



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	48KB (48K 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/st72f561ar7t6

Email: info@E-XFL.COM

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

FLASH PROGRAM MEMORY (Cont'd)

4.4 ICC INTERFACE

ICC (In-Circuit Communication) needs a minimum of four and up to six pins to be connected to the programming tool (see Figure 7). These pins are:

- RESET: device reset
- V_{SS}: device power supply ground

Figure 7. Typical ICC Interface

- ICCCLK: ICC output serial clock pin
- ICCDATA: ICC input/output serial data pin
- ICCSEL/V_{PP}: programming voltage
- OSC1(or OSCIN): main clock input for external source (optional)
- V_{DD}: application board power supply (see Figure 7, Note 3)



Notes:

1. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the Programming Tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.

2. During the IC<u>C</u> session, the programming tool must control the RESET pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5mA at high level (push pull output or pull-up resistor<1K). A schottky diode can be used to isolate the application RESET circuit in this case. When using a classical RC network with R > 1K or a reset man-

agement IC with open drain output and pull-up resistor > 1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.

3. The use of Pin 7 of the ICC connector depends on the Programming Tool architecture. This pin must be connected when using most ST Programming Tools (it is used to monitor the application power supply). Please refer to the Programming Tool manual.

4. Pin 9 has to be connected to the OSC1 or OS-CIN pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte. ST7 devices with multi-oscillator capability need to have OSC2 grounded in this case.



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>/</u>ک

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.



INTERRUPTS (Cont'd)

7.3 INTERRUPTS AND LOW POWER MODES

All interrupts allow the processor to exit the WAIT low power mode. On the contrary, only external and other specified interrupts allow the processor to exit from the HALT modes (see column "Exit from HALT" in "Interrupt Mapping" table). When several pending interrupts are present while exiting HALT mode, the first one serviced can only be an interrupt with exit from HALT mode capability and it is selected through the same decision process shown in Figure 18.

Note: If an interrupt, that is not able to Exit from HALT mode, is pending with the highest priority when exiting HALT mode, this interrupt is serviced after the first one serviced.

Figure 19. Concurrent Interrupt Management

7.4 CONCURRENT & NESTED MANAGEMENT

The following Figure 19 and Figure 20 show two different interrupt management modes. The first is called concurrent mode and does not allow an interrupt to be interrupted, unlike the nested mode in Figure 20. The interrupt hardware priority is given in this order from the lowest to the highest: MAIN, IT4, IT3, IT2, IT1, IT0, TLI. The software priority is given for each interrupt.

Warning: A stack overflow may occur without notifying the software of the failure.



Figure 20. Nested Interrupt Management

<u>ل</u>رک



I/O PORTS (Cont'd)

9.6 I/O PORT REGISTER CONFIGURATIONS

The I/O port register configurations are summarized as follows.

9.6.1 Standard Ports

PB7:6, PC0, PC3, PC7:5, PD3:2, PD5, PE7:0, PF7:0

MODE	DDR	OR
floating input	0	0
pull-up input	0	1
open drain output	1	0
push-pull output	1	1

9.6.2 Interrupt Ports PA0,2,4,6; PB0,2,4; PC1; PD0,6

(with pull-up)

MODE	DDR	OR
floating input	0	0
pull-up interrupt input	0	1
open drain output	1	0
push-pull output	1	51
obsolete Produ		

PA1,3,5,7; PB1,3,5; PC2; PD1,4,7

(without pull-up)

MODE	DDR	OR
floating input	0	0
floating interrupt input	0	1
open drain output	1	0
push-pull output	1	1

9.6.3 Pull-up Input Port (CANTX requirement) PC4

MODE
pull-up input

The PC4 port cannot operate as a general purpose output. The CAN peripheral controls it directly when enabled. Otherwise, PC4 is a pull-up input.

If DDR = 1 it is still possible to read the port through the DR register.



PWM AUTO-RELOAD TIMER (Cont'd)

Input Capture Function

Input Capture mode allows the measurement of external signal pulse widths through ARTICRx registers.

Each input capture can generate an interrupt independently on a selected input signal transition. This event is flagged by a set of the corresponding CFx bits of the Input Capture Control/Status register (ARTICCSR).

These input capture interrupts are enabled through the CIEx bits of the ARTICCSR register.

The active transition (falling or rising edge) is software programmable through the CSx bits of the ARTICCSR register.

The read only input capture registers (ARTICRx) are used to latch the auto-reload counter value when a transition is detected on the ARTICx pin (CFx bit set in ARTICCSR register). After fetching the interrupt vector, the CFx flags can be read to identify the interrupt source.

Note: After a capture detection, data transfer in the ARTICRx register is inhibited until the next read (clearing the CFx bit).

The timer interrupt remains pending while the CFx flag is set when the interrupt is enabled (CIEx bit

set). This means, the ARTICRx register has to be read at each capture event to clear the CFx flag.

The timing resolution is given by auto-reload counter cycle time $(1/f_{COUNTER})$.

Note: During HALT mode, input capture is inhibited (the ARTICRx is never reloaded) and only the external interrupt capability can be used.

Note: The ARTICx signal is synchronized on CPU clock. It takes two rising edges until ARTICRx is latched with the counter value. Depending on the prescaler value and the time when the ICAP event occurs, the value loaded in the ARTICRx register may be different.

If the counter is clocked with the CPU clock, the value latched in ARTICRx is always the next counter value after the event on ARTICx occurred (Figure 45).

If the counter clock is prescaled, it depends on the position of the ARTICx event within the counter cycle (Figure 46).



Figure 45. Input Capture Timing Diagram, f_{COUNTER} = f_{CPU}

16-BIT TIMER (Cont'd)

10.4.3.4 Output Compare

In this section, the index, *i*, may be 1 or 2 because there are two output compare functions in the 16-bit timer.

This function can be used to control an output waveform or indicate when a period of time has elapsed.

When a match is found between the Output Compare register and the free running counter, the output compare function:

- Assigns pins with a programmable value if the OC*i*E bit is set
- Sets a flag in the status register
- Generates an interrupt if enabled

Two 16-bit registers Output Compare Register 1 (OC1R) and Output Compare Register 2 (OC2R) contain the value to be compared to the counter register each timer clock cycle.

	MS Byte	LS Byte
OC <i>i</i> R	OC <i>i</i> HR	OC <i>i</i> LR

These registers are readable and writable and are not affected by the timer hardware. A reset event changes the OC*i*R value to 8000h.

Timing resolution is one count of the free running counter: $(f_{CPU/CC[1:0]})$.

Procedure:

To use the output compare function, select the following in the CR2 register:

- Set the OC*i*E bit if an output is needed then the OCMP*i* pin is dedicated to the output compare *i* signal.
- Select the timer clock (CC[1:0]) (see Table 17 Clock Control Bits).

And select the following in the CR1 register:

- Select the OLVL*i* bit to applied to the OCMP*i* pins after the match occurs.
- Set the OCIE bit to generate an interrupt if it is needed.

When a match is found between OC*i*R register and CR register:

OCF*i* bit is set.

- The OCMP*i* pin takes OLVL*i* bit value (OCMP*i* pin latch is forced low during reset).
- A timer interrupt is generated if the OCIE bit is set in the CR1 register and the I bit is cleared in the CC register (CC).

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

$$\Delta \text{ OC}_{i}\text{R} = \frac{\Delta t * f_{CPU}}{\text{PRESC}}$$

Where:

1

- Δt = Output compare 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:

$$\Delta \text{ OC}_{i} = \Delta t \star f_{EXT}$$

Where:

 Δt = Output compare period (in seconds)

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

Clearing the output compare interrupt request (that is, clearing the OCF*i* bit) is done by:

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

The following procedure is recommended to prevent the OCF*i* bit from being set between the time it is read and the write to the OC*i*R register:

- Write to the OC*i*HR register (further compares are inhibited).
- Read the SR register (first step of the clearance of the OCF*i* bit, which may be already set).
- Write to the OCiLR register (enables the output compare function and clears the OCF*i* bit).

57

16-BIT TIMER (Cont'd)

OUTPUT COMPARE 2 HIGH REGISTER (OC2HR)

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 2 LOW REGISTER (OC2LR)

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.

7				0	
MSB				LSB	

COUNTER HIGH REGISTER (CHR)

Read Only

Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

7	000	0
MSB		LSB

COUNTER LOW REGISTER (CLR)

Read Only

Reset Value: 1111 1100 (FCh)

This is an 8-bit register that contains the low part of 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 HIGH REGISTER (ACHR)

Read Only

Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

7				0
MSB				LSB

ALTERNATE COUNTER LOW REGISTER (ACLR)

Read Only

Reset Value: 1111 1100 (FCh)

This is an 8-bit register that contains the low part of 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.

7	~			0
MSB				LSB

INPUT CAPTURE 2 HIGH REGISTER (IC2HR)

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 2 event).

7				0
MSB				LSB

INPUT CAPTURE 2 LOW REGISTER (IC2LR)

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 2 event).

1				0
MSB				LSB



SERIAL PERIPHERAL INTERFACE (cont'd)

10.6.3.3 Master Mode Operation

In master mode, the serial clock is output on the SCK pin. The clock frequency, polarity and phase are configured by software (refer to the description of the SPICSR register).

Note: The idle state of SCK must correspond to the polarity selected in the SPICSR register (by pulling up SCK if CPOL = 1 or pulling down SCK if CPOL = 0).

How to operate the SPI in master mode

To operate the SPI in master mode, perform the following steps in order:

- Write to the SPICR register:
 - Select the clock frequency by configuring the SPR[2:0] bits.
 - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits. Figure 74 shows the four possible configurations. **Note:** The slave must have the same CPOL and CPHA settings as the master.
- 2. Write to the SPICSR register:
 - Either set the SSM bit and set the SSI bit or clear the SSM bit and tie the SS pin high for the complete byte transmit sequence.
- 3. Write to the SPICR register:

 - Set the MSTR and SPE bits
 Note: MSTR and SPE bits remain set only if SS is high).

Important note: if the SPICSR register is not written first, the SPICR register setting (MSTR bit) may be not taken into account.

The transmit sequence begins when software writes a byte in the SPIDR register.

10.6.3.4 Master Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MOSI pin most significant bit first.

When data transfer is complete:

- The SPIF bit is set by hardware.
- An interrupt request is generated if the SPIE bit is set and the interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

- 1. An access to the SPICSR register while the SPIF bit is set
- 2. A read to the SPIDR register

Note: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

10.6.3.5 Slave Mode Operation

In slave mode, the serial clock is received on the SCK pin from the master device.

To operate the SPI in slave mode:

- 1. Write to the SPICSR register to perform the following actions:
 - Select the clock polarity and clock phase by configuring the CPOL and CPHA bits (see **Note:** The slave must have the same CPOL and CPHA settings as the master.
 - Manage the \overline{SS} pin as described in Section 10.6.3.2 and Figure 72. If CPHA = 1 SS must be held low continuously. If CPHA = 0 SS must be held low during byte transmission and pulled up between each byte to let the slave write in the shift register.
- 2. Write to the SPICR register to clear the MSTR bit and set the SPE bit to enable the SPI I/O functions.

10.6.3.6 Slave Mode Transmit Sequence

When software writes to the SPIDR register, the data byte is loaded into the 8-bit shift register and then shifted out serially to the MISO pin most significant bit first.

The transmit sequence begins when the slave device receives the clock signal and the most significant bit of the data on its MOSI pin.

When data transfer is complete:

- The SPIF bit is set by hardware.
- An interrupt request is generated if SPIE bit is set and interrupt mask in the CCR register is cleared.

Clearing the SPIF bit is performed by the following software sequence:

- 1. An access to the SPICSR register while the SPIF bit is set
- A write or a read to the SPIDR register

Notes: While the SPIF bit is set, all writes to the SPIDR register are inhibited until the SPICSR register is read.

The SPIF bit can be cleared during a second transmission; however, it must be cleared before the second SPIF bit in order to prevent an Overrun condition (see Section 10.6.5.2).



SERIAL PERIPHERAL INTERFACE (cont'd)

10.6.5 Error Flags

10.6.5.1 Master Mode Fault (MODF)

Master mode fault occurs when the master device's SS pin is pulled low.

When a Master mode fault occurs:

- The MODF bit is set and an SPI interrupt request is generated if the SPIE bit is set.
- The SPE bit is reset. This blocks all output from the device and disables the SPI peripheral.
- The MSTR bit is reset, thus forcing the device into slave mode.

Clearing the MODF bit is done through a software sequence:

1. A read access to the SPICSR register while the MODF bit is set.

2. A write to the SPICR register.

Notes: To avoid any conflicts in an application with multiple slaves, the SS pin must be pulled high during the MODF bit clearing sequence. The SPE and MSTR bits may be restored to their original state during or after this clearing sequence.

Hardware does not allow the user to set the SPE and MSTR bits while the MODF bit is set except in the MODF bit clearing sequence.

In a slave device, the MODF bit can not be set, but in a multimaster configuration the device can be in slave mode with the MODF bit set.

The MODF bit indicates that there might have been a multimaster conflict and allows software to handle this using an interrupt routine and either perform a reset or return to an application default state.

10.6.5.2 Overrun Condition (OVR)

An overrun condition occurs when the master device has sent a data byte and the slave device has not cleared the SPIF bit issued from the previously transmitted byte.

When an Overrun occurs:

 The OVR bit is set and an interrupt request is generated if the SPIE bit is set.

In this case, the receiver buffer contains the byte sent after the SPIF bit was last cleared. A read to the SPIDR register returns this byte. All other bytes are lost.

The OVR bit is cleared by reading the SPICSR register.

10.6.5.3 Write Collision Error (WCOL)

A write collision occurs when the software tries to write to the SPIDR register while a data transfer is taking place with an external device. When this happens, the transfer continues uninterrupted and the software write will be unsuccessful.

Write collisions can occur both in master and slave mode. See also Section 10.6.3.2 "Slave Select Management".

Note: A "read collision" will never occur since the received data byte is placed in a buffer in which access is always synchronous with the CPU operation.

The WCOL bit in the SPICSR register is set if a write collision occurs.

No SPI interrupt is generated when the WCOL bit is set (the WCOL bit is a status flag only).

Clearing the WCOL bit is done through a software sequence (see Figure 75).

Figure 75. Clearing the WCOL Bit (Write Collision Flag) Software Sequence





10.7 LINSCI SERIAL COMMUNICATION INTERFACE (LIN MASTER/SLAVE)

10.7.1 Introduction

The Serial Communications Interface (SCI) offers a flexible means of full-duplex data exchange with external equipment requiring an industry standard NRZ asynchronous serial data format. The SCI offers a very wide range of baud rates using two baud rate generator systems.

The LIN-dedicated features support the LIN (Local Interconnect Network) protocol for both master and slave nodes.

This chapter is divided into SCI Mode and LIN mode sections. For information on general SCI communications, refer to the SCI mode section. For LIN applications, refer to both the SCI mode and LIN mode sections.

10.7.2 SCI Features

- Full duplex, asynchronous communications
- NRZ standard format (Mark/Space)
- Independently programmable transmit and receive baud rates up to 500K baud
- Programmable data word length (8 or 9 bits)
- Receive buffer full, Transmit buffer empty and End of Transmission flags
- 2 receiver wake-up modes:
 - Address bit (MSB)
 - Idle line
- Muting function for multiprocessor configurations
- Separate enable bits for Transmitter and Receiver
- Overrun, Noise and Frame error detection

- 6 interrupt sources
 - Transmit data register empty
 - Transmission complete
 - Receive data register full
 - Idle line received
 - Overrun error
 - Parity interrupt
- Parity control:
 - Transmits parity bit
 - Checks parity of received data byte
- Reduced power consumption mode

10.7.3 LIN Features

- LIN Master
 - 13-bit LIN Synch Break generation
- LIN Slave
 - Automatic Header Handling
 - Automatic baud rate resynchronization based on recognition and measurement of the LIN Synch Field (for LIN slave nodes)
 - Automatic baud rate adjustment (at CPU frequency precision)
 - 11-bit LIN Synch Break detection capability
 - LIN Parity check on the LIN Identifier Field (only in reception)
 - LIN Error management
 - LIN Header Timeout
 - Hot plugging support



LINSCI™ SERIAL COMMUNICATION INTERFACE (SCI Mode) (cont'd) EXTENDED RECEIVE PRESCALER DIVISION REGISTER (SCIERPR) EXTENDED TRAN REGISTER (SCIERPR)

Read/Write

Reset Value: 0000 0000 (00h)

7							0
ERPR							
7	6	5	4	3	2	1	0

Bits 7:0 = **ERPR[7:0]** 8-bit Extended Receive Prescaler Register

The extended Baud Rate Generator is activated when a value other than 00h is stored in this register. The clock frequency from the 16 divider (see Figure 3) is divided by the binary factor set in the SCIERPR register (in the range 1 to 255).

The extended baud rate generator is not active after a reset.

EXTENDED TRANSMIT PRESCALER DIVISION REGISTER (SCIETPR)

Read/Write

Reset Value:0000 0000 (00h)

7							0
ETPR							
7	6	5	4	3	2	1	0

Bits 7:0 = **ETPR[7:0]** 8-bit Extended Transmit Prescaler Register

The extended Baud Rate Generator is activated when a value other than 00h is stored in this register. The clock frequency from the 16 divider (see Figure 3) is divided by the binary factor set in the SCIETPR register (in the range 1 to 255).

The extended baud rate generator is not active after a reset.

Note: In LIN slave mode, the Conventional and Extended Baud Rate Generators are disabled.







<u>ل</u>رک

LINSCI™ SERIAL COMMUNICATION INTERFACE (LIN Master Only) (Cont'd) CONTROL REGISTER 1 (SCICR1)

Read/Write

Reset Value: x000 0000 (x0h)

7							0
R8	Т8	SCID	М	WAKE	PCE	PS	PIE

Bit 7 = R8 Receive data bit 8.

This bit is used to store the 9th bit of the received word when M = 1.

Bit 6 = T8 Transmit data bit 8.

This bit is used to store the 9th bit of the transmitted word when M = 1.

Bit 5 = **SCID** *Disabled for low power consumption* When this bit is set the SCI prescalers and outputs are stopped and the end of the current byte transfer in order to reduce power consumption. This bit is set and cleared by software. 0: SCI enabled

1: SCI prescaler and outputs disabled

Bit 4 = M Word length.
This bit determines the word length. It is set or cleared by software.
0: 1 Start bit, 8 Data bits, 1 Stop bit
1: 1 Start bit, 9 Data bits, 1 Stop bit

Note: The M bit must not be modified during a data transfer (both transmission and reception).

Bit 3 = WAKE Wake-Up method.

This bit determines the SCI Wake-Up method, it is set or cleared by software. 0: Idle Line

1: Address Mark

Bit 2 = **PCE** Parity control enable.

This bit selects the hardware parity control (generation and detection). When the parity control is enabled, the computed parity is inserted at the MSB position (9th bit if M = 1; 8th bit if M = 0) and parity is checked on the received data. This bit is set and cleared by software. Once it is set, PCE is active after the current byte (in reception and in transmission).

0: Parity control disabled

1: Parity control enabled

Bit 1 = **PS** Parity selection.

This bit selects the odd or even parity when the parity generation/detection is enabled (PCE bit set). It is set and cleared by software. The parity is selected after the current byte.

0: Even parity

1: Odd parity

Bit 0 = **PIE** Parity interrupt enable.

This bit enables the interrupt capability of the hardware parity control when a parity error is detected (PE bit set). It is set and cleared by software. 0: Parity error interrupt disabled

1: Parity error interrupt enabled

10501



beCAN CONTROLLER (Cont'd)

10.9.5 Interrupts

Two interrupt vectors are dedicated to beCAN. Each interrupt source can be independently ena-

bled or disabled by means of the CAN Interrupt Enable Register (CIER) and CAN Error Interrupt Enable register (CEIER).





beCAN CONTROLLER (Cont'd)

10.9.8 Register Description

10.9.8.1 Control and Status Registers

CAN MASTER CONTROL REGISTER (CMCR) Reset Value: 0000 0010 (02h)

7							0
0	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ

Bit 7 = Reserved, must be kept cleared.

Bit 6 = **ABOM** Automatic Bus-Off Management - Read/Set/Clear

This bit controls the behaviour of the CAN hardware on leaving the Bus-Off state.

- 0: The Bus-Off state is left on software request. Refer to Section 0.1.4.5 Error Management, Bus-Off recovery.
- 1: The Bus-Off state is left automatically by hardware once 128 x 11 recessive bits have been monitored.

For detailed information on the Bus-Off state please refer to Section 0.1.4.5 Error Management.

Bit 5 = **AWUM** Automatic Wake-Up Mode - Read/Set/Clear

This bit controls the behaviour of the CAN hardware on message reception during sleep mode. 0: The sleep mode is left on software request by

- clearing the SLEEP bit of the CMCR register.
- 1: The sleep mode is left automatically by hardware on CAN message detection. The SLEEP bit of the CMCR register and the SLAK bit of the CMSR register are cleared by hardware.

Bit 4 = **NART** No Automatic Retransmission - Read/Set/Clear

- 0: The CAN hardware will automatically retransmit the message until it has been successfully transmitted according to the CAN standard.
- 1: A message will be transmitted only once, independently of the transmission result (successful, error or arbitration lost).

Bit 3 = **RFLM** Receive FIFO Locked Mode

- Read/Set/Clear

- 0: Receive FIFO not locked on overrun. Once a receive FIFO is full the next incoming message will overwrite the previous one.
- 1: Receive FIFO locked against overrun. Once a receive FIFO is full the next incoming message will be discarded.

Bit 2 = **TXFP** Transmit FIFO Priority

- Read/Set/Clear

This bit controls the transmission order when several mailboxes are pending at the same time.

- 0: Priority driven by the identifier of the message
- 1: Priority driven by the request order (chronologically)

Bit 1 = **SLEEP** Sleep Mode Request - Read/Set/Clear

This bit is set by software to request the CAN hardware to enter the sleep mode. Sleep mode will be entered as soon as the current CAN activity (transmission or reception of a CAN frame) has been completed.

This bit is cleared by software to exit sleep mode.

This bit is cleared by hardware when the AWUM bit is set and a SOF bit is detected on the CAN Rx signal.

Bit 0 = INRQ Initialization Request

- Read/Set/Clear

The software clears this bit to switch the hardware into normal mode. Once 11 consecutive recessive bits have been monitored on the Rx signal the CAN hardware is synchronized and ready for transmission and reception. Hardware signals this event by clearing the INAK bit if the CMSR register.

Software sets this bit to request the CAN hardware to enter initialization mode. Once software has set the INRQ bit, the CAN hardware waits until the current CAN activity (transmission or reception) is completed before entering the initialization mode. Hardware signals this event by setting the INAK bit in the CMSR register.

beCAN CONTROLLER (Cont'd)

10.9.8.3 CAN Filter Registers

CAN FILTER CONFIGURATION REG.0 (CFCR0)

All bits of this register are set and cleared by software. Read / Write

Reset Value: 0000 0000 (00h)

7							0	
0	FSC11	FSC10	FACT1	0	FSC01	FSC00	FACT0	

Note: To modify the FFAx and FSCx bits, the be-CAN must be in INIT mode.

Bit 7 = Reserved. Forced to 0 by hardware.

Bits 6:5 = **FSC1[1:0]** *Filter Scale Configuration* These bits define the scale configuration of Filter 1.

Bit 4 = **FACT1** *Filter Active* The software sets this bit to activate Filter 1. To modify the Filter 1 registers (CF1R[7:0]), the FACT1 bit must be cleared. 0: Filter 1 is not active 1: Filter 1 is active

Bit 3 = Reserved. Forced to 0 by hardware.

Bits 2:1 = **FSC0[1:0]** *Filter Scale Configuration* These bits define the scale configuration of Filter 0.

Bit 0 = **FACT0** *Filter Active* The software sets this bit to activate Filter 0. To modify the Filter 0 registers (CF0R[0:7]), the FACT0 bit must be cleared. 0: Filter 0 is not active 1: Filter 0 is active

CAN FILTER CONFIGURATION REG.1 (CFCR1)

All bits of this register are set and cleared by software. Read / Write

Reset Value: 0000 0000 (00h)

7							0	
0	FSC31	FSC30	FACT3	0	FSC21	FSC20	FACT2	

Bit 7 = Reserved. Forced to 0 by hardware.

Bits 6:5 = **FSC3[1:0]** *Filter Scale Configuration* These bits define the scale configuration of Filter 3.

Bit 4 = FACT3 Filter Active

The software sets this bit to activate filter 3. To modify the Filter 3 registers (CF3R[0:7]) the FACT3 bit must be cleared. 0: Filter 3 is not active 1: Filter 3 is active

Bit 3 = Reserved. Forced to 0 by hardware.

Bits 2:1 = **FSC2[1:0]** *Filter Scale Configuration* These bits define the scale configuration of Filter 2.

Bit 0 = FACT2 Filter Active

The software sets this bit to activate Filter 2. To modify the Filter 2 registers (CF2R[0:7]), the FACT2 bit must be cleared. 0: Filter 2 is not active

1: Filter 2 is active



COMMUNICATIONS INTERFACE CHARACTERISTICS (Cont'd)

12.12.2 CAN - Controller Area Network Interface

Subject to general operating condition for V_{DD} , f_{O-SC} , and T_A unless otherwise specified. Refer to I/O port characteristics for more details on

the input/output alternate function characteristics (CANTX and CANRX).

Symbol	Parameter	Conditions	Min	Тур	Max	Unit
t _{p(RX:TX)}	CAN controller propagation time ¹⁾				60	ns

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

12.13 10-BIT ADC CHARACTERISTICS

Subject to general operating conditions for V_{DD} , f_{CPU} , and T_A unless otherwise specified.

Symbol	Parameter	Conditions	Min ¹⁾	Тур	Max ¹⁾	Unit
f _{ADC}	ADC clock frequency		0.4		4	MHz
V _{AIN}	Conversion voltage range ²⁾	×	V _{SSA}		V _{DDA}	V
R _{AIN}	External input impedance	26			see Figure	kΩ
C _{AIN}	External capacitor on analog input	S			141 and	pF
f _{AIN}	Variation frequency of analog input signal	0,02			Figure 142	Hz
l _{lkg}	Negative input leakage current on ro- bust analog pins (refer to Table 2 on page 8)	$V_{IN} < V_{SS,}$ I _{IN} < 400µA on adjacent robust analog pin		5	6	μA
C _{ADC}	Internal sample and hold capacitor			6		pF
taaruu	Conversion time	f _{ADC} = 4 MHz	3.5			μS
CONV						1/f _{ADC}
1	Analog part	Sunk on V _{DDA} ²⁾		3.6		m۸
'ADC	Digital part	Sunk on V _{DD}			0.2	

Notes:

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

2. When V_{DDA} and V_{SSA} pins are not available on the pinout, the ADC refers to V_{DD} and V_{SS} .



13 PACKAGE CHARACTERISTICS

13.1 PACKAGE MECHANICAL DATA

57

Figure 146. 64-Pin Low Profile Quad Flat Package (14x14)



249/265

DEVICE CONFIGURATION AND ORDER INFORMATION (cont'd)

14.2 TRANSFER OF CUSTOMER CODE

Customer code is made up of the ROM/FAS-TROM contents and the list of the selected options (if any). The ROM/FASTROM contents are to be sent on diskette, or by electronic means, with the S19 hexadecimal file generated by the development tool. All unused bytes must be set to FFh.

The selected options are communicated to STMicroelectronics using the correctly completed OPTION LIST appended.

Refer to application note AN1635 for information on the counter listing returned by ST after code has been transferred.

The STMicroelectronics Sales Organization will be pleased to provide detailed information on contractual points.





uctls

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2008 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan -Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com

57