Welcome to E-XFL.COM

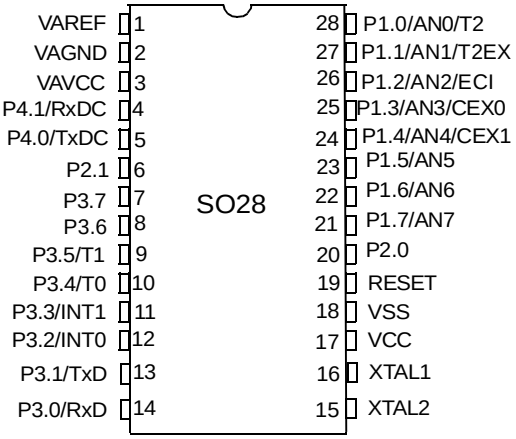
What is "Embedded - Microcontrollers"?

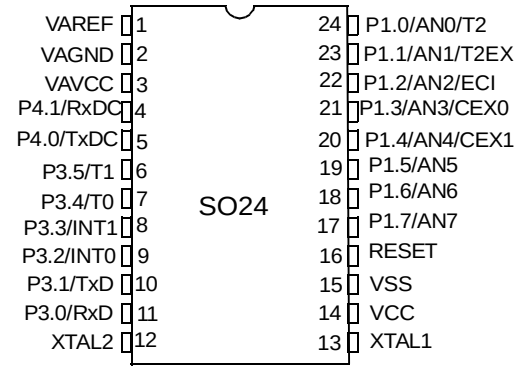
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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.

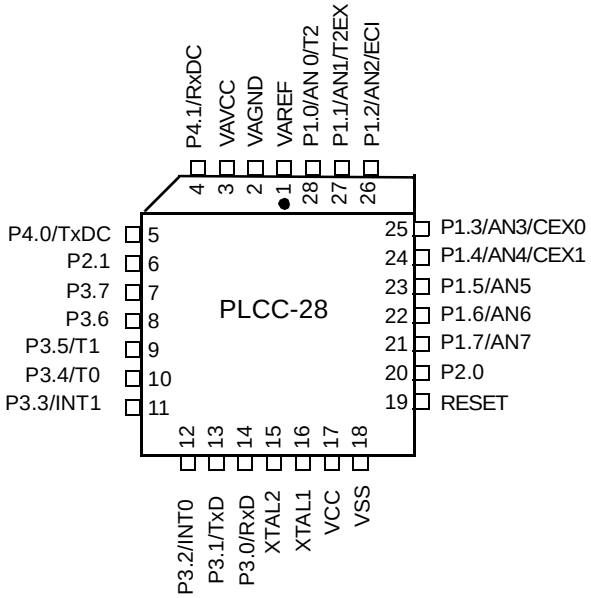
Applications of "Embedded - Microcontrollers"

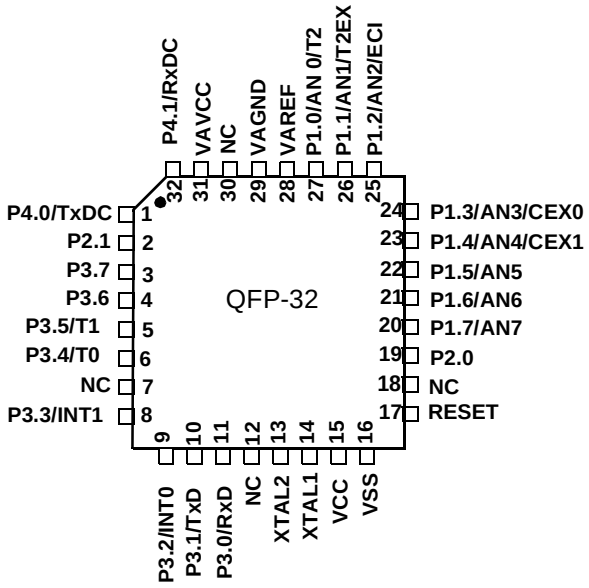
Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	20
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SO
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/t89c51cc02ca-6ksim

Pin Configurations








P4.1/RxDC
VAVCC
NC
VAGND
VAREF
P1.0/AN 0/T2
P1.1/AN1/T2EX
P1.2/AN2/ECI
32
31
30
29
28
27
26
25
P4.0/TxDC 1
P2.1 2
P3.7 3
P3.6 4
P3.5/T1 5
P3.4/T0 6
NC 7
P3.3/INT1 8
QFP-32
24 P1.3/AN3/CEX0
23 P1.4/AN4/CEX1
22 P1.5/AN5
21 P1.6/AN6
20 P1.7/AN7
19 P2.0
18 NC
17 RESET
9
10
11
12
13
14
15
16
P3.2/INT0
P3.1/TxD
P3.0/RxD
NC
XTAL2
XTAL1
VCC
VSS

Clock

The T89C51CC02 core needs only 6 clock periods per machine cycle. This feature, called “X2”, provides the following advantages:

- Divides frequency crystals by 2 (cheaper crystals) while keeping the same CPU power.
- Saves power consumption while keeping the same CPU power (oscillator power saving).
- Saves power consumption by dividing dynamic operating frequency by 2 in operating and idle modes.
- Increases CPU power by 2 while keeping the same crystal frequency.

In order to keep the original C51 compatibility, a divider-by-2 is inserted between the XTAL1 signal and the main clock input of the core (phase generator). This divider may be disabled by the software.

An extra feature is available to start after Reset in the X2 Mode. This feature can be enabled by a bit X2B in the Hardware Security Byte. This bit is described in the section ’In-System Programming’.

Description

The X2 bit in the CKCON register (See Table 12) allows switching from 12 clock cycles per instruction to 6 clock cycles and vice versa. At reset, the standard speed is activated (STD mode).

Setting this bit activates the X2 feature (X2 Mode) for the CPU Clock only (See Figure 3).

The Timers 0, 1 and 2, Uart, PCA, watchdog or CAN switch in X2 Mode only if the corresponding bit is cleared in the CKCON register.

The clock for the whole circuit and peripheral is first divided by two before being used by the CPU core and peripherals. This allows any cyclic ratio to be accepted on the XTAL1 input. In X2 Mode, as this divider is bypassed, the signals on XTAL1 must have a cyclic ratio between 40 to 60%. Figure 3. shows the clock generation block diagram. The X2 bit is validated on the XTAL1 ÷ 2 rising edge to avoid glitches when switching from the X2 to the STD mode. Figure 4 shows the mode switching waveforms.

Figure 3. Clock CPU Generation Diagram

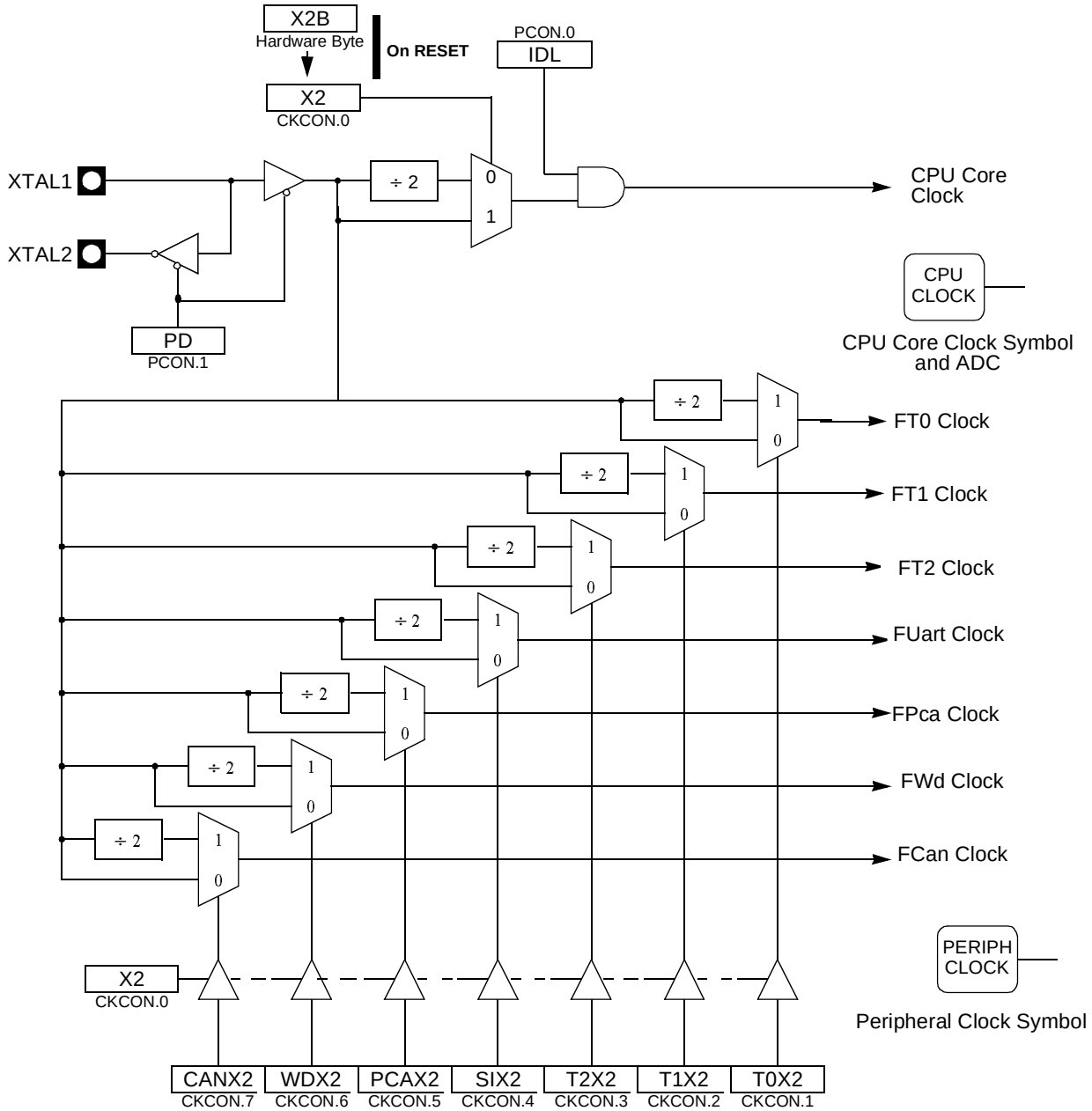


Table 14. Minimum Reset Capacitor for a 15k Pull-down Resistor

oscrst/vddrst	1ms	10ms	100ms
5ms	2.7µF	4.7µF	47µF
20ms	10µF	15µF	47µF

Note: These values assume VDD starts from 0v to the nominal value. If the time between two on/off sequences is too fast, the power-supply decoupling capacitors may not be fully discharged, leading to a bad reset sequence.

During a Normal Operation (Warm Reset)

Reset pin must be maintained for at least 2 machine cycles (24 oscillator clock periods) to apply a reset sequence during normal operation. The number of clock periods is mode independent (X2 or X1).

Watchdog Reset

A 1K resistor must be added in series with the capacitor to allow the use of watchdog reset pulse output on the RST pin or when an external power-supply supervisor is used. Figure 6 shows the reset circuitry when a capacitor is used.

Figure 6. Reset Circuitry for a Watchdog Configuration

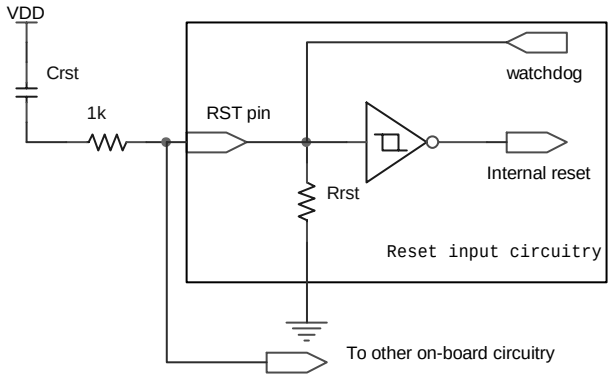
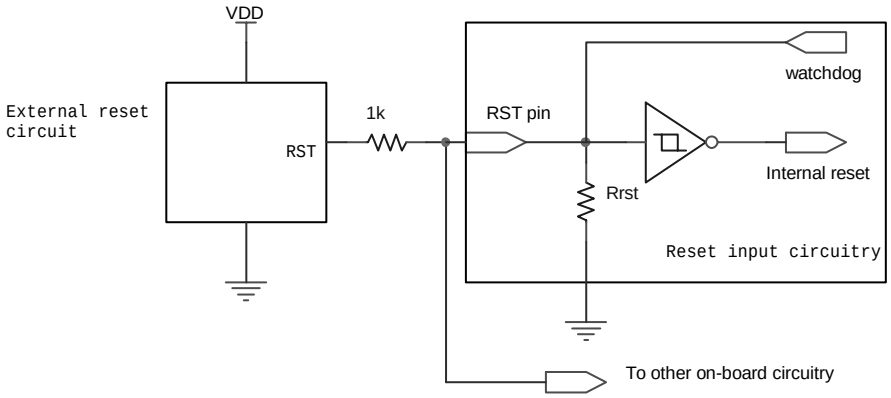


Figure 7 shows the reset circuitry when an external reset circuit is used.

Figure 7. Reset Circuitry Example Using an External Reset Circuit

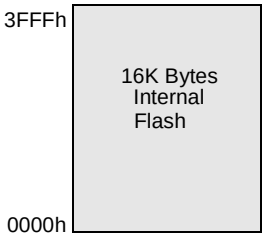


Program/Code Memory

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

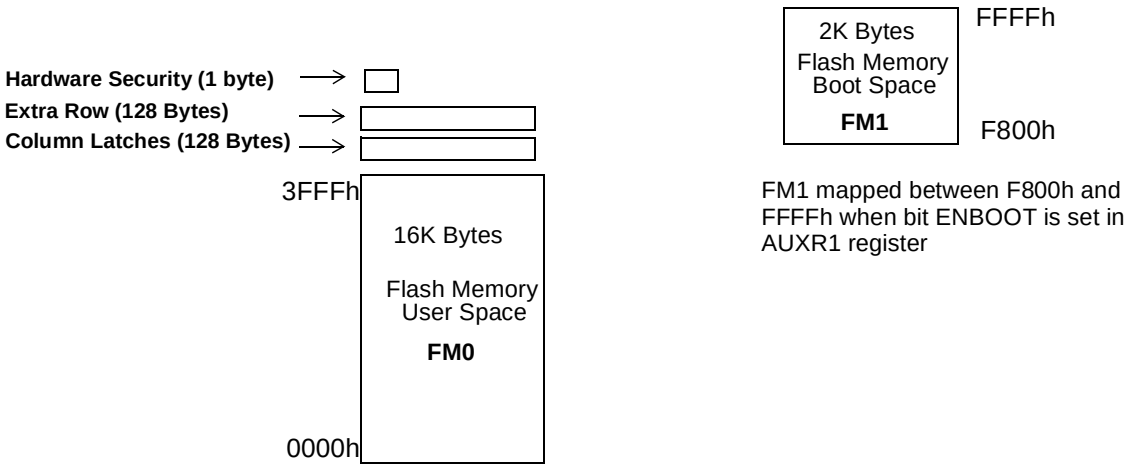
T89C51CC02 features two on-chip Flash memories:

- 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



Overview of FM0 Operations

The CPU interfaces the Flash memory through the FCON register and AUXR1 register.

These registers are used to:

- Map the memory spaces in the adressable space
- Launch the programming of the memory spaces
- Get the status of the Flash memory (busy/not busy)

Mapping of the Memory Space

By default, the user space is accessed by MOVC instruction for read only. The column latches space is made accessible by setting the FPS bit in FCON register. Writing is possible from 0000h to 3FFFh, address bits 6 to 0 are used to select an address within a page while bits 14 to 7 are used to select the programming address of the page.
Setting FPS bit takes precedence on the EEE bit in EECON register.

The other memory spaces (user, extra row, hardware security) are made accessible in the code segment by programming bits FMOD0 and FMOD1 in FCON register in accordance with Table 21. A MOVC instruction is then used for reading these spaces.

Table 21. FM0 blocks Select bits

FMOD1	FMOD0	FM0 Adressable Space
0	0	User (0000h-3FFFh)
0	1	Extra Row(FF80h-FFFFh)
1	0	Hardware Security Byte (0000h)
1	1	Reserved

Launching Programming

FPL3:0 bits in FCON register are used to secure the launch of programming. A specific sequence must be written in these bits to unlock the write protection and to launch the programming. This sequence is 5xh followed by Axh. Table 22 summarizes the memory spaces to program according to FMOD1:0 bits.

Table 22. Programming Spaces

	Write to FCON				
	FPL3:0	FPS	FMOD1	FMOD0	Operation
User	5	x	0	0	No action
	A	x	0	0	Write the column latches in user space
Extra Row	5	x	0	1	No action
	A	x	0	1	Write the column latches in extra row space
Hardware Security Byte	5	x	1	0	No action
	A	x	1	0	Write the fuse bits space
Reserved	5	x	1	1	No action
	A	x	1	1	No action

Note: The sequence 5xh and Axh must be executing without instructions between them otherwise the programming is aborted.
Interrupts that may occur during programming time must be disabled to avoid any spurious exit of the programming mode.

Status of the Flash Memory

The bit FBUSY in FCON register is used to indicate the status of programming.

FBUSY is set when programming is in progress.

Selecting FM1

The bit ENBOOT in AUXR1 register is used to map FM1 from F800h to FFFFh.

Loading the Column Latches

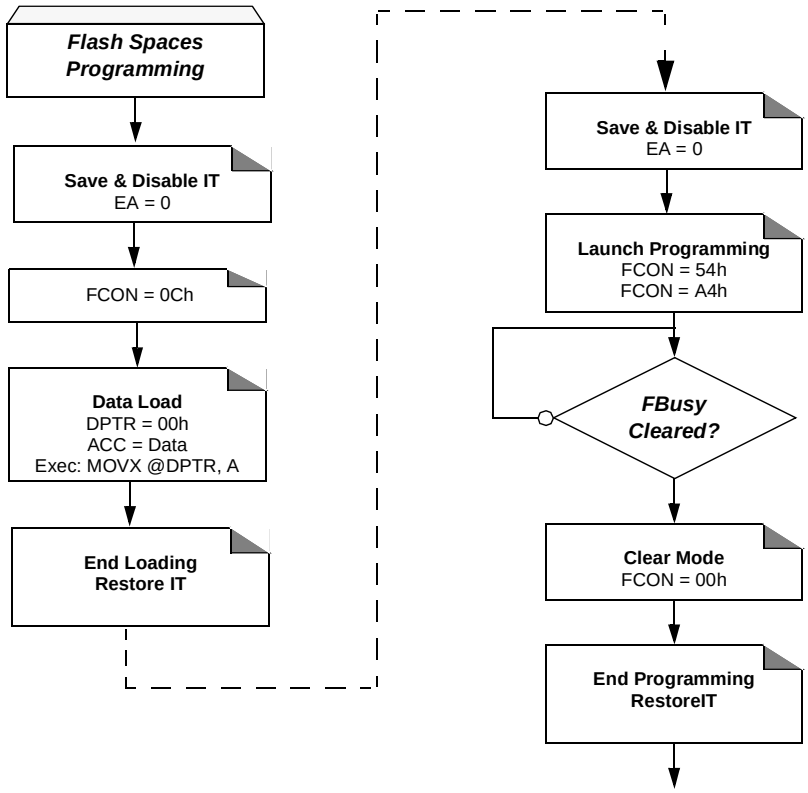
Any number of data from 1 byte to 128 Bytes can be loaded in the column latches. This provides the capability to program the whole memory by byte, by page or by any number of Bytes in a page.

When programming is launched, an automatic erase of the locations loaded in the column latches is first performed, then programming is effectively done. Thus no page or block erase is needed and only the loaded data are programmed in the corresponding page.

The following procedure is used to load the column latches and is summarized in Figure 14:

- Save then disable interrupt and map the column latch space by setting FPS bit.
- Load the DPTR with the address to load.
- Load Accumulator register with the data to load.
- Execute the MOVX @DPTR, A instruction.
- If needed loop the three last instructions until the page is completely loaded.
- unmap the column latch and Restore Interrupt

Figure 16. Hardware Programming Procedure



Reading the Flash Spaces

User

The following procedure is used to read the User space:

- Read one byte in Accumulator by executing MOVC A,@A+DPTR with A+DPTR is the address of the code byte to read.

Note: FCON must be cleared (00h) when not used.

Extra Row

The following procedure is used to read the Extra Row space and is summarized in Figure 17:

- Map the Extra Row space by writing 02h in FCON register.
- Read one byte in Accumulator by executing MOVC A,@A+DPTR with A= 0 & DPTR= FF80h to FFFFh.
- Clear FCON to unmap the Extra Row.

Hardware Security Byte

The following procedure is used to read the Hardware Security Byte and is summarized in Figure 17:

- Map the Hardware Security space by writing 04h in FCON register.
- Read the byte in Accumulator by executing MOVC A,@A+DPTR with A= 0 & DPTR= 0000h.
- Clear FCON to unmap the Hardware Security Byte.

Table 40. TL0 Register
TL0 (S:8Ah)
Timer 0 Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 0**

Reset Value = 0000 0000b

Table 41. TH1 Register
TH1 (S:8Dh)
Timer 1 High Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**High Byte of Timer 1**

Reset Value = 0000 0000b

Table 42. TL1 Register
TL1 (S:8Bh)
Timer 1 Low Byte Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 1**

Reset Value = 0000 0000b

Table 43. T2CON Register
T2CON (S:C8h)
Timer 2 Control Register

7	6	5	4	3	2	1	0
TF2	**EXF2**	**RCLK**	**TCLK**	**EXEN2**	**TR2**	**C/T2#**	**CP/RL2#**

Bit Number	Bit Mnemonic	Description
7	TF2	**Timer 2 Overflow Flag** TF2 is not set if RCLK=1 or TCLK = 1. Must be cleared by software. Set by hardware on Timer 2 overflow.
6	EXF2	**Timer 2 External Flag** Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2=1. Set to cause the CPU to vector to Timer 2 interrupt routine when Timer 2 interrupt is enabled. Must be cleared by software.
5	RCLK	**Receive Clock bit** Clear to use timer 1 overflow as receive clock for serial port in mode 1 or 3. Set to use Timer 2 overflow as receive clock for serial port in mode 1 or 3.
4	TCLK	**Transmit Clock bit** Clear to use timer 1 overflow as transmit clock for serial port in mode 1 or 3. Set to use Timer 2 overflow as transmit clock for serial port in mode 1 or 3.
3	EXEN2	**Timer 2 External Enable bit** Clear to ignore events on T2EX pin for Timer 2 operation. Set to cause a capture or reload when a negative transition on T2EX pin is detected, if Timer 2 is not used to clock the serial port.
2	TR2	**Timer 2 Run Control bit** Clear to turn off Timer 2. Set to turn on Timer 2.
1	C/T2#	**Timer/Counter 2 Select bit** Clear for timer operation (input from internal clock system: f_{OSC}). Set for counter operation (input from T2 input pin).
0	CP/RL2#	**Timer 2 Capture/Reload bit** If RCLK=1 or TCLK=1, CP/RL2# is ignored and timer is forced to auto-reload on Timer 2 overflow. Clear to auto-reload on Timer 2 overflows or negative transitions on T2EX pin if EXEN2=1. Set to capture on negative transitions on T2EX pin if EXEN2=1.

Reset Value = 0000 0000b
bit addressable

Table 60. 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 35.

Reset Value = xx00 000xb

Table 63. CANIE Register

CANIE (S:C3h) – CAN Enable Interrupt message object Registers 2

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

Bit Number	Bit Mnemonic	Description
7 - 4	-	**Reserved** The values read from these bits are indeterminate. Do not set these bits.
3 - 0	IECH3:0	**Enable Interrupt by Message Object** 0 - disable IT. 1 - enable IT. IECH3:0 = 0b 0000 1100 -> Enable IT's of message objects 3 & 2.

Reset Value = xxxx 0000b

Table 64. CANBT1 Register

CANBT1 (S:B4h) – CAN bit Timing Registers 1

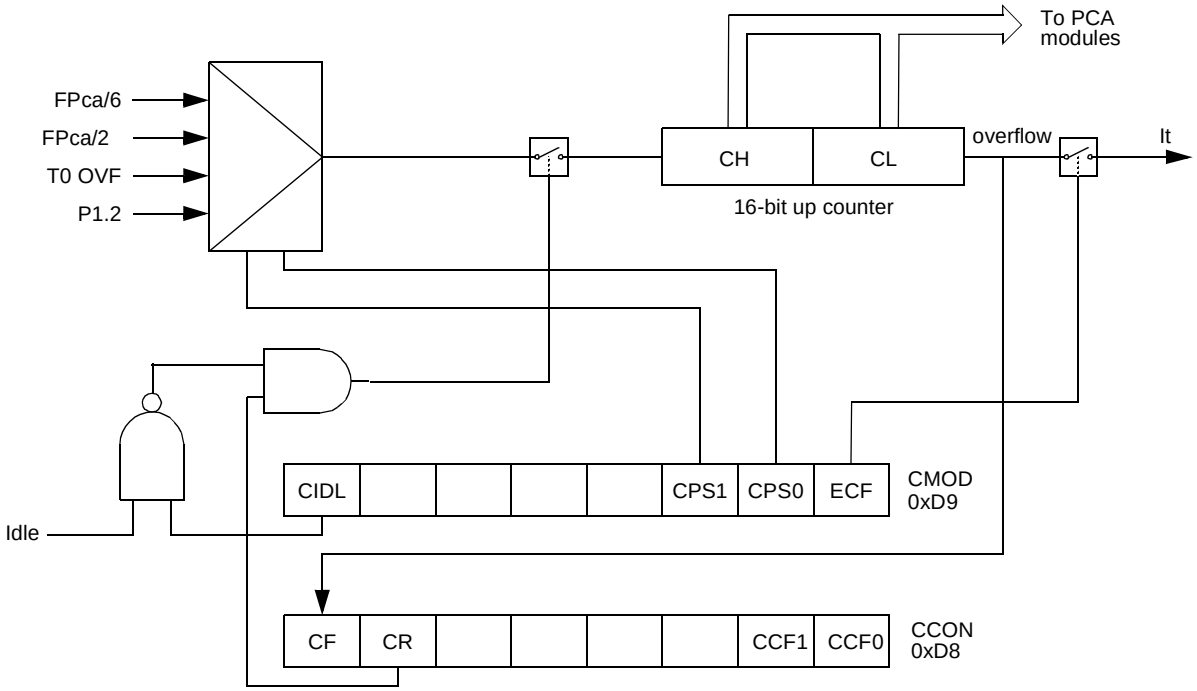
7	6	5	4	3	2	1	0
-	BRP 5	BRP 4	BRP 3	BRP 2	BRP 1	BRP 0	-

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6 - 1	BRP5:0	**Baud Rate Prescaler** **The period of the CAN** controller system clock Tscl is programmable and determines the individual bit timing.(1) $Tscl = \frac{BRP[5..0] + 1}{F_{CAN}}$
0	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.

Note: 1. The CAN controller bit timing registers must be accessed only if the CAN controller is disabled with the ENA bit of the CANGCON register set to 0.
See Figure 37.

No default value after reset.

Figure 42. PCA Timer/Counter



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

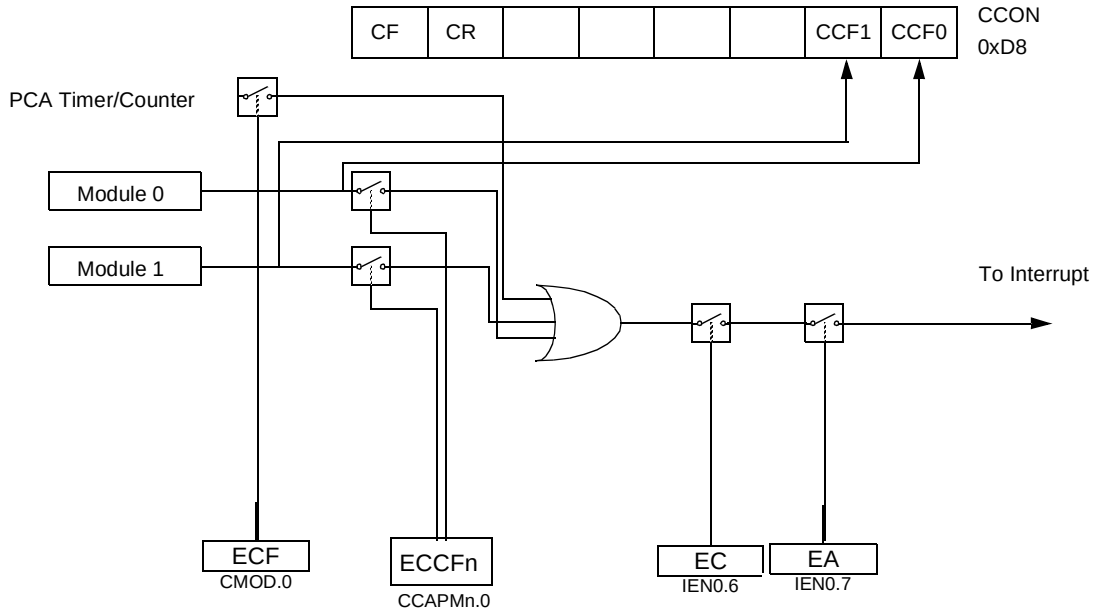
- The CIDL bit which allows the PCA to stop during idle mode.
- The ECF bit which when set causes an interrupt and the PCA overflow flag CF in CCON register to be set when the PCA timer overflows.

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.

PCA Interrupt

Figure 43. PCA Interrupt System



PCA Capture Mode

To use one of the PCA modules in capture mode either one or both of the CCAPM bits CAPN and CAPP for that module must be set. The external CEX input for the module (on port 1) is sampled for a transition. When a valid transition occurs the PCA hardware loads the value of the PCA counter registers (CH and CL) into the module's capture registers (CCAPnL and CCAPnH). If the CCFn bit for the module in the CCON SFR and the ECCFn bit in the CCAPMn SFR are set then an interrupt will be generated.

Figure 44. PCA Capture Mode

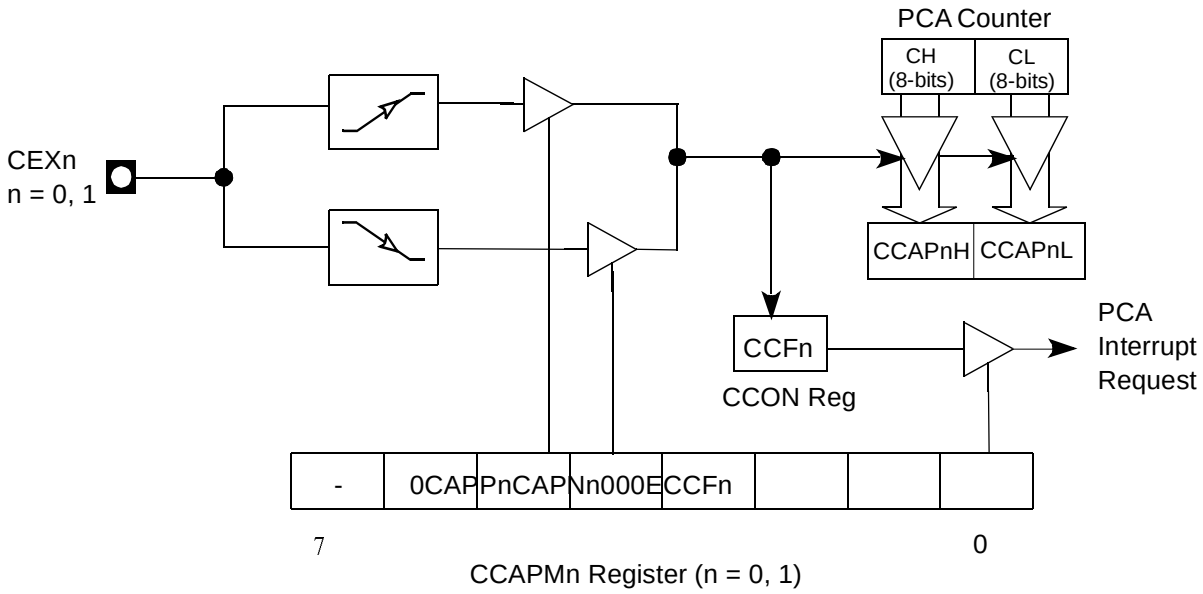
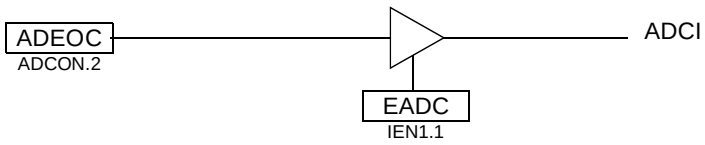


Figure 51. ADC interrupt structure



Routine Examples

```
1. Configure P1.2 and P1.3 in ADC channels
// configure channel P1.2 and P1.3 for ADC
 ADCF = 0Ch

// Enable the ADC
 ADCON = 20h
2. Start a standard conversion
// The variable 'channel' contains the channel to convert
// The variable 'value_converted' is an unsigned int
// Clear the field SCH[2:0]
 ADCON &= F8h
// Select channel
 ADCON |= channel
// Start conversion in standard mode
 ADCON |= 08h
// Wait flag End of conversion
 while((ADCON & 01h)!= 01h)
// Clear the End of conversion flag
 ADCON &= EFh
// read the value
 value_converted = (ADDH << 2)+(ADDL)

3. Start a precision conversion (need interrupt ADC)
// The variable 'channel' contains the channel to convert
// Enable ADC
 EADC = 1
// clear the field SCH[2:0]
 ADCON &= F8h
// Select the channel
 ADCON |= channel
// Start conversion in precision mode
 ADCON |= 48h
```

Note: To enable the ADC interrupt: EA = 1

Figure 54. IEN1 Register

IEN1 (S:E8h)
Interrupt Enable Register

7	6	5	4	3	2	1	0
-	-	-	-		ETIM	EADC	ECAN

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	ETIM	**TImer overrun Interrupt Enable bit** Clear to disable the timer overrun interrupt. Set to enable the timer overrun interrupt.
1	EADC	**ADC Interrupt Enable bit** Clear to disable the ADC interrupt. Set to enable the ADC interrupt.
0	ECAN	**CAN Interrupt Enable bit** Clear to disable the CAN interrupt. Set to enable the CAN interrupt.

Reset Value = xxxx x000b
bit addressable

Ordering Information

Part Number	Bootloader	Temperature Range	Package	Packing	Product Marking
T89C51CC02CA-RATIM	CAN(2)	Industrial	VQFP32	Tray	89C51CC02CA-IM
T89C51CC02CA-SISIM	CAN(2)	Industrial	PLCC28	Stick	89C51CC02CA-IM
T89C51CC02CA-TDSIM	CAN(2)	Industrial	SOIC24	Stick	89C51CC02CA-IM
T89C51CC02CA-TISIM	CAN(2)	Industrial	SOIC28	Stick	89C51CC02CA-IM
T89C51CC02UA-RATIM	UART(2)	Industrial	VQFP32	Tray	89C51CC02UA-IM
T89C51CC02UA-SISIM	UART(2)	Industrial	PLCC28	Stick	89C51CC02UA-IM
T89C51CC02UA-TDSIM	UART(2)	Industrial	SOIC24	Stick	89C51CC02UA-IM
T89C51CC02UA-TISIM	UART(2)	Industrial	SOIC28	Stick	89C51CC02UA-IM

Factory default programming for T89C51CC02CA-xxxx is Bootloader CAN and HSB = BBh:

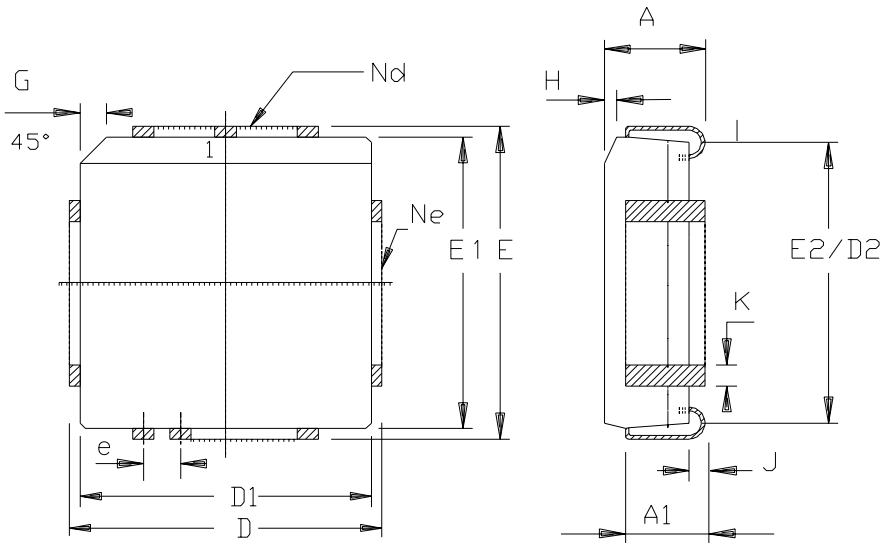
- X1 mode
- BLJB = 0 : jump to Bootloader
- LB2 = 0 : Security Level 3.(1)

Factory default programming for T89C51CC02UA-xxxx is Bootloader UART and HSB = BBh:

- X1 mode
- BLJB = 0 : jump to Bootloader
- LB2 = 0 : Security Level 3.(1)

Notes: 1. LB2 = 0 is not described in Table 22 Program load bit. LB2 = 0 is equivalent to LB1 = 0: Security Level 3.
2. Customer can change these modes by re-programming with a parallel programmer, this can be done by an Atmel distributor.

PLCC28



	MM		INCH	
A	4.20	4.57	.165	.180
A1	2.29	3.04	.090	.120
D	12.32	12.57	.485	.495
D1	11.43	11.58	.450	.456
D2	9.91	10.92	.390	.430
E	12.32	12.57	.485	.495
E1	11.43	11.58	.450	.456
E2	9.91	10.92	.390	.430
e	1.27	BSC	.050	BSC
G	1.07	1.22	.042	.048
H	1.07	1.42	.042	.056
J	0.51	-	.020	-
K	0.33	0.53	.013	.021
Nd	7		7	
Ne	7		7	
PKG STD	00			