E·XFL



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	32
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 11x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f561j7t6

Email: info@E-XFL.COM

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

SYSTEM INTEGRITY MANAGEMENT (Cont'd)

6.4.4 Register Description

SYSTEM INTEGRITY (SI) CONTROL/STATUS REGISTER (SICSR)

Read/Write

Reset Value: 000x 000x (00h)

7							0
0	AVD IE	AVD F	LVD RF	0	0	0	WDG RF

Bit 7 = Reserved, must be kept cleared.

Bit 6 = **AVDIE** Voltage Detector interrupt enable This bit is set and cleared by software. It enables an interrupt to be generated when the AVDF flag changes (toggles). The pending interrupt information is automatically cleared when software enters the AVD interrupt routine. 0: AVD interrupt disabled

1: AVD interrupt enabled

Bit 5 = **AVDF** Voltage Detector flag

This read-only bit is set and cleared by hardware. If the AVDIE bit is set, an interrupt request is generated when the AVDF bit changes value. Refer to Figure 16 and to Section 6.4.2.1 for additional details.

0: V_{DD} over $V_{IT+(AVD)}$ threshold 1: V_{DD} under $V_{IT-(AVD)}$ threshold

Bit 4 = LVDRF LVD reset flag

This bit indicates that the last Reset was generated by the LVD block. It is set by hardware (LVD reset) and cleared by software (writing zero). See WDGRF flag description for more details. When the LVD is disabled by OPTION BYTE, the LVDRF bit value is undefined. Bits 3:1 = Reserved, must be kept cleared.

Bit 0 = WDGRF Watchdog reset flag

This bit indicates that the last Reset was generated by the Watchdog peripheral. It is set by hardware (watchdog reset) and cleared by software (writing zero) or an LVD Reset (to ensure a stable cleared state of the WDGRF flag when CPU starts).

Combined with the LVDRF flag information, the flag description is given by the following table.

•	
0	0
0	1
1	Х
	0 0 1

Application notes

The LVDRF flag is not cleared when another RE-SET type occurs (external or watchdog), the LVDRF flag remains set to keep trace of the original failure.

In this case, a watchdog reset can be detected by software while an external reset can not.

CAUTION: When the LVD is not activated with the associated option byte, the WDGRF flag can not be used in the application.

I/O PORTS (Cont'd)

Table 14. Port Configuration

Port	Din nomo	Input		Out	put
FUIL	Fininame	OR = 0	OR = 1	OR = 0	OR = 1
	PA0		pull-up interrupt (ei0)		
	PA1		floating interrupt (ei0)		
	PA2		pull-up interrupt (ei0)		
Port A	PA3	floating	floating interrupt (ei0)	opon drain	puch pull
FUILA	PA4	noating	pull-up interrupt (ei0)	openulain	pusii-puli
	PA5		floating interrupt (ei0)		
	PA6		pull-up interrupt (ei0)		
	PA7		floating interrupt (ei0)		.(5)
	PB0		pull-up interrupt (ei1)		
	PB1		floating interrupt (ei1)		
Dort D	PB2	flooting	pull-up interrupt (ei1)	anon drain	
FULE	PB3	noaung	floating interrupt (ei1)	openurain	pusn-puii
	PB4		pull-up interrupt (ei1)		
	PB5		floating interrupt (ei1)	0	
	PC0		pull-up	$\langle \Theta \rangle$	
	PC1	floating	pull-up interrupt (ei2)	opon drain	puch pull
Port C	PC2	noating	floating interrupt (ei2)	openulain	pusii-puli
10110	PC3		pull-up		
	PC4	pu	ll-up	controlled b	y CANTX *
	PC7:5	floating	pull-up	open drain	push-pull
	PD0		pull-up interrupt (ei3)		
	PD1	15	floating interrupt (ei3)		
	PD3:2		pull-up		
Port D	PD4	floating	floating interrupt (ei3)	open drain	push-pull
	PD5		pull-up		
	PD6		pull-up interrupt (ei3)		
	PD7		floating interrupt (ei3)		
Port E	PE7:0	floating (TTL)	pull-up (TTL)	open drain	push-pull
Port F	PF7:0	floating (TTL)	pull-up (TTL)	open drain	push-pull
16.					

* Note: When the CANTX alternate function is selected, the I/O port operates in output push-pull mode.

57

PWM AUTO-RELOAD TIMER (Cont'd)

10.3.2 Functional Description

Counter

The free running 8-bit counter is fed by the output of the prescaler, and is incremented on every rising edge of the clock signal.

It is possible to read or write the contents of the counter on the fly by reading or writing the Counter Access register (ARTCAR).

When a counter overflow occurs, the counter is automatically reloaded with the contents of the ARTARR register (the prescaler is not affected).

Counter clock and prescaler

The counter clock frequency is given by:

 $f_{COUNTER} = f_{INPUT} / 2^{CC[2:0]}$

The timer counter's input clock (f_{INPUT}) feeds the 7-bit programmable prescaler, which selects one of the eight available taps of the prescaler, as defined by CC[2:0] bits in the Control/Status Register (ARTCSR). Thus the division factor of the prescaler can be set to 2ⁿ (where n = 0, 1,..7).

This f_{INPUT} frequency source is selected through the EXCL bit of the ARTCSR register and can be either the f_{CPU} or an external input frequency f_{FXT} .

The clock input to the counter is enabled by the TCE (Timer Counter Enable) bit in the ARTCSR register. When TCE is reset, the counter is stopped and the prescaler and counter contents are frozen. When TCE is set, the counter runs at the rate of the selected clock source.

Counter and Prescaler Initialization

After RESET, the counter and the prescaler are cleared and $f_{INPUT} = f_{CPU}$.

The counter can be initialized by:

- Writing to the ARTARR register and then setting the FCRL (Force Counter Re-Load) and the TCE (Timer Counter Enable) bits in the ARTCSR register.
- Writing to the ARTCAR counter access register,

In both cases the 7-bit prescaler is also cleared, whereupon counting will start from a known value.

Direct access to the prescaler is not possible.

Output compare control

The timer compare function is based on four different comparisons with the counter (one for each PWMx output). Each comparison is made between the counter value and an output compare register (OCRx) value. This OCRx register can not be accessed directly, it is loaded from the duty cycle register (PWMDCRx) at each overflow of the counter.

This double buffering method avoids glitch generation when changing the duty cycle on the fly.



Figure 41. Output Compare Control

/۲۸

16-bit read sequence: (from either the Counter Register or the Alternate Counter Register).

Beginning of the sequence



Sequence completed

The user must read the MS Byte first, then the LS Byte value is buffered automatically.

This buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MS Byte several times.

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LS Byte of the count value at the time of the read.

Whatever the timer mode used (input capture, output compare, One Pulse mode or PWM mode) an overflow occurs when the counter rolls over from FFFFh to 0000h 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 remains pending to be issued as soon as they are both true.



1. Reading the SR register while the TOF bit is set. 2. An access (read or write) to the CLR register.

Notes: The TOF bit is not cleared by accesses to ACLR register. The advantage of accessing the ACLR register rather than the CLR 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).

10.4.3.2 External Clock

The external clock (where available) is selected if CC0 = 1 and CC1 = 1 in the CR2 register.

The status of the EXEDG bit in the CR2 register determines the type of level transition on the external clock pin EXTCLK that will trigger the free running counter.

The counter is synchronized with the falling edge of the internal CPU clock.

A minimum of four falling edges of the CPU clock must occur between two consecutive active edges of the external clock; thus the external clock frequency must be less than a quarter of the CPU clock frequency.

<u>///</u>

Figure 49. Counter Timing Diagram, Internal Clock Divided by 2

CPU CLOCK	mmmmmmm
INTERNAL RESET	
TIMER CLOCK	
- COUNTER REGISTER _	
TIMER OVERFLOW FLAG (TOF)	, ictts

Figure 50. Counter Timing Diagram, Internal Clock Divided by 4



Figure 51. Counter Timing Diagram, Internal Clock Divided By 8

	CPU CLOCK	
1	INTERNAL RESET	1
	TIMER CLOCK	/
	COUNTER REGISTER	FFFC FFFD 0000
	TIMER OVERFLOW FLAG (TOF)	

Note: The MCU is in reset state when the internal reset signal is high, when it is low the MCU is running.









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

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.

SERIAL PERIPHERAL INTERFACE (cont'd) SPI CONTROL/STATUS REGISTER (SPICSR)

Read/Write (some bits Read Only) Reset Value: 0000 0000 (00h)

7							0
SPIF	WCOL	OVR	MODF	-	SOD	SSM	SSI

Bit 7 = **SPIF** Serial Peripheral Data Transfer Flag (Read only)

This bit is set by hardware when a transfer has been completed. An interrupt is generated if SPIE = 1 in the SPICR register. It is cleared by a software sequence (an access to the SPICSR register followed by a write or a read to the SPIDR register).

- 0: Data transfer is in progress or the flag has been cleared.
- 1: Data transfer between the device and an external device has been completed.

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

Bit 6 = **WCOL** *Write Collision status (Read only)* This bit is set by hardware when a write to the SPIDR register is done during a transmit sequence. It is cleared by a software sequence (see Figure 75).

0: No write collision occurred

1: A write collision has been detected

Bit 5 = **OVR** SPI Overrun error (Read only) This bit is set by hardware when the byte currently being received in the shift register is ready to be transferred into the SPIDR register while SPIF = 1 (See Section 10.6.5.2). An interrupt is generated if SPIE = 1 in the SPICR register. The OVR bit is cleared by software reading the SPICSR register. 0: No overrun error

1: Overrun error detected

Bit 4 = MODF Mode Fault flag (Read only)

This bit is set by hardware when the SS pin is pulled low in master mode (see Section 10.6.5.1 "Master Mode Fault (MODF)"). An SPI interrupt can be generated if SPIE = 1 in the SPICR register. This bit is cleared by a software sequence (An access to the SPICSR register while MODF = 1 followed by a write to the SPICR register). 0: No master mode fault detected

1: A fault in master mode has been detected

Bit 3 = Reserved, must be kept cleared.

Bit 2 = SOD SPI Output Disable

This bit is set and cleared by software. When set, it disables the alternate function of the SPI output (MOSI in master mode / MISO in slave mode) 0: SPI output enabled (if SPE = 1) 1: SPI output disabled

Bit 1 = **SSM** SS Management

This bit is set and cleared by software. When set, it disables the alternate function of the SPI SS pin and uses the SSI bit value instead. See Section 10.6.3.2 "Slave Select Management".

- 0: Hardware management (SS managed by external pin)
- 1: Software management (internal SS signal controlled by SSI bit. External SS pin free for general-purpose I/O)

Bit 0 = SSI SS Internal Mode

This bit is set and cleared by software. It acts as a 'chip select' by controlling the level of the \overline{SS} slave select signal when the SSM bit is set.

0: Slave selected

1: Slave deselected

SPI DATA I/O REGISTER (SPIDR)

Read/Write Reset Value: Undefined

7							0
D7	D6	D5	D4	D3	D2	D1	D0

The SPIDR register is used to transmit and receive data on the serial bus. In a master device, a write to this register will initiate transmission/reception of another byte.

Notes: During the last clock cycle the SPIF bit is set, a copy of the received data byte in the shift register is moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read.

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

Warning: A write to the SPIDR register places data directly into the shift register for transmission.

A read to the SPIDR register returns the value located in the buffer and not the content of the shift register (see Figure 70).



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



57

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

SCICR2 register is set, the LHDM bit selects the Wake-Up method (replacing the WAKE bit). 0: LIN Synch Break Detection Method 1: LIN Identifier Field Detection Method

Bit 2 = LHIE LIN Header Interrupt Enable

This bit is set and cleared by software. It is only usable in LIN Slave mode.

0: LIN Header Interrupt is inhibited.

1: An SCI interrupt is generated whenever LHDF = 1.

Bit 1 = LHDF LIN Header Detection Flag

This bit is set by hardware when a LIN Header is detected and cleared by a software sequence (an access to the SCISR register followed by a read of the SCICR3 register). It is only usable in LIN Slave mode.

0: No LIN Header detected.

1: LIN Header detected.

Notes: The header detection method depends on the LHDM bit:

- If LHDM = 0, a header is detected as a LIN Synch Break.
- If LHDM = 1, a header is detected as a LIN Identifier, meaning that a LIN Synch Break Field + a LIN Synch Field + a LIN Identifier Field have been consecutively received.

Bit 0 = LSF LIN Synch Field State

This bit indicates that the LIN Synch Field is being analyzed. It is only used in LIN Slave mode. In Auto Synchronization Mode (LASE bit = 1), when the SCI is in the LIN Synch Field State it waits or counts the falling edges on the RDI line.

It is set by hardware as soon as a LIN Synch Break is detected and cleared by hardware when the LIN Synch Field analysis is finished (see Figure 11). This bit can also be cleared by software to exit LIN Synch State and return to idle mode.

0: The current character is not the LIN Synch Field

1: LIN Synch Field State (LIN Synch Field undergoing analysis)





LIN DIVIDER REGISTERS

LDIV is coded using the two registers LPR and LP-FR. In LIN Slave mode, the LPR register is accessible at the address of the SCIBRR register and the LPFR register is accessible at the address of the SCIETPR register.

LIN PRESCALER REGISTER (LPR) Read/Write

Reset Value: 0000 0000 (00h)

75	0.						0
LPR7	LPR6	LPR5	LPR4	LPR3	LPR2	LPR1	LPR0

LPR[7:0] LIN Prescaler (mantissa of LDIV)

These 8 bits define the value of the mantissa of the LIN Divider (LDIV):

LPR[7:0]	Rounded Mantissa (LDIV)
00h	SCI clock disabled
01h	1
FEh	254
FFh	255

Caution: LPR and LPFR registers have different meanings when reading or writing to them. Consequently bit manipulation instructions (BRES or BSET) should never be used to modify the LPR[7:0] bits, or the LPFR[3:0] bits.

57

10.8 LINSCI SERIAL COMMUNICATION INTERFACE (LIN Master Only)

10.8.1 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.

10.8.2 Main Features

- Full duplex, asynchronous communications
- NRZ standard format (Mark/Space)
- Dual baud rate generator systems
- 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
- 4 error detection flags:
 - Overrun error
 - Noise error
 - Frame error
 - Parity error
- 5 interrupt sources with flags:
 - Transmit data register empty
 - Transmission complete
 - Receive data register full
 - Idle line received
 - Overrun error detected
- Transmitter clock output
- Parity control:
 - Transmits parity bit
 - Checks parity of received data byte
- Reduced power consumption mode
- LIN Synch Break send capability

10.8.3 General Description

The interface is externally connected to another device by three pins (see Figure 88 on page 153). Any SCI bidirectional communication requires a minimum of two pins: Receive Data In (RDI) and Transmit Data Out (TDO):

- SCLK: Transmitter clock output. This pin outputs the transmitter data clock for synchronous transmission (no clock pulses on start bit and stop bit, and a software option to send a clock pulse on the last data bit). This can be used to control peripherals that have shift registers (e.g. LCD drivers). The clock phase and polarity are software programmable.
- TDO: Transmit Data Output. When the transmitter is disabled, the output pin returns to its I/O port configuration. When the transmitter is enabled and nothing is to be transmitted, the TDO pin is at high level.
- RDI: Receive Data Input is the serial data input. Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise.

Through these pins, serial data is transmitted and received as frames comprising:

- An Idle Line prior to transmission or reception
- A start bit
- A data word (8 or 9 bits) least significant bit first
- A Stop bit indicating that the frame is complete.
- This interface uses two types of baud rate generator:
- A conventional type for commonly-used baud rates,
- An extended type with a prescaler offering a very wide range of baud rates even with non-standard oscillator frequencies.

67/

beCAN CONTROLLER (Cont'd)

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

```
Figure 114. Workaround 2
```

```
a, CRFR
    Ld
    And
             a,#3
    Ср
             a,#2
                         ; test FMP=2 ?
             RELEASE
    Jrne
                        ; if not release
    Btjf
             CMSR, #5, _RELEASE ; test if reception on going.
                         ; if not release
Obsolete Product(s)
```

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.

INSTRUCTION SET OVERVIEW (Cont'd)

11.1.6 Indirect Indexed (Short, Long)

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

The indirect indexed addressing mode consists of two submodes:

Indirect Indexed (Short)

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - 1FE addressing space, and requires 1 byte after the opcode.

Indirect Indexed (Long)

The pointer address is a byte, the pointer size is a word, thus allowing 64 Kbyte addressing space, and requires 1 byte after the opcode.

Table 38. InstructionsSupportingDirect,Indexed,IndirectandIndirectIndexedAddressing ModesIndexedIndexedIndexedIndexed

Long and Short Instructions	Function		
LD	Load		
СР	Compare		
AND, OR, XOR	Logical Operations		
ADC, ADD, SUB, SBC	Arithmetic Additions/Sub- stractions operations		
BCP	Bit Compare		

Short Instructions Only	Function				
CLR	Clear				
INC, DEC	Increment/Decrement				
TNZ	Test Negative or Zero				
CPL, NEG	1 or 2 Complement				
BSET, BRES	Bit Operations				
BTJT, BTJF	Bit Test and Jump Opera- tions				
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations				
SWAP	Swap Nibbles				
CALL, JP	Call or Jump subroutine				

11.1.7 Relative mode (Direct, Indirect)

This addressing mode is used to modify the PC register value, by adding an 8-bit signed offset to it.

Available Relative Direct/Indirect Instructions	Function		
JRxx	Conditional Jump		
CALLR	Call Relative		
	*		

The relative addressing mode consists of two submodes:

Relative (Direct)

The offset is following the opcode.

Relative (Indirect)

The offset is defined in memory, which address follows the opcode.

CLOCK CHARACTERISTICS (Cont'd)

12.5.4 PLL Characteristics

Operating conditions: V_{DD} 3.8 to 5.5V @ T_A 0 to 70°C¹⁾ or V_{DD} 4.5 to 5.5V @ T_A -40 to 125°C

Symbol	Parameter	Conditions	Min	Тур	Мах	Unit
V _{DD(PLL)}	PLL Voltage Bange	$T_A = 0$ to $+70^{\circ}C$	3.8		5.5	
		$T_{A} = -40 \text{ to } +125^{\circ}\text{C}$	4.5		5.5	
f _{OSC}	PLL input frequency range		2		4	MHz
$\Delta f_{CPU}/f_{CPU}$	PLL jitter ¹⁾	$f_{OSC} = 4 \text{ MHz}, V_{DD} = 4.5 \text{ to } 5.5 \text{V}$		Note 2		0/_
		$f_{OSC} = 2 \text{ MHz}, \text{ V}_{DD} = 4.5 \text{ to } 5.5 \text{ V}$				/0

Notes:

- 1. Data characterized but not tested.
- 2. Under characterization





The user must take the PLL jitter into account in the application (for example in serial communication or sampling of high frequency signals). The PLL jitter is a periodic effect, which is integrated over several CPU cycles. Therefore, the longer the period of the application signal, the less it is impacted by the PLL jitter.

Figure 123 shows the PLL jitter integrated on application signals in the range 125 kHz to 2 MHz. At frequencies of less than 125 kHz, the jitter is negligible.

Notes:

1. Measurement conditions: $f_{CPU} = 4$ MHz, $T_A = 25^{\circ}C$



I/O PORT PIN CHARACTERISTICS (Cont'd)

Figure 131. Typical V_{OL} vs V_{DD} (Standard I/Os)





57



237/265

TRANSFER OF CUSTOMER CODE (cont'd)

Figure 151. FASTROM Factory Coded Device Types



Figure 152. FLASH Device Types



SCIBRR = 0xC9), the wrong break duration occurrence is around 1%.

Workaround

If this wrong duration is not compliant with the communication protocol in the application, software can request that an Idle line be generated before the break character. In this case, the break duration is always correct assuming the application is not doing anything between the idle and the break. This can be ensured by temporarily disabling interrupts.

- Disable interrupts
- Reset and Set TE (IDLE request)
- Set and Reset SBK (Break Request)
- Re-enable interrupts

LIN mode

If the LINE bit in the SCICR3 is set and the M bit in the SCICR1 register is reset, the LINSCI is in LIN master mode. A single break character is sent by obsolete Production setting and resetting the SBK bit in the SCICR2 register. In some cases, the break character may



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