



Welcome to [E-XFL.COM](#)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, LINbusSCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	32
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 11x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	-
Purchase URL	<a href="https://www.e-xfl.com/product-detail/stmicroelectronics/st72f561j9t3">https://www.e-xfl.com/product-detail/stmicroelectronics/st72f561j9t3</a>

Address	Block	Register Label	Register Name	Reset Status	Remarks
000Fh 0010h 0011h	Port F	PFDR PFDDR PFOR	Port F Data Register Port F Data Direction Register Port F Option Register	00h <sup>1)</sup> 00h 00h	R/W <sup>2)</sup> R/W <sup>2)</sup> R/W <sup>2)</sup>
0012h to 0020h	Reserved Area (15 bytes)				
0021h 0022h 0023h	SPI	SPIDR SPICR SPICSR	SPI Data I/O Register SPI Control Register SPI Control/Status Register	xxh 0xh 00h	R/W R/W R/W
0024h	FLASH	FCSR	Flash Control/Status Register	00h	R/W
0025h 0026h 0027h 0028h 0029h 002Ah	ITC	ISPR0 ISPR1 ISPR2 ISPR3 EICR0 EICR1	Interrupt Software Priority Register 0 Interrupt Software Priority Register 1 Interrupt Software Priority Register 2 Interrupt Software Priority Register 3 External Interrupt Control Register 0 External Interrupt Control Register 1	FFh FFh FFh FFh 00h 00h	R/W R/W R/W R/W R/W R/W
002Bh 002Ch	AWU	AWUCSR AWUPR	Auto Wake up f. Halt Control/Status Register Auto Wake Up From Halt Prescaler	00h FFh	R/W R/W
002Dh 002Eh	CKCTRL	SICSR MCCSR	System Integrity Control / Status Register Main Clock Control / Status Register	0xh 00h	R/W R/W
002Fh 0030h	WWDG	WDGCR WDGWR	Watchdog Control Register Watchdog Window Register	7Fh 7Fh	R/W R/W
0031h 0032h 0033h 0034h 0035h 0036h 0037h 0038h 0039h 003Ah 003Bh	PWM ART	PWMDCR3 PWMDCR2 PWMDCR1 PWMDCR0 PWMCR ARTCSR ARTCAR ARTARR ARTICCSR ARTICR1 ARTICR2	Pulse Width Modulator Duty Cycle Register 3 PWM Duty Cycle Register 2 PWM Duty Cycle Register 1 PWM Duty Cycle Register 0 PWM Control register Auto-Reload Timer Control/Status Register Auto-Reload Timer Counter Access Register Auto-Reload Timer Auto-Reload Register ART Input Capture Control/Status Register ART Input Capture Register 1 ART Input Capture register 2	00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h	R/W R/W R/W R/W R/W R/W R/W R/W R/W Read Only Read Only
003Ch 003Dh 003Eh 003Fh 0040h 0041h 0042h 0043h 0044h	8-BIT TIMER	T8CR2 T8CR1 T8CSR T8IC1R T8OC1R T8CTR T8ACTR T8IC2R T8OC2R	Timer Control Register 2 Timer Control Register 1 Timer Control/Status Register Timer Input Capture 1 Register Timer Output Compare 1 Register Timer Counter Register Timer Alternate Counter Register Timer Input Capture 2 Register Timer Output Compare 2 Register	00h 00h 00h xxh 00h FCh FCh xxh 00h	R/W R/W Read Only Read Only R/W Read Only Read Only Read Only R/W
0045h 0046h 0047h	ADC	ADCCSR ADCDRH ADCDDL	Control/Status Register Data High Register Data Low Register	00h 00h 00h	R/W Read Only Read Only

Address	Block	Register Label	Register Name	Reset Status	Remarks
0068h 0069h 006Ah 006Bh 006Ch 006Dh 006Eh 006Fh	Active CAN	CMCR CMSR CTSR CTPR CRFR CIER CDGR CPSR	CAN Master Control Register CAN Master Status Register CAN Transmit Status Register CAN Transmit Priority Register CAN Receive FIFO Register CAN Interrupt Enable Register CAN Diagnosis Register CAN Page Selection Register		R/W R/W R/W R/W R/W R/W R/W R/W
0070h 0071h 0072h 0073h 0074h 0075h 0076h 0077h 0078h 0079h 007Ah 007Bh 007Ch 007Dh 007Eh 007Fh		PAGES	PAGE REGISTER 0 PAGE REGISTER 1 PAGE REGISTER 2 PAGE REGISTER 3 PAGE REGISTER 4 PAGE REGISTER 5 PAGE REGISTER 6 PAGE REGISTER 7 PAGE REGISTER 8 PAGE REGISTER 9 PAGE REGISTER 10 PAGE REGISTER 11 PAGE REGISTER 12 PAGE REGISTER 13 PAGE REGISTER 14 PAGE REGISTER 15		R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W

**Legend:** x = undefined, R/W = read/write

**Notes:**

1. The contents of the I/O port DR registers are readable only in output configuration. In input configuration, the values of the I/O pins are returned instead of the DR register contents.
2. The bits associated with unavailable pins must always keep their reset value.

## FLASH PROGRAM MEMORY (Cont'd)

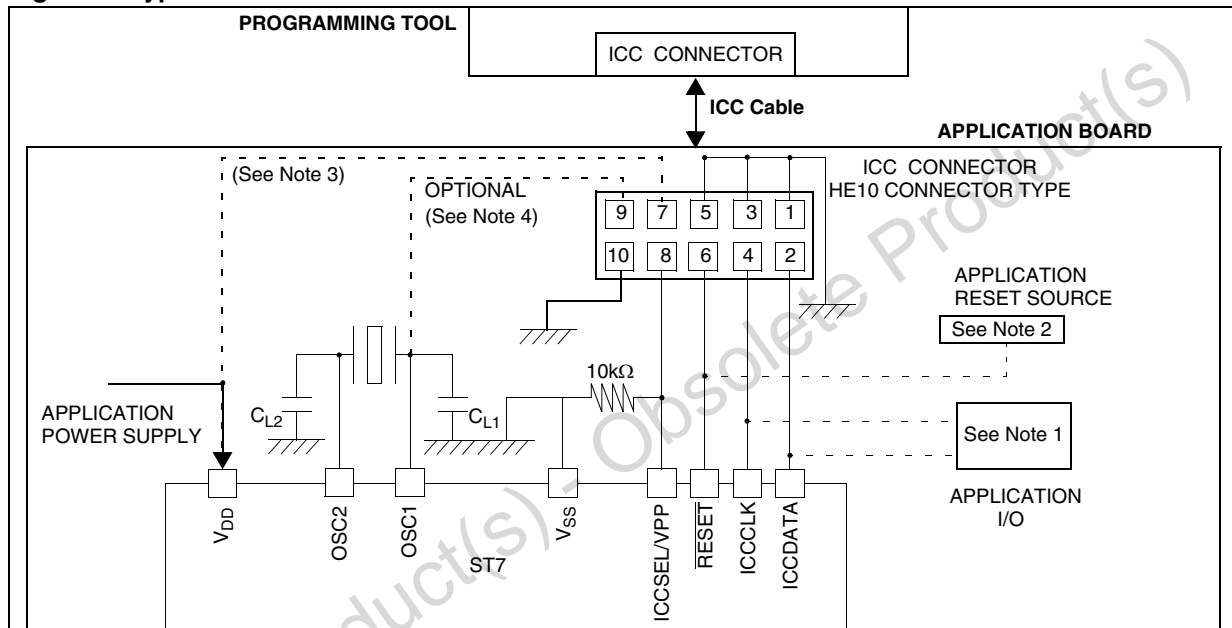
## 4.4 ICC INTERFACE

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:

- $\overline{\text{RESET}}$ : device reset
- $V_{SS}$ : device power supply ground

- 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)

Figure 7. Typical ICC Interface



## 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 be implemented in case another device forces the signal. Refer to the Programming Tool documentation for recommended resistor values.

2. During the ICC session, the programming tool must control the  $\overline{\text{RESET}}$  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 OSCIN 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.

# 5 CENTRAL PROCESSING UNIT

## 5.1 INTRODUCTION

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

## 5.2 MAIN FEATURES

- Enable executing 63 basic instructions
- Fast 8-bit by 8-bit multiply
- 17 main addressing modes (with indirect addressing mode)
- Two 8-bit index registers
- 16-bit stack pointer
- Low power HALT and WAIT modes
- Priority maskable hardware interrupts
- Non-maskable software/hardware interrupts

## 5.3 CPU REGISTERS

The six CPU registers shown in Figure 8 are not present in the memory mapping and are accessed by specific instructions.

### Accumulator (A)

The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations and to manipulate data.

### Index Registers (X and Y)

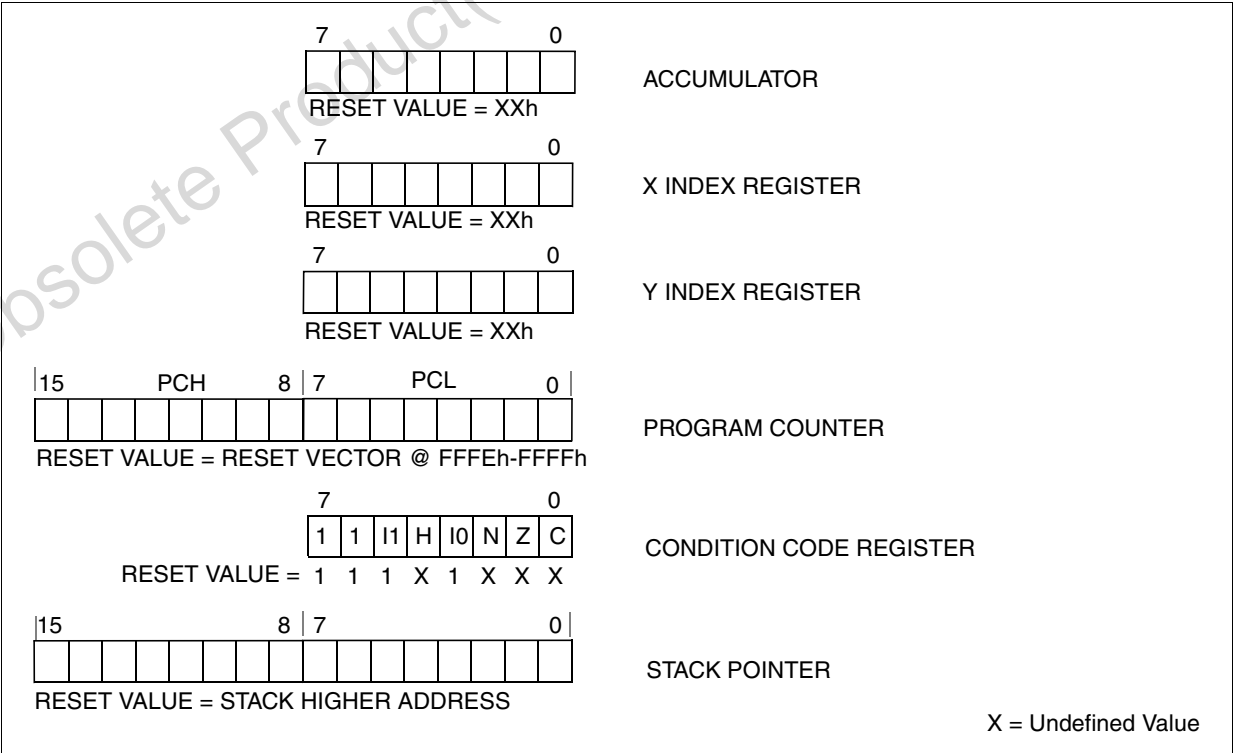
These 8-bit registers are used to create effective addresses or as temporary storage areas for data manipulation. (The Cross-Assembler generates a precede instruction (PRE) to indicate that the following instruction refers to the Y register.)

The Y register is not affected by the interrupt automatic procedures.

### Program Counter (PC)

The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU. It is made of two 8-bit registers PCL (Program Counter Low which is the LSB) and PCH (Program Counter High which is the MSB).

Figure 8. CPU Registers



**INTERRUPTS (Cont'd)****7.6.2 Register Description****EXTERNAL INTERRUPT CONTROL REGISTER 0 (EICR0)**

Read/Write

Reset Value: 0000 0000 (00h)

7							0
IS31	IS30	IS21	IS20	IS11	IS10	IS01	IS00

Bits 7:6 = **IS3[1:0]** *ei3 sensitivity*

The interrupt sensitivity, defined using the IS3[1:0] bits, is applied to the ei3 external interrupts:

IS31	IS30	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

Bits 5:4 = **IS2[1:0]** *ei2 sensitivity*

The interrupt sensitivity, defined using the IS2[1:0] bits, is applied to the ei2 external interrupts:

IS21	IS20	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

Bits 3:2 = **IS1[1:0]** *ei1 sensitivity*

The interrupt sensitivity, defined using the IS1[1:0] bits, is applied to the ei1 external interrupts:

IS11	IS10	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

Bits 1:0 = **IS0[1:0]** *ei0 sensitivity*

The interrupt sensitivity, defined using the IS0[1:0] bits, is applied to the ei0 external interrupts:

IS01	IS00	External Interrupt Sensitivity
0	0	Falling edge & low level
0	1	Rising edge only
1	0	Falling edge only
1	1	Rising and falling edge

These 2 bits can be written only when I1 and I0 of the CC register are both set to 1 (level 3).

**EXTERNAL INTERRUPT CONTROL REGISTER 1 (EICR1)**

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	0	TLIS	TLIE

Bits 7:2 = Reserved

Bit 1 = **TLIS** *Top Level Interrupt sensitivity*

This bit configures the TLI edge sensitivity. It can be set and cleared by software only when TLIE bit is cleared.

0: Falling edge

1: Rising edge

Bit 0 = **TLIE** *Top Level Interrupt enable*

This bit allows to enable or disable the TLI capability on the dedicated pin. It is set and cleared by software.

0: TLI disabled

1: TLI enabled

**Notes:**

- A parasitic interrupt can be generated when clearing the TLIE bit.
- In some packages, the TLI pin is not available. In this case, the TLIE bit must be kept low to avoid parasitic TLI interrupts.

**WINDOW WATCHDOG (Cont'd)****10.1.9 Interrupts**

None.

**10.1.10 Register Description****CONTROL REGISTER (WDGCR)**

Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0

Bit 7 = **WDGA** Activation bit.

This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.

0: Watchdog disabled

1: Watchdog enabled

**Note:** This bit is not used if the hardware watchdog option is enabled by option byte.

Bits 6:0 = **T[6:0]** 7-bit counter (MSB to LSB).

These bits contain the value of the watchdog counter. It is decremented every 16384  $f_{OSC2}$  cycles (approx.). A reset is produced when it rolls over from 40h to 3Fh (T6 becomes cleared).

**WINDOW REGISTER (WDGWR)**

Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
-	W6	W5	W4	W3	W2	W1	W0

Bit 7 = Reserved

Bits 6:0 = **W[6:0]** 7-bit window value

These bits contain the window value to be compared to the downcounter.

**Figure 38. Watchdog Timer Register Map and Reset Values**

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
2F	<b>WDGCR</b>	WDGA	T6	T5	T4	T3	T2	T1	T0
	Reset Value	0	1	1	1	1	1	1	1
30	<b>WDGWR</b>	-	W6	W5	W4	W3	W2	W1	W0
	Reset Value	0	1	1	1	1	1	1	1

**MAIN CLOCK CONTROLLER WITH REAL TIME CLOCK (Cont'd)**

Bit 0 = **OIF** *Oscillator interrupt flag*

This bit is set by hardware and cleared by software reading the CSR register. It indicates when set that the main oscillator has reached the selected elapsed time (TB1:0).

0: Timeout not reached

1: Timeout reached

**CAUTION:** The BRES and BSET instructions must not be used on the MCCR register to avoid unintentionally clearing the OIF bit.

**Table 16. Main Clock Controller Register Map and Reset Values**

Address (Hex.)	Register Label	7	6	5	4	3	2	1	0
002Dh	<b>SICSR</b> Reset Value	0	AVDIE	AVDF	LVDRF	0	0	0	WDGRF x
002Eh	<b>MCCR</b> Reset Value	MCO 0	CP1 0	CP0 0	SMS 0	TB1 0	TB0 0	OIE 0	OIF 0

## ON-CHIP PERIPHERALS (Cont'd)

## 10.3 PWM AUTO-RELOAD TIMER (ART)

## 10.3.1 Introduction

The Pulse Width Modulated Auto-Reload Timer on-chip peripheral consists of an 8-bit auto reload counter with compare/capture capabilities and of a 7-bit prescaler clock source.

These resources allow five possible operating modes:

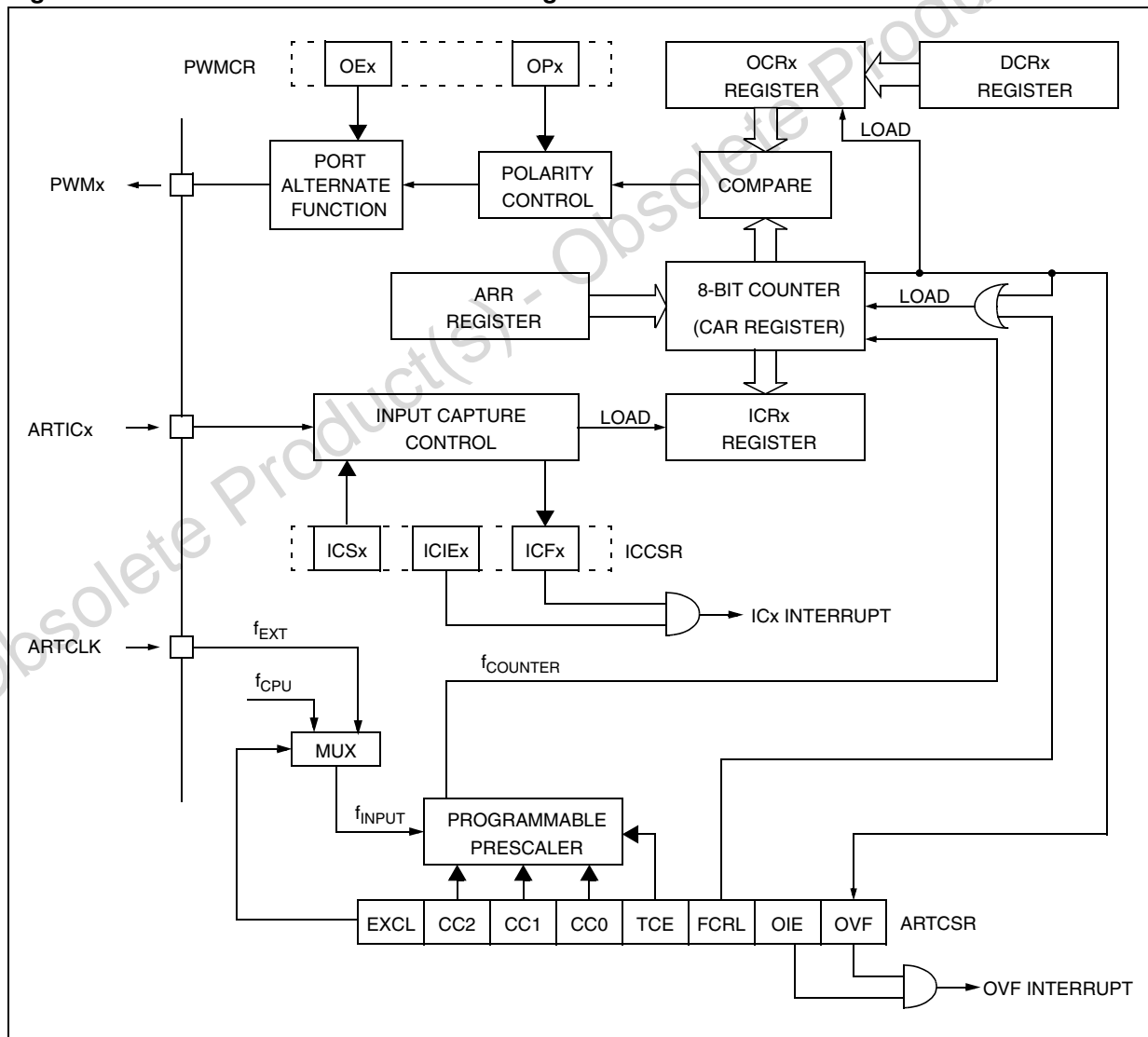
- Generation of up to four independent PWM signals

- Output compare and Time base interrupt
- Up to two input capture functions
- External event detector
- Up to two external interrupt sources

The three first modes can be used together with a single counter frequency.

The timer can be used to wake up the MCU from WAIT and HALT modes.

Figure 40. PWM Auto-Reload Timer Block Diagram



# PWM AUTO-RELOAD TIMER (Cont'd)

## Independent PWM signal generation

This mode allows up to four Pulse Width Modulated signals to be generated on the PWMx output pins with minimum core processing overhead. This function is stopped during HALT mode.

Each PWMx output signal can be selected independently using the corresponding OEx bit in the PWM Control register (PWMCR). When this bit is set, the corresponding I/O pin is configured as output push-pull alternate function.

The PWM signals all have the same frequency which is controlled by the counter period and the ARTARR register value.

$$f_{\text{PWM}} = f_{\text{COUNTER}} / (256 - \text{ARTARR})$$

When a counter overflow occurs, the PWMx pin level is changed depending on the corresponding OPx (output polarity) bit in the PWMCR register.

When the counter reaches the value contained in one of the output compare register (OCRx) the corresponding PWMx pin level is restored.

It should be noted that the reload values will also affect the value and the resolution of the duty cycle of the PWM output signal. To obtain a signal on a PWMx pin, the contents of the OCRx register must be greater than the contents of the ARTARR register.

The maximum available resolution for the PWMx duty cycle is:

$$\text{Resolution} = 1 / (256 - \text{ARTARR})$$

**Note:** To get the maximum resolution (1/256), the ARTARR register must be 0. With this maximum resolution, 0% and 100% can be obtained by changing the polarity.

Figure 42. PWM Auto-reload Timer Function

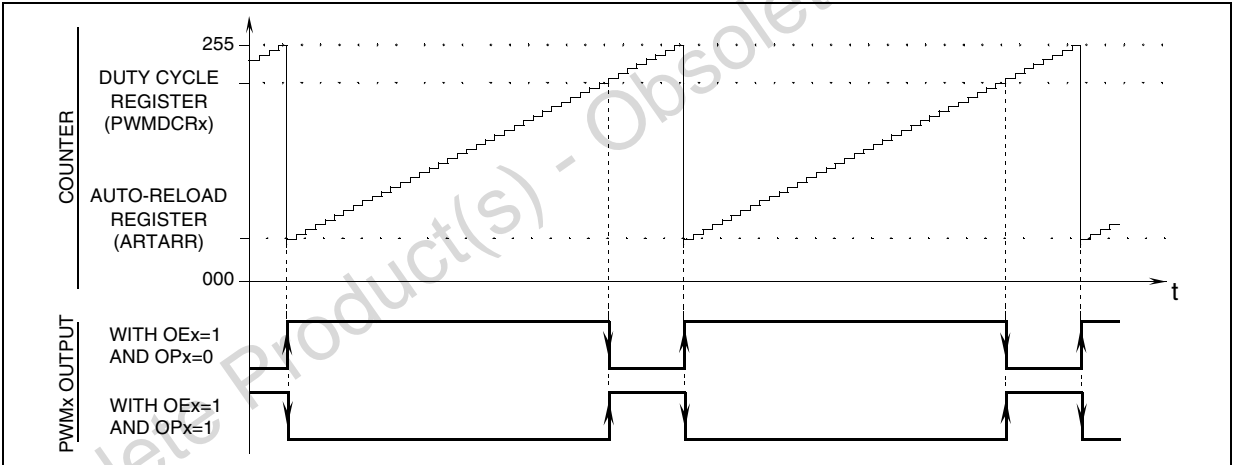
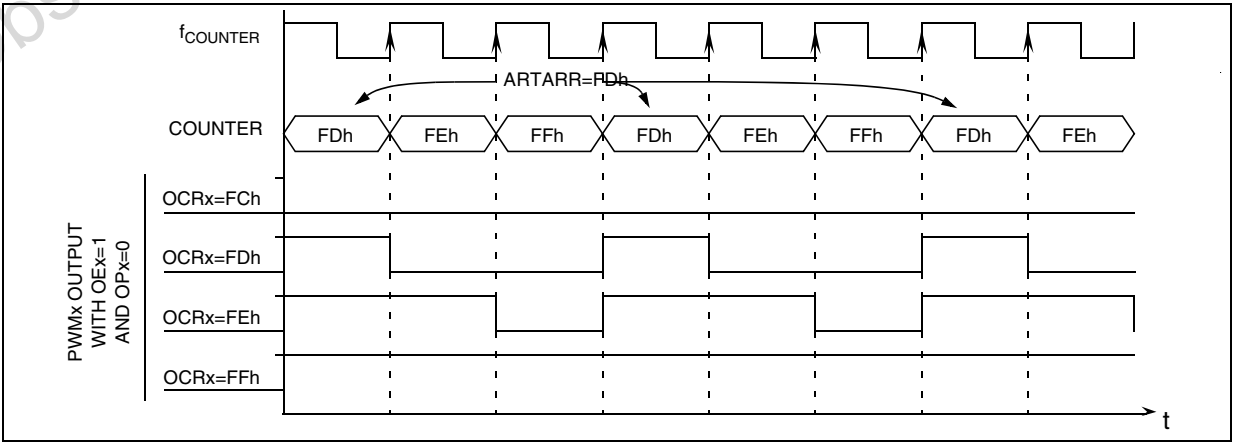


Figure 43. PWM Signal from 0% to 100% Duty Cycle



**ON-CHIP PERIPHERALS (Cont'd)****INPUT CAPTURE CONTROL / STATUS REGISTER (ARTICCSR)**

Read/Write

Reset Value: 0000 0000 (00h)

7							0
0	0	CS2	CS1	CIE2	CIE1	CF2	CF1

Bit 7:6 = Reserved, always read as 0.

**Bit 5:4 = CS[2:1] Capture Sensitivity**

These bits are set and cleared by software. They determine the trigger event polarity on the corresponding input capture channel.

0: Falling edge triggers capture on channel x.

1: Rising edge triggers capture on channel x.

**Bit 3:2 = CIE[2:1] Capture Interrupt Enable**

These bits are set and cleared by software. They enable or disable the Input capture channel interrupts independently.

0: Input capture channel x interrupt disabled.

1: Input capture channel x interrupt enabled.

**Bit 1:0 = CF[2:1] Capture Flag**

These bits are set by hardware and cleared by software reading the corresponding ARTICRx register. Each CFx bit indicates that an input capture x has occurred.

0: No input capture on channel x.

1: An input capture has occurred on channel x.

**INPUT CAPTURE REGISTERS (ARTICRx)**

Read only

Reset Value: 0000 0000 (00h)

7							0
IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0

**Bit 7:0 = IC[7:0] Input Capture Data**

These read only bits are set and cleared by hardware. An ARTICRx register contains the 8-bit auto-reload counter value transferred by the input capture channel x event.

**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.

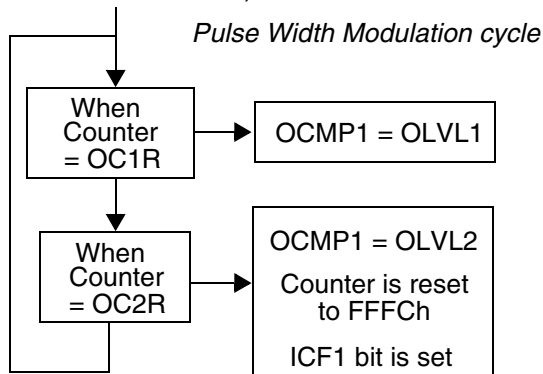
Pulse Width Modulation mode uses the complete Output Compare 1 function plus the OC2R register, and so this functionality can not be used when PWM mode is activated.

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/R register value required for a specific timing application can be calculated using the following formula:

$$\text{OC/R Value} = \frac{t \cdot f_{\text{CPU}}}{\text{PRESC}} - 5$$

Where:

$t$  = Signal or pulse period (in seconds)

$f_{\text{CPU}}$  = 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:

$$\text{OC/R} = t \cdot f_{\text{EXT}} - 5$$

Where:

$t$  = Signal or pulse period (in seconds)

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

**16-BIT TIMER (Cont'd)****CONTROL REGISTER 2 (CR2)**

Read/Write

Reset Value: 0000 0000 (00h)

7							0
OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG

Bit 7 = **OC1E** *Output Compare 1 Pin Enable*.

This bit is used only to output the signal from the timer on the OCMP1 pin (OLV1 in Output Compare mode, both OLV1 and OLV2 in PWM and one-pulse mode). Whatever the value of the OC1E bit, the Output Compare 1 function of the timer remains active.

0: OCMP1 pin alternate function disabled (I/O pin free for general-purpose I/O).

1: OCMP1 pin alternate function enabled.

Bit 6 = **OC2E** *Output Compare 2 Pin Enable*.

This bit is used only to output the signal from the timer on the OCMP2 pin (OLV2 in Output Compare mode). Whatever the value of the OC2E bit, the Output Compare 2 function of the timer remains active.

0: OCMP2 pin alternate function disabled (I/O pin free for general-purpose I/O).

1: OCMP2 pin alternate function enabled.

Bit 5 = **OPM** *One Pulse Mode*.

0: One Pulse mode is not active.

1: One Pulse mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.

Bit 4 = **PWM** *Pulse Width Modulation*.

0: PWM mode is not active.

1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.

Bit 3, 2 = **CC[1:0]** *Clock Control*.

The timer clock mode depends on these bits:

**Table 18. Clock Control Bits**

Timer Clock	CC1	CC0
$f_{CPU} / 4$	0	0
$f_{CPU} / 2$		1
$f_{CPU} / 8$	1	0
External Clock (where available)		1

**Note:** If the external clock pin is not available, programming the external clock configuration stops the counter.

Bit 1 = **IEDG2** *Input Edge 2*.

This bit determines which type of level transition on the ICAP2 pin will trigger the capture.

0: A falling edge triggers the capture.

1: A rising edge triggers the capture.

Bit 0 = **EXEDG** *External Clock Edge*.

This bit determines which type of level transition on the external clock pin EXTCLK will trigger the counter register.

0: A falling edge triggers the counter register.

1: A rising edge triggers the counter register.

**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.

**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.

**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.

**COUNTER LOW REGISTER (CLR)**

Read Only

Reset Value: 1111 1100 (FCh)

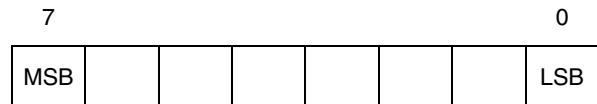
This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after accessing the CSR register clears the TOF bit.

**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.

**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.

**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).

**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).



**LINSPI™ SERIAL COMMUNICATION INTERFACE (LIN Mode)** (cont'd)**10.7.10 LIN Mode Register Description****STATUS REGISTER (SCISR)**

Read Only

Reset Value: 1100 0000 (C0h)

7							0
TDRE	TC	RDRF	IDLE	LHE	NF	FE	PE

Bits 7:4 = Same function as in SCI mode; please refer to Section 0.1.8 SCI Mode Register Description.

Bit 3 = **LHE** LIN Header Error.

During LIN Header this bit signals three error types:

- The LIN Synch Field is corrupted and the SCI is blocked in LIN Synch State (LSF bit = 1).
- A timeout occurred during LIN Header reception
- An overrun error was detected on one of the header field (see OR bit description in Section 0.1.8 SCI Mode Register Description).

An interrupt is generated if RIE = 1 in the SCICR2 register. If blocked in the LIN Synch State, the LSF bit must first be reset (to exit LIN Synch Field state and then to be able to clear LHE flag). Then it is cleared by the following software sequence: An access to the SCISR register followed by a read to the SCIDR register.

0: No LIN Header error

1: LIN Header error detected

**Note:**

Apart from the LIN Header this bit signals an Overrun Error as in SCI mode (see description in Section 0.1.8 SCI Mode Register Description).

Bit 2 = **NF** Noise flag

In LIN Master mode (LINE bit = 1 and LSLV bit = 0), this bit has the same function as in SCI mode; please refer to Section 0.1.8 SCI Mode Register Description.

In LIN Slave mode (LINE bit = 1 and LSLV bit = 1) this bit has no meaning.

Bit 1 = **FE** Framing error.

In LIN slave mode, this bit is set only when a real

framing error is detected (if the stop bit is dominant (0) and at least one of the other bits is recessive (1). It is not set when a break occurs, the LHDF bit is used instead as a break flag (if the LHDM bit = 0). It is cleared by a software sequence (an access to the SCISR register followed by a read to the SCIDR register).

0: No Framing error

1: Framing error detected

Bit 0 = **PE** Parity error.

This bit is set by hardware when a LIN parity error occurs (if the PCE bit is set) in receiver mode. It is cleared by a software sequence (a read to the status register followed by an access to the SCIDR data register). An interrupt is generated if PIE = 1 in the SCICR1 register.

0: No LIN parity error

1: LIN Parity error detected

**CONTROL REGISTER 1 (SCICR1)**

Read/Write

Reset Value: x000 0000 (x0h)

7							0
R8	T8	SCID	M	WAKE	PCE	PS	PIE

Bits 7:3 = Same function as in SCI mode; please refer to Section 0.1.8 SCI Mode Register Description.

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

This bit is set and cleared by software. It selects the hardware parity control for LIN identifier parity check.

0: Parity control disabled

1: Parity control enabled

When a parity error occurs, the PE bit in the SCISR register is set.

Bit 1 = Reserved

Bit 0 = Same function as in SCI mode; please refer to Section 0.1.8 SCI Mode Register Description.

**beCAN CONTROLLER (Cont'd)****TRANSMIT ERROR COUNTER REG. (TECR)**

Read Only

Reset Value: 00h

7							0
TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

**TEC[7:0]** is the least significant byte of the 9-bit Transmit Error Counter implementing part of the fault confinement mechanism of the CAN protocol.

**RECEIVE ERROR COUNTER REG. (RECR)**

Page: 00h — Read Only

Reset Value: 00h

7							0
REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0

**REC[7:0]** is the Receive Error Counter implementing part of the fault confinement mechanism of the CAN protocol. In case of an error during reception, this counter is incremented by 1 or by 8 depending on the error condition as defined by the CAN standard. After every successful reception the counter is decremented by 1 or reset to 120 if its value was higher than 128. When the counter value exceeds 127, the CAN controller enters the error passive state.

**CAN DIAGNOSIS REGISTER (CDGR)**

All bits of this register are set and clear by software.

Read / Write

Reset Value: 0000 1100 (0Ch)

7							0
0	0	0	0	RX	SAMP	SILM	LBKM

Bit 3 = **RX** CAN Rx Signal

- Read

Monitors the actual value of the **CAN\_RX** Pin.Bit 2 = **SAMP** Last Sample Point

- Read

The value of the last sample point.

Bit 1 = **SILM** Silent Mode

- Read/Set/Clear

0: Normal operation

1: Silent Mode

Bit 0 = **LBKM** Loop Back Mode

- Read/Set/Clear

0: Loop Back Mode disabled

1: Loop Back Mode enabled

**CAN BIT TIMING REGISTER 0 (CBTR0)**

This register can only be accessed by the software when the CAN hardware is in configuration mode.

Read / Write

Reset Value: 0000 0000 (00h)

7							0
SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0

Bit 7:6 **SJW[1:0]** Resynchronization Jump Width

These bits define the maximum number of time quanta the CAN hardware is allowed to lengthen or shorten a bit to perform the resynchronization. Resynchronization Jump Width = (SJW+1).

Bit 5:0 **BRP[5:0]** Baud Rate Prescaler

These bits define the length of a time quantum.

 $tq = (BRP+1)/f_{CPU}$ 

For more information on bit timing, please refer to Section 0.1.4.6 Bit Timing.

**CAN BIT TIMING REGISTER 1 (CBTR1)**

Read / Write

Reset Value: 0001 0011 (23h)

7							0
0	BS22	BS21	BS20	BS13	BS12	BS11	BS10

Bit 7 = Reserved. Forced to 0 by hardware.

Bits 6:4 **BS2[2:0]** Time Segment 2

These bits define the number of time quanta in Time Segment 2.

Time Segment 2 = (BS2+1)

**beCAN CONTROLLER (Cont'd)****10.9.8.2 Mailbox Registers**

This chapter describes the registers of the transmit and receive mailboxes. Refer to Section 0.1.4.4 Message Storage for detailed register mapping.

Transmit and receive mailboxes have the same registers except:

- MCSR register in a transmit mailbox is replaced by MFMI register in a receive mailbox.
- A receive mailbox is always write protected.
- A transmit mailbox is write enable only while empty, corresponding TME bit in the CTPR register set.

**MAILBOX CONTROL STATUS REGISTER  
(MCSR)**

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	0	TERR	ALST	TXOK	RQCP	ABRQ	TXRQ

Bits 7:6 = Reserved. Forced to 0 by hardware.

**Bit 5 = TERR Transmission Error**

- Read

This bit is updated by hardware after each transmission attempt.

0: The previous transmission was successful

1: The previous transmission failed due to an error

**Bit 4 = ALST Arbitration Lost**

- Read

This bit is updated by hardware after each transmission attempt.

0: The previous transmission was successful

1: The previous transmission failed due to an arbitration lost

**Bit 3 = TXOK Transmission OK**

- Read

The hardware updates this bit after each transmission attempt.

0: The previous transmission failed

1: The previous transmission was successful

**Note:** This bit has the same value as the corresponding TXOKx bit in the CTSR register.

**Bit 2 = RQCP Request Completed**

- Read/Clear

Set by hardware when the last request (transmit or abort) has been performed.

Cleared by software writing a "1" or by hardware on transmission request.

**Note:** This bit has the same value as the corresponding RQCPx bit of the CTSR register.

**Clearing this bit clears all the status bits (TXOK, ALST and TERR) in the MCSR register and the RQCP and TXOK bits in the CTSR register.**

**Bit 1 = ABRQ Abort Request for Mailbox**

- Read/Set

Set by software to abort the transmission request for the corresponding mailbox.

Cleared by hardware when the mailbox becomes empty.

Setting this bit has no effect when the mailbox is not pending for transmission.

**Bit 0 = TXRQ Transmit Mailbox Request**

- Read/Set

Set by software to request the transmission for the corresponding mailbox.

Cleared by hardware when the mailbox becomes empty.

**Note:** This register is implemented only in transmit mailboxes. In receive mailboxes, the MFMI register is mapped at this location.

**beCAN CONTROLLER (Cont'd)****CAN FILTER MODE REGISTER (CFMR1)**

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

Read / Write

Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	FMH5	FML5	FMH4	FML4

Bits 7:4 = Reserved. Forced to 0 by hardware.

Bit 3 = **FMH5** *Filter Mode High*

Mode of the high registers of Filter 5.

0: High registers are in mask mode

1: High registers are in identifier list mode

Bit 2 = **FML5** *Filter Mode Low*

Mode of the low registers of filter 5.

0: Low registers are in mask mode

1: Low registers are in identifier list mode

Bit 1 = **FMH4** *Filter Mode High*

Mode of the high registers of filter 4.

0: High registers are in mask mode

1: High registers are in identifier list mode

Bit 0 = **FML4** *Filter Mode Low*

Mode of the low registers of filter 4.

0: Low registers are in mask mode

1: Low registers are in identifier list mode

**FILTER x REGISTER[7:0] (CFxR[7:0])**

Read / Write

Reset Value: Undefined

7							0
FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0

**In all configurations:**

Bits 7:0 = **FB[7:0]** *Filter Bits*

**Identifier**

Each bit of the register specifies the level of the corresponding bit of the expected identifier.

0: Dominant bit is expected

1: Recessive bit is expected

**Mask**

Each bit of the register specifies whether the bit of the associated identifier register must match with the corresponding bit of the expected identifier or not.

0: Don't care, the bit is not used for the comparison

1: Must match, the bit of the incoming identifier must have the same level as specified in the corresponding identifier register of the filter.

**Note:** Each filter x is composed of 8 registers, CFxR[7:0]. Depending on the scale and mode configuration of the filter the function of each register can differ. For the filter mapping, functions description and mask registers association, refer to Section 0.1.4.3 Identifier Filtering.

A Mask/Identifier register in **mask mode** has the same bit mapping as in **identifier list** mode.

**Note:** To modify these registers, the corresponding FACT bit in the CFCR register must be cleared.

## 10.10 10-BIT A/D CONVERTER (ADC)

### 10.10.1 Introduction

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

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

### 10.10.2 Main Features

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

The block diagram is shown in Figure 116.

### 10.10.3 Functional Description

#### 10.10.3.1 Digital A/D Conversion Result

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

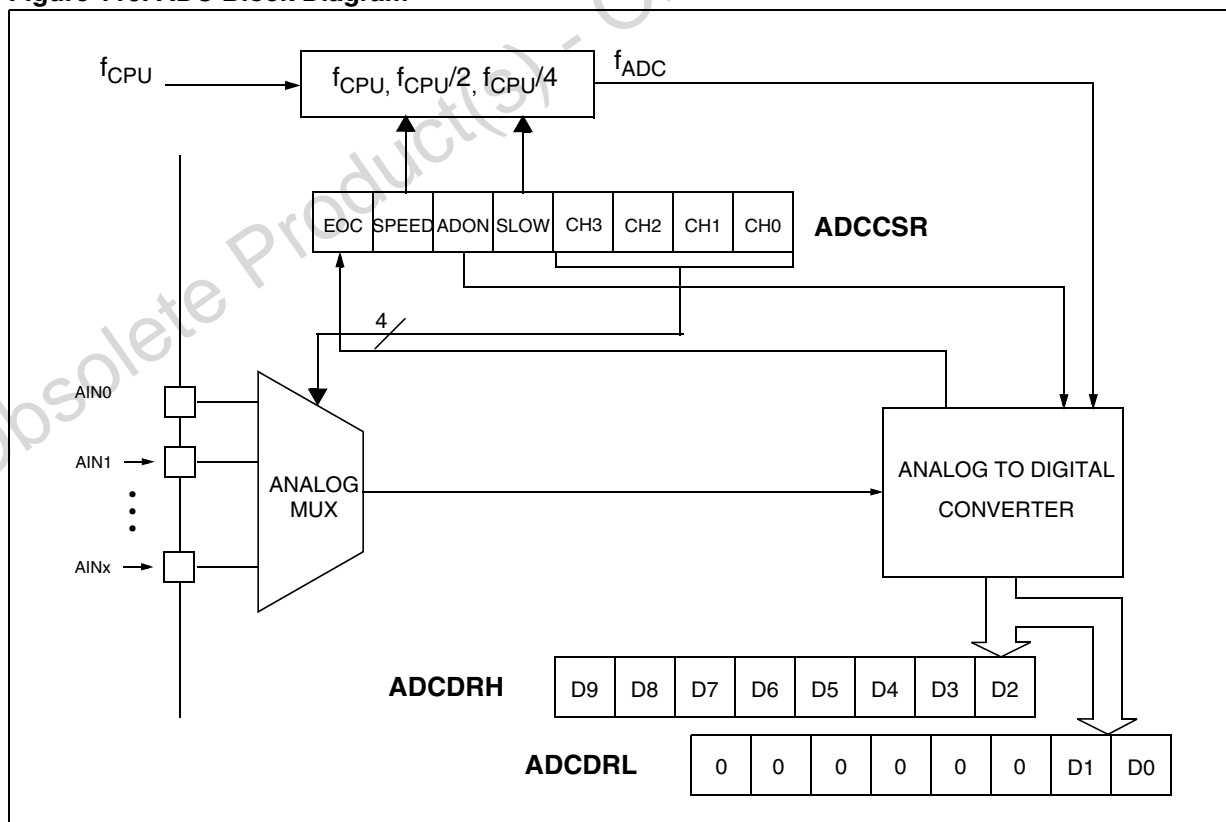
If the input voltage ( $V_{AIN}$ ) is greater than  $V_{DDA}$  (high-level voltage reference) then the conversion result is FFh in the ADCDRH register and 03h in the ADCDRL register (without overflow indication).

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

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

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

Figure 116. ADC Block Diagram



## 10-BIT A/D CONVERTER (ADC) (Cont'd)

### 10.10.3.2 A/D Conversion

The analog input ports must be configured as input, no pull-up, no interrupt. Refer to the "I/O ports" chapter. Using these pins as analog inputs does not affect the ability of the port to be read as a logic input.

In the ADCCSR register:

- Select the CS[3:0] bits to assign the analog channel to convert.

### ADC Conversion mode

In the ADCCSR register:

- Set the ADON bit to enable the A/D converter and to start the conversion. From this time on, the ADC performs a continuous conversion of the selected channel.

When a conversion is complete:

- The EOC bit is set by hardware.
- The result is in the ADCDR registers.

A read to the ADCDRH resets the EOC bit.

To read the 10 bits, perform the following steps:

1. Poll EOC bit
2. Read the ADCDRL register
3. Read the ADCDRH register. This clears EOC automatically.

To read only 8 bits, perform the following steps:

1. Poll EOC bit
2. Read the ADCDRH register. This clears EOC automatically.

### 10.10.3.3 Changing the conversion channel

The application can change channels during conversion. When software modifies the CH[3:0] bits

in the ADCCSR register, the current conversion is stopped, the EOC bit is cleared, and the A/D converter starts converting the newly selected channel.

### 10.10.3.4 ADCDR consistency

If an End Of Conversion event occurs after software has read the ADCDRLSB but before it has read the ADCDRMSB, there would be a risk that the two values read would belong to different samples.

To guarantee consistency:

- The ADCDRL and the ADCDRH registers are locked when the ADCCRL is read
- The ADCDRL and the ADCDRH registers are unlocked when the ADCDRH register is read or when ADON is reset.

This is important, as the ADCDR register will not be updated until the ADCDRH register is read.

### 10.10.4 Low Power Modes

**Note:** The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed and between single shot conversions.

Mode	Description
WAIT	No effect on A/D Converter
HALT	A/D Converter disabled. After wakeup from Halt mode, the A/D Converter requires a stabilization time $t_{STAB}$ (see Electrical Characteristics) before accurate conversions can be performed.

### 10.10.5 Interrupts

None.

## 16 IMPORTANT NOTES

### 16.1 ALL DEVICES

#### 16.1.1 RESET Pin Protection with LVD Enabled

As mentioned in note 2 below Figure 134 on page 240, when the LVD is enabled, it is recommended not to connect a pull-up resistor or capacitor. A 10nF pull-down capacitor is required to filter noise on the reset line.

#### 16.1.2 Clearing Active Interrupts Outside Interrupt Routine

When an active interrupt request occurs at the same time as the related flag or interrupt mask is being cleared, the CC register may be corrupted.

##### Concurrent interrupt context

The symptom does not occur when the interrupts are handled normally, that is, when:

- The interrupt request is cleared (flag reset or interrupt mask) within its own interrupt routine
- The interrupt request is cleared (flag reset or interrupt mask) within any interrupt routine
- The interrupt request is cleared (flag reset or interrupt mask) in any part of the code while this interrupt is disabled

If these conditions are not met, the symptom can be avoided by implementing the following sequence:

Perform SIM and RIM operation before and after resetting an active interrupt request

Example:

SIM

reset flag or interrupt mask

RIM

##### Nested interrupt context

The symptom does not occur when the interrupts are handled normally, that is, when:

- The interrupt request is cleared (flag reset or interrupt mask) within its own interrupt routine
- The interrupt request is cleared (flag reset or interrupt mask) within any interrupt routine with higher or identical priority level
- The interrupt request is cleared (flag reset or interrupt mask) in any part of the code while this interrupt is disabled

If these conditions are not met, the symptom can be avoided by implementing the following sequence:

PUSH CC

SIM

reset flag or interrupt mask

POP CC

#### 16.1.3 External Interrupt Missed

To avoid any risk of generating a parasitic interrupt, the edge detector is automatically disabled for one clock cycle during an access to either DDR and OR. Any input signal edge during this period will not be detected and will not generate an interrupt.

This case can typically occur if the application refreshes the port configuration registers at intervals during runtime.

##### Workaround

The workaround is based on software checking the level on the interrupt pin before and after writing to the PxOR or PxDDR registers. If there is a level change (depending on the sensitivity programmed for this pin) the interrupt routine is invoked using the call instruction with three extra PUSH instructions before executing the interrupt routine (this is to make the call compatible with the IRET instruction at the end of the interrupt service routine).

But detection of the level change does ensure that edge occurs during the critical 1 cycle duration and the interrupt has been missed. This may lead to occurrence of same interrupt twice (one hardware and another with software call).

To avoid this, a semaphore is set to '1' before checking the level change. The semaphore is changed to level '0' inside the interrupt routine. When a level change is detected, the semaphore status is checked and if it is '1' this means that the last interrupt has been missed. In this case, the interrupt routine is invoked with the call instruction.

There is another possible case, that is, if writing to PxOR or PxDDR is done with global interrupts disabled (interrupt mask bit set). In this case, the semaphore is changed to '1' when the level change is detected. Detecting a missed interrupt is done after the global interrupts are enabled (interrupt mask bit reset) and by checking the status of the semaphore. If it is '1' this means that the last interrupt was missed and the interrupt routine is invoked with the call instruction.

To implement the workaround, the following software sequence is to be followed for writing into the PxOR/PxDDR registers. The example is for Port PF1 with falling edge interrupt sensitivity. The