



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	24
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 6x10b
Oscillator Type	External
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/st72f561k9ta

Email: info@E-XFL.COM

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

1 DESCRIPTION

The ST72561 devices are members of the ST7 microcontroller family designed for mid-range applications with CAN (Controller Area Network) and LIN (Local Interconnect Network) interface.

All devices are based on a common industrystandard 8-bit core, featuring an enhanced instruction set and are available with Flash or ROM program memory. The ST7 family architecture offers both power and flexibility to software developers, enabling the design of highly efficient and compact application code. The on-chip peripherals include an A/D converter, a PWM Autoreload timer, 2 general purpose timers, 2 asynchronous serial interfaces, and an SPI interface.

For power economy, microcontroller can switch dynamically into WAIT, SLOW, Active-Halt, Auto Wake-up from HALT (AWU) or HALT mode when the application is in idle or stand-by state.

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

<u>لرکم</u>



CENTRAL PROCESSING UNIT (Cont'd)

Stack Pointer (SP)

Read/Write

Reset Value: 01 FFh



The Stack Pointer is a 16-bit register which is always pointing to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack (see Figure 9).

Since the stack is 256 bytes deep, the 8 most significant bits are forced by hardware. Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer contains its reset value (the SP7 to SP0 bits are set) which is the stack higher address.

Figure 9. Stack Manipulation Example

The least significant byte of the Stack Pointer (called S) can be directly accessed by a LD instruction.

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

The stack is used to save the return address during a subroutine call and the CPU context during an interrupt. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt, the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations as shown in Figure 9.

- 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.

47/



6.3 RESET SEQUENCE MANAGER (RSM)

6.3.1 Introduction

The reset sequence manager includes three RE-SET sources as shown in Figure 2:

- External RESET source pulse
- Internal LVD RESET (Low Voltage Detection)
- Internal WATCHDOG RESET

These sources act on the RESET pin and it is always kept low during the delay phase.

The RESET service routine vector is fixed at addresses FFFEh-FFFFh in the ST7 memory map.

The basic RESET sequence consists of three phases as shown in Figure 1:

- Active Phase depending on the RESET source
- 256 or 4096 CPU clock cycle delay (selected by option byte)
- RESET vector fetch

<u>ل</u>رک

The 256 or 4096 CPU clock cycle delay allows the oscillator to stabilize and ensures that recovery has taken place from the Reset state. The shorter or longer clock cycle delay should be selected by option byte to correspond to the stabilization time of the external oscillator used in the application.

The RESET vector fetch phase duration is two clock cycles.



Figure 12. RESET Sequence Phases



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

6.3.2 Asynchronous External RESET pin

The $\overline{\text{RESET}}$ pin is both an input and an open-drain output with integrated R_{ON} weak pull-up resistor. This pull-up has no fixed value but varies in accordance with the input voltage. It can be pulled low by external circuitry to reset the device. See Electrical Characteristic section for more details.

A RESET signal originating from an external source must have a duration of at least $t_{h(RSTL)in}$ in order to be recognized (see Figure 3). This detection is asynchronous and therefore the MCU can enter reset state even in HALT mode.



6.4 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.

6.4.1 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 poweron 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

لركم

The LVD function is illustrated in Figure 15.

Figure 15. Low Voltage Detector vs Reset

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 RESET pin is held low, thus permitting the MCU to reset other devices.

Notes:

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

The LVD is an optional function which can be selected by option byte.

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.



SYSTEM INTEGRITY MANAGEMENT (Cont'd)

6.4.2 Auxiliary Voltage Detector (AVD)

The Voltage Detector function (AVD) is based on an analog comparison between a $V_{IT-(AVD)}$ and $V_{IT+(AVD)}$ reference value and the V_{DD} main supply. The $V_{IT-(AVD)}$ reference value for falling voltage is lower than the $V_{IT+(AVD)}$ reference value for rising voltage in order to avoid parasitic detection (hysteresis).

The output of the AVD comparator is directly readable by the application software through a real time status bit (AVDF) in the SICSR register. This bit is read only.

Caution: The AVD function is active only if the LVD is enabled through the option byte.

6.4.2.1 Monitoring the V_{DD} Main Supply

If the AVD interrupt is enabled, an interrupt is generated when the voltage crosses the $V_{IT+(AVD)}$ or $V_{IT-(AVD)}$ threshold (AVDF bit toggles).

In the case of a drop in voltage, the AVD interrupt acts as an early warning, allowing software to shut

Figure 16. Using the AVD to Monitor V_{DD}

down safely before the LVD resets the microcontroller. See Figure 16.

The interrupt on the rising edge is used to inform the application that the V_{DD} warning state is over.

If the voltage rise time t_{rv} is less than 256 or 4096 CPU cycles (depending on the reset delay selected by option byte), no AVD interrupt will be generated when $V_{\rm IT+(AVD)}$ is reached.

If t_{rv} is greater than 256 or 4096 cycles then:

- If the AVD interrupt is enabled before the $V_{IT+(AVD)}$ threshold is reached, then two AVD interrupts will be received: The first when the AVDIE bit is set and the second when the threshold is reached.
- If the AVD interrupt is enabled after the V_{IT+(AVD)} threshold is reached, then only one AVD interrupt occurs.



INTERRUPTS (Cont'd)

Instruction	New Description	Function/Example	11	н	10	Ν	z	С
HALT	Entering Halt mode		1		0			
IRET	Interrupt routine return	Pop CC, A, X, PC	11	Н	10	Ν	Z	С
JRM	Jump if I1:0 = 11 (level 3)	l1:0 = 11 ?						
JRNM	Jump if I1:0 <> 11	1:0 <> 11 ?						
POP CC	Pop CC from the Stack	Mem => CC	11	Н	10	Ν	Z	С
RIM	Enable interrupt (level 0 set)	Load 10 in I1:0 of CC	1		0		S	
SIM	Disable interrupt (level 3 set)	Load 11 in I1:0 of CC	1		1	X		
TRAP	Software trap	Software NMI	1		1	Ś	1	
WFI	Wait for interrupt		1		0	5		

Table 8. Dedicated Interrupt Instruction Set

Note: During the execution of an interrupt routine, the HALT, POPCC, RIM, SIM and WFI instructions change the current software priority up to the next IRET instruction or one of the previously mentioned instructions.

57

POWER SAVING MODES (Cont'd)

8.3 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[1:0] bits of the CC register are forced to '10', 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.

Jbsolete Producils

Refer to Figure 24

OSCILLATOR ON PERIPHERALS ON WFI INSTRUCTION CPU OFF I[1:0] BITS 10 Ν RESET Y Ν INTERRUPT Y OSCILLATOR ON PERIPHERALS OFF CPU ON I[1:0] BITS 10 256 OR 4096 CPU CLOCK CYCLE DELAY OSCILLATOR ON PERIPHERALS ON CPU ON I[1:0] BITS XX 1) FETCH RESET VECTOR **OR SERVICE INTERRUPT**

Note:

1. Before servicing an interrupt, the CC register is pushed on the stack. The I[1:0] bits of the CC register are set to the current software priority level of the interrupt routine and recovered when the CC register is popped.

Figure 24. WAIT Mode Flow-chart



16-BIT TIMER (Cont'd)

Figure 48. Timer Block Diagram



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:

$$OC_{i}R Value = \frac{t \cdot t_{CPU}}{PRESC} - 5$$

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.

8-BIT TIMER (Cont'd) INPUT CAPTURE 1 REGISTER (IC1R)

Read Only

_

Reset Value: Undefined

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

7				0	
MSB				LSB	

OUTPUT COMPARE 1 REGISTER (OC1R)

Read/Write

Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the value to be compared to the CTR register.

7				0	
MSB				LSB	

OUTPUT COMPARE 2 REGISTER (OC2R)

Read/Write

Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the value to be compared to the CTR register.



COUNTER REGISTER (CTR)

Read Only

Reset Value: 1111 1100 (FCh)

This is an 8-bit register that contains the counter value. A write to this register resets the counter. An access to this register after accessing the CSR register clears the TOF bit.

7				0
MSB				LSB

ALTERNATE COUNTER REGISTER (ACTR)

Read Only Reset Value: 1111 1100 (FCh)

This is an 8-bit register that contains the counter value. A write to this register resets the counter. An access to this register after an access to CSR register does not clear the TOF bit in the CSR register.

9				0
MSB				LSB

INPUT CAPTURE 2 REGISTER (IC2R)

Read Only

Reset Value: Undefined

This is an 8-bit read only register that contains the counter value (transferred by the Input Capture 2 event).

7				0
MSB				LSB



LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Mode) (cont'd)

10.7.9.3 LIN Reception

In LIN mode the reception of a byte is the same as in SCI mode but the LINSCI has features for handling the LIN Header automatically (identifier detection) or semiautomatically (Synch Break detection) depending on the LIN Header detection mode. The detection mode is selected by the LHDM bit in the SCICR3.

Additionally, an automatic resynchronization feature can be activated to compensate for any clock deviation, for more details please refer to Section 0.1.9.5 LIN Baud Rate.

LIN Header Handling by a Slave

Depending on the LIN Header detection method the LINSCI will signal the detection of a LIN Header after the LIN Synch Break or after the Identifier has been successfully received.

Note:

It is recommended to combine the Header detection function with Mute mode. Putting the LINSCI in Mute mode allows the detection of Headers only and prevents the reception of any other characters.

This mode can be used to wait for the next Header without being interrupted by the data bytes of the current message in case this message is not relevant for the application.

Synch Break Detection (LHDM = 0):

When a LIN Synch Break is received:

- The RDRF bit in the SCISR register is set. It indicates that the content of the shift register is transferred to the SCIDR register, a value of 0x00 is expected for a Break.
- The LHDF flag in the SCICR3 register indicates that a LIN Synch Break Field has been detected.

 An interrupt is generated if the LHIE bit in the SCICR3 register is set and the I[1:0] bits are cleared in the CCR register.

- Then the LIN Synch Field is received and measured.
 - If automatic resynchronization is enabled (LA-SE bit = 1), the LIN Synch Field is not transferred to the shift register: There is no need to clear the RDRF bit.
 - If automatic resynchronization is disabled (LA-SE bit = 0), the LIN Synch Field is received as a normal character and transferred to the SCIDR register and RDRF is set.

Note:

In LIN slave mode, the FE bit detects all frame error which does not correspond to a break.

Identifier Detection (LHDM = 1):

This case is the same as the previous one except that the LHDF and the RDRF flags are set only after the entire header has been received (this is true whether automatic resynchronization is enabled or not). This indicates that the LIN Identifier is available in the SCIDR register.

Notes:

During LIN Synch Field measurement, the SCI state machine is switched off: No characters are transferred to the data register.

LIN Slave parity

In LIN Slave mode (LINE and LSLV bits are set) LIN parity checking can be enabled by setting the PCE bit.

In this case, the parity bits of the LIN Identifier Field are checked. The identifier character is recognized as the third received character after a break character (included):



The bits involved are the two MSB positions (7th and 8th bits if M = 0; 8th and 9th bits if M = 1) of the identifier character. The check is performed as specified by the LIN specification:



47/

LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Mode) (cont'd)

10.7.9.4 LIN Error Detection

LIN Header Error Flag

The LIN Header Error Flag indicates that an invalid LIN Header has been detected.

When a LIN Header Error occurs:

- The LHE flag is set
- An interrupt is generated if the RIE bit is set and the I[1:0] bits are cleared in the CCR register.

If autosynchronization is enabled (LASE bit = 1), this can mean that the LIN Synch Field is corrupted, and that the SCI is in a blocked state (LSF bit is set). The only way to recover is to reset the LSF bit and then to clear the LHE bit.

- The LHE bit is reset by an access to the SCISR register followed by a read of the SCIDR register.

LHE/OVR Error Conditions

When Auto Resynchronization is disabled (LASE bit = 0), the LHE flag detects:

- That the received LIN Synch Field is not equal to 55h.
- That an overrun occurred (as in standard SCI mode)
- Furthermore, if LHDM is set it also detects that a LIN Header Reception Timeout occurred (only if LHDM is set).

When the LIN auto-resynchronization is enabled (LASE bit = 1), the LHE flag detects:

- That the deviation error on the Synch Field is outside the LIN specification which allows up to +/-15.5% of period deviation between the slave and master oscillators.
- A LIN Header Reception Timeout occurred.
 If T_{HEADER} > T_{HEADER_MAX} then the LHE flag is set. Refer to Figure 6. (only if LHDM is set to 1)
- An overflow during the Synch Field Measurement, which leads to an overflow of the divider registers. If LHE is set due to this error then the SCI goes into a blocked state (LSF bit is set).
- That an overrun occurred on Fields other than the Synch Field (as in standard SCI mode)

Deviation Error on the Synch Field

The deviation error is checking by comparing the current baud rate (relative to the slave oscillator) with the received LIN Synch Field (relative to the master oscillator). Two checks are performed in parallel:

 The first check is based on a measurement between the first falling edge and the last falling edge of the Synch Field. Let us refer to this period deviation as D:

If the LHE flag is set, it means that:

D > 15.625%

If LHE flag is not set, it means that:

D < 16.40625%

If $15.625\% \le D < 16.40625\%$, then the flag can be either set or reset depending on the dephasing between the signal on the RDI line and the CPU clock.

 The second check is based on the measurement of each bit time between both edges of the Synch Field: this checks that each of these bit times is large enough compared to the bit time of the current baud rate.

When LHE is set due to this error then the SCI goes into a blocked state (LSF bit is set).

LIN Header Time-out Error

When the LIN Identifier Field Detection Method is used (by configuring LHDM to 1) or when LIN auto-resynchronization is enabled (LASE bit = 1), the LINSCI automatically monitors the T_{HEADER_MAX} condition given by the LIN protocol.

If the entire Header (up to and including the STOP bit of the LIN Identifier Field) is not received within the maximum time limit of 57 bit times then a LIN Header Error is signalled and the LHE bit is set in the SCISR register.

Figure 82. LIN Header Reception Timeout



The time-out counter is enabled at each break detection. It is stopped in the following conditions:

- A LIN Identifier Field has been received

- An LHE error occurred (other than a timeout error).

- A software reset of LSF bit (transition from high to low) occurred during the analysis of the LIN Synch Field or

If LHE bit is set due to this error during the LIN Synchr Field (if LASE bit = 1) then the SCI goes into a blocked state (LSF bit is set).

beCAN CONTROLLER (Cont'd)

10.9.4.2 Reception Handling

For the reception of CAN messages, three mailboxes organized as a FIFO are provided. In order to save CPU load, simplify the software and guarantee data consistency, the FIFO is managed completely by hardware. The application accesses the messages stored in the FIFO through the FIFO output mailbox.

A received message is considered as valid when it has been received correctly according to the CAN protocol (no error until the last but one bit of the EOF field) and It passed through the identifier filtering successfully, see Section 0.1.4.3 Identifier Filterina.

Valid Message

Figure 101. Receive FIFO states



beCAN CONTROLLER (Cont'd)

10.9.4.6 Bit Timing

The bit timing logic monitors the serial bus-line and performs sampling and adjustment of the sample point by synchronizing on the start-bit edge and resynchronizing on the following edges.

Its operation may be explained simply by splitting nominal bit time into three segments as follows:

- Synchronization segment (SYNC SEG): a bit change is expected to occur within this time segment. It has a fixed length of one time quantum (1 x t_{CAN}).
- Bit segment 1 (BS1): defines the location of the sample point. It includes the PROP_SEG and PHASE SEG1 of the CAN standard. Its duration is programmable between 1 and 16 time guanta but may be automatically lengthened to compensate for positive phase drifts due to differences in the frequency of the various nodes of the network.
- Bit segment 2 (BS2): defines the location of the transmit point. It represents the PHASE SEG2 of the CAN standard. Its duration is programmable between 1 and 8 time quanta but may also be automatically shortened to compensate for negative phase drifts.
- Resynchronization Jump Width (RJW): defines an upper bound to the amount of lengthening or shortening of the bit segments. It is programmable between 1 and 4 time quanta.

To guarantee the correct behaviour of the CAN controller, SYNC_SEG + BS1 + BS2 must be greater than or equal to 5 time quanta.

For a detailed description of the CAN resynchronization mechanism and other bit timing configuration constraints, please refer to the Bosch CAN standard 2.0.

As a safeguard against programming errors, the configuration of the Bit Timing Registers CBTR1 and CBTR0 is only possible while the device is in Initialization mode.



Figure 105. Bit Timing

<u>لرک</u>

Figure 106. CAN Frames (Part 1of 2)



beCAN CONTROLLER (Cont'd)

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: While loop	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).





SUPPLY CURRENT CHARACTERISTICS (Cont'd)

12.4.1 Supply and Clock Managers

The previous current consumption specified for the ST7 functional operating modes over temperature range does not take into account the clock source current consumption. To obtain the total device consumption, the two current values must be added (except for HALT mode).

Symbol	Parameter	Conditions	Тур	Max ¹⁾	Unit
I _{DD(RES)}	Supply current of resonator oscillator ²⁾³⁾		See Section 12.	5.3 on page 227	
I _{DD(PLL)}	PLL supply current	$V_{DD} = 5V$	360		μA
I _{DD(LVD)}	LVD supply current	HALT mode, $V_{DD} = 5V$	150	300	

Notes:

1. Data based on characterization results, not tested in production.

2. Data based on characterization results done with the external components specified in Section 12,5.3, not tested in and on the obsolete Product (S) - Obsolete Pr production.

3. As the oscillator is based on a current source, the consumption does not depend on the voltage.



I/O PORT PIN CHARACTERISTICS (Cont'd)

Figure 133. Typical V_{OH} vs V_{DD}





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.



ADC CHARACTERISTICS (Cont'd)



Figure 143. Typical Application with ADC



Notes:

1. $C_{PARASITIC}$ represents the capacitance of the PCB (dependent on soldering and PCB layout quality) plus the pad capacitance (3pF). A high $C_{PARASITIC}$ value will downgrade conversion accuracy. To remedy this, f_{ADC} should be reduced. 2. Any added external serial resistor will downgrade the ADC accuracy (especially for resistance greater than 10k Ω). Data based on characterization results, not tested in production.

3. This graph shows that depending on the input signal variation (f_{AIN}), C_{AIN} can be increased for stabilization time and reduced to allow the use of a larger serial resistor (R_{AIN}). It is valid for all f_{ADC} frequencies ≤ 4 MHz.

ADC CHARACTERISTICS (Cont'd)

ADC Accuracy with f_{CPU} = 8 MHz, f_{ADC} = 4 MHz R_{AIN} < 10k $\Omega,\,V_{DD}$ = 5V

Symbol	Parameter	Conditions	Тур	Max	Unit
IE _T I	Total unadjusted error ¹⁾		3.2	5	
IE _O I	Offset error ¹⁾		1	1	
IE _G I	Gain Error ¹⁾		0.7	4	LSB
IE _D I	Differential linearity error ¹⁾		1.5	2.3	
IELI	Integral linearity error ¹⁾		1.2	3.6	



Notes:

1. Data based on characterization results, not tested in production. ADC Accuracy vs. Negative Injection Current: Injecting negative current on any of the standard (non-robust) analog input pins should be avoided as this significantly reduces the accuracy of the conversion being performed on another analog input. It is recommended to add a Schottky diode (pin to ground) to standard analog pins which may potentially inject negative current. The effect of negative injection current on robust pins is specified in Section 12.9.

Any positive injection current within the limits specified for $I_{INJ(PIN)}$ and $\Sigma I_{INJ(PIN)}$ in Section 12.9 does not affect the ADC accuracy.

