STMicroelectronics - ST92F150CV1TB Datasheet

Welcome to E-XFL.COM

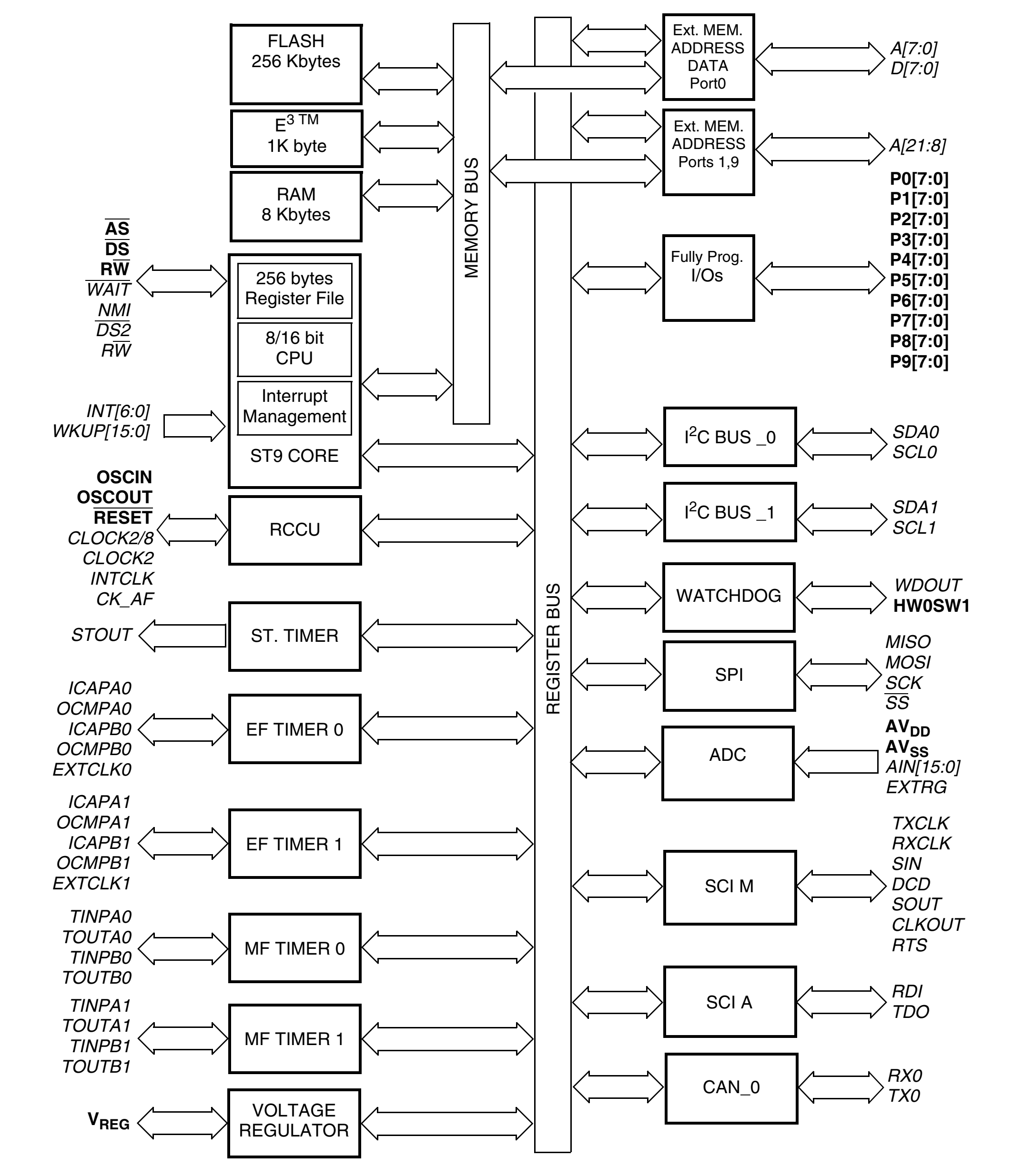
What is "Embedded - Microcontrollers"?

"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"

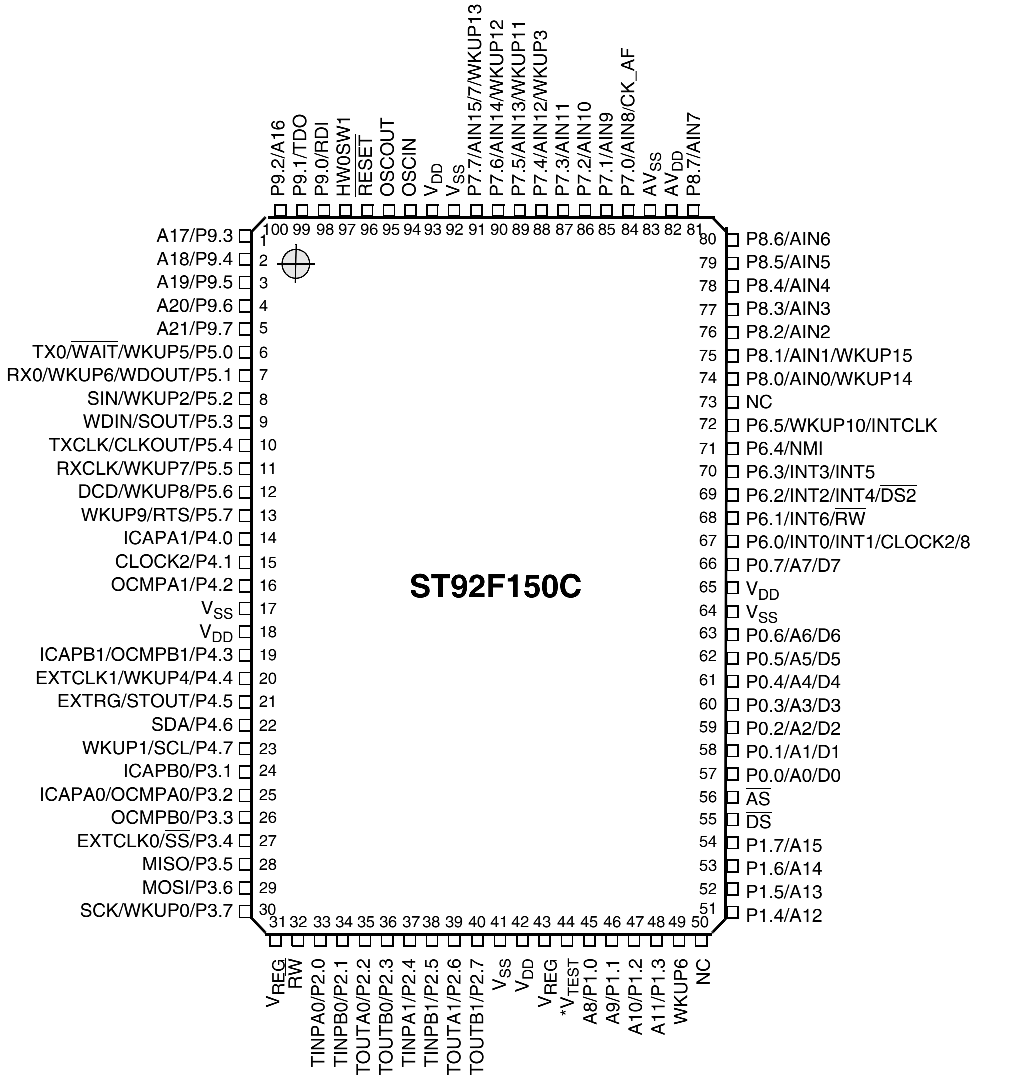
Details	
Product Status	Obsolete
Core Processor	ST9
Core Size	8/16-Bit
Speed	24MHz
Connectivity	CANbus, EBI/EMI, I²C, LINbus, SCI, SPI
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	77
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st92f150cv1tb

Figure 5. ST92F250CV2: Architectural block diagram



The alternate functions (*Italic characters*) are mapped on Port0, Port1, Port2, Port3, Port4, Port5, Port6, Port7, Port8 and Port9.

Figure 10. ST92F150C: Pin configuration (top-view PQFP100)



* V_{TEST} must be kept low in standard operating mode.

Bit 6 = **Z:** *Zero Flag*. The Zero flag is affected by:

Addition (`add, addw, adc, adcw`),
Subtraction (`sub, subw, sbc, sbcw`),
Compare (`cp, cpw`),
Shift Right Arithmetic (`sra, sraw`),
Shift Left Arithmetic (`sla, slaw`),
Swap Nibbles (`swap`),
Rotate (rrc`, rrcw, rlc, rlcw, ror, rol),`
Decimal Adjust (`da`),
Multiply and Divide (`mul, div, divws`),
Logical (`and, andw, or, orw, xor, xorw, cpl`),
Increment and Decrement (`inc, incw, dec, decw`),

Test (`tm, tmw, tcm, tcmw, btset`).

In most cases, the Zero flag is set when the contents of the register being used as an accumulator become zero, following one of the above operations.

Bit 5 = **S**: *Sign Flag*.
The Sign flag is affected by the same instructions as the Zero flag.

The Sign flag is set when bit 7 (for a byte operation) or bit 15 (for a word operation) of the register used as an accumulator is one.

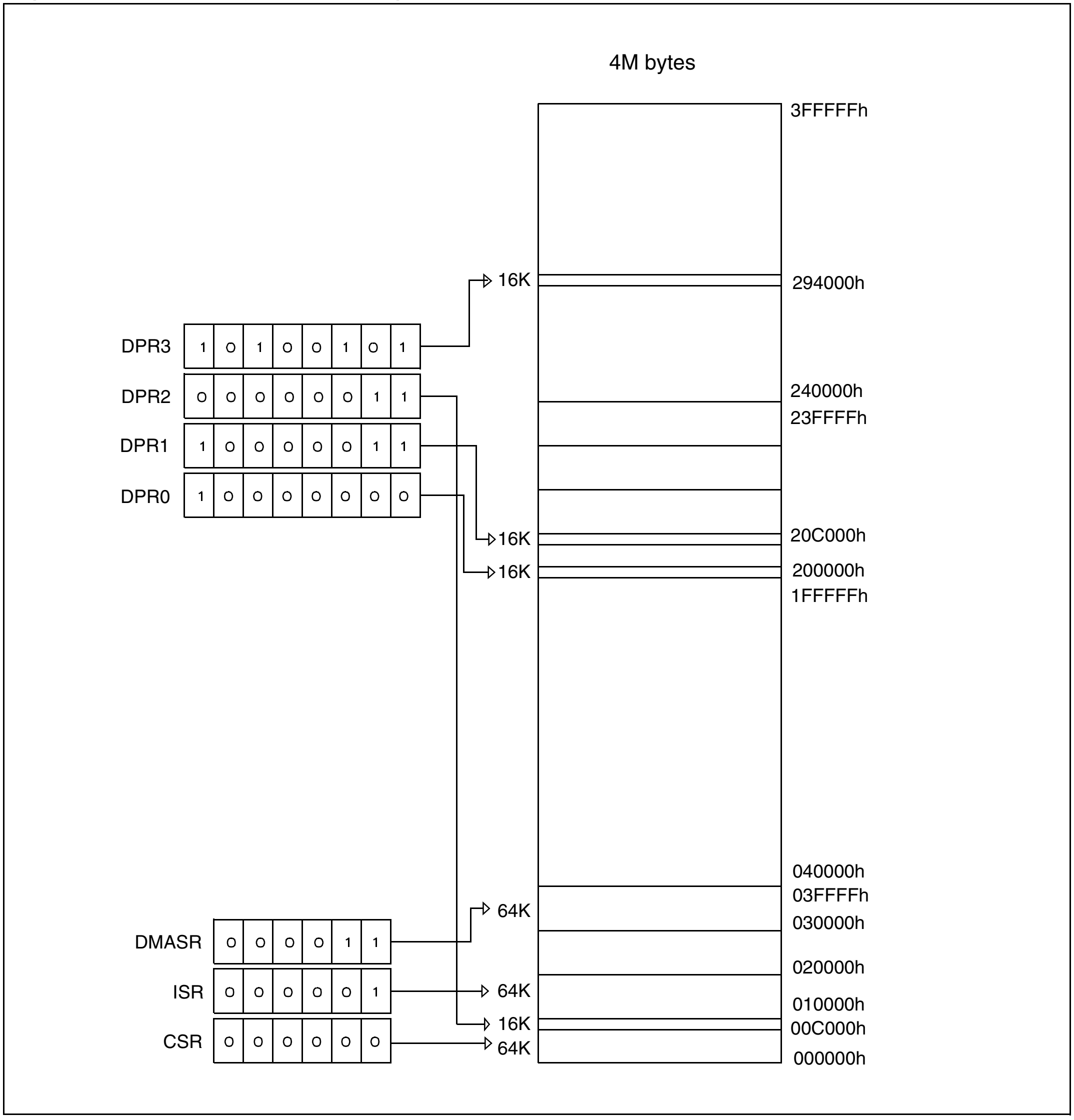
Bit 4 = **V**: *Overflow Flag*.
The Overflow flag is affected by the same instructions as the Zero and Sign flags.

When set, the Overflow flag indicates that a two's-complement number, in a result register, is in error, since it has exceeded the largest (or is less than the smallest), number that can be represented in two’s-complement notation.

Bit 3 = **DA**: *Decimal Adjust Flag*.
The DA flag is used for BCD arithmetic. Since the algorithm for correcting BCD operations is different for addition and subtraction, this flag is used to specify which type of instruction was executed last, so that the subsequent Decimal Adjust (`da`) operation can perform its function correctly. The DA flag cannot normally be used as a test condition by the programmer.

Bit 2 = **H**: *Half Carry Flag.*
The H flag indicates a carry out of (or a borrow into) bit 3, as the result of adding or subtracting two 8-bit bytes, each representing two BCD digits. The H flag is used by the Decimal Adjust (`da`) instruction to convert the binary result of a previous addition or subtraction into the correct BCD result. Like the DA flag, this flag is not normally accessed by the user.

Figure 29. Memory addressing scheme (example)



6.8 MMU usage

6.8.1 Normal program execution

Program memory is organized as a set of 64-Kbyte segments. The program can span as many segments as needed, but a procedure cannot stretch across segment boundaries. `jps`, `calls` and `rets` instructions, which automatically modify the CSR, must be used to jump across segment boundaries. Writing to the CSR is forbidden during normal program execution because it is not synchronized with the opcode fetch. This could result in fetching the first byte of an instruction from one memory segment and the second byte from another. Writing to the CSR is allowed when it is not being used, i.e during an interrupt service routine if ENCSR is reset.

Table 11. Memory structure for 256K Flash device (continued)

Sector	Addresses	Max size
Hardware Emulated EEPROM sectors (reserved)	228000h to 22CFFFh	8 Kbytes
Emulated EEPROM	220000h to 2203FFh	1 Kbyte

7.2.3 Operation

The memory has a register interface mapped in memory space (segment 22h). All operations are enabled through the FCR (Flash Control Register), ECR (E$^{3\ \mathrm{TM}}$ Control Register).

All operations on the Flash must be executed from another memory (internal RAM, E$^{3\ \mathrm{TM}}$, external memory).

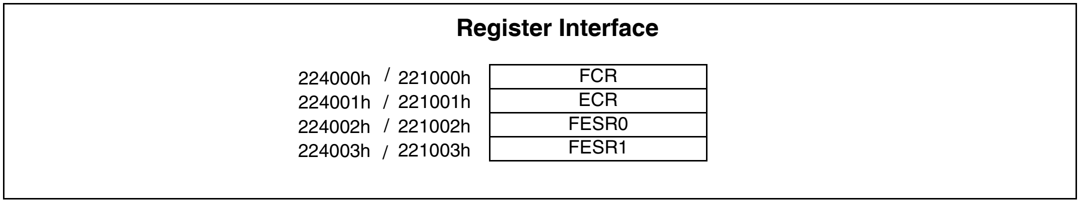
Flash (including TestFlash) and E$^{3\ \mathrm{TM}}$ are independent, this means that one can be read while the other is written. However simultaneous Flash and E$^{3\ \mathrm{TM}}$ write operations are forbidden.

An interrupt can be generated at the end of a Flash or an E$^{3\ \mathrm{TM}}$ write operation: this interrupt is multiplexed with an external interrupt EXTINTx (device dependent) to generate an interrupt INTx.

The status of a write operation inside the Flash and the E$^{3\ \mathrm{TM}}$ memories can be monitored through the FESR[1:0] registers.

Control and Status registers are mapped in memory (segment 22h), as shown in the following figure.

Figure 32. Control and status register map



Register Interface

224000h / 221000h	FCR
224001h / 221001h	ECR
224002h / 221002h	FESR0
224003h / 221003h	FESR1

In order to use the same data pointer register (DPR) to point both to the E$^{3\ \mathrm{TM}}$ (220000h-2203FFh) and to these control and status registers, the Flash and E$^{3\ \mathrm{TM}}$ control registers are mapped not only at page 0x89 (224000h-224003h) but also on page 0x88 (221000h-221003h).

If the $\overline{\text{RESET}}$ pin is activated during a write operation, the write operation is interrupted. In this case the user must repeat this last write operation following power on or reset. If the internal supply voltage drops below the V_{IT-} threshold, a reset sequence is generated automatically by hardware.

7.2.4 E$^{3\ \mathrm{TM}}$ update operation

The update of the E$^{3\ \mathrm{TM}}$ content can be made by pages of 16 consecutive bytes. The Page Update operation allows up to 16 bytes to be loaded into the RAM buffer that replace the ones already contained in the specified address.

Note: *The ITExx bits must be set to enable the CAN interrupts as the CAN interrupt events are rising edge events.*

If either a rising or a falling edge occurs on the interrupt lines during a write access to the ITER register, the pending bit will not be set.

INTERRUPT PENDING REGISTER HIGH (SIPRH)
R249 - Read/Write
Register Page: 60
Reset value: 0000 0000 (00h)

7							0
-	-	-	-	-	-	-	IPI0

Bits 7:1 = Reserved.

Bit 0 = **IPI0** *Channel I0 Pending bit*

The IPI0 bit is set by hardware on occurrence of the trigger event. (as specified in the ITR register) and is cleared by hardware on interrupt acknowledge.

0: No interrupt pending

1: Interrupt pending

INTERRUPT PENDING REGISTER LOW (SIPRL)
R250 - Read/Write
Register Page: 60
Reset value: 0000 0000 (00h)

7							0
IPH1	IPH0	IPG1	IPG0	IPF1	IPF0	IPE1	IPE0

Bits 7:0 = **IPxx** *Channel E-H Pending bits*
The IPxx bits are set by hardware on occurrence of the trigger event. (as specified in the ITR register) and are cleared by hardware on interrupt acknowledge.

0: No interrupt pending

1: Interrupt pending

Note: *IPR bits may be set by the user to implement a software interrupt.*

STANDARD INTERRUPT VECTOR REGISTER (SIVR)
R251 - Read/Write
Register Page: 60
Reset value: xxx1 1110 (xE)

7							0
V7	V6	V5	W3	W2	W1	W0	0

Bits 7:5 = **V[7:5]** *MSBs of Channel E to L interrupt vector address*

10.6 DMA registers

As each peripheral DMA channel has its own specific control registers, the following register list should be considered as a general example. The names and register bit allocations shown here may be different from those found in the peripheral chapters.

DMA COUNTER POINTER REGISTER (DCPR)
Read/Write
Address set by Peripheral
Reset value: undefined

7							0
C7	C6	C5	C4	C3	C2	C1	RM

Bit 7:1 = **C[7:1]**: *DMA Transaction Counter Pointer.*

Software should write the pointer to the DMA Transaction Counter in these bits.

Bit 0 = **RM**: *Register File/Memory Selector.*
This bit is set and cleared by software.

0: DMA transactions are with memory (see also DAPR.DP)

1: DMA transactions are with the Register File

GENERIC EXTERNAL PERIPHERAL INTERRUPT AND DMA CONTROL (IDCR)
Read/Write
Address set by Peripheral
Reset value: undefined

7							0
		IP	DM	IM	PRL2	PRL1	PRL0

Bit 5 = **IP**: *Interrupt Pending*.

This bit is set by hardware when the Trigger Event occurs. It is cleared by hardware when the request is acknowledged. It can be set/cleared by software in order to generate/cancel a pending request.

0: No interrupt pending

1: Interrupt pending

Bit 4 = **DM**: *DMA Request Mask*.

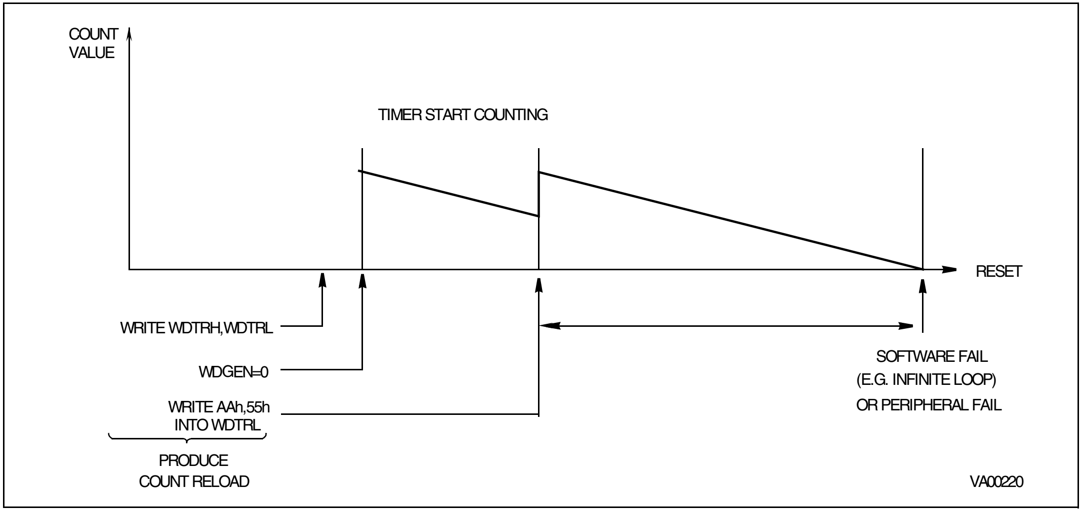
This bit is set and cleared by software. It is also cleared when the transaction counter reaches zero (unless SWAP mode is active).

0: No DMA request is generated when IP is set.

1: DMA request is generated when IP is set

Bit 3 = **IM**: *End of block Interrupt Mask*.
This bit is set and cleared by software.

Figure 86. Watchdog timer mode



14.1.4 WDT interrupts

The Timer/Watchdog issues an interrupt request at every End of Count, when this feature is enabled.

A pair of control bits, IA0S (EIVR.1, Interrupt A0 selection bit) and TLIS (EIVR.2, Top Level Input Selection bit) allow the selection of 2 interrupt sources (Timer/Watchdog End of Count, or External Pin) handled in two different ways, as a Top Level Non Maskable Interrupt (Software Reset), or as a source for channel A0 of the external interrupt logic.

A block diagram of the interrupt logic is given in *Figure 87*.

Note: Software traps can be generated by setting the appropriate interrupt pending bit.

Table 45 below shows all the possible configurations of interrupt/reset sources which relate to the Timer/Watchdog.

A reset caused by the watchdog will set bit 6, WDGRES of R242 - Page 55 (Clock Flag Register). See *Section 11.4: Clock control registers*.

WDTCR: Timer/Watchdog Control Register

Three additional control bits are mapped in the following registers on Page 0:

Watchdog Mode Enable, (WCR.6)

Top Level Interrupt Selection, (EIVR.2)

Interrupt A0 Channel Selection, (EIVR.1)

Note: *The registers containing these bits also contain other functions. Only the bits relevant to the operation of the Timer/Watchdog are shown here.*

Counter Register

This 16-bit register (WDTLR, WDTHR) is used to load the 16-bit counter value. The registers can be read or written “on the fly”.

TIMER/WATCHDOG HIGH REGISTER (WDTHR)
R248 - Read/Write
Register Page: 0
Reset value: 1111 1111 (FFh)

7							0
R15	R14	R13	R12	R11	R10	R9	R8

Bits 7:0 = **R[15:8]** *Counter Most Significant Bits*.

TIMER/WATCHDOG LOW REGISTER (WDTLR)
R249 - Read/Write
Register Page: 0
Reset value: 1111 1111b (FFh)

7							0
R7	R6	R5	R4	R3	R2	R1	R0

Bits 7:0 = **R[7:0]** *Counter Least Significant Bits.*

TIMER/WATCHDOG PRESCALER REGISTER (WDTPR)
R250 - Read/Write
Register Page: 0
Reset value: 1111 1111 (FFh)

7							0
PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0

Bits 7:0 = **PR[7:0]** *Prescaler value.*
A programmable value from 1 (00h) to 256 (FFh).

Bit 7 = **ST-SP**: *Start-Stop Bit.*
This bit is set and cleared by software.

0: Stop counting

1: Start counting

Bit 6 = **S-C**: *Single-Continuous Mode Select.*
This bit is set and cleared by software.

0: Continuous Mode

1: Single Mode

Bits 5:4 = **INMD[1:2]**
Bit 3 = **INEN**
These 3 bits select the clock source.

Table 47. Clock input

INMD1	INMD2	INEN	Clock input
0	0	1	CLOCK2/1024
X	X	0	INTCLK/4

Bit 2 = **INTS**: *Interrupt Selection.*

0: Standard Timer interrupt enabled

1: Standard Timer interrupt is disabled and the external interrupt pin is enabled.

Bits 1:0 = **OUTMD[1:2]**: Output Mode Selection.
These bits select the output functions as described in *Standard timer output modes*.

Table 48. Output modes

OUTMD1	OUTMD2	Mode
0	0	No output mode
0	1	Square wave output mode
1	x	PWM output mode

14.3 Extended function timer (EFT)

14.3.1 Introduction

The timer consists of a 16-bit free-running counter driven by a programmable prescaler.

It may be used for a variety of purposes, including pulse length measurement of up to two input signals (*input capture*) or generation of up to two output waveforms (*output compare* and *PWM*).

INPUT CAPTURE 2 HIGH REGISTER (IC2HR)

R242 - Read Only
Register Page: 28
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)

R243 - Read Only
Register Page: 28
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).

7							0
MSB							LSB

COUNTER HIGH REGISTER (CHR)

R244 - Read Only
Register Page: 28
Reset Value: 1111 1111 (FFh)

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

7							0
MSB							LSB

COUNTER LOW REGISTER (CLR)

R245 - Read/Write
Register Page: 28
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 SR register clears the TOF bit.

7							0
MSB							LSB

ALTERNATE COUNTER HIGH REGISTER (ACHR)

R246 - Read Only
Register Page: 28
Reset Value: 1111 1111 (FFh)

Table 52. Bi-value modes

TMR bits			Timer operating modes
RM0	RM1	BM	
0 1	X X	1 1	Bi-Load mode Bi-Capture Mode

a) Biload Mode

The Bi-load Mode is entered by selecting the Bivalue Mode (BM set in TMR) and programming REG0R as a reload register (RM0 reset in TMR).

At any End Of Count, counter reloading is performed alternately from REG0R and REG1R, (a low level for BM bit always sets REG0R as the current register, so that, after a Low to High transition of BM bit, the first reload is always from REG0R).

Every software or external trigger event on REG0R performs a reload from REG0R resetting the Biload cycle. In One Shot mode (reload initiated by software or by an external trigger), reloading is always from REG0R.

b) Bicapture Mode

The Bicapture Mode is entered by selecting the Bivalue Mode (the BM bit in TMR is set) and by programming REG0R as a capture register (the RM0 bit in TMR is set).

Interrupt generation can be configured as an AND or OR function of the two Capture events. This is configured by the A0 bit in the T_FLAGR register.

Every capture event, software simulated (by setting the CP0 flag) or coming directly from the TxINA input line, captures the current counter value alternately into REG0R and REG1R. When the BM bit is reset, REG0R is the current register, so that the first capture, after resetting the BM bit, is always into REG0R.

Parallel mode

When two MFTs are present on an ST9 device, the parallel mode is entered when the ECK bit in the TMR register of Timer 1 is set. The Timer 1 prescaler input is internally connected to the Timer 0 prescaler output. Timer 0 prescaler input is connected to the system clock line.

By loading the Prescaler Register of Timer 1 with the value 00h the two timers (Timer 0 and Timer 1) are driven by the same frequency in parallel mode. In this mode the clock frequency may be divided by a factor in the range from 1 to 2^{16}.

Autodiscriminator mode

The phase difference sign of two overlapping pulses (respectively on TxINB and TxINA) generates a one step up/down count, so that the up/down control and the counter clock are both external. The setting of the UDC bit in the TCR register has no effect in this configuration.

1: Configure the capture interrupt as an AND function of REG0R/REG1R captures

Note: When A0 is set, both CP0I and CP1I in the IDMR register must be set to enable both capture interrupts.

INTERRUPT/DMA MASK REGISTER (IDMR)
R255 - Read/Write
Register Page: 10
Reset value: 0000 0000 (00h)

7							0
GTIEN	CP0D	CP0I	CP1I	CM0D	CM0I	CM1I	OUI

Bit 7 = **GTIEN**: *Global timer interrupt enable*.
This bit is set and cleared by software.

0: Disable all Timer interrupts

1: Enable all timer Timer Interrupts from enabled sources

Bit 6 = **CP0D**: *Capture 0 DMA mask.*
This bit is set by software to enable a Capt0 DMA transfer and cleared by hardware at the end of the block transfer.

0: Disable capture on REG0R DMA

1: Enable capture on REG0R DMA

Bit 5 = **CP0I**: *Capture 0 interrupt mask*.

0: Disable capture on REG0R interrupt

1: Enable capture on REG0R interrupt (or Capt0 DMA End of Block interrupt if CP0D=1)

Bit 4 = **CP1I**: *Capture 1 interrupt mask*.
This bit is set and cleared by software.

0: Disable capture on REG1R interrupt

1: Enable capture on REG1R interrupt

Bit 3 = **CM0D**: *Compare 0 DMA mask.*
This bit is set by software to enable a Comp0 DMA transfer and cleared by hardware at the end of the block transfer.

0: Disable compare on CMP0R DMA

1: Enable compare on CMP0R DMA

Bit 2 = **CM0I**: *Compare 0 Interrupt mask*.
This bit is set and cleared by software.

Note: *Bit 5 =* ***DCTS****: DMA capture transfer source.*
This bit is set and cleared by software. It selects the source of the DMA operation related to the channel associated with the Capture 0.
The I/O port source is available only on specific devices.

0: REG0R register

1: I/O port.

Bit 4 = **DCTD**: DMA compare transfer destination.
This bit is set and cleared by software. It selects the destination of the DMA operation related to the channel associated with Compare 0.
The I/O port destination is available only on specific devices.

0: CMP0R register

1: I/O port

Bit 3 = **SWEN**: *Swap function enable*.
This bit is set and cleared by software.

0: Disable Swap mode

1: Enable Swap mode for both DMA channels.

Bits 2:0 = **PL[2:0]:** *Interrupt/DMA priority level*.
With these three bits it is possible to select the Interrupt and DMA priority level of each timer, as one of eight levels (see Interrupt/DMA chapter).

I/O CONNECTION REGISTER (IOCR)

R248 - Read/Write
Register Page: 9
Reset value: 1111 1100 (FCh)

7							0
						SC1	SC0

Bits 7:2 = not used.

Bit 1 = **SC1**: *Select connection odd.*
This bit is set and cleared by software. It selects if the TxOUTA and TxINA pins for Timer 1 and Timer 3 are connected on-chip or not.

0: T1OUTA / T1INA and T3OUTA/ T3INA unconnected

1: T1OUTA connected internally to T1INA and T3OUTA connected internally to T3INA

Bit 0 = **SC0**: *Select connection even.*
This bit is set and cleared by software. It selects if the TxOUTA and TxINA pins for Timer 0 and Timer 2 are connected on-chip or not.

ADDRESS/9TH: The Address/9th Bit is optional and may be added to any word format. It is used in both Serial Expansion and Asynchronous modes to indicate that the data is an address (bit set).

The ADDRESS/9TH bit is useful when several microcontrollers are exchanging data on the same serial bus. Individual microcontrollers can stay idle on the serial bus, waiting for a transmitted address. When a microcontroller recognizes its own address, it can begin Data Reception, likewise, on the transmit side, the microcontroller can transmit another address to begin communication with a different microcontroller.

The ADDRESS/9TH bit can be used as an additional data bit or to mark control words (9th bit).

STOP: Indicates the end of a data frame in Asynchronous modes. A dummy STOP bit is generated in Serial Expansion mode. The STOP bit can be programmed to be 1, 1.5, 2, 2.5 or 3 bits long, depending on the mode. It returns the SCI to the quiescent marking state (i.e., a constant high-state condition) which lasts until a new start bit indicates an incoming word. The STOP bit is not generated in Synchronous mode.

Figure 125. SCI signal

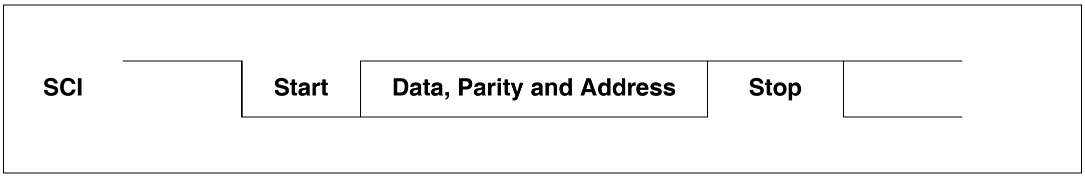


Table 63. SCI character formats

Data frame	START [1]	DATA [2]	PARITY [3]	ADDRESS [1]	STOP [1]	Mode
# Bits	1	5, 6, 7, 8	0, 1	0, 1	1, 1.5, 2, 2.5, 1, 2, 3	16X 1X
States			NONE ODD EVEN	ON OFF		

1. Not available in Synchronous mode
2. LSB First
3. Not available in Serial Expansion mode and Synchronous mode

Data transfer

Data to be transmitted by the SCI is first loaded by the program into the Transmitter Buffer Register. The SCI will transfer the data into the Transmitter Shift Register when the Shift Register becomes available (empty). The Transmitter Shift Register converts the parallel data into serial format for transmission via the SCI Alternate Function output, Serial Data Out. On completion of the transfer, the transmitter buffer register interrupt pending bit will be updated. If the selected word length is less than 8 bits, the unused most significant bits do not need to be defined.

Incoming serial data from the Serial Data Input pin is converted into parallel format by the Receiver Shift Register. At the end of the input data frame, the valid data portion of the received word is transferred from the Receiver Shift Register into the Receiver Buffer

Figure 156. CAN block diagram

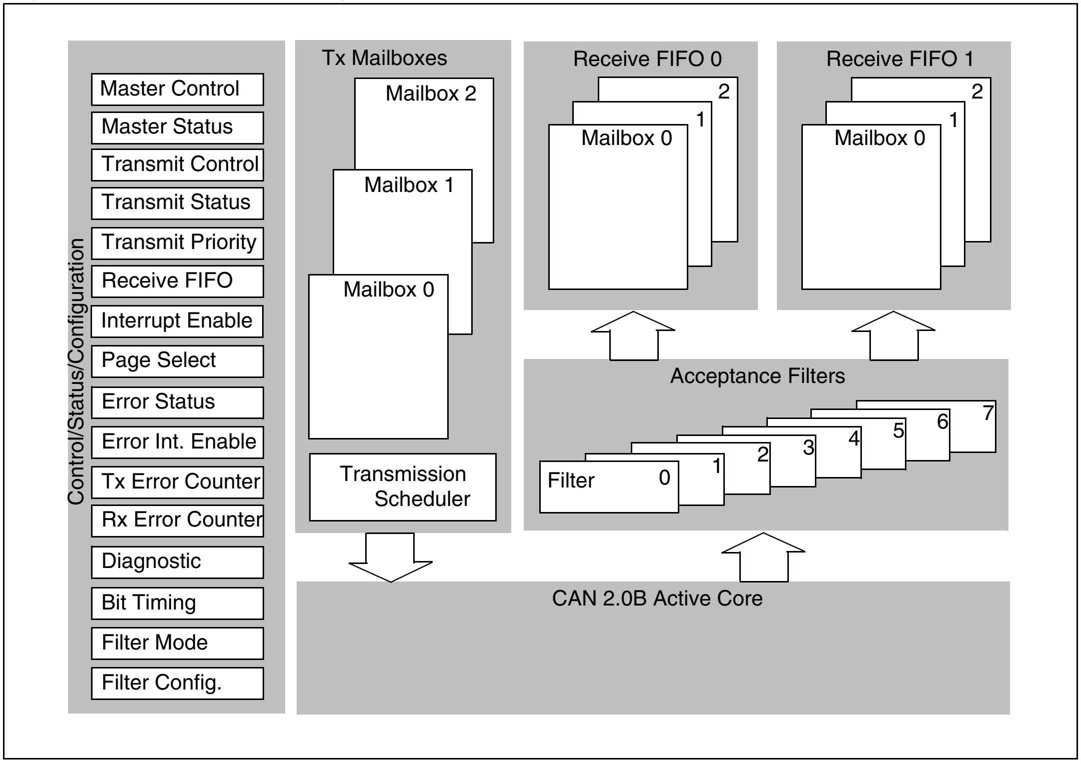
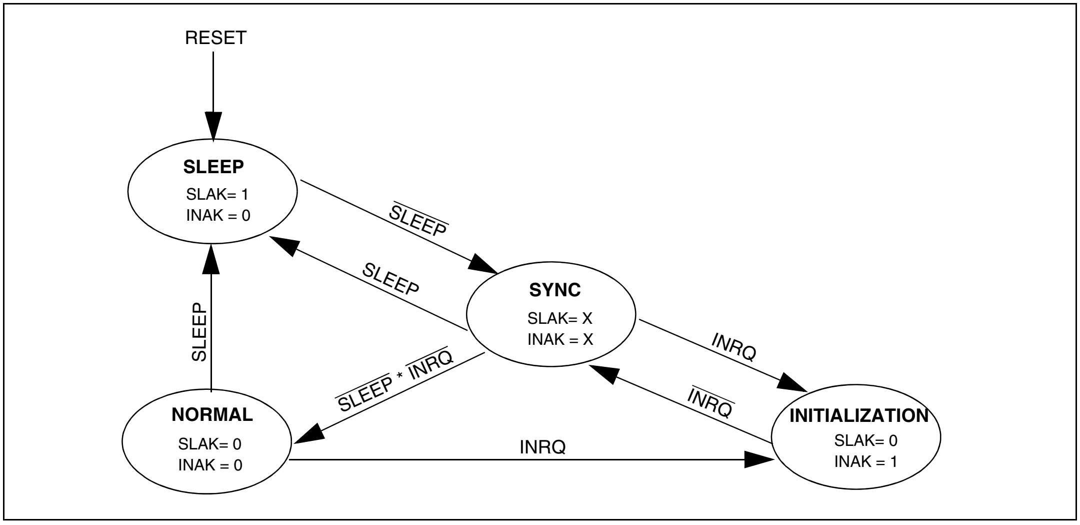


Figure 157. bxCAN operating modes



14.10.4 Operating modes

bxCAN has three main operating modes: **initialization**, **normal** and **sleep**. After a hardware reset, bxCAN is in sleep mode to reduce power consumption and an internal pull-up is active on RX1. The software requests bxCAN to enter **initialization** or **sleep** mode by setting the INRQ or SLEEP bits in the CMCR register. Once the mode has been entered, bxCAN confirms it by setting the INAK or SLAK bits in the CMSR register and the internal pull-up is disabled. When neither INAK nor SLAK are set, bxCAN is in **normal** mode. Before entering **normal** mode bxCAN always has to **synchronize** on the CAN bus. To synchronize,

Page mapping for CAN 0 / CAN 1

Figure 169. Page mapping for CAN 0 / CAN 1

	PAGE 48 / 36	PAGE 49 / 37	PAGE 50 / 38	PAGE 51 / 39	PAGE 52 / 40
240	CMCR	MFMI	MFMI	MCSR	MCSR
241	CMSR	MDLC	MDLC	MDLC	MDLC
242	CTSR	MIDR0	MIDR0	MIDR0	MIDR0
243	CTPR	MIDR1	MIDR1	MIDR1	MIDR1
244	CRFR0	MIDR2	MIDR2	MIDR2	MIDR2
245	CRFR1	MIDR3	MIDR3	MIDR3	MIDR3
246	CIER	MDAR0	MDAR0	MDAR0	MDAR0
247	CESR	MDAR1	MDAR1	MDAR1	MDAR1
248	CEIER	MDAR2	MDAR2	MDAR2	MDAR2
249	TEC	MDAR3	MDAR3	MDAR3	MDAR3
250	REC	MDAR4	MDAR4	MDAR4	MDAR4
251	CDGR	MDAR5	MDAR5	MDAR5	MDAR5
252	CBTR0	MDAR6	MDAR6	MDAR6	MDAR6
253	CBTR1	MDAR7	MDAR7	MDAR7	MDAR7
254	Reserved	MTSLR	MTSLR	MTSLR	MTSLR
255	CFPSR	MTSHR	MTSHR	MTSHR	MTSHR
	Control/Status	Receive FIFO 0	Receive FIFO 1	Tx Mailbox 0	Tx Mailbox 1

	PAGE 53 / 41	PAGE 54/4 42/4	PAGE 54/0 42/0	PAGE 54/1 42/1	PAGE 54/2 42/2
240	MCSR	CFMR0	CF0R0	CF2R0	CF4R0
241	MDLC	CFMR1	CF0R1	CF2R1	CF4R1
242	MIDR0	CFCR0	CF0R2	CF2R2	CF4R2
243	MIDR1	CFCR1	CF0R3	CF2R3	CF4R3
244	MIDR2	CFCR2	CF0R4	CF2R4	CF4R4
245	MIDR3	CFCR3	CF0R5	CF2R5	CF4R5
246	MDAR0	Reserved	CF0R6	CF2R6	CF4R6
247	MDAR1	Reserved	CF0R7	CF2R7	CF4R7
248	MDAR2	Reserved	CF1R0	CF3R0	CF5R0
249	MDAR3	Reserved	CF1R1	CF3R1	CF5R1
250	MDAR4	Reserved	CF1R2	CF3R2	CF5R2
251	MDAR5	Reserved	CF1R3	CF3R3	CF5R3
252	MDAR6	Reserved	CF1R4	CF3R4	CF5R4
253	MDAR7	Reserved	CF1R5	CF3R5	CF5R5
254	MTSLR	Reserved	CF1R6	CF3R6	CF5R6
255	MTSHR	Reserved	CF1R7	CF3R7	CF5R7
	Tx Mailbox 2	Filter Configuration	Acceptance Filter 0:1	Acceptance Filter 2:3	Acceptance Filter 4:5

Table 107. Prescaler programming

PR[2:0]	$T_{A/D}$ clock/T_{INTCLK}	f_{ADC} (MHz)	T_{Sample} (µs)	T_{Conv} (µs)	f_{ADC} (MHz)	T_{Sample} (µs)	T_{Conv} (µs)	f_{ADC} (MHz)	T_{Sample} (µs)	T_{Conv} (µs)
		@T_{INTCLK}= 8MHz			@T_{INTCLK}= 20MHz			@T_{INTCLK}=24MHz		
000	2	4.00	2	7	10.00	Not Allowed		12.00	Not Allowed	
001	4	2.00	4	14	5.00	Not Allowed		6.00	Not Allowed	
010	6	1.33	6	21	3.33	2.4	8.4	4.00	2	7
011	8	1.00	8	28	2.50	3.2	11.2	3.00	2.66	9.33
100	10	0.80	Not Allowed		2.00	4	14	2.40	3.33	11.66
101	12	0.66	Not Allowed		1.66	4.8	16.8	2.00	4	14
110	14	0.57	Not Allowed		1.43	5.6	19.6	1.71	4.66	16.33
111	16	0.50	Not Allowed		1.25	6.4	22.4	1.50	5.33	18.66

Bit 4 = **EXTG**: *External Trigger Enable*.
This bit is set and cleared by software.

0: External trigger disabled.

1: External trigger enabled. Allows a conversion sequence to be started on the subsequent edge of the external signal applied to the EXTRG pin (when enabled as an Alternate Function).

Bit 3 = **INTG**: *Internal Trigger Enable.*
This bit is set and cleared by software.

0: Internal trigger disabled.

1: Internal trigger enabled. Allows a conversion sequence to be started, synchronized by an internal signal (On-chip Event signal) from a Multifunction Timer peripheral.

Both External and Internal Trigger inputs are internally ORed, thus avoiding Hardware conflicts; however, the correct procedure is to enable only one alternate synchronization input at a time.

Note: The effect of either synchronization mode is to set the START/STOP bit, which is reset by hardware when in SINGLE mode, at the end of each sequence of conversions.

Requirements: The External Synchronisation Input must receive a low level pulse wider than an INTCLK period and, for both External and On-Chip Event synchronisation, the repetition period must be greater than the time required for the selected sequence of conversions.

Bit 2 = **POW**: *Power Up/Power Down.*
This bit is set and cleared by software.

0: Power down mode: all power-consuming logic is disabled, thus selecting a low power idle mode.