Welcome to E-XFL.COM

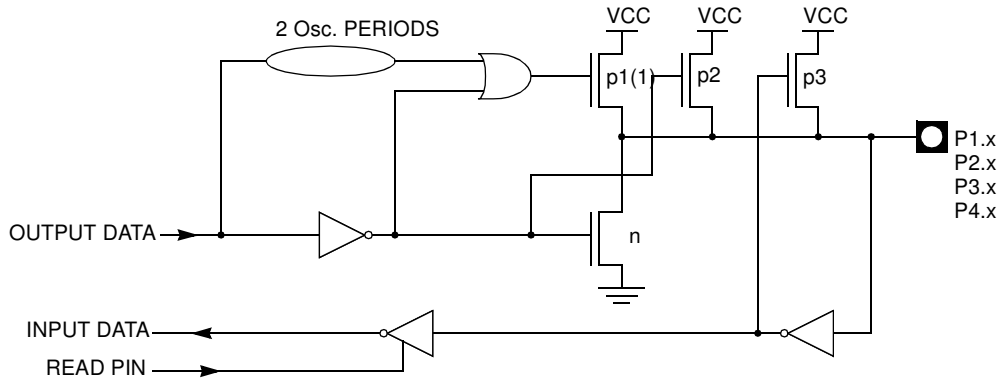
What is "Embedded - Microcontrollers"?

"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"

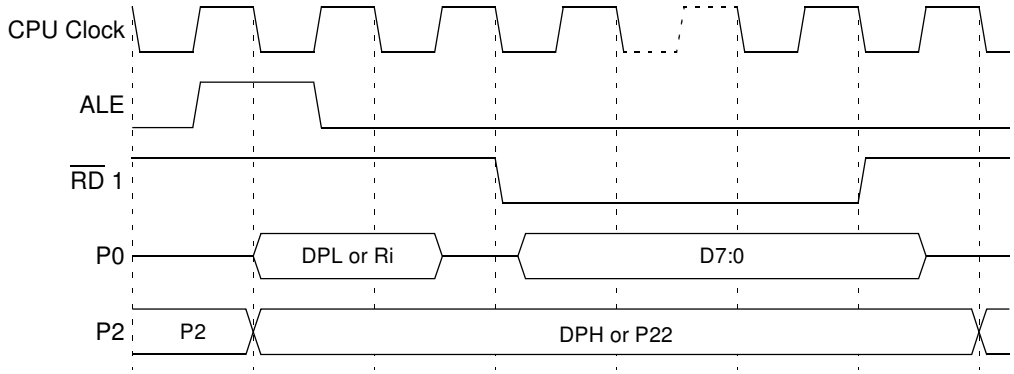
Details	
Product Status	Obsolete
Core Processor	80C51
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	32
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	1.25K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	44-VQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/t89c51cc01ua-rltim

Figure 4. Internal Pull-Up Configurations



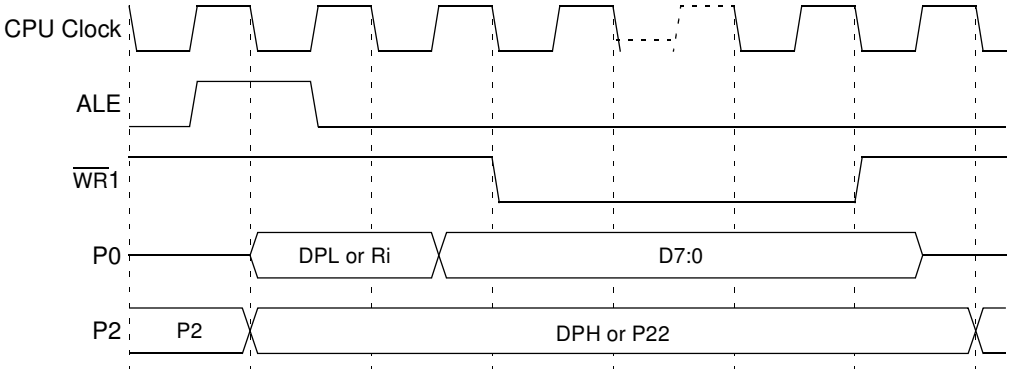
Note: Port 2 p1 assists the logic-one output for memory bus cycles.

Figure 14. External Data Read Waveforms



Notes: 1. $\overline{RD}$ signal may be stretched using M0 bit in AUXR register.
2. When executing MOVX @Ri instruction, P2 outputs SFR content.

Figure 15. External Data Write Waveforms



Notes: 1. $\overline{WR}$ signal may be stretched using M0 bit in AUXR register.
2. When executing MOVX @Ri instruction, P2 outputs SFR content.

Examples

```
;*F****************************************************************************
;* NAME: api_rd_eeprom_byte
;* DPTR contain address to read.
;* Acc contain the reading value
;* NOTE: before execute this function, be sure the EEPROM is not BUSY
;*******************************************************************************
api_rd_eeprom_byte:
; Save and clear EA
 MOV   EECON, #02h; map EEPROM in XRAM space
 MOVX A, @DPTR
 MOV   EECON, #00h; unmap EEPROM
; Restore EA
ret

;*F****************************************************************************
;* NAME: api_ld_eeprom_cl
;* DPTR contain address to load
;* Acc contain value to load
;* NOTE: in this example we load only 1 byte, but it is possible upto
;* 128 Bytes.
;* before execute this function, be sure the EEPROM is not BUSY
;*******************************************************************************
api_ld_eeprom_cl:
; Save and clear EA
 MOV   EECON, #02h ; map EEPROM in XRAM space
 MOVX @DPTR, A
 MOVEECON, #00h; unmap EEPROM
; Restore EA
ret

;*F****************************************************************************
;* NAME: api_wr_eeprom
;* NOTE: before execute this function, be sure the EEPROM is not BUSY
;*******************************************************************************
api_wr_eeprom:
; Save and clear EA
 MOV   EECON, #050h
 MOV   EECON, #0A0h
; Restore EA
ret
```

Registers

Table 21. EECON Register

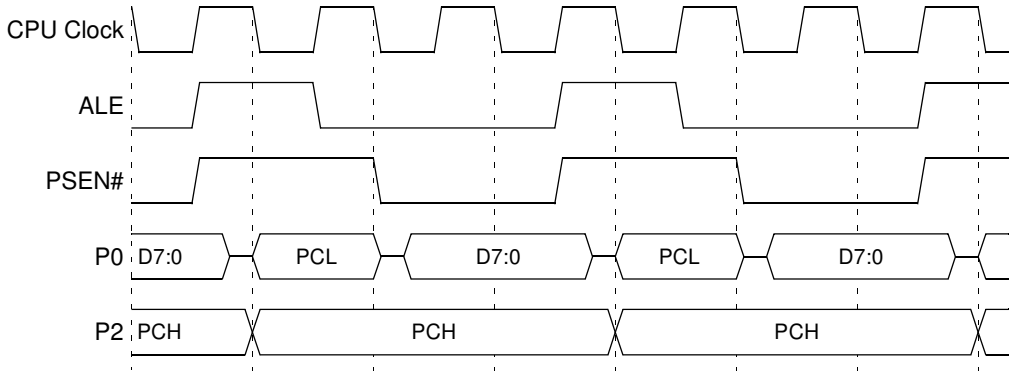
EECON (S:0D2h)
EEPROM Control Register

7	6	5	4	3	2	1	0
EEPL3	**EEPL2**	**EEPL1**	**EEPL0**	**-**	**-**	**EEE**	**EEBUSY**

Bit Number	Bit Mnemonic	Description
7-4	EEPL3-0	**Programming Launch command bits** Write 5Xh followed by AXh to EEPL to launch the programming.
3	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
2	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
1	EEE	**Enable EEPROM Space bit** Set to map the EEPROM space during MOVX instructions (Write in the column latches) Clear to map the XRAM space during MOVX.
0	EEBUSY	**Programming Busy flag** Set by hardware when programming is in progress. Cleared by hardware when programming is done. Can not be set or cleared by software.

Reset Value = XXXX XX00b
Not bit addressable

Figure 19. External Code Fetch Waveforms



Flash Memory Architecture

T89C51CC01 features two on-chip Flash memories:

- Flash memory FM0:
  containing 32K Bytes of program memory (user space) organized into 128 byte 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 In-System-Programming (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 20. Flash Memory Architecture

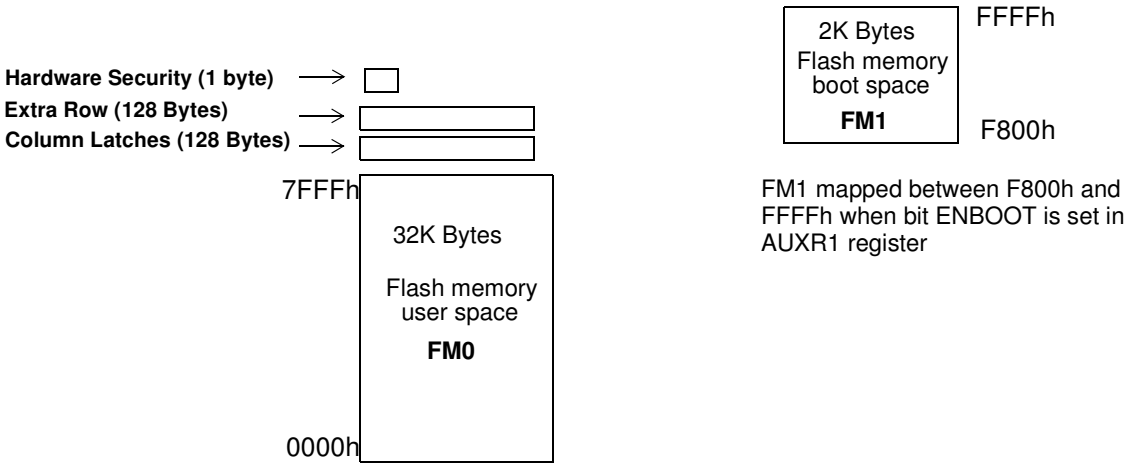
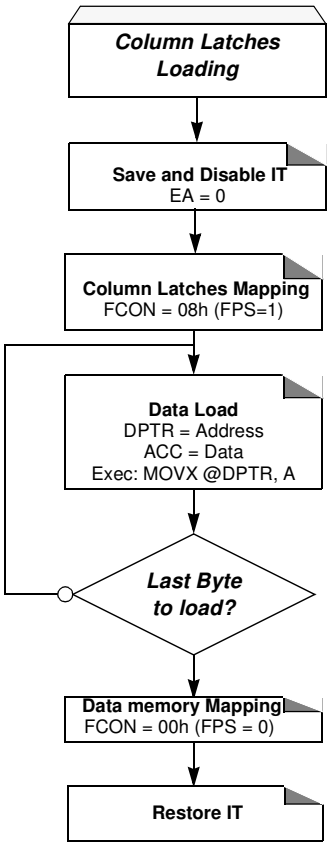


Figure 21. Column Latches Loading Procedure



Note: 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 22:

- Load up to one page of data in the column latches from address 0000h to 7FFFh.
- Save then disable the interrupts.
- Launch the programming by writing the data sequence 50h followed by A0h in FCON register (only 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 22:

- 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.
  The end of the programming indicated by the FBUSY flag cleared.
- Restore the interrupts.

For slaves A and B, bit 2 is a don't care bit; for slave C, bit 2 is set. To communicate with all of the slaves, the master must send an address FFh. To communicate with slaves A and B, but not slave C, the master can send and address FBh.

Registers

Table 35. SCON Register

SCON (S:98h)
Serial Control Register

7	6	5	4	3	2	1	0
FE/SM0	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**

Bit Number	Bit Mnemonic	Description
7	FE	**Framing Error bit (SMOD0=1**) Clear to reset the error state, not cleared by a valid stop bit. Set by hardware when an invalid stop bit is detected.
	SM0	**Serial port Mode bit 0 (SMOD0=0)** Refer to SM1 for serial port mode selection.
6	SM1	**Serial port Mode bit 1** SM0 SM1 Mode Baud Rate 0 0 Shift Register F_{XTAL}/12 (or F_{XTAL} /6 in mode X2) 0 1 8-bit UART Variable 1 0 9-bit UART F_{XTAL}/64 or F_{XTAL}/32 1 1 9-bit UART Variable
5	SM2	**Serial port Mode 2 bit/Multiprocessor Communication Enable bit** Clear to disable multiprocessor communication feature. Set to enable multiprocessor communication feature in mode 2 and 3.
4	REN	**Reception Enable bit** Clear to disable serial reception. Set to enable serial reception.
3	TB8	**Transmitter Bit 8/Ninth bit to transmit in modes 2 and 3** Clear to transmit a logic 0 in the 9th bit. Set to transmit a logic 1 in the 9th bit.
2	RB8	**Receiver Bit 8/Ninth bit received in modes 2 and 3** Cleared by hardware if 9th bit received is a logic 0. Set by hardware if 9th bit received is a logic 1.
1	TI	**Transmit Interrupt flag** Clear to acknowledge interrupt. Set by hardware at the end of the 8th bit time in mode 0 or at the beginning of the stop bit in the other modes.
0	RI	**Receive Interrupt flag** Clear to acknowledge interrupt. Set by hardware at the end of the 8th bit time in mode 0, see Figure 29. and Figure 30. in the other modes.

Reset Value = 0000 0000b
Bit addressable

Registers

Table 46. 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

- ACK Errors
  As already mentioned frames received are acknowledged by all receivers through positive acknowledgement. If no acknowledgement is received by the transmitter of the message an ACK error is indicated.

Error at Bit Level

- Monitoring
  The ability of the transmitter to detect errors is based on the monitoring of bus signals. Each node which transmits also observes the bus level and thus detects differences between the bit sent and the bit received. This permits reliable detection of global errors and errors local to the transmitter.
- Bit Stuffing
  The coding of the individual bits is tested at bit level. The bit representation used by CAN is "Non Return to Zero (NRZ)" coding, which guarantees maximum efficiency in bit coding. The synchronization edges are generated by means of bit stuffing.

Error Signalling

If one or more errors are discovered by at least one node using the above mechanisms, the current transmission is aborted by sending an "error flag". This prevents other nodes accepting the message and thus ensures the consistency of data throughout the network. After transmission of an erroneous message that has been aborted, the sender automatically re-attempts transmission.

CAN Controller Description

The CAN Controller accesses are made through SFR.
Several operations are possible by SFR:

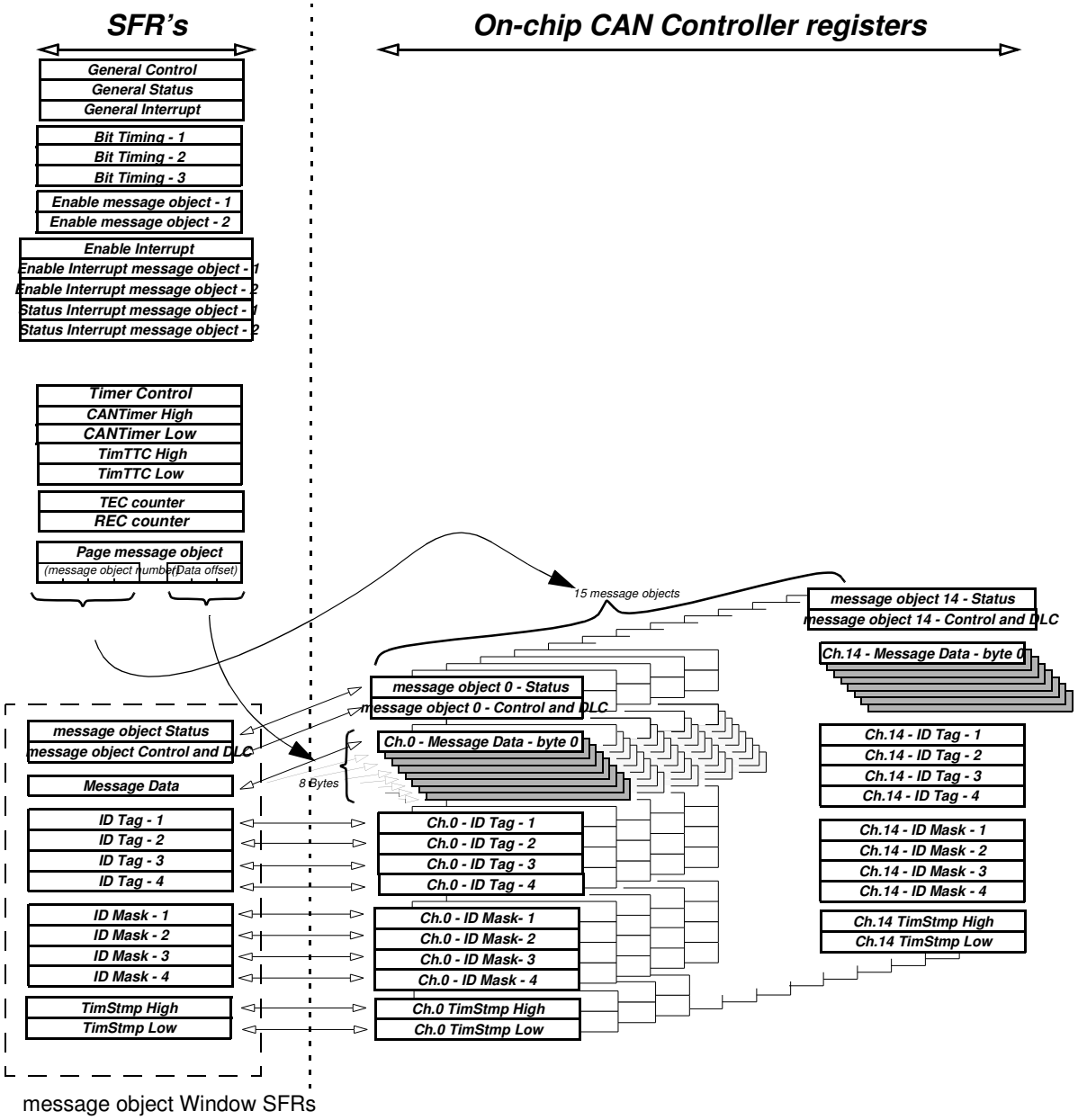
- arithmetic and logic operations, transfers and program control (SFR is accessible by direct addressing).
- 15 independent message objects are implemented, a pagination system manages their accesses.

Any message object can be programmed in a reception buffer block (even non-consecutive buffers). For the reception of defined messages one or several receiver message objects can be masked without participating in the buffer feature. An IT is generated when the buffer is full. The frames following the buffer-full interrupt will not be taken into account until at least one of the buffer message objects is re-enabled in reception. Higher priority of a message object for reception or transmission is given to the lower message object number.

The programmable 16-bit Timer (CANTIMER) is used to stamp each received and sent message in the CANSTMP register. This timer starts counting as soon as the CAN controller is enabled by the ENA bit in the CANGCON register.

The Time Trigger Communication (TTC) protocol is supported by the T89C51CC01.

Figure 44. CAN Controller Memory Organization



Bit Timing and Baud Rate

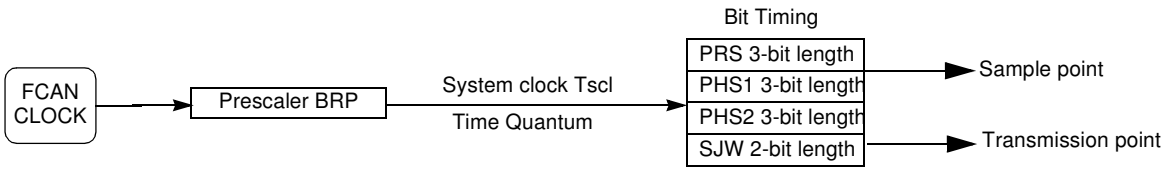
FSM's (Finite State Machine) of the CAN channel need to be synchronous to the time quantum. So, the input clock for bit timing is the clock used into CAN channel FSM's.

Field and segment abbreviations:

- BRP: Baud Rate Prescaler.
- TQ: Time Quantum (output of Baud Rate Prescaler).
- SYNS: SYNchronization Segment is 1 TQ long.
- PRS: PRopagation time Segment is programmable to be 1, 2, ..., 8 TQ long.
- PHS1: PHase Segment 1 is programmable to be 1, 2, ..., 8 TQ long.
- PHS2: PHase Segment 2 is programmable to be superior or equal to the INFORMATION PROCESSING TIME and inferior or equal to TPHS1.
- INFORMATION PROCESSING TIME is 2 TQ.
- SJW: (Re) Synchronization Jump Width is programmable to be minimum of PHS1 and 4.

The total number of TQ in a bit time has to be programmed at least from 8 to 25.

Figure 47. Sample And Transmission Point



The baud rate selection is made by Tbit calculation:

Tbit = Tsyns + Tprs + Tphs1 + Tphs2

1. Tsyns = Tscl = (BRP[5..0]+ 1)/Fcan = 1TQ.
2. Tprs = (1 to 8) * Tscl = (PRS[2..0]+ 1) * Tscl
3. Tphs1 = (1 to 8) * Tscl = (PHS1[2..0]+ 1) * Tscl
4. Tphs2 = (1 to 8) * Tscl = (PHS2[2..0]+ 1) * Tscl
   Tphs2 = Max of (Tphs1 and 2TQ)
5. Tsjw = (1 to 4) * Tscl = (SJW[1..0]+ 1) * Tscl

The total number of Tscl (Time Quanta) in a bit time must be comprised between **8 to 25**.

Table 59. CANGSTA Register

CANGSTA (S:AAh Read Only)
CAN General Status Register

7	6	5	4	3	2	1	0
-	OVFG	-	TBSY	RBSY	ENFG	BOFF	ERRP

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The values read from this bit is indeterminate. Do not set this bit.
6	OVFG	**Overload Frame Flag** This status bit is set by the hardware as long as the produced overload frame is sent. This flag does not generate an interrupt
5	-	**Reserved** The values read from this bit is indeterminate. Do not set this bit.
4	TBSY	**Transmitter Busy** This status bit is set by the hardware as long as the CAN transmitter generates a frame (remote, data, overload or error frame) or an ack field. This bit is also active during an InterFrame Spacing if a frame must be sent. This flag does not generate an interrupt.
3	RBSY	**Receiver Busy** This status bit is set by the hardware as long as the CAN receiver acquires or monitors a frame. This flag does not generate an interrupt.
2	ENFG	**Enable On-chip CAN Controller Flag** Because an enable/disable command is not effective immediately, this status bit gives the true state of a chosen mode. This flag does not generate an interrupt.
1	BOFF	**Bus Off Mode** **see Figure 49**
0	ERRP	**Error Passive Mode** **see Figure 49**

Reset Value = x0x0 0000b

Table 69. CANIE2 Register

CANIE2 (S:C3h)
CAN Enable Interrupt Message Object Registers 2

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

Bit Number	Bit Mnemonic	Description
7-0	IECH7:0	**Enable interrupt by Message Object** 0 - disable IT. 1 - enable IT. IECH7:0 = 0b 0000 1100 -> Enable IT's of message objects 3 and 2.

Reset Value = 0000 0000b

Table 70. 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. $Tscl = \frac{BRP[5..0] + 1}{Fcan}$
0	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.

Note: 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 48.

No default value after reset.

Table 93. CANTCON Register

CANTCON (S:A1h)
CAN Timer ClockControl

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

Bit Number	Bit Mnemonic	Description
7-0	TPRESC7:0	**Timer Prescaler of CAN Timer** This register is a prescaler for the main timer upper counter range = 0 to 255. See Figure 51.

Reset Value = 00h

Table 94. CANTIMH Register

CANTIMH (S:ADh)
CAN Timer High

7	6	5	4	3	2	1	0
CANGTIM 15	CANGTIM 14	CANGTIM 13	CANGTIM 12	CANGTIM 11	CANGTIM 10	CANGTIM 9	CANGTIM 8

Bit Number	Bit Mnemonic	Description
7-0	CANGTIM15:8	**High byte of Message Timer** See Figure 51.

Reset Value = 0000 0000b

Table 95. CANTIML Register

CANTIML (S:ACh)
CAN Timer Low

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

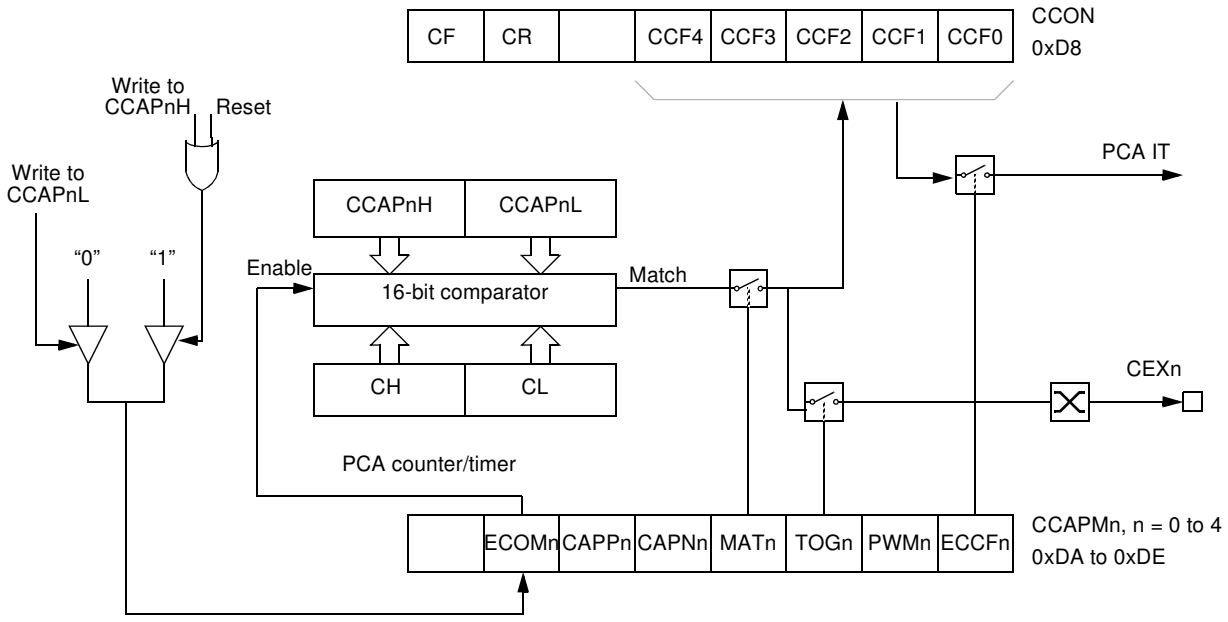
Bit Number	Bit Mnemonic	Description
7-0	CANGTIM7:0	**Low byte of Message Timer** See Figure 51.

Reset Value = 0000 0000b

High Speed Output Mode

In this mode the CEX output (on port 1) associated with the PCA module will toggle each time a match occurs between the PCA counter and the module's capture registers. To activate this mode the TOG, MAT, and ECOM bits in the module's CCAPMn SFR must be set.

Figure 57. PCA High Speed Output Mode



Pulse Width Modulator Mode

All the PCA modules can be used as PWM outputs. The output frequency depends on the source for the PCA timer. All the modules will have the same output frequency because they all share the PCA timer. The duty cycle of each module is independently variable using the module's capture register CCAPLn. When the value of the PCA CL SFR is less than the value in the module's CCAPLn SFR the output will be low, when it is equal to or greater than it, the output will be high. When CL overflows from FF to 00, CCAPLn is reloaded with the value in CCAPHn. the allows the PWM to be updated without glitches. The PWM and ECOM bits in the module's CCAPMn register must be set to enable the PWM mode.

Table 104. CCAPMn Registers

CCAPM0 (S:DAh)
CCAPM1 (S:DBh)
CCAPM2 (S:DCh)
CCAPM3 (S:DDh)
CCAPM4 (S:DEh)
PCA Compare/Capture Module n Mode registers (n=0..4)

7	6	5	4	3	2	1	0
-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The Value read from this bit is indeterminate. Do not set this bit.
6	ECOMn	**Enable Compare Mode Module x bit** Clear to disable the Compare function. Set to enable the Compare function. The Compare function is used to implement the software Timer, the high-speed output, the Pulse Width Modulator (PWM) and the Watchdog Timer (WDT).
5	CAPPn	**Capture Mode (Positive) Module x bit** Clear to disable the Capture function triggered by a positive edge on CEXx pin. Set to enable the Capture function triggered by a positive edge on CEXx pin
4	CAPNn	**Capture Mode (Negative) Module x bit** Clear to disable the Capture function triggered by a negative edge on CEXx pin. Set to enable the Capture function triggered by a negative edge on CEXx pin.
3	MATn	**Match Module x bit** Set when a match of the PCA Counter with the Compare/Capture register sets CCFx bit in CCON register, flagging an interrupt.
2	TOGn	**Toggle Module x bit** The toggle mode is configured by setting ECOMx, MATx and TOGx bits. Set when a match of the PCA Counter with the Compare/Capture register toggles the CEXx pin.
1	PWMn	**Pulse Width Modulation Module x Mode bit** Set to configure the module x as an 8-bit Pulse Width Modulator with output waveform on CEXx pin.
0	ECCFn	**Enable CCFx Interrupt bit** Clear to disable CCFx bit in CCON register to generate an interrupt request. Set to enable CCFx bit in CCON register to generate an interrupt request.

Reset Value = X000 0000b

Analog-to-Digital Converter (ADC)

This section describes the on-chip 10 bit analog-to-digital converter of the T89C51CC01. Eight ADC channels are available for sampling of the external sources AN0 to AN7. An analog multiplexer allows the single ADC converter to select one from the 8 ADC channels as ADC input voltage (ADCIN). ADCIN is converted by the 10-bit cascaded potentiometric ADC.

Two modes of conversion are available:
- Standard conversion (8 bits).
- Precision conversion (10 bits).

For the precision conversion, set bit PSIDLE in ADCON register and start conversion. The device is in a pseudo-idle mode, the CPU does not run but the peripherals are always running. This mode allows digital noise to be as low as possible, to ensure high precision conversion.

For this mode it is necessary to work with end of conversion interrupt, which is the only way to wake the device up.

If another interrupt occurs during the precision conversion, it will be served only after this conversion is completed.

Features

- 8 channels with multiplexed inputs
- 10-bit cascaded potentiometric ADC
- Conversion time 16 micro-seconds (typ.)
- Zero Error (offset) ± 2 LSB max
- Positive External Reference Voltage Range (VAREF) 2.4 to 3.0 Volt (typ.)
- ADCIN Range 0 to 3Volt
- Integral non-linearity typical 1 LSB, max. 2 LSB
- Differential non-linearity typical 0.5 LSB, max. 1 LSB
- Conversion Complete Flag or Conversion Complete Interrupt
- Selectable ADC Clock

ADC Port 1 I/O Functions

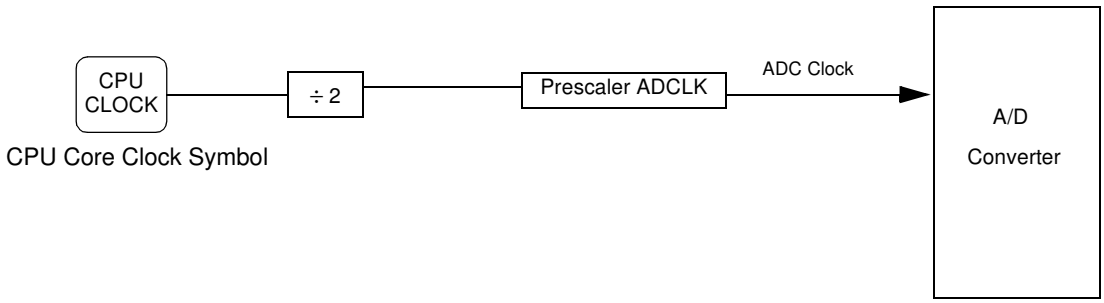
Port 1 pins are general I/O that are shared with the ADC channels. The channel select bit in ADCF register define which ADC channel/port1 pin will be used as ADCIN. The remaining ADC channels/port1 pins can be used as general-purpose I/O or as the alternate function that is available.

A conversion launched on a channel which are not selected on ADCF register will not have any effect.

VAREF

VAREF should be connected to a low impedance point and must remain in the range specified in Table 122. If the ADC is not used, it is recommended to connect VAREF to VAGND.

Figure 61. A/D Converter clock



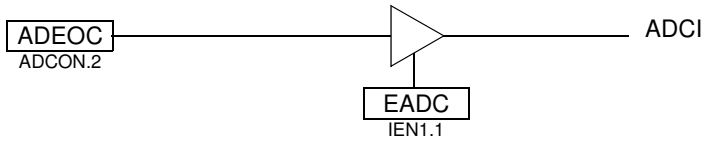
ADC Standby Mode

When the ADC is not used, it is possible to set it in standby mode by clearing bit ADEN in ADCON register. In this mode its power dissipation is reduced.

IT ADC Management

An interrupt end-of-conversion will occurs when the bit ADEOC is activated and the bit EADC is set. For re-arming the interrupt the bit ADEOC must be cleared by software.

Figure 62. ADC Interrupt Structure



Routines 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 and = F8h
// Select channel
 ADCON | = channel
// Start conversion in standard mode
 ADCON | = 08h
// Wait flag End of conversion
 while((ADCON and 01h)! = 01h)
// Clear the End of conversion flag
 ADCON and = 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
```

Table 116. 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

Datasheet Change Log for T89C51CC01

Changes from 4129F - 11/02 to 4129G - 04/03

1. Changed the endurance of Flash to 100, 000 Write/Erase cycles.
2. Added note on Flash retention formula for V_{IH1}, in Section "DC Parameters for Standard Voltage", page 144.

Changes from 4129G - 04/03 to 4129H - 10/03

1. Updated “Electrical Characteristics” on page 144.
2. Corrected Figure 46 on page 84.

Changes from 4129H - 10/03 to 4129I - 12/03

1. Correction in Registers CPA and CPS0.
2. Added note regarding PSEN during power On see Section “Hardware Boot Process”, page 48.

Changes from 4129I - 12/03 to 4129J - 08/04

1. Figure clock-out mode modified see, Figure 37 on page 67.
2. Added explanation on the CAN protocol, see Section “CAN Controller”, page 75.
3. Corrected error in Table 53 on page 72, (1.25ms to 1.25s) for Time-out Computation.

Changes from 4129J - 08/04 to 4129K 01/05

1. Minor corrections throughout the document.
2. Clarification to Mode Switching Waveforms diagram. See page 16.

Changes from 4129K 01/05 to 4129L 08/05

1. Added green product ordering information.

Changes from 4129L 08/05 to 4129M 02/08

1. Removed non-green packages from ordering information.

Changes from 4129M 02/08 to 4129N 03/08

1. Removed CA-BGA package offering from ordering information.
2. Updated package drawings.