



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, LINbusSCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	48
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f561ar9t6

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

6 SUPPLY, RESET AND CLOCK MANAGEMENT

The device includes a range of utility features for securing the application in critical situations (for example, in case of a power brown-out), and reducing the number of external components. An overview is shown in Figure 11.

For more details, refer to dedicated parametric section.

Main features

/رک

- Optional PLL for multiplying the frequency by 2
- Reset Sequence Manager (RSM)
- Multi-Oscillator Clock Management (MO)
 4 Crystal/Ceramic resonator oscillators
- System Integrity Management (SI)
 - Main supply Low voltage detection (LVD)
 - Auxiliary Voltage detector (AVD) with interrupt capability for monitoring the main supply

6.1 PHASE LOCKED LOOP

If the clock frequency input to the PLL is in the range 2 to 4 MHz, the PLL can be used to multiply the frequency by two to obtain an f_{OSC2} of 4 to 8 MHz. The PLL is enabled by option byte. If the PLL is disabled, then $f_{OSC2} = f_{OSC}/2$.

Caution: The PLL is not recommended for applications where timing accuracy is required. See "PLL Characteristics" on page 228.

Figure 10. PLL Block Diagram



Figure 11. Clock, Reset and Supply Block Diagram



9 I/O PORTS

9.1 INTRODUCTION

The I/O ports offer different functional modes: - transfer of data through digital inputs and outputs

and for specific pins:

- external interrupt generation
- alternate signal input/output for the on-chip peripherals.

An I/O port contains up to 8 pins. Each pin can be programmed independently as digital input (with or without interrupt generation) or digital output.

9.2 FUNCTIONAL DESCRIPTION

Each port has two main registers:

- Data Register (DR)

- Data Direction Register (DDR)

and one optional register:

- Option Register (OR)

Each I/O pin may be programmed using the corresponding register bits in the DDR and OR registers: Bit X corresponding to pin X of the port. The same correspondence is used for the DR register.

The following description takes into account the OR register, (for specific ports which do not provide this register refer to the I/O Port Implementation section). The generic I/O block diagram is shown in Figure 32

9.2.1 Input Modes

The input configuration is selected by clearing the corresponding DDR register bit.

In this case, reading the DR register returns the digital value applied to the external I/O pin.

Different input modes can be selected by software through the OR register.

Notes:

1. Writing the DR register modifies the latch value but does not affect the pin status.

2. When switching from input to output mode, the DR register has to be written first to drive the correct level on the pin as soon as the port is configured as an output.

3. Do not use read/modify/write instructions (BSET or BRES) to modify the DR register as this might corrupt the DR content for I/Os configured as input.

External interrupt function

When an I/O is configured as Input with Interrupt, an event on this I/O can generate an external interrupt request to the CPU. Each pin can independently generate an interrupt request. The interrupt sensitivity is independently programmable using the sensitivity bits in the EICR register.

Each external interrupt vector is linked to a dedicated group of I/O port pins (see pinout description and interrupt section). If several input pins are selected simultaneously as interrupt sources, these are first detected according to the sensitivity bits in the EICR register and then logically ORed.

The external interrupts are hardware interrupts, which means that the request latch (not accessible directly by the application) is automatically cleared when the corresponding interrupt vector is fetched. To clear an unwanted pending interrupt by software, the sensitivity bits in the EICR register must be modified.

9.2.2 Output Modes

The output configuration is selected by setting the corresponding DDR register bit. In this case, writing the DR register applies this digital value to the I/O pin through the latch. Then reading the DR register returns the previously stored value.

Two different output modes can be selected by software through the OR register: Output push-pull and open-drain.

DR register value and output pin status:

DR	Push-pull	Open-drain		
0	V _{SS}	Vss		
1	V _{DD}	Floating		

9.2.3 Alternate Functions

When an on-chip peripheral is configured to use a pin, the alternate function is automatically selected. This alternate function takes priority over the standard I/O programming.

When the signal is coming from an on-chip peripheral, the I/O pin is automatically configured in output mode (push-pull or open drain according to the peripheral).

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

Note: Input pull-up configuration can cause unexpected value at the input of the alternate peripheral input. When an on-chip peripheral use a pin as input and output, this pin has to be configured in input floating mode.



I/O PORTS (Cont'd)

Table 13. I/O Port Configurations



Notes:

- 1. When the I/O port is in input configuration and the associated alternate function is enabled as an output, reading the DR register will read the alternate function output status.
- When the I/O port is in output configuration and the associated alternate function is enabled as an input, the alternate function reads the pin status given by the DR register content.

ON-CHIP PERIPHERALS (Cont'd)

PWM CONTROL REGISTER (PWMCR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
OE3	OE2	OE1	OE0	OP3	OP2	OP1	OP0

Bit 7:4 = **OE[3:0]** *PWM Output Enable*

These bits are set and cleared by software. They enable or disable the PWM output channels independently acting on the corresponding I/O pin. 0: PWM output disabled.

1: PWM output enabled.

Bit 3:0 = OP[3:0] PWM Output Polarity

These bits are set and cleared by software. They independently select the polarity of the four PWM output signals.

PWMx ou	OPv	
Counter <= OCRx	Counter > OCRx	
1	0	0
0	1	1

Note: When an OPx bit is modified, the PWMx output signal polarity is immediately reversed.

DUTY CYCLE REGISTERS (PWMDCRx)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0

Bit 7:0 = DC[7:0] Duty Cycle Data

These bits are set and cleared by software.

A PWMDCRx register is associated with the OCRx register of each PWM channel to determine the second edge location of the PWM signal (the first edge location is common to all channels and given by the ARTARR register). These PWMDCR registers allow the duty cycle to be set independently for each PWM channel.





16-BIT TIMER (Cont'd)

Figure 52. Input Capture Block Diagram



Figure 53. Input Capture Timing Diagram

TIMER CLOCK	- duct (S)			7
COUNTER REGISTER	FF01 X FF02	FF	03	
ICAPI PIN ICAPI FLAG				
ICAPI REGISTER			FF03	
Note: The rising e	edge is the active edge.			



16-BIT TIMER (Cont'd)

10.4.3.6 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 17 Clock Control Bits).



If OLVL1 = 1 and OLVL2 = 0 the length of the positive pulse is the difference between the OC2R and OC1R registers.

If OLVL1 = OLVL2 a continuous signal will be seen on the OCMP1 pin.

The OC*i*R register value required for a specific timing application can be calculated using the following formula:

$$OC_{iR} Value = \frac{t \cdot f_{CPU}}{PRESC} - 5$$

Where:

t = Signal or pulse period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits, see Table 17 Clock Control Bits)

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

Where:

= Signal or pulse period (in seconds)

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

The Output Compare 2 event causes the counter to be initialized to FFFCh (See Figure 58)

Notes:

- 1. After a write instruction to the OC*i*HR register, the output compare function is inhibited until the OC*i*LR register is also written.
- 2. The OCF1 and OCF2 bits cannot be set by hardware in PWM mode therefore the Output Compare interrupt is inhibited.
- 3. The ICF1 bit is set by hardware when the counter reaches the OC2R value and can produce a timer interrupt if the ICIE bit is set and the I bit is cleared.
- 4. In PWM mode the ICAP1 pin can not be used to perform input capture because it is disconnected to the timer. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each period and ICF1 can also generates interrupt if ICIE is set.
- 5. When the Pulse Width Modulation (PWM) and One Pulse mode (OPM) bits are both set, the PWM mode is the only active one.

16-BIT TIMER (Cont'd) INPUT CAPTURE 1 HIGH REGISTER (IC1HR)

Read Only

Reset Value: Undefined

This is an 8-bit read only register that contains the high part of the counter value (transferred by the input capture 1 event).

7				0	
MSB				LSB	

INPUT CAPTURE 1 LOW REGISTER (IC1LR)

Read Only

Reset Value: Undefined

This is an 8-bit read only register that contains the low part of the counter value (transferred by the input capture 1 event).

	7						0	
	MSB						LSB	
							_	
						19		
					, _C			
			~	00				
			21					
		10						
	S	0,						
\bigcirc	0							

OUTPUT COMPARE 1 HIGH REGISTER (OC1HR)

Read/Write

_

Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

7				0
MSB				LSB

OUTPUT COMPARE 1 LOW REGISTER (OC1LR)

Read/Write

Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.



<u>ل</u>رک

8-BIT TIMER (Cont'd)

Whatever the timer mode used (input capture, output compare, one pulse mode or PWM mode) an overflow occurs when the counter rolls over from FFh to 00h then:

- The TOF bit of the SR register is set.
- A timer interrupt is generated if:
 - TOIE bit of the CR1 register is set and
 - I bit of the CC register is cleared.

If one of these conditions is false, the interrupt revyan .vakenedi obsolete Production mains pending to be issued as soon as they are

Notes: The TOF bit is not cleared by accesses to ACTR register. The advantage of accessing the ACTR register rather than the CTR register is that it allows simultaneous use of the overflow function and reading the free running counter at random times (for example, to measure elapsed time) without the risk of clearing the TOF bit erroneously.

The timer is not affected by WAIT mode.

In HALT mode, the counter stops counting until the mode is exited. Counting then resumes from the previous count (MCU awakened by an interrupt) or from the reset count (MCU awakened by a Reset).



8-BIT TIMER (Cont'd)

10.5.3.2 Input Capture

In this section, the index, *i*, may be 1 or 2 because there are two input capture functions in the 8-bit timer.

The two 8-bit input capture registers (IC1R and IC2R) are used to latch the value of the free running counter after a transition is detected on the ICAP*i* pin (see Figure 63).

IC*i*R register is a read-only register.

The active transition is software programmable through the IEDG*i* bit of Control Registers (CR*i*).

Timing resolution is one count of the free running counter (see Table 19 Clock Control Bits).

Procedure:

To use the input capture function select the following in the CR2 register:

- Select the timer clock (CC[1:0]) (see Table 19 Clock Control Bits).
- Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit (the ICAP2 pin must be configured as floating input or input with pull-up without interrupt if this configuration is available).

And select the following in the CR1 register:

- Set the ICIE bit to generate an interrupt after an input capture coming from either the ICAP1 pin or the ICAP2 pin
- Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin must be configured as floating input or input with pull-up without interrupt if this configuration is available).

When an input capture occurs:

- ICF*i* bit is set.
- The IC*i*R register contains the value of the free running counter on the active transition on the ICAP*i* pin (see Figure 64).
- A timer interrupt is generated if the ICIE bit is set and the interrupt mask is cleared in the CC register. Otherwise, the interrupt remains pending until both conditions become true.

Clearing the Input Capture interrupt request (that is, clearing the ICF*i* bit) is done in two steps:

- 1. Reading the SR register while the ICF*i* bit is set.
- 2. An access (read or write) to the ICiR register.

Notes:

1. The IC/R register contains the free running counter value which corresponds to the most recent input capture.

2. The two input capture functions can be used together even if the timer also uses the two output compare functions.

3. Once the ICIE bit is set both input capture features may trigger interrupt requests. If only one is needed in the application, the interrupt routine software needs to discard the unwanted capture interrupt. This can be done by checking the ICF1 and ICF2 flags and resetting them both.

4. In One pulse Mode and PWM mode only Input Capture 2 can be used.

5. The alternate inputs (ICAP1 and ICAP2) are always directly connected to the timer. So any transitions on these pins activates the input capture function.

Moreover if one of the ICAP*i* pins is configured as an input and the second one as an output, an interrupt can be generated if the user toggles the output pin and if the ICIE bit is set.

6. The TOF bit can be used with interrupt generation in order to measure events that go beyond the timer range (FFh).



LINSCI[™] SERIAL COMMUNICATION INTERFACE (SCI Mode) (cont^{*}d) Figure 79. SCI Baud Rate and Extended Prescaler Block Diagram

57

129/265

LINSCITM SERIAL COMMUNICATION INTERFACE (LIN Master Only) (Cont'd)

10.8.4.7 Parity control

Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the PCE bit in the SCICR1 register. Depending on the frame length defined by the M bit, the possible SCI frame formats are as listed in Table 24.

Table 25. Frame Formats

M bit	PCE bit	SCI frame
0	0	SB 8 bit data STB
0	1	SB 7-bit data PB STB
1	0	SB 9-bit data STB
I	1	SB 8-bit data PB STB

Legend:

SB: Start Bit

STB: Stop Bit

PB: Parity Bit

Note: In case of wake up by an address mark, the MSB bit of the data is taken into account and not the parity bit

Even parity: The parity bit is calculated to obtain an even number of "1s" inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit is 0 if even parity is selected (PS bit = 0).

Odd parity: The parity bit is calculated to obtain an odd number of "1s" inside the frame made of the 7 or 8 LSB bits (depending on whether M is equal to 0 or 1) and the parity bit.

Example: data = 00110101; 4 bits set => parity bit is 1 if odd parity is selected (PS bit = 1).

<u>**Transmission mode:**</u> If the PCE bit is set then the MSB bit of the data written in the data register is not transmitted but is changed by the parity bit.

<u>Reception mode:</u> If the PCE bit is set then the interface checks if the received data byte has an

even number of "1s" if even parity is selected (PS = 0) or an odd number of "1s" if odd parity is selected (PS = 1). If the parity check fails, the PE flag is set in the SCISR register and an interrupt is generated if PIE is set in the SCICR1 register.

10.8.5 Low Power Modes

Mode	Description
	No effect on SCI.
WAIT	SCI interrupts cause the device to exit from Wait mode.
	SCI registers are frozen.
HALT	In Halt mode, the SCI stops transmitting/re- ceiving until Halt mode is exited.

10.8.6 Interrupts

Interrupt Event	Event Flag	Enable Control Bit	Exit from Wait	Exit from Halt
Transmit Data Register Empty	TDRE	TIE		
Transmission Com- plete	тс	TCIE		
Received Data Ready to be Read	RDRF	DIE	Yes	No
Overrun Error Detect- ed	OR	111		
Idle Line Detected	IDLE	ILIE		
Parity Error	PE	PIE		

The SCI interrupt events are connected to the same interrupt vector.

These events generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).



Figure 103. Filtering Mechanism - example



The example above shows the filtering principle of the beCAN. On reception of a message, the identifier is compared first with the filters configured in identifier list mode. If there is a match, the message is stored in the FIFO and the index of the matching filter is stored in the Filter Match Index. As shown in the example, the identifier matches with Identifier #2 thus the message content and MFMI 2 is stored in the FIFO. If there is no match, the incoming identifier is then compared with the filters configured in mask mode.

If the identifier does not match any of the identifiers configured in the filters, the message is discarded by hardware without software intervention.



10.9.4.4 Message Storage

The interface between the software and the hardware for the CAN messages is implemented by means of mailboxes. A mailbox contains all information related to a message; identifier, data, control and status information.

Transmit Mailbox

The software sets up the message to be transmitted in an empty transmit mailbox. The status of the transmission is indicated by hardware in the MCSR register.

Offset to Transmit Mailbox base ad- dress (bytes)	Register Name
0	MCSR
1	MDLC
2	MIDR0
3	MIDR1
4	MIDR2
5	MIDR3
6	MDAR0
7	MDAR1
8	MDAR2
9	MDAR3
10	MDAR4
11	MDAR5
12	MDAR6
13	MDAR7
14	Reserved
	Reserved

Transmit Mailbox Mapping

Receive Mailbox

When a message has been received, it is available to the software in the FIFO output mailbox. Once the software has handled the message (e.g. read it) the software must release the FIFO output mailbox by means of the RFOM bit in the CRFR register to make the next incoming message available. The filter match index is stored in the MFMI register.

Receive Mailbox Mapping

Offset to Receive Mailbox base ad- dress (bytes)	Register Name
0	MFML
1	MDLC
2	MIDR0
3	MIDR1
4	MIDR2
5	MIDR3
6	MDAR0
7	MDAR1
8	MDAR2
9	MDAR3
10	MDAR4
11	MDAR5
12	MDAR6
13	MDAR7
14	Reserved
15	Reserved

- The FIFO interrupt can be generated by the following events:
 - Reception of a new message, FMP bits in the CRFR0 register incremented.
 - FIFO0 full condition, FULL bit in the CRFR0 register set.
 - FIFO0 overrun condition, FOVR bit in the CRFR0 register set.
- The transmit, error and status change interrupt can be generated by the following events:
 - Transmit mailbox 0 becomes empty, RQCP0 bit in the CTSR register set.
 - Transmit mailbox 1 becomes empty, RQCP1 bit in the CTSR register set.
 - Error condition, for more details on error conditions please refer to the CAN Error Status register (CESR).
- Wake-up condition, SOF monitored on the done by s

CAN Rx signal.

10.9.6 Register Access Protection

Erroneous access to certain configuration registers can cause the hardware to temporarily disturb the whole CAN network. Therefore the following registers can be modified by software only while the hardware is in initialization mode:

CBTR0, CBTR1, CFCR0, CFCR1, CFMR and CDGR registers.

Although the transmission of incorrect data will not cause problems at the CAN network level, it can severely disturb the application. A transmit mailbox can be only modified by software while it is in empty state (refer to Figure 7. Transmit Mailbox States).

The filters must be deactivated before their value can be modified by software. The modification of the filter configuration (scale or mode) can be done by software only in initialization mode.



10.9.7 BeCAN Cell Limitations

10.9.7.1 FIFO Corruption

FIFO corruption occurs in the following case:

WHEN the beCAN RX FIFO already holds two messages (that is, FMP == 2)

AND the application releases the FIFO (with the instruction CRFR = B RFOM;)

WHILE the beCAN requests the transfer of a new receive message into the FIFO (this lasts one CPU cycle)

THEN the internal FIFO pointer is not updated

BUT the FMP bits are updated correctly



As the FIFO pointer is not updated correctly, this causes the last message received to be overwritten by any incoming message. This means one message is lost as shown in the example in Figure 16. The beCAN will not recover normal operation until a device reset occurs.



Side-effect of Workround 1

Because the while loop lasts 10 CPU cycles, at high baud rate, it is possible to miss a dominant state on the bus if it lasts just one CAN bit time and the bus speed is high enough (see Table 1).

Table 29. While Loop Timing

f _{CPU}	Software timing:	Minimum baud rate for possible missed dominant bit				
8 MHz	1.25 µs	800 Kbaud				
4 MHz	2.5 µs	400 Kbaud				
f _{CPU}	10/f _{CPU}	f _{CPU} /10				

If this happens, we will continue waiting in the while loop instead of releasing the FIFO immediately. The workaround is still valid because we will not release the FIFO during the critical period. But the application may lose additional time waiting in the while loop as we are no longer able to guarantee a maximum of 6 CAN bit times spent in the workaround.

In this particular case the time the application can spend in the workaround may increase up to a full CAN frame, depending of the frame contents. This

Figure 113. Reception at Maximum CAN Baud Rate

case is very rare but happens when a specific sequence is present on in the CAN frame.

The example in Figure 20 shows reception at maximum CAN baud rate: In this case t_{CAN} is $8/f_{CPU}$ and the sampling time is $10/f_{CPU}$.

If the application is using the maximum baud rate and the possible delay caused by the workaround is not acceptable, there is another workaround which reduces the Rx pin sampling time.

Workaround 2 (see Figure 21) first tests that FMP = 2 and the CAN cell is receiving, if not the FIFO can be released immediately. If yes, the program goes through a sequence of test instructions on the RX pin that last longer than the time between the acknowledge dominant bit and the critical time slot. If the Rx pin is in recessive state for more than 8 CAN bit times, it means we are now after the acknowledge and the critical slot. If a dominant bit is read on the bus, we can release the FIFO immediately. This workaround has to be written in assembly language to avoid the compiler optimizing the test sequence.

The implementation shown here is for the CAN bus maximum speed (1 Mbaud @ 8 MHz CPU clock).





10.10 10-BIT A/D CONVERTER (ADC)

10.10.1 Introduction

The on-chip Analog to Digital Converter (ADC) peripheral is a 10-bit, successive approximation converter with internal sample and hold circuitry. This peripheral has up to 16 multiplexed analog input channels (refer to device pin out description) that allow the peripheral to convert the analog voltage levels from up to 16 different sources.

The result of the conversion is stored in a 10-bit Data Register. The A/D converter is controlled through a Control/Status Register.

10.10.2 Main Features

- 10-bit conversion
- Up to 16 channels with multiplexed input
- Linear successive approximation
- Data register (DR) which contains the results
- Conversion complete status flag
- On/off bit (to reduce consumption)

The block diagram is shown in Figure 116.

Figure 116. ADC Block Diagram

10.10.3 Functional Description

10.10.3.1 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 AD-CDRL 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 allotted time.



12.4 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).

Symbol	Deveneter	Conditions	Flash [Devices	ROM	l lmit		
Symbol	Parameter	Conditions	Typ ¹⁾	Max ²⁾	Typ ¹	Max ²⁾	Unit	
I _{DD}	Supply current in RUN mode ³⁾	$ \begin{array}{l} f_{OSC} = 2 \ \text{MHz}, \ f_{CPU} = 1 \ \text{MHz} \\ f_{OSC} = 4 \ \text{MHz}, \ f_{CPU} = 2 \ \text{MHz} \\ f_{OSC} = 8 \ \text{MHz}, \ f_{CPU} = 4 \ \text{MHz} \\ f_{OSC} = 16 \ \text{MHz}, \ f_{CPU} = 8 \ \text{MHz} \end{array} $	1.8 3.2 6 10	3 5 8 15	1.1 2.2 4.4 8.9	2 3.5 6 12	mA	
	Supply current in SLOW mode ³⁾		0.5 0.6 0.85 1.25	2.7 3 3.6 4	0.1 0.2 0.4 0.8	0.2 0.4 0.8 1.5		
	Supply current in WAIT mode ³⁾	$ \begin{array}{l} f_{OSC} = 2 \ \text{MHz}, \ f_{CPU} = 1 \ \text{MHz} \\ f_{OSC} = 4 \ \text{MHz}, \ f_{CPU} = 2 \ \text{MHz} \\ f_{OSC} = 8 \ \text{MHz}, \ f_{CPU} = 4 \ \text{MHz} \\ f_{OSC} = 16 \ \text{MHz}, \ f_{CPU} = 8 \ \text{MHz} \end{array} $	1 1.8 3.4 6.4	3 4 5 7	0.7 1.4 2.9 5.7	3 4 5 7		
	Supply current in SLOW WAIT mode ²⁾		0.4 0.5 0.6 0.8	1.2 1.3 1.8 2	0.07 0.14 0.28 0.56	0.12 0.25 0.5 1		
	Supply current in HALT mode ⁴⁾	$V_{DD} = 5.5V \frac{-40^{\circ}C \le T_{A} \le +85^{\circ}C}{-40^{\circ}C \le T_{A} \le +125^{\circ}C}$	<1	10 50	<1	10 50	μA	
	Supply current in ACTIVE HALT mode ⁴⁾⁵⁾		0.5	1.2	0.18	0.25	mA	
	Supply current in AWUFH	$V_{DD} = 5.5V$ $-40^{\circ}C \le T_A \le +85^{\circ}C$	25	30	25	30	μA	
	mode ^{-1,3)}	-40°C ≤ T _A ≤ +125°C		70	-	70		

Notes:

1. Typical data are based on T_A = 25°C, V_{DD} = 5V (4.5V $\leq V_{DD} \leq$ 5.5V range).

2. Data based on characterization results, tested in production at V_{DD} max., f_{CPU} max. and T_A max.

3. Measurements are done in the following conditions:

- Program executed from Flash, CPU running with Flash (for flash devices).
- All I/O pins in input mode with a static value at $V_{DD} \, \text{or} \, V_{SS}$ (no load)

- All peripherals in reset state.

- Clock input (OSC1) driven by external square wave.

- In SLOW and SLOW WAIT mode, f_{CPU} is based on f_{OSC} divided by 32.

To obtain the total current consumption of the device, add the clock source (Section 12.5.3) and the peripheral power consumption (Section 12.4.2).

4. All I/O pins in input mode with a static value at V_{DD} or V_{SS} (no load). Data based on characterization results, tested in production at V_{DD} max., f_{CPU} max. and T_A max.

5. This consumption refers to the Halt period only and not the associated run period which is software dependent.

<u>/</u>ک

CONTROL PIN CHARACTERISTICS (Cont'd)

Figure 134. RESET Pin Protection When LVD Is Enabled¹⁾²⁾



Figure 135. RESET Pin Protection When LVD Is Disabled¹⁾



Note 1:

1.1 The reset network protects the device against parasitic resets.

1.2 The output of the external reset circuit must have an open-drain output to drive the ST7 reset pad. Otherwise the device can be damaged when the ST7 generates an internal reset (LVD or watchdog).

1.3 Whatever the reset source is (internal or external), the user must ensure that the level on the $\overline{\text{RESET}}$ pin can go below the V_{IL} max. level specified in Section 12.10.1. Otherwise the reset will not be taken into account internally.

1.4 Because the reset circuit is designed to allow the internal RESET to be output in the RESET pin, the user must ensure that the current sunk on the RESET pin (by an external pull-up for example) is less than the absolute maximum value specified for I_{INJ(RESET)} in Section 12.2.2 on page 220.

Note 2:

2.1 When the LVD is enabled, it is mandatory not to connect a pull-up resistor. A 10nF pull-down capacitor is recommended to filter noise on the reset line.

2.2. In case a capacitive power supply is used, it is recommended to connect a1MW pull-down resistor to the $\overrightarrow{\text{RESET}}$ pin to discharge any residual voltage induced by this capacitive power supply (this will add 5µA to the power consumption of the MCU).

2.3. Tips when using the LVD:

- 1. Check that all recommendations related to reset circuit have been applied (see notes above)
- 2. Check that the power supply is properly decoupled (100nF + 10µF close to the MCU). Refer to AN1709. If this cannot be done, it is recommended to put a 100nF + 1MW pull-down on the RESET pin.
- 3. The capacitors connected on the RESET pin and also the power supply are key to avoiding any start-up marginality. In most cases, steps 1 and 2 above are sufficient for a robust solution. Otherwise: Replace 10nF pull-down on the RESET pin with a 5µF to 20µF capacitor.



14 DEVICE CONFIGURATION AND ORDERING INFORMATION

Each device is available for production in user programmable versions (FLASH) as well as in factory coded versions (ROM/FASTROM).

ST72561 devices are ROM versions. ST72P561 devices are Factory Advanced Service Technique ROM (FASTROM) versions: They are factory-programmed HDFlash devices.

ST72F561 FLASH devices are shipped to customers with a default content (FFh), while ROM factory coded parts contain the code supplied by the customer. This implies that FLASH devices have to be configured by the customer using the Option Bytes while the ROM devices are factory-configured.

14.1 FLASH OPTION BYTES

The option bytes allows the hardware configuration of the microcontroller to be selected. They have no address in the memory map and can be accessed only in programming mode (for example using a standard ST7 programming tool). The default content of the FLASH is fixed to FFh. To program directly the FLASH devices using ICP, FLASH devices are shipped to customers with a reserved internal clock source enabled. In masked ROM devices, the option bytes are fixed in hardware by the ROM code (see option list).

OPTION BYTE 0

OPT7 = **WDGHALT** Watchdog reset on HALT This option bit determines if a RESET is generated when entering HALT mode while the Watchdog is active.

- 0: No Reset generation when entering Halt mode
- 1: Reset generation when entering Halt mode

OPT6 = **WDGSW** *Hardware or software watchdog* This option bit selects the watchdog type. 0: Hardware (watchdog always enabled)

1: Software (watchdog to be enabled by software)

OPT5 = Reserved, must be kept at default value.

OPT4 = **LVD** Voltage detection This option bit enables the voltage detection block (LVD).

Selected Low Voltage Detector	VD
LVD Off	1
LVD On	0

OPT3 = PLL OFF PLL activation

This option bit activates the PLL which allows multiplication by two of the main input clock frequency. The PLL is guaranteed only with an input frequency between 2 and 4 MHz.

0: PLL x2 enabled

1: PLL x2 disabled

Caution: The PLL can be enabled only if the "OSC RANGE" (OPT11:10) bits are configured to "MP - 2~4 MHz". Otherwise, the device functionality is not guaranteed.

		STATIC OPTION BYTE 0 7 0									STATIC OPTION BYTE 1						
											7						0
WDG					PKG 🕰		AFI_MAP		OSCTYPE		OSCRANGE		irved	тс			
	02	HALT	SW	Rese	LVD	PLLO	1	0	FMF	1	0	1	0	1	0	Rese	RS
	De- fault(*)	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1

(*): Option bit values programmed by ST

