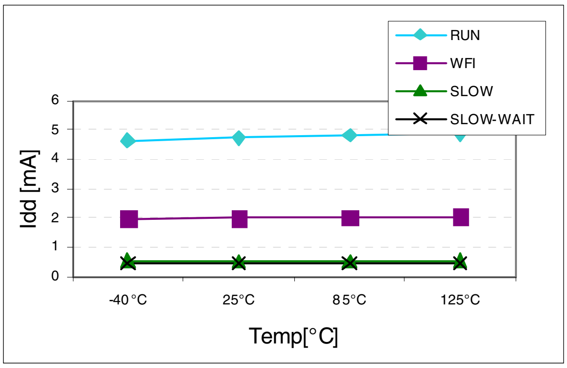


Figure 73. Typical application with a crystal or ceramic resonator

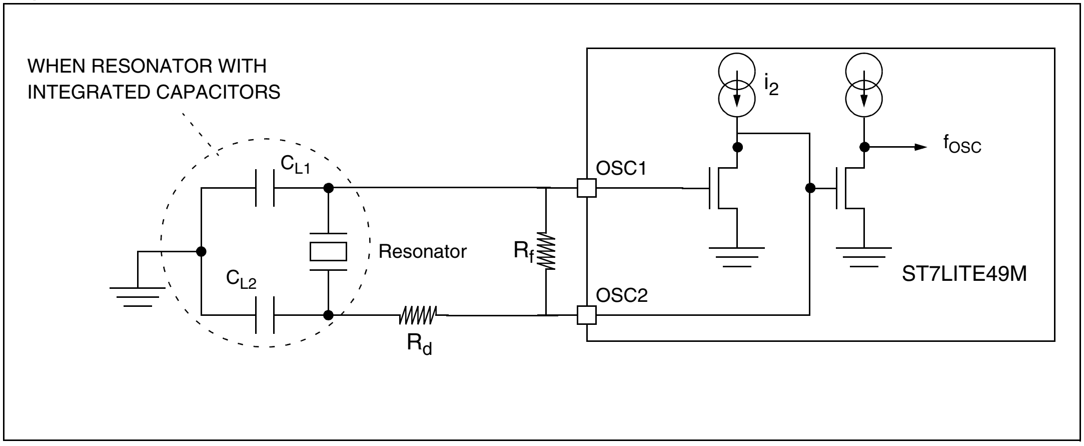


Figure 94. Typical V_{DD}-V_{OH} vs. V_{DD} at I_{IO} = 2 mA (high sink)

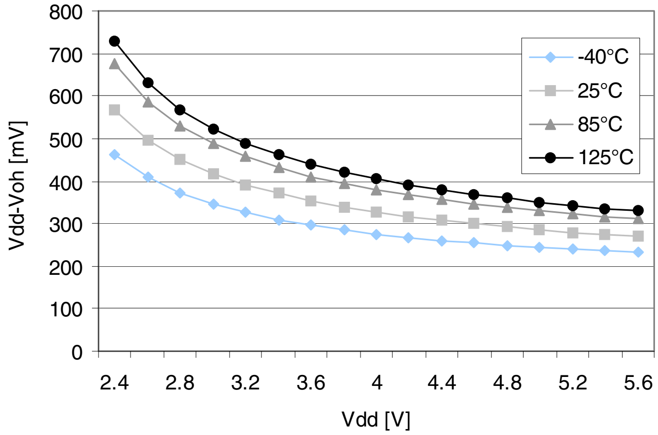
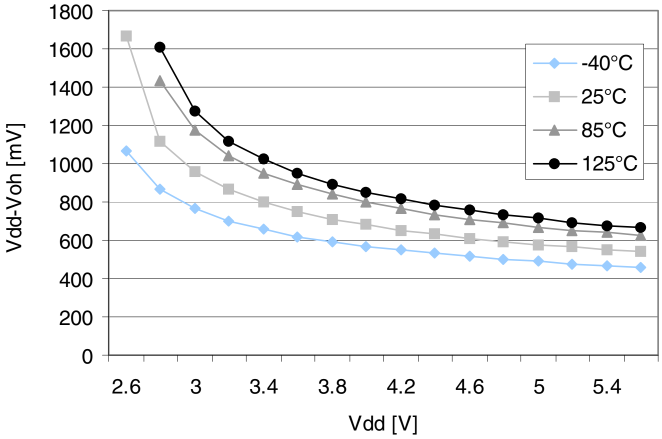


Figure 95. Typical V_{DD}-V_{OH} vs. V_{DD} at I_{IO} = 4 mA (high sink)


Vdd-Voh [mV]
Vdd [V]
-40°C
25°C
85°C
125°C