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, SPI, UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	36
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	2.25K 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	64-LQFP
Supplier Device Package	64-VQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89c51cc03ca-rdtum

write the new byte back to the latch. These Read-Modify-Write instructions are directed to the latch rather than the pin in order to avoid possible misinterpretation of voltage (and therefore, logic) levels at the pin. For example, a Port bit used to drive the base of an external bipolar transistor can not rise above the transistor's base-emitter junction voltage (a value lower than VIL). With a logic one written to the bit, attempts by the CPU to read the Port at the pin are misinterpreted as logic zero. A read of the latch rather than the pins returns the correct logic-one value.

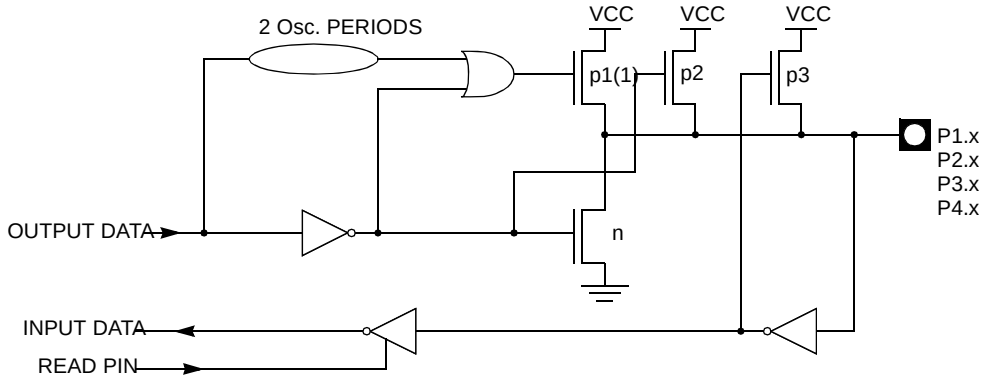
Quasi-Bidirectional Port Operation

Port 1, Port 2, Port 3 and Port 4 have fixed internal pull-ups and are referred to as "quasi-bidirectional" Ports. When configured as an input, the pin impedance appears as logic one and sources current in response to an external logic zero condition. Port 0 is a "true bidirectional" pin. The pins float when configured as input. Resets write logic one to all Port latches. If logical zero is subsequently written to a Port latch, it can be returned to input conditions by a logical one written to the latch.

Note: Port latch values change near the end of Read-Modify-Write instruction cycles. Output buffers (and therefore the pin state) update early in the instruction after Read-Modify-Write instruction cycle.

Logical zero-to-one transitions in Port 1, Port 2, Port 3 and Port 4 use an additional pull-up (p1) to aid this logic transition (see Figure 4.). This increases switch speed. This extra pull-up sources 100 times normal internal circuit current during 2 oscillator clock periods. The internal pull-ups are field-effect transistors rather than linear resistors. Pull-ups consist of three p-channel FET (pFET) devices. A pFET is on when the gate senses logical zero and off when the gate senses logical one. pFET #1 is turned on for two oscillator periods immediately after a zero-to-one transition in the Port latch. A logical one at the Port pin turns on pFET #3 (a weak pull-up) through the inverter. This inverter and pFET pair form a latch to drive logical one. pFET #2 is a very weak pull-up switched on whenever the associated nFET is switched off. This is traditional CMOS switch convention. Current strengths are 1/10 that of pFET #3.

Figure 4. Internal Pull-Up Configurations



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

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
T2CON	C8h	Timer/Counter 2 control	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#
T2MOD	C9h	Timer/Counter 2 Mode	–	–	–	–	–	–	T2OE	DCEN
RCAP2H	CBh	Timer/Counter 2 Reload/Capture High byte	–	–	–	–	–	–	–	–
RCAP2L	CAh	Timer/Counter 2 Reload/Capture Low byte	–	–	–	–	–	–	–	–
WDTRST	A6h	WatchDog Timer Reset	–	–	–	–	–	–	–	–
WDTPRG	A7h	WatchDog Timer Program	–	–	–	–	–	S2	S1	S0

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
SCON	98h	Serial Control	FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SBUF	99h	Serial Data Buffer	–	–	–	–	–	–	–	–
SADEN	B9h	Slave Address Mask	–	–	–	–	–	–	–	–
SADDR	A9h	Slave Address	–	–	–	–	–	–	–	–

Mnemonic	Add	Name	7	6	5	4	3	2	1	0
CCON	D8h	PCA Timer/Counter Control	CF	CR	–	CCF4	CCF3	CCF2	CCF1	CCF0
CMOD	D9h	PCA Timer/Counter Mode	CIDL	WDTE	–	–	–	CPS1	CPS0	ECF
CL	E9h	PCA Timer/Counter Low byte	–	–	–	–	–	–	–	–
CH	F9h	PCA Timer/Counter High byte	–	–	–	–	–	–	–	–
CCAPM0	DAh	PCA Timer/Counter Mode 0	–	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0
CCAPM1	DBh	PCA Timer/Counter Mode 1		ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1
CCAPM2	DCh	PCA Timer/Counter Mode 2		ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2
CCAPM3	DDh	PCA Timer/Counter Mode 3		ECOM3	CAPP3	CAPN3	MAT3	TOG3	PWM3	ECCF3
CCAPM4	DEh	PCA Timer/Counter Mode 4		ECOM4	CAPP4	CAPN4	MAT4	TOG4	PWM4	ECCF4
CCAP0H	FAh	PCA Compare Capture Module 0 H	CCAP0H7	CCAP0H6	CCAP0H5	CCAP0H4	CCAP0H3	CCAP0H2	CCAP0H1	CCAP0H0
CCAP1H	FBh	PCA Compare Capture Module 1 H	CCAP1H7	CCAP1H6	CCAP1H5	CCAP1H4	CCAP1H3	CCAP1H2	CCAP1H1	CCAP1H0
CCAP2H	FCh	PCA Compare Capture Module 2 H	CCAP2H7	CCAP2H6	CCAP2H5	CCAP2H4	CCAP2H3	CCAP2H2	CCAP2H1	CCAP2H0
CCAP3H	FDh	PCA Compare Capture Module 3 H	CCAP3H7	CCAP3H6	CCAP3H5	CCAP3H4	CCAP3H3	CCAP3H2	CCAP3H1	CCAP3H0
CCAP4H	FEh	PCA Compare Capture Module 4 H	CCAP4H7	CCAP4H6	CCAP4H5	CCAP4H4	CCAP4H3	CCAP4H2	CCAP4H1	CCAP4H0
CCAP0L	EAh	PCA Compare Capture Module 0 L	CCAP0L7	CCAP0L6	CCAP0L5	CCAP0L4	CCAP0L3	CCAP0L2	CCAP0L1	CCAP0L0
CCAP1L	EBh	PCA Compare Capture Module 1 L	CCAP1L7	CCAP1L6	CCAP1L5	CCAP1L4	CCAP1L3	CCAP1L2	CCAP1L1	CCAP1L0
CCAP2L	ECh	PCA Compare Capture Module 2 L	CCAP2L7	CCAP2L6	CCAP2L5	CCAP2L4	CCAP2L3	CCAP2L2	CCAP2L1	CCAP2L0
CCAP3L	EDh	PCA Compare Capture Module 3 L	CCAP3L7	CCAP3L6	CCAP3L5	CCAP3L4	CCAP3L3	CCAP3L2	CCAP3L1	CCAP3L0
CCAP4L	EEh	PCA Compare Capture Module 4 L	CCAP4L7	CCAP4L6	CCAP4L5	CCAP4L4	CCAP4L3	CCAP4L2	CCAP4L1	CCAP4L0

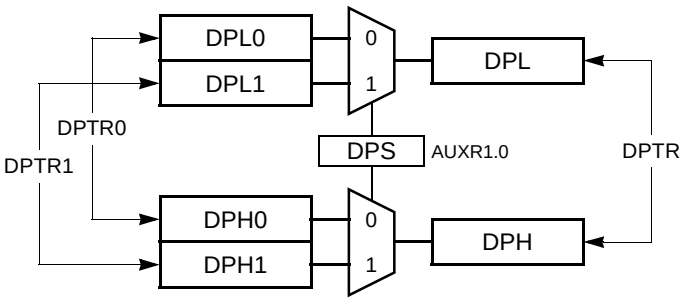
Mnemonic	Add	Name	7	6	5	4	3	2	1	0
CANIE2	C3h	CAN Interrupt Enable Channel byte 2	IECH7	IECH6	IECH5	IECH4	IECH3	IECH2	IECH1	IECH0
CANSIT1	BAh	CAN Status Interrupt Channel byte1	–	SIT14	SIT13	SIT12	SIT11	SIT10	SIT9	SIT8
CANSIT2	BBh	CAN Status Interrupt Channel byte2	SIT7	SIT6	SIT5	SIT4	SIT3	SIT2	SIT1	SIT0
CANTCON	A1h	CAN Timer Control	TPRESC 7	TPRESC 6	TPRESC 5	TPRESC 4	TPRESC 3	TPRESC 2	TPRESC 1	TPRESC 0
CANTIMH	ADh	CAN Timer high	CANTIM 15	CANTIM 14	CANTIM 13	CANTIM 12	CANTIM 11	CANTIM 10	CANTIM 9	CANTIM 8
CANTIML	ACh	CAN Timer low	CANTIM 7	CANTIM 6	CANTIM 5	CANTIM 4	CANTIM 3	CANTIM 2	CANTIM 1	CANTIM 0
CANSTMPH	AFh	CAN Timer Stamp high	TIMSTMP 15	TIMSTMP 14	TIMSTMP 13	TIMSTMP 12	TIMSTMP 11	TIMSTMP 10	TIMSTMP 9	TIMSTMP 8
CANSTMPL	AEh	CAN Timer Stamp low	TIMSTMP7	TIMSTMP 6	TIMSTMP 5	TIMSTMP 4	TIMSTMP 3	TIMSTMP 2	TIMSTMP 1	TIMSTMP 0
CANTTCH	A5h	CAN Timer TTC high	TIMTTC 15	TIMTTC 14	TIMTTC 13	TIMTTC 12	TIMTTC 11	TIMTTC 10	TIMTTC 9	TIMTTC 8
CANTTCL	A4h	CAN Timer TTC low	TIMTTC 7	TIMTTC 6	TIMTTC 5	TIMTTC 4	TIMTTC 3	TIMTTC 2	TIMTTC 1	TIMTTC 0
CANTEC	9Ch	CAN Transmit Error Counter	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
CANREC	9Dh	CAN Receive Error Counter	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
CANPAGE	B1h	CAN Page	CHNB3	CHNB2	CHNB1	CHNB0	AINC	INDX2	INDX1	INDX0
CANSTCH	B2h	CAN Status Channel	DLCW	TXOK	RXOK	BERR	SERR	CERR	FERR	AERR
CANCONCH	B3h	CAN Control Channel	CONCH1	CONCH0	RPLV	IDE	DLC3	DLC2	DLC1	DLC0
CANMSG	A3h	CAN Message Data	MSG7	MSG6	MSG5	MSG4	MSG3	MSG2	MSG1	MSG0
CANIDT1	BCh	CAN Identifier Tag byte 1(Part A) CAN Identifier Tag byte 1(PartB)	IDT10 IDT28	IDT9 IDT27	IDT8 IDT26	IDT7 IDT25	IDT6 IDT24	IDT5 IDT23	IDT4 IDT22	IDT3 IDT21
CANIDT2	BDh	CAN Identifier Tag byte 2 (PartA) CAN Identifier Tag byte 2 (PartB)	IDT2 IDT20	IDT1 IDT19	IDT0 IDT18	– IDT17	– IDT16	– IDT15	– IDT14	– IDT13
CANIDT3	BEh	CAN Identifier Tag byte 3(PartA) CAN Identifier Tag byte 3(PartB)	– IDT12	– IDT11	– IDT10	– IDT9	– IDT8	– IDT7	– IDT6	– IDT5

Dual Data Pointer

Description

The AT89C51CC03 implements a second data pointer for speeding up code execution and reducing code size in case of intensive usage of external memory accesses.
DPTR 0 and DPTR 1 are seen by the CPU as DPTR and are accessed using the SFR addresses 83h and 84h that are the DPH and DPL addresses. The DPS bit in AUXR1 register (see Figure 8) is used to select whether DPTR is the data pointer 0 or the data pointer 1 (see Figure 13).

Figure 13. Dual Data Pointer Implementation



Application

Software can take advantage of the additional data pointers to both increase speed and reduce code size, for example, block operations (copy, compare…) are well served by using one data pointer as a “source” pointer and the other one as a “destination” pointer. Hereafter is an example of block move implementation using the two pointers and coded in assembler. The latest C compiler takes also advantage of this feature by providing enhanced algorithm libraries.

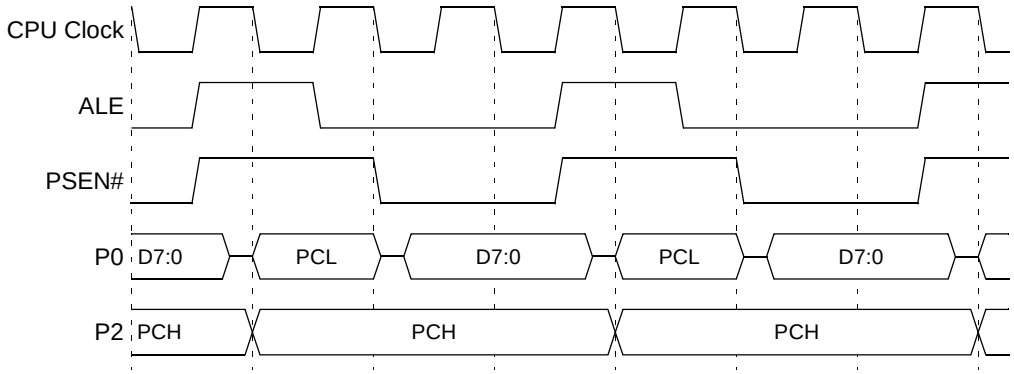
The INC instruction is a short (2 Bytes) and fast (6 machine cycle) way to manipulate the DPS bit in the AUXR1 register. However, note that the INC instruction does not directly force the DPS bit to a particular state, but simply toggles it. In simple routines, such as the block move example, only the fact that DPS is toggled in the proper sequence matters, not its actual value. In other words, the block move routine works the same whether DPS is '0' or '1' on entry.

```
; ASCII block move using dual data pointers
; Modifies DPTR0, DPTR1, A and PSW
; Ends when encountering NULL character
; Note: DPS exits opposite to the entry state unless an extra INC AUXR1 is added

AUXR1EQU0A2h

move:movDPTR,#SOURCE ; address of SOURCE
   incAUXR1 ; switch data pointers
   movDPTR,#DEST ; address of DEST
mv_loop:incAUXR1; switch data pointers
   movxA,@DPTR; get a byte from SOURCE
   incDPTR; increment SOURCE address
   incAUXR1; switch data pointers
   movx@DPTR,A; write the byte to DEST
   incDPTR; increment DEST address
   jnzmv_loop; check for NULL terminator
end_move:
```

Figure 22. External Code Fetch Waveforms



Flash Memory Architecture

AT89C51CC03 features two on-chip Flash memories:

- Flash memory FM0:
  containing 64K 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.

The bit ENBOOT in AUXR1 register is used to map FM1 from F800h to FFFFh. Figure 23 and Figure 24 show the Flash memory configuration with ENBOOT=1 and ENBOOT=0.

Figure 23. Flash Memory Architecture with ENBOOT=1 (boot mode)

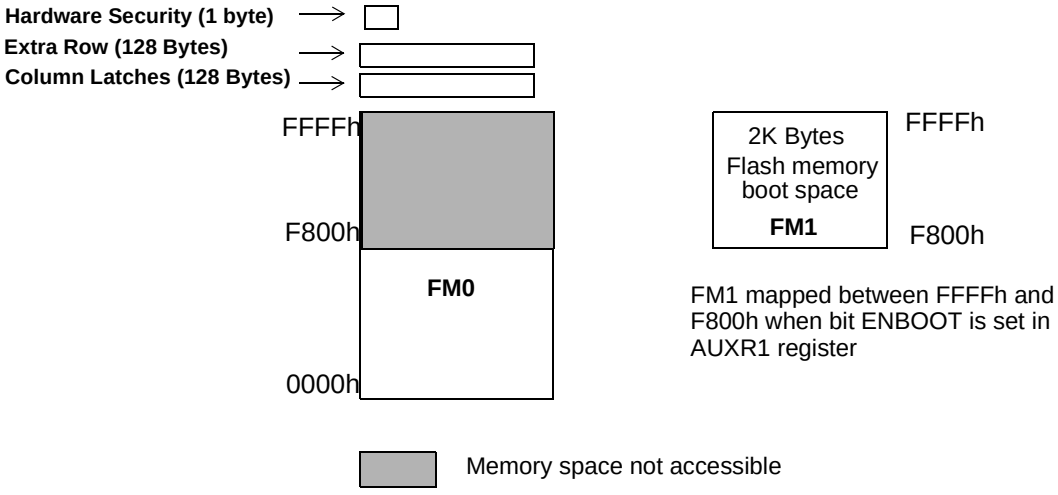
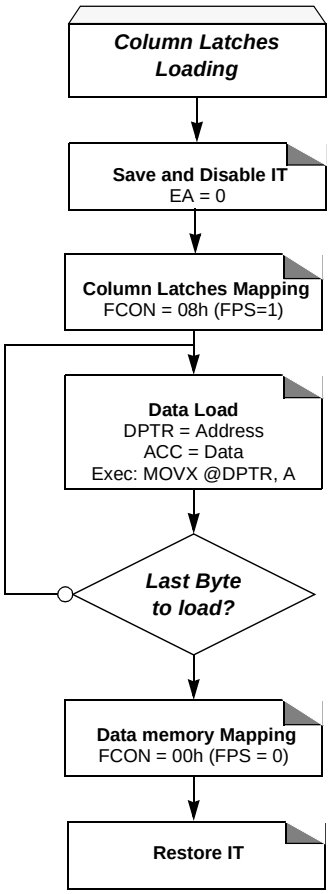


Figure 25. Column Latches Loading Procedure



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 26:

- Load up to one page of data in the column latches from address 0000h to FFFFh.
- Save and 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 26:

- Load data in the column latches from address FF80h to FFFFh.
- Save and Disable the interrupts.
- Launch the programming by writing the data sequence 52h followed by A2h in FCON register (only from FM1).
  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 25. 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 33. and Figure 34. in the other modes.

Reset Value = 0000 0000b
Bit addressable

Table 32. TH0 Register

TH0 (S:8Ch)
Timer 0 High Byte Register

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

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

Reset Value = 0000 0000b

Table 33. 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 34. 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 35. TL1 Register

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

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

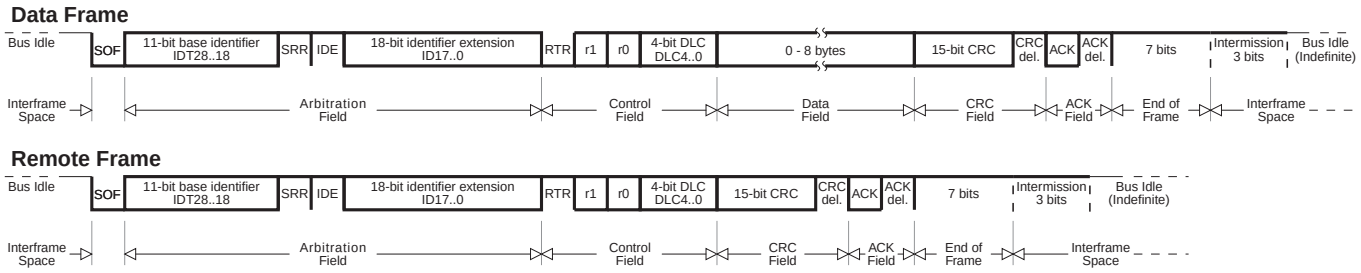
Bit Number	Bit Mnemonic	Description
7:0		**Low Byte of Timer 1.**

Reset Value = 0000 0000b

number of following data bytes in the "Data field". In a remote frame, the DLC contains the number of requested data bytes. The "Data field" that follows can hold up to 8 data bytes. The frame integrity is guaranteed by the following "Cyclic Redundant Check (CRC)" sum. The "ACKnowledge (ACK) field" compromises the ACK slot and the ACK delimiter. The bit in the ACK slot is sent as a recessive bit and is overwritten as a dominant bit by the receivers which have at this time received the data correctly. Correct messages are acknowledged by the receivers regardless of the result of the acceptance test. The end of the message is indicated by "End Of Frame (EOF)". The "Intermission Frame Space (IFS)" is the minimum number of bits separating consecutive messages. If there is no following bus access by any node, the bus remains idle.

CAN Extended Frame

Figure 44. CAN Extended Frames



A message in the CAN extended frame format is likely the same as a message in CAN standard frame format. The difference is the length of the identifier used. The identifier is made up of the existing 11-bit identifier (base identifier) and an 18-bit extension (identifier extension). The distinction between CAN standard frame format and CAN extended frame format is made by using the IDE bit which is transmitted as dominant in case of a frame in CAN standard frame format, and transmitted as recessive in the other case.

Format Co-existence

As the two formats have to co-exist on one bus, it is laid down which message has higher priority on the bus in the case of bus access collision with different formats and the same identifier / base identifier: The message in CAN standard frame format always has priority over the message in extended format.

There are three different types of CAN modules available:

- 2.0A - Considers 29 bit ID as an error
- 2.0B Passive - Ignores 29 bit ID messages
- 2.0B Active - Handles both 11 and 29 bit ID Messages

Bit Timing

To ensure correct sampling up to the last bit, a CAN node needs to re-synchronize throughout the entire frame. This is done at the beginning of each message with the falling edge SOF and on each recessive to dominant edge.

Bit Construction

One CAN bit time is specified as four non-overlapping time segments. Each segment is constructed from an integer multiple of the Time Quantum. The Time Quantum or TQ is the smallest discrete timing resolution used by a CAN node.

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 53. Line Error Mode

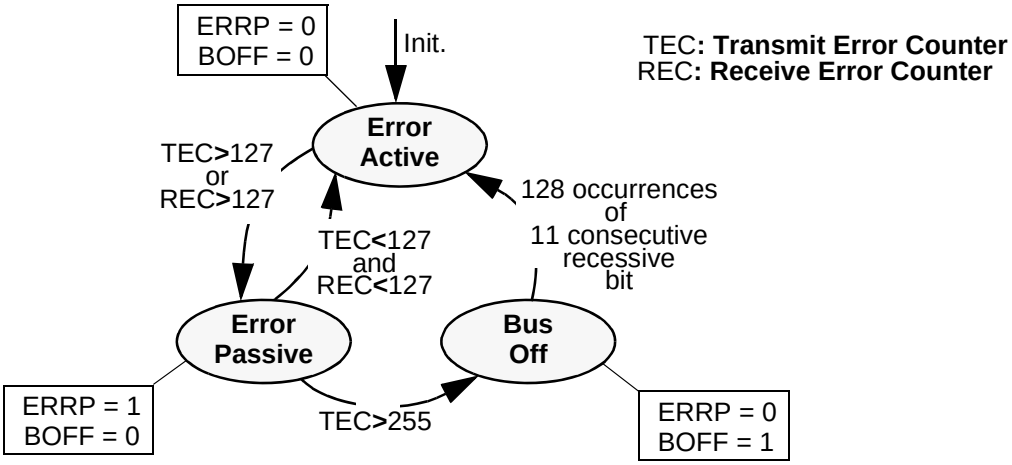


Table 53. CANGIE Register

CANGIE (S:C1h)
CAN General Interrupt Enable

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.

Note: See Figure 50

Reset Value = xx00 000xb

Table 62. CANBT3 Register

CANBT3 (S:B6h)
CAN Bit Timing Registers 3

7	6	5	4	3	2	1	0
-	PHS2 2	PHS2 1	PHS2 0	PHS1 2	PHS1 1	PHS1 0	SMP

Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
6-4	PHS2 2:0	**Phase Segment 2** This phase is used to compensate for phase edge errors. This segment can be shortened by the re-synchronization jump width. Tphs2 = Tscl x (PHS2[2..0] + 1) Phase segment 2 is the maximum of Phase segment 1 and the Information Processing Time (= 2TQ).
3-1	PHS1 2:0	**Phase Segment 1** This phase is used to compensate for phase edge errors. This segment can be lengthened by the re-synchronization jump width. Tphs1 = Tscl x (PHS1[2..0] + 1)
0	SMP	**Sample Type** 0 - once, at the sample point. 1 - three times, the threefold sampling of the bus is the sample point and twice over a distance of a 1/2 period of the Tscl. The result corresponds to the majority decision of the three values.

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 52.

No default value after reset.

Table 81. CANIDM4 Register for V2.0 part B

CANIDM4 for V2.0 part B (S:C7h)
CAN Identifier Mask Registers 4

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

Bit Number	Bit Mnemonic	Description
7-3	IDMSK4:0	**IDentifier Mask Value** 0 - comparison true forced. 1 - bit comparison enabled. See Figure 54.
2	RTRMSK	**Remote Transmission Request Mask Value** 0 - comparison true forced. 1 - bit comparison enabled.
1	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
0	IDEMSK	**IDentifier Extension Mask Value** 0 - comparison true forced. 1 - bit comparison enabled.

Note: The ID Mask is only used for reception.

No default value after reset.

Table 82. CANMSG Register

CANMSG (S:A3h)
CAN Message Data Register

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

Bit Number	Bit Mnemonic	Description
7-0	MSG7:0	**Message Data** This register contains the mailbox data byte pointed at the page message object register. After writing in the page message object register, this byte is equal to the specified message location (in the mailbox) of the pre-defined identifier + index. If auto-incrementation is used, at the end of the data register writing or reading cycle, the mailbox pointer is auto-incremented. The range of the counting is 8 with no end loop (0, 1,..., 7, 0,...)

No default value after reset.

SPCON, SPSTA and SPDAT registers may be read and written at any time while there is no on-going exchange. However, special care should be taken when writing to them while a transmission is on-going:

- Do not change SPR2, SPR1 and SPR0
- Do not change CPHA and CPOL
- Do not change MSTR
- Clearing SPEN would immediately disable the peripheral
- Writing to the SPDAT will cause an overflow.

Table 115. IPH1 Register

IPH1 (S:F7h)
Interrupt High Priority Register 1

7	6	5	4	3	2	1	0
-	-	-	-	SPIH	POVRH	PADCH	PCANH

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	SPIH	**SPI Interrupt Priority Level Most Significant bit** SPIH SPIL Priority level 0 0 Lowest 0 1 1 0 1 1 Highest
2	POVRH	**Timer overrun Interrupt Priority Level Most Significant bit** POVRH POVRL Priority level 0 0 Lowest 0 1 1 0 1 1 Highest
1	PADCH	**ADC Interrupt Priority Level Most Significant bit** PADCH PADCL Priority level 0 0 Lowest 0 1 1 0 1 1 Highest
0	PCANH	**CAN Interrupt Priority Level Most Significant bit** PCANH PCANL Priority level 0 0 Lowest 0 1 1 0 1 1 Highest

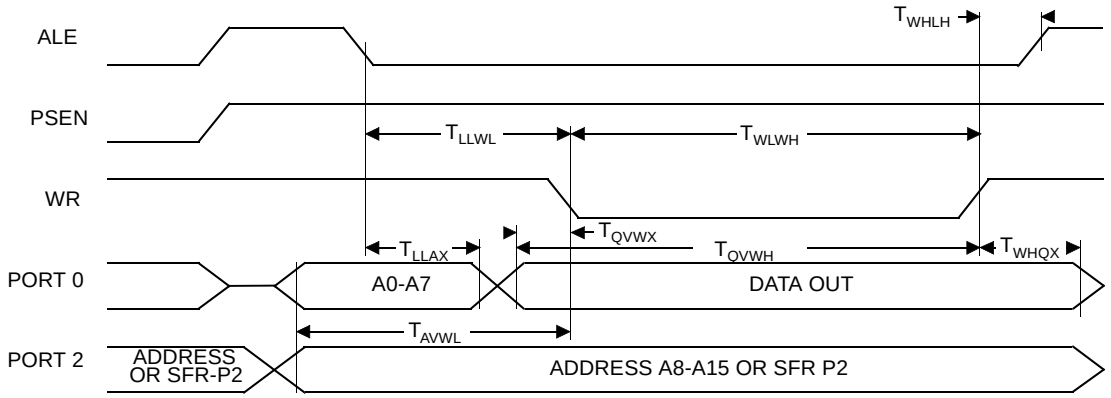
Reset Value = XXXX 0000b

Table 116. DC Parameters in Standard Voltage (Continued)

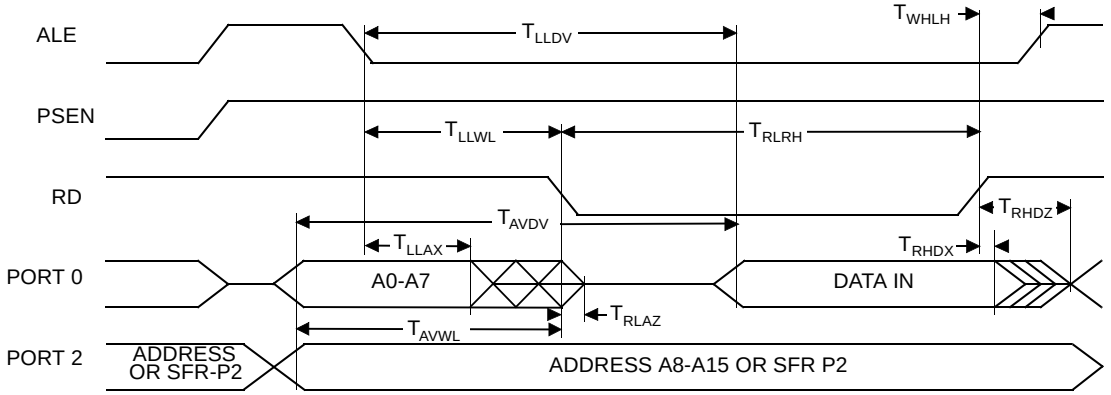
Symbol	Parameter	Min	Typ[5]	Max	Unit	Test Conditions
V_{OH}	Output High Voltage, ports 1, 2, 3, and 4	V_{CC} - 0.3 V_{CC} - 0.7 V_{CC} - 1.5			V V V	I_{OH} = -10 µA I_{OH} = -30 µA I_{OH} = -60 µA V_{CC} = 3V to 5.5V
V_{OH1}	Output High Voltage, port 0, ALE, $\overline{PSEN}$	V_{CC} - 0.3 V_{CC} - 0.7 V_{CC} - 1.5			V V V	I_{OH} = -200 µA I_{OH} = -3.2 mA I_{OH} = -7.0 mA V_{CC} = 5V ± 10%
R_{RST}	RST Pulldown Resistor	20	100	200	kΩ	
I_{IL}	Logical 0 Input Current ports 1, 2, 3 and 4			-50	µA	Vin = 0.45V
I_{LI}	Input Leakage Current			±10	µA	0.45V < Vin < V_{CC}
I_{TL}	Logical 1 to 0 Transition Current, ports 1, 2, 3 and 4			-650	µA	Vin = 2.0V
C_{IO}	Capacitance of I/O Buffer			10	pF	Fc = 1 MHz T_A = 25°C
I_{PD}	Power-down Current Industrial		75	150	µA	3V < V_{CC} < 5.5V[3]
	Power-down Current Automotive		100	350	µA	3V < V_{CC} < 5.5V[3]
I_{CC}	Power Supply Current	I_{CCOP} = 0.4 Frequency (MHz) + 8 I_{CCIDLE} = 0.2 Frequency (MHz) + 8			mA	Vcc = 5.5V[1][2]
$I_{CCWRITE}$	Power Supply Current on flash or EEdata write			0.8 x Frequency (MHz) + 15	mA	V_{CC} = 5.5V

Notes:
1. Operating I_{CC} is measured with all output pins disconnected; XTAL1 driven with T_{CLCH}, T_{CHCL} = 5 ns (see Figure 81.), V_{IL} = V_{SS} + 0.5V,
   V_{IH} = V_{CC} - 0.5V; XTAL2 N.C.; $\overline{EA}$ = RST = Port 0 = V_{CC}. I_{CC} would be slightly higher if a crystal oscillator used (see Figure 78.).
2. Idle I_{CC} is measured with all output pins disconnected; XTAL1 driven with T_{CLCH}, T_{CHCL} = 5 ns, V_{IL} = V_{SS} + 0.5V, V_{IH} = V_{CC} - 0.5V; XTAL2 N.C; Port 0 = V_{CC}; $\overline{EA}$ = RST = V_{SS} (see Figure 79.).
3. Power-down I_{CC} is measured with all output pins disconnected; $\overline{EA}$ = V_{CC}, PORT 0 = V_{CC}; XTAL2 NC.; RST = V_{SS} (see Figure 80.). In addition, the WDT must be inactive and the POF flag must be set.
4. Capacitance loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}s of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1 to 0 transitions during bus operation. In the worst cases (capacitive loading 100pF), the noise pulse on the ALE line may exceed 0.45V with maxi V_{OL} peak 0.6V. A Schmitt Trigger use is not necessary.
5. Typicals are based on a limited number of samples and are not guaranteed. The values listed are at room temperature.
6. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port:
   Port 0: 26 mA
   Ports 1, 2, 3 and 4: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

External Data Memory Write Cycle



External Data Memory Read Cycle



Serial Port Timing – Shift Register Mode

Table 124. Symbol Description (F = 40 MHz)

Symbol	Parameter
T_{XLXL}	Serial port clock cycle time
T_{QVHX}	Output data set-up to clock rising edge
T_{XHQX}	Output data hold after clock rising edge
T_{XHDX}	Input data hold after clock rising edge
T_{XHDV}	Clock rising edge to input data valid

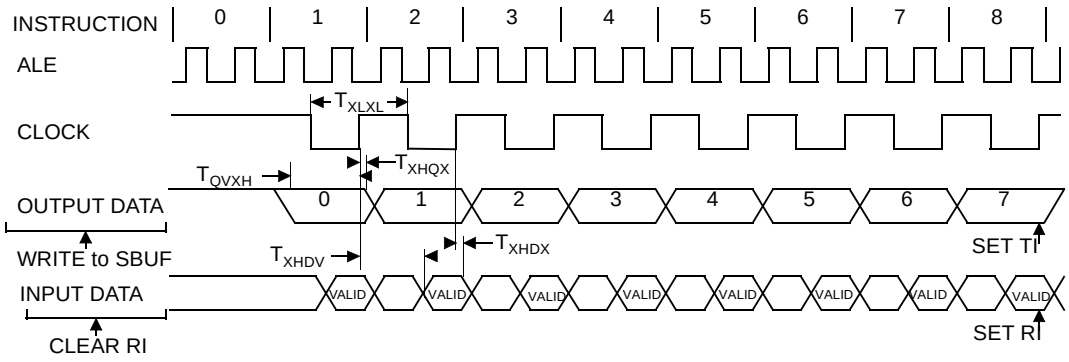
Table 125. AC Parameters for a Fix Clock (F = 40 MHz)

Symbol	Min	Max	Units
T_{XLXL}	300		ns
T_{QVHX}	200		ns
T_{XHQX}	30		ns
T_{XHDX}	0		ns
T_{XHDV}		117	ns

Table 126. AC Parameters for a Variable Clock

Symbol	Type	Standard Clock	X2 Clock	X parameter for -M range	Units
T_{XLXL}	Min	12 T	6 T		ns
T_{QVHX}	Min	10 T - x	5 T - x	50	ns
T_{XHQX}	Min	2 T - x	T - x	20	ns
T_{XHDX}	Min	x	x	0	ns
T_{XHDV}	Max	10 T - x	5 T- x	133	ns

Shift Register Timing Waveforms



External Clock Drive Characteristics (XTAL1)

Table 127. AC Parameters

Symbol	Parameter	Min	Max	Units
T_{CLCL}	Oscillator Period	25		ns
T_{CHCX}	High Time	5		ns
T_{CLCX}	Low Time	5		ns
T_{CLCH}	Rise Time		5	ns
T_{CHCL}	Fall Time		5	ns
T_{CHCX}/T_{CLCX}	Cyclic ratio in X2 mode	40	60	%

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.