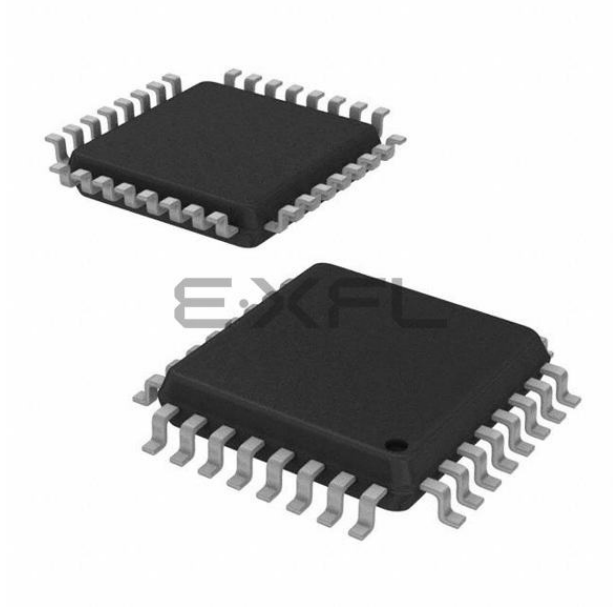


Welcome to E-XFL.COM

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	Active
Core Processor	80C51
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	20
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-VQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89c51cc02ua-ratum

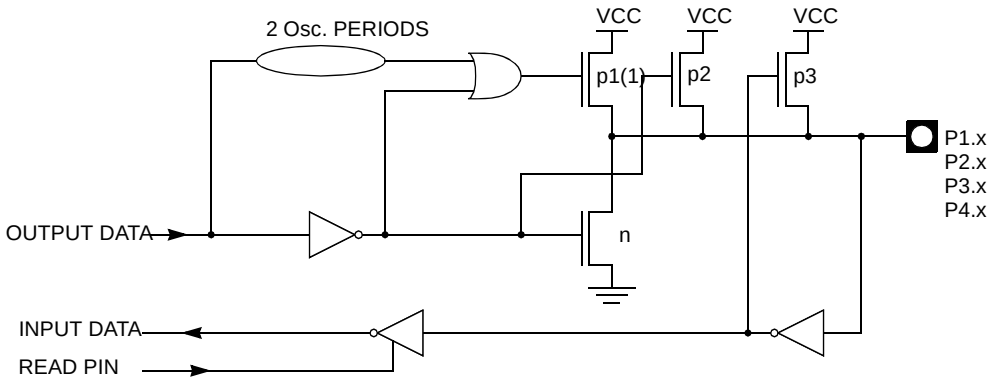
Pin Description

Pin Name	Type	Description
VSS	GND	Circuit ground
VCC		**Supply Voltage**
VAREF		Reference Voltage for ADC (input)
VAVCC		Supply Voltage for ADC
VAGND		Reference Ground for ADC (internaly connected with the VSS)
P1.0:7	I/O	**Port 1:** Is an 8-bit bi-directional I/O port with internal pull-ups. Port 1 pins can be used for digital input/output or as analog inputs for the Analog Digital Converter (ADC). Port 1 pins that have 1's written to them are pulled high by the internal pull-up transistors and can be used as inputs in this state. As inputs, Port 1 pins that are being pulled low externally will be the source of current (I_{IL}, See section 'Electrical Characteristic') because of the internal pull-ups. Port 1 pins are assigned to be used as analog inputs via the ADCCF register (in this case the internal pull-ups are disconnected). As a secondary digital function, port 1 contains the Timer 2 external trigger and clock input; the PCA external clock input and the PCA module I/O. P1.0/AN0/T2 Analog input channel 0, External clock input for Timer/counter2. P1.1/AN1/T2EX Analog input channel 1, Trigger input for Timer/counter2. P1.2/AN2/ECI Analog input channel 2, PCA external clock input. P1.3/AN3/CEX0 Analog input channel 3, PCA module 0 Entry of input/PWM output. P1.4/AN4/CEX1 Analog input channel 4, PCA module 1 Entry of input/PWM output. P1.5/AN5 Analog input channel 5, P1.6/AN6 Analog input channel 6, P1.7/AN7 Analog input channel 7, It can drive CMOS inputs without external pull-ups.
P2.0:1	I/O	**Port 2:** Is an 2-bit bi-directional I/O port with internal pull-ups. Port 2 pins that have 1's written to them are pulled high by the internal pull-ups and can be used as inputs in this state. As inputs, Port 2 pins that are being pulled low externally will be a source of current (IIL, on the datasheet) because of the internal pull-ups. In the T89C51CC02 Port 2 can sink or source 5mA. It can drive CMOS inputs without external pull-ups.

associated nFET is switched off. This is traditional CMOS switch convention. Current strengths are 1/10 that of pFET #3.

Note: During Reset, pFET#1 is not avtivated. During Reset, only the weak pFET#3 pull up the pin.

Figure 2. Internal Pull-up Configurations

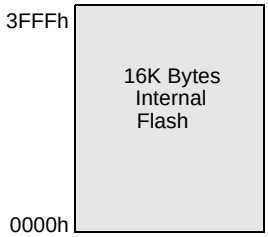


Program/Code Memory

The T89C51CC02 implement 16K Bytes of on-chip program/code memory.

The Flash memory increases EPROM and ROM functionality by in-circuit electrical erasure and programming. Thanks to the internal charge pump, the high voltage needed for programming or erasing Flash cells is generated on-chip using the standard V_{DD} voltage. Thus, the Flash memory can be programmed using only one voltage and allows In-System Programming (ISP). Hardware programming mode is also available using specific programming tool.

Figure 12. Program/Code Memory Organization



Flash Memory Architecture

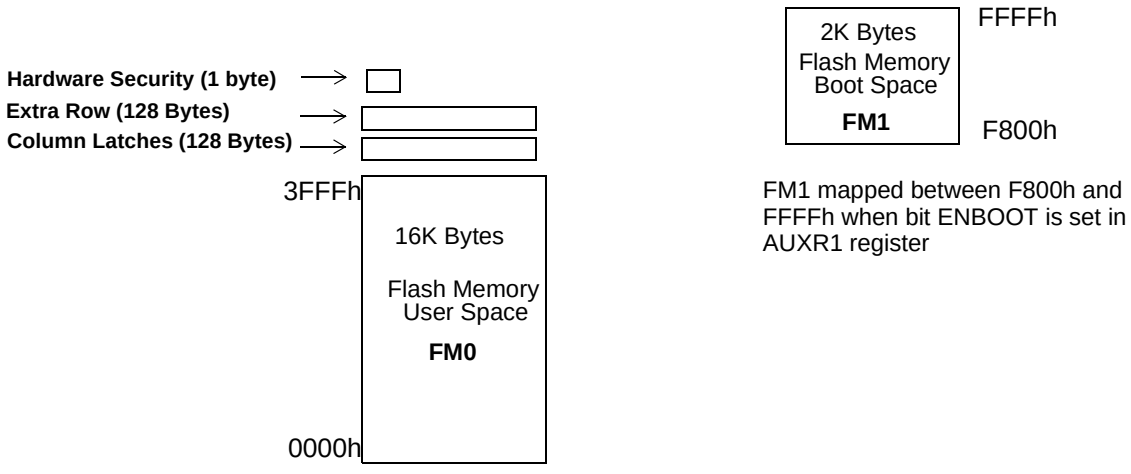
T89C51CC02 features two on-chip Flash memories:

- Flash memory FM0:
  containing 16K Bytes of program memory (user space) organized into 128 bytes pages,
- Flash memory FM1:
  2K Bytes for boot loader and Application Programming Interfaces (API).

The FM0 can be program by both parallel programming and Serial ISP whereas FM1 supports only parallel programming by programmers. The ISP mode is detailed in the 'In-System Programming' section.

All Read/Write access operations on Flash memory by user application are managed by a set of API described in the 'In-System Programming' section.

Figure 13. Flash Memory Architecture



FM0 Memory Architecture

The Flash memory is made up of 4 blocks (See Figure 13):

1. The memory array (user space) 16K Bytes
2. The Extra Row
3. The Hardware security bits
4. The column latch registers

User Space

This space is composed of a 16K Bytes Flash memory organized in 128 pages of 128 Bytes. It contains the user's application code.

Extra Row (XRow)

This row is a part of FM0 and has a size of 128 Bytes. The extra row may contain information for boot loader usage.

Hardware Security Byte

The Hardware security Byte space is a part of FM0 and has a size of 1 byte.
The 4 MSB can be read/written by software, the 4 LSB can only be read by software and written by hardware in parallel mode.

Column Latches

The column latches, also part of FM0, have a size of full page (128 Bytes).
The column latches are the entrance buffers of the three previous memory locations (user array, XROW and Hardware security byte).

Cross Flash Memory Access Description

The FM0 memory can be programmed as describe on Table 21. Programming FM0 from FM0 is impossible.

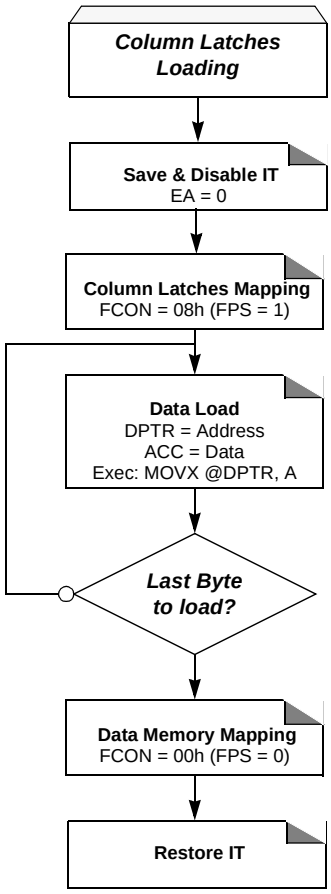
The FM1 memory can be program only by parallel programming.

Table 21 show all software Flash access allowed.

Table 21. Cross Flash Memory Access

Code executing from		Action	FM0 (user Flash)	FM1 (boot Flash)
	FM0 (user Flash)	Read	ok	-
		Load column latch	ok	-
		Write	-	-
	FM1 (boot Flash)	Read	ok	ok
		Load column latch	ok	-
		Write	ok	-

Figure 14. Column Latches Loading Procedure(1)



Note: 1. The last page address used when loading the column latch is the one used to select the page programming address.

Programming the Flash Spaces

User

The following procedure is used to program the User space and is summarized in Figure 15:

- Load up to one page of data in the column latches from address 0000h to 3FFFh.
- Save then disable the interrupts.
- Launch the programming by writing the data sequence 50h followed by A0h in FCON register.This step must be executed from FM1.
  The end of the programming indicated by the FBUSY flag cleared.
- Restore the interrupts.

Extra Row

The following procedure is used to program the Extra Row space and is summarized in Figure 15:

- Load data in the column latches from address FF80h to FFFFh.
- Save then disable the interrupts.
- Launch the programming by writing the data sequence 52h followed by A2h in FCON register. This step of the procedure must be executed from FM1.
  The end of the programming indicated by the FBUSY flag cleared.
- Restore the interrupts.

Operation Cross Memory Access

Space addressable in read and write are:

- RAM
- ERAM (Expanded RAM access by movx)
- EEPROM DATA
- FM0 (user flash)
- Hardware byte
- XROW
- Boot Flash
- Flash Column latch

The table below provides the different kind of memory which can be accessed from different code location.

Table 26. Cross Memory Access

	Action	RAM	ERAM	Boot FLASH	FM0	E² Data	Hardware Byte	XROW
boot FLASH	Read			OK	OK	OK	OK	-
	Write			-	OK(1)	OK(1)	OK(1)	OK(1)
FM0	Read			OK	OK	OK	-OK	-
	Write			-	OK (idle)	OK(1)	-	-OK

Note: 1. RWW: Read While Write

Timers/Counters

The T89C51CC02 implements two general-purpose, 16-bit Timers/Counters. Such are identified as Timer 0 and Timer 1, and can be independently configured to operate in a variety of modes as a Timer or an event Counter. When operating as a Timer, the Timer/Counter runs for a programmed length of time, then issues an interrupt request. When operating as a Counter, the Timer/Counter counts negative transitions on an external pin. After a preset number of counts, the Counter issues an interrupt request.
The various operating modes of each Timer/Counter are described in the following sections.

Timer/Counter Operations

A basic operation is Timer registers THx and TLx (x = 0, 1) connected in cascade to form a 16-bit Timer. Setting the run control bit (TRx) in TCON register (See Figure 38) turns the Timer on by allowing the selected input to increment TLx. When TLx overflows it increments THx; when THx overflows it sets the Timer overflow flag (TFx) in TCON register. Setting the TRx does not clear the THx and TLx Timer registers. Timer registers can be accessed to obtain the current count or to enter preset values. They can be read at any time but TRx bit must be cleared to preset their values, otherwise the behavior of the Timer/Counter is unpredictable.

The C/Tx# control bit selects Timer operation or Counter operation by selecting the divided-down peripheral clock or external pin Tx as the source for the counted signal. TRx bit must be cleared when changing the mode of operation, otherwise the behavior of the Timer/Counter is unpredictable.
For Timer operation (C/Tx# = 0), the Timer register counts the divided-down peripheral clock. The Timer register is incremented once every peripheral cycle (6 peripheral clock periods). The Timer clock rate is $f_{PER}/6$, i.e. $f_{OSC}/12$ in standard mode or $f_{OSC}/6$ in X2 Mode.

For Counter operation (C/Tx# = 1), the Timer register counts the negative transitions on the Tx external input pin. The external input is sampled every peripheral cycles. When the sample is high in one cycle and low in the next one, the Counter is incremented. Since it takes 2 cycles (12 peripheral clock periods) to recognize a negative transition, the maximum count rate is $f_{PER}/12$, i.e. $f_{OSC}/24$ in standard mode or $f_{OSC}/12$ in X2 Mode. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it should be held for at least one full peripheral cycle.

Timer 0

Timer 0 functions as either a Timer or event Counter in four modes of operation. Figure 24 through Figure 27 show the logical configuration of each mode.

Timer 0 is controlled by the four lower bits of TMOD register (See Figure 39) and bits 0, 1, 4 and 5 of TCON register (See Figure 38). TMOD register selects the method of Timer gating (GATE0), Timer or Counter operation (T/C0#) and mode of operation (M10 and M00). TCON register provides Timer 0 control functions: overflow flag (TF0), run control bit (TR0), interrupt flag (IE0) and interrupt type control bit (IT0).
For normal Timer operation (GATE0 = 0), setting TR0 allows TL0 to be incremented by the selected input. Setting GATE0 and TR0 allows external pin INT0# to control Timer operation.

Timer 0 overflow (count rolls over from all 1s to all 0s) sets TF0 flag generating an interrupt request.

It is important to stop Timer/Counter before changing mode.

Timer 2

The T89C51CC02 Timer 2 is compatible with Timer 2 in the 80C52.

It is a 16-bit timer/counter: the count is maintained by two eightbit timer registers, TH2 and TL2 that are cascade-connected. It is controlled by T2CON register (See Table 45) and T2MOD register (See Table 46). Timer 2 operation is similar to Timer 0 and Timer 1. C/$\overline{T2}$ selects $F_{T2\ clock}$/6 (timer operation) or external pin T2 (counter operation) as timer clock. Setting TR2 allows TL2 to be incremented by the selected input.

Timer 2 includes the following enhancements:

- Auto-reload mode (up or down counter)
- Programmable clock-output

Auto-Reload Mode

The auto-reload mode configures Timer 2 as a 16-bit timer or event counter with automatic reload. This feature is controlled by the DCEN bit in T2MOD register (See Table 45). Setting the DCEN bit enables Timer 2 to count up or down as shown in Figure 29. In this mode the T2EX pin controls the counting direction.

When T2EX is high, Timer 2 counts up. Timer overflow occurs at FFFFh which sets the TF2 flag and generates an interrupt request. The overflow also causes the 16-bit value in RCAP2H and RCAP2L registers to be loaded into the timer registers TH2 and TL2.

When T2EX is low, Timer 2 counts down. Timer underflow occurs when the count in the timer registers TH2 and TL2 equals the value stored in RCAP2H and RCAP2L registers. The underflow sets TF2 flag and reloads FFFFh into the timer registers.

The EXF2 bit toggles when Timer 2 overflow or underflow, depending on the direction of the count. EXF2 does not generate an interrupt. This bit can be used to provide 17-bit resolution.

Figure 29. Auto-Reload Mode Up/Down Counter

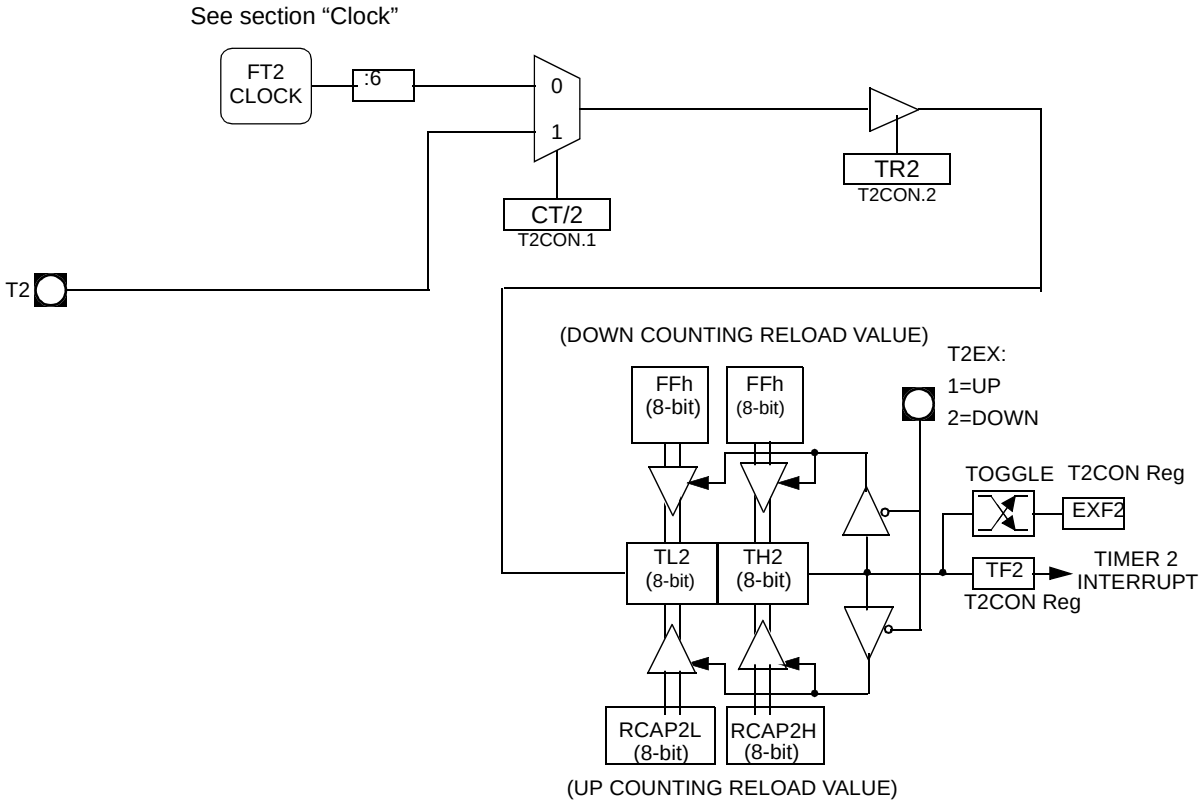
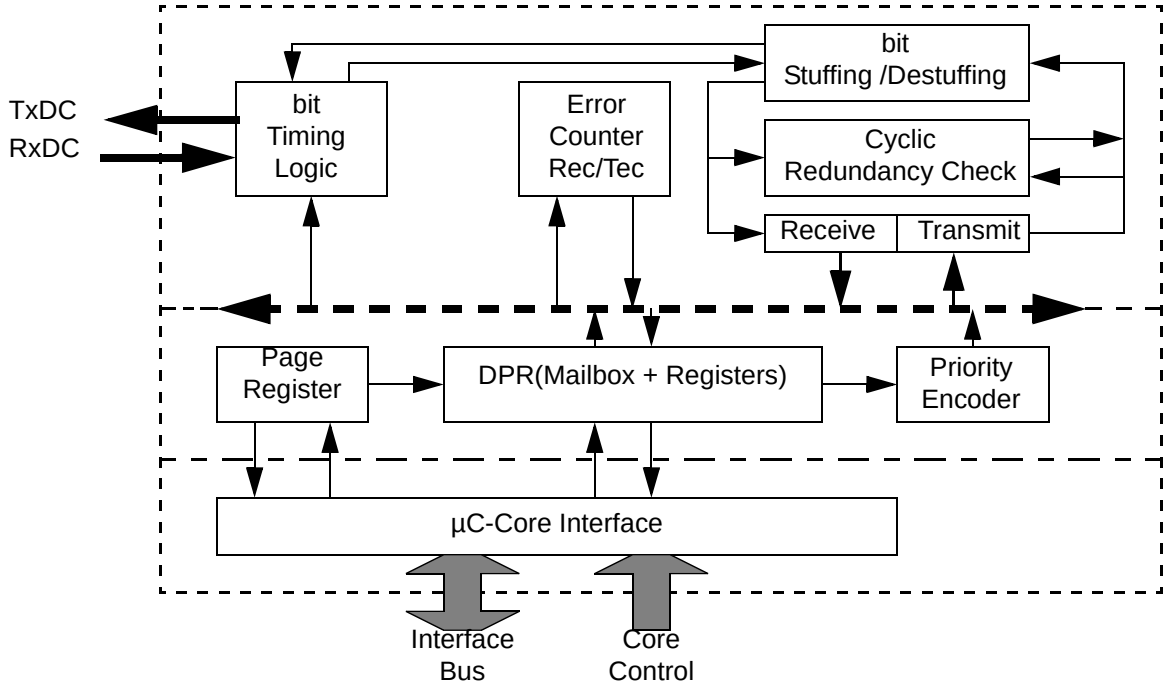


Figure 36. CAN Controller Block Diagram



CAN Controller Mailbox and Registers Organization

The pagination allows management of the 91 registers including 80(4 x 20) Bytes of mailbox via 32 SFRs.

All actions on the message object window SFRs apply to the corresponding message object registers pointed by the message object number find in the Page message object register (CANPAGE) as illustrate in Figure 37.

Fault Confinement

With respect to fault confinement, a unit may be in one of the three following status:

- Error active
- Error passive
- Bus off

An error active unit takes part in bus communication and can send an active error frame when the CAN macro detects an error.

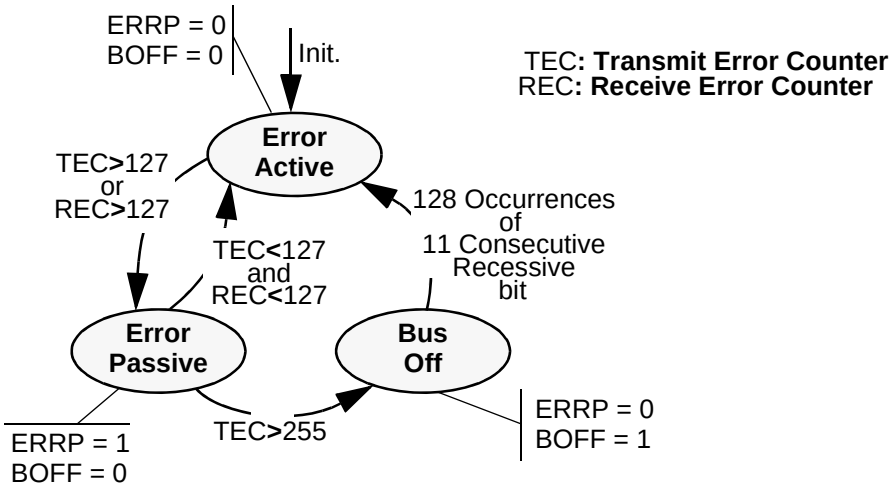
An error passive unit cannot send an active error frame. It takes part in bus communication, but when an error is detected, a passive error frame is sent. Also, after a transmission, an error passive unit will wait before initiating further transmission.

A bus off unit is not allowed to have any influence on the bus.

For fault confinement, two error counters (TEC and REC) are implemented.

See CAN Specification for details on Fault confinement.

Figure 42. Line Error Mode



```
// Find the first message object which generate an interrupt in CANSIT
// Select the corresponding message object

// Analyse the CANSTCH register to identify which kind of interrupt is
generated

// Manage the interrupt

// Clear the status register CANSTCH = 00h;

// if it is not a channel interrupt but a general interrupt
// Manage the general interrupt and clear CANGIT register

// restore the old CANPAGE
```

Table 61. CANGIE Register

CANGIE (S:C1h) – CAN

7	6	5	4	3	2	1	0
-	-	ENRX	ENTX	ENERCH	ENBUF	ENERG	-

Bit Number	Bit Mnemonic	Description
7 - 6	-	**Reserved** The values read from these bits are indeterminate. Do not set these bits.
5	ENRX	**Enable Receive Interrupt** 0 - Disable 1 - Enable
4	ENTX	**Enable Transmit Interrupt** 0 - Disable 1 - Enable
3	ENERCH	**Enable Message Object Error Interrupt** 0 - Disable 1 - Enable
2	ENBUF	**Enable BUF Interrupt** 0 - Disable 1 - Enable
1	ENERG	**Enable General Error Interrupt** 0 - Disable 1 - Enable
0	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit. See Figure 39.

Reset Value = xx00 000xb

Table 77. CANIDT3 Register for V2.0 Part B
CANIDT3 for V2.0 Part B (S:BEh)
CAN Identifier Tag Registers 3

7	6	5	4	3	2	1	0
IDT 12	**IDT 11**	**IDT 10**	**IDT 9**	**IDT 8**	**IDT 7**	**IDT 6**	**IDT 5**

Bit Number	Bit Mnemonic	Description
7 - 0	IDT12:5	**IDentifier Tag Value** See Figure 43.

No default value after reset.

Table 78. CANIDT4 Register for V2.0 Part B
CANIDT4 for V2.0 Part B (S:BFh)
CAN Identifier Tag Registers 4

7	6	5	4	3	2	1	0
IDT 4	**IDT 3**	**IDT 2**	**IDT 1**	**IDT 0**	**RTRTAG**	**RB1TAG**	**RB0TAG**

Bit Number	Bit Mnemonic	Description
7 - 3	IDT4:0	**IDentifier Tag Value** See Figure 43.
2	RTRTAG	**Remote Transmission Request Tag Value**
1	RB1TAG	Reserved bit 1 tag value.
0	RB0TAG	Reserved bit 0 tag value.

No default value after reset.

Table 79. CANIDM1 Register for V2.0 part A
CANIDM1 for V2.0 part A (S:C4h)
CAN Identifier Mask Registers 1

7	6	5	4	3	2	1	0
IDMSK 10	IDMSK 9	IDMSK 8	IDMSK 7	IDMSK 6	IDMSK 5	IDMSK 4	IDMSK 3

Bit Number	Bit Mnemonic	Description
7 - 0	IDTMSK10:3	**IDentifier Mask Value** 0 - comparison true forced. 1 - bit comparison enabled. See Figure 43.

No default value after reset.

Table 80. CANIDM2 Register for V2.0 part A
CANIDM2 for V2.0 part A (S:C5h)
CAN Identifier Mask Registers 2

7	6	5	4	3	2	1	0
IDMSK 2	IDMSK 1	IDMSK 0	-	-	-	-	-

Bit Number	Bit Mnemonic	Description
7 - 5	IDTMSK2:0	**IDentifier Mask Value** 0 - comparison true forced. 1 - bit comparison enabled. See Figure 43.
4 -0	-	**Reserved** The values read from these bits are indeterminate. Do not set these bits.

No default value after reset.

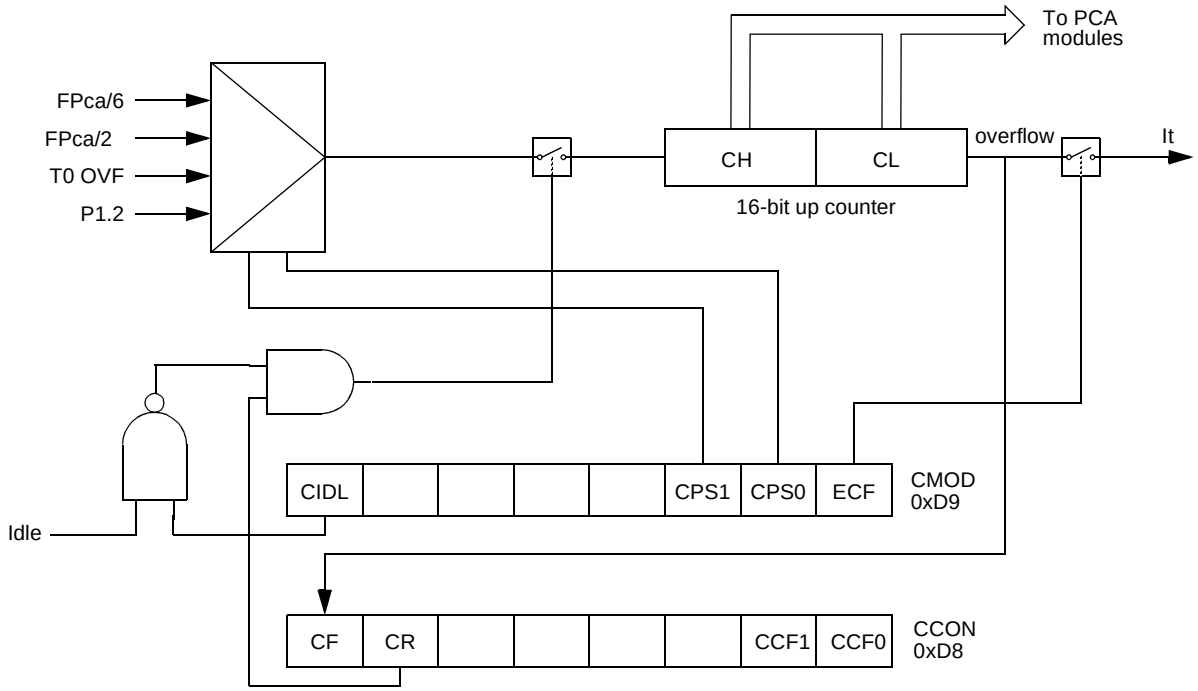
Table 81. CANIDM3 Register for V2.0 part A
CANIDM3 for V2.0 part A (S:C6h)
CAN Identifier Mask Registers 3

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-

Bit Number	Bit Mnemonic	Description
7 - 0	-	**Reserved** The values read from these bits are indeterminate.

No default value after reset.

Figure 46. PCA Timer/Counter



The CMOD register includes three additional bits associated with the PCA.

- The CIDL bit which allows the PCA to stop during idle mode.
- The ECF bit which when set causes an interrupt and the PCA overflow flag CF in CCON register to be set when the PCA timer overflows.

The CCON register contains the run control bit for the PCA and the flags for the PCA timer and each module.

- The CR bit must be set to run the PCA. The PCA is shut off by clearing this bit.
- The CF bit is set when the PCA counter overflows and an interrupt will be generated if the ECF bit in CMOD register is set. The CF bit can only be cleared by software.
- The CCF0:1 bits are the flags for the modules (CCF0 for module0...) and are set by hardware when either a match or a capture occurs. These flags also can be cleared by software.

Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the Interrupt Enable register. This register also contains a global disable bit which must be cleared to disable all the interrupts at the same time.

Each interrupt source can also be individually programmed to one of four priority levels by setting or clearing a bit in the Interrupt Priority registers. The Table below shows the bit values and priority levels associated with each combination.

Table 108. Priority Level bit Values

IPH.x	IPL.x	Interrupt Level Priority
0	0	0 (Lowest)
0	1	1
1	0	2
1	1	3 (Highest)

A low-priority interrupt can be interrupted by a high priority interrupt but not by another low-priority interrupt. A high-priority interrupt cannot be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of the higher priority level is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence, See Table 109.

Table 109. Interrupt Priority Within Level

Interrupt Name	Interrupt Address Vector	Interrupt Number	Polling Priority
External interrupt (INT0)	0003h	1	1
Timer0 (TF0)	000Bh	2	2
External interrupt (INT1)	0013h	3	3
Timer 1 (TF1)	001Bh	4	4
PCA (CF or CCFn)	0033h	7	5
UART (RI or TI)	0023h	5	6
Timer 2 (TF2)	002Bh	6	7
CAN (Txok, Rxok, Err or OvrBuf)	003Bh	8	8
ADC (ADCI)	0043h	9	9
CAN Timer Overflow (OVRTIM)	004Bh	10	10

For timing purposes as port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs. $I_{OL}/I_{OH} \geq \pm 20mA$.

Clock Waveforms

Valid in normal clock mode. In X2 Mode XTAL2 must be changed to XTAL2/2.

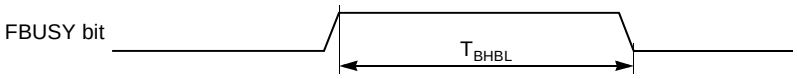
Flash/EEPROM Memory

Table 120. Memory AC Timing

V_{cc} = 3.0V to 5.5V, T_A = -40°C to +85°C

Symbol	Parameter	Min	Typ	Max	Unit
T_{BHBL}	Flash/EEPROM Internal Busy (Programming) Time		13	17	ms
N_{FCY}	Number of Flash/EEPROM Erase/Write Cycles	100 000			cycles
T_{FDR}	Flash/EEPROM Data Retention Time	10			years

Figure 63. Flash Memory - Internal Busy Waveforms



A/D Converter

Table 121. AC Parameters for A/D Conversion

Symbol	Parameter	Min	Typ	Max	Unit
T_{SETUP}		4			µs
ADC Clock Frequency			700		KHz

STANDARD NOTES FOR PQFP/ VQFP / TQFP / DQFP

1/ CONTROLLING DIMENSIONS : INCHES

2/ ALL DIMENSIONING AND TOLERANCING CONFORM TO ANSI Y 14.5M - 1982.

3/ "D1 AND E1" DIMENSIONS DO NOT INCLUDE MOLD PROTUSIONS. MOLD PROTUSIONS SHALL NOT EXCEED 0.25 mm (0.010 INCH). THE TOP PACKAGE BODY SIZE MAY BE SMALLER THAN THE BOTTOM PACKAGE BODY SIZE BY AS MUCH AS 0.15 mm.

4/ DATUM PLANE "H" LOCATED AT MOLD PARTING LINE AND COINCIDENT WITH LEAD, WHERE LEAD EXITS PLASTIC BODY AT BOTTOM OF PARTING LINE.

5/ DATUM "A" AND "D" TO BE DETERMINED AT DATUM PLANE H.

6/ DIMENSION " f " DOES NOT INCLUDE DAMBAR PROTUSION ALLOWABLE DAMBAR PROTUSION SHALL BE 0.08mm/.003" TOTAL IN EXCESS OF THE " f " DIMENSION AT MAXIMUM MATERIAL CONDITION . DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

STANDARD NOTES FOR PLCC

1/ CONTROLLING DIMENSIONS : INCHES

2/ DIMENSIONING AND TOLERANCING PER ANSI Y 14.5M - 1982.

3/ "D" AND "E1" DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTUSIONS. MOLD FLASH OR PROTUSIONS SHALL NOT EXCEED 0.20 mm (.008 INCH) PER SIDE.

NOTES: SOIC STANDARD NOTES

1. DIMENSIONING & TOLERANCING CONFORM TO ASME Y14.5M. – 1982.

2. "D" AND "E" DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTUSIONS.
MOLD FLASH OR PROTUSIONS SHALL NOT EXCEED 0.15mm (0.006 INCH) PER SIDE.

3. THE CHAMFER "h" IS OPTIONAL.