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	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at89c51cc02ca-tisum

Registers

Table 16. PCON Register
PCON (S:87h)
Power Control Register

7	6	5	4	3	2	1	0
SMOD1	**SMOD0**	**-**	**POF**	**GF1**	**GF0**	**PD**	**IDL**

Bit Number	Bit Mnemonic	Description
7	SMOD1	**Serial port Mode bit 1** Set to select double baud rate in mode 1, 2 or 3.
6	SMOD0	**Serial port Mode bit 0** Clear to select SM0 bit in SCON register. Set to select FE bit in SCON register.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	POF	**Power-off Flag** Clear to recognize next reset type. Set by hardware when V_{CC} rises from 0 to its nominal voltage. Can also be set by software.
3	GF1	**General purpose Flag** Cleared by user for general purpose usage. Set by user for general purpose usage.
2	GF0	**General purpose Flag** Cleared by user for general purpose usage. Set by user for general purpose usage.
1	PD	**Power-down Mode bit** Cleared by hardware when reset occurs. Set to enter power-down mode.
0	IDL	**Idle Mode bit** Clear by hardware when interrupt or reset occurs. Set to enter idle mode.

Reset Value = 00X1 0000b
Not bit addressable

Registers

Table 18. PSW Register
PSW (S:D0h)
Program Status Word Register

7	6	5	4	3	2	1	0
CY	**AC**	**F0**	**RS1**	**RS0**	**OV**	**F1**	**P**

Bit Number	Bit Mnemonic	Description
7	CY	**Carry Flag** Carry out from bit 1 of ALU operands.
6	AC	**Auxiliary Carry Flag** Carry out from bit 1 of addition operands.
5	F0	**User Definable Flag 0**
4 - 3	RS1:0	**Register Bank Select bits** Refer to Table 17 for bits description.
2	OV	**Overflow Flag** Overflow set by arithmetic operations.
1	F1	**User Definable Flag 1**
0	P	**Parity bit** Set when ACC contains an odd number of 1's. Cleared when ACC contains an even number of 1's.

Reset Value = 0000 0000b

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
```

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.

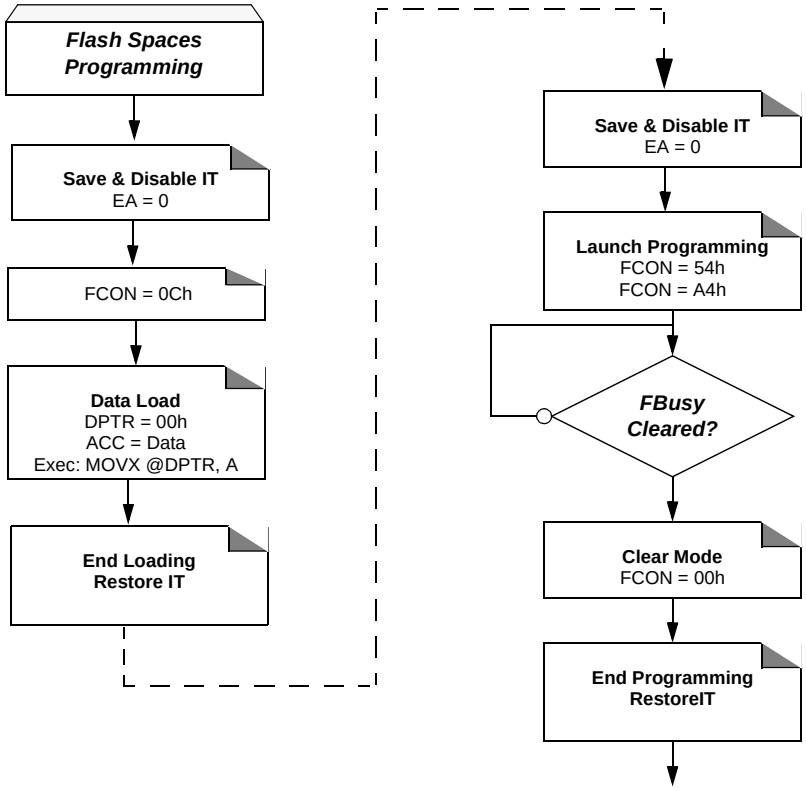
The FM1 memory can be program only by parallel programming.

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 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.

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

Table 34. SADEN Register
SADEN (S:B9h)
Slave Address Mask Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7 - 0		**Mask Data for Slave Individual Address**

Reset Value = 0000 0000b
Not bit addressable

Table 35. SADDR Register
SADDR (S:A9h)
Slave Address Register

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7 - 0		**Slave Individual Address**

Reset Value = 0000 0000b
Not bit addressable

Table 36. SBUF Register
SBUF (S:99h)
Serial Data Buffer

7	6	5	4	3	2	1	0

Bit Number	Bit Mnemonic	Description
7 - 0		**Data sent/received by Serial I/O Port**

Reset Value = 0000 0000b
Not bit addressable

Table 37. PCON Register

PCON (S:87h)

Power Control Register

7	6	5	4	3	2	1	0
SMOD1	**SMOD0**	**-**	**POF**	**GF1**	**GF0**	**PD**	**IDL**

Bit Number	Bit Mnemonic	Description
7	SMOD1	**Serial port Mode bit 1** Set to select double baud rate in mode 1, 2 or 3.
6	SMOD0	**Serial port Mode bit 0** Clear to select SM0 bit in SCON register. Set to select FE bit in SCON register.
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	POF	**Power-off Flag** Clear to recognize next reset type. Set by hardware when V_{CC} rises from 0 to its nominal voltage. Can also be set by software.
3	GF1	**General purpose Flag** Cleared by user for general purpose usage. Set by user for general purpose usage.
2	GF0	**General purpose Flag** Cleared by user for general purpose usage. Set by user for general purpose usage.
1	PD	**Power-down Mode bit** Cleared by hardware when reset occurs. Set to enter power-down mode.
0	IDL	**Idle Mode bit** Clear by hardware when interrupt or reset occurs. Set to enter idle mode.

Reset Value = 00X1 0000b

Not bit addressable

Table 39. TMOD Register
TMOD (S:89h)
Timer/Counter Mode Control Register

7	6	5	4	3	2	1	0
GATE1	**C/T1#**	**M11**	**M01**	**GATE0**	**C/T0#**	**M10**	**M00**

Bit Number	Bit Mnemonic	Description
7	GATE1	**Timer 1 Gating Control bit** Clear to enable Timer 1 whenever TR1 bit is set. Set to enable Timer 1 only while INT1# pin is high and TR1 bit is set.
6	C/T1#	**Timer 1 Counter/Timer Select bit** Clear for Timer operation: Timer 1 counts the divided-down system clock. Set for Counter operation: Timer 1 counts negative transitions on external pin T1.
5	M11	**Timer 1 Mode Select bits** M11 M01 Operating mode 0 0 Mode 0: 8-bit Timer/Counter (TH1) with 5bit prescaler (TL1). 0 1 Mode 1: 16-bit Timer/Counter. 1 1 Mode 3: Timer 1 halted. Retains count. 1 0 Mode 2: 8-bit auto-reload Timer/Counter (TL1).(1)
4	M01	
3	GATE0	**Timer 0 Gating Control bit** Clear to enable Timer 0 whenever TR0 bit is set. Set to enable Timer/Counter 0 only while INT0# pin is high and TR0 bit is set.
2	C/T0#	**Timer 0 Counter/Timer Select bit** Clear for Timer operation: Timer 0 counts the divided-down system clock. Set for Counter operation: Timer 0 counts negative transitions on external pin T0.
1	M10	**Timer 0 Mode Select bit** M10 M00 Operating mode 0 0 Mode 0: 8-bit Timer/Counter (TH0) with 5bit prescaler (TL0). 0 1 Mode 1: 16-bit Timer/Counter. 1 0 Mode 2: 8-bit auto-reload Timer/Counter (TL0).(2) 1 1 Mode 3: TL0 is an 8-bit Timer/Counter. TH0 is an 8-bit Timer using Timer 1's TR0 and TF0 bits.
0	M00	

Reset Value = 0000 0000b

Notes: 1. Reloaded from TH1 at overflow.
2. Reloaded from TH0 at overflow.

Table 40. 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

Programmable Clock-Output

In clock-out mode, Timer 2 operates as a 50%-duty-cycle, programmable clock generator (Figure 30). The input clock increments TL2 at frequency $f_{OSC}/2$. The timer repeatedly counts to overflow from a loaded value. At overflow, the contents of RCAP2H and RCAP2L registers are loaded into TH2 and TL2. In this mode, Timer 2 overflows do not generate interrupts. The formula gives the clock-out frequency depending on the system oscillator frequency and the value in the RCAP2H and RCAP2L registers:

$$Clock-OutFrequency = \frac{FT2clock}{4 \times (65536 - RCAP2H/RCAP2L)}$$

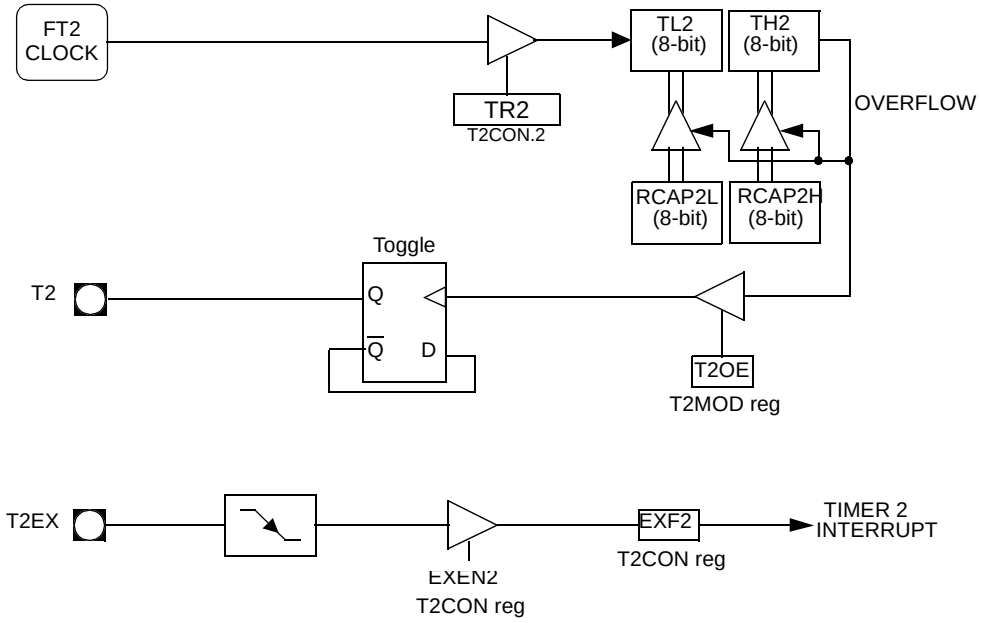
For a 16 MHz system clock in x1 mode, Timer 2 has a programmable frequency range of 61 Hz ($f_{OSC}/2^{16}$) to 4 MHz ($f_{OSC}/4$). The generated clock signal is brought out to T2 pin (P1.0).

Timer 2 is programmed for the clock-out mode as follows:

- Set T2OE bit in T2MOD register.
- Clear C/$\overline{T2}$ bit in T2CON register.
- Determine the 16-bit reload value from the formula and enter it in RCAP2H/RCAP2L registers.
- Enter a 16-bit initial value in timer registers TH2/TL2. It can be the same as the reload value or different depending on the application.
- To start the timer, set TR2 run control bit in T2CON register.

It is possible to use Timer 2 as a baud rate generator and a clock generator simultaneously. For this configuration, the baud rates and clock frequencies are not independent since both functions use the values in the RCAP2H and RCAP2L registers.

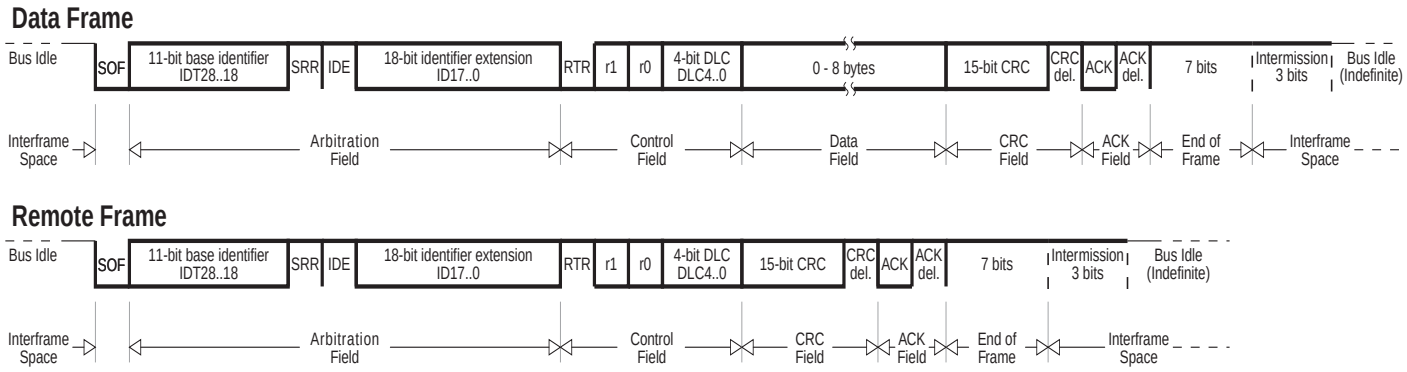
Figure 30. Clock-Out Mode



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 33. 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

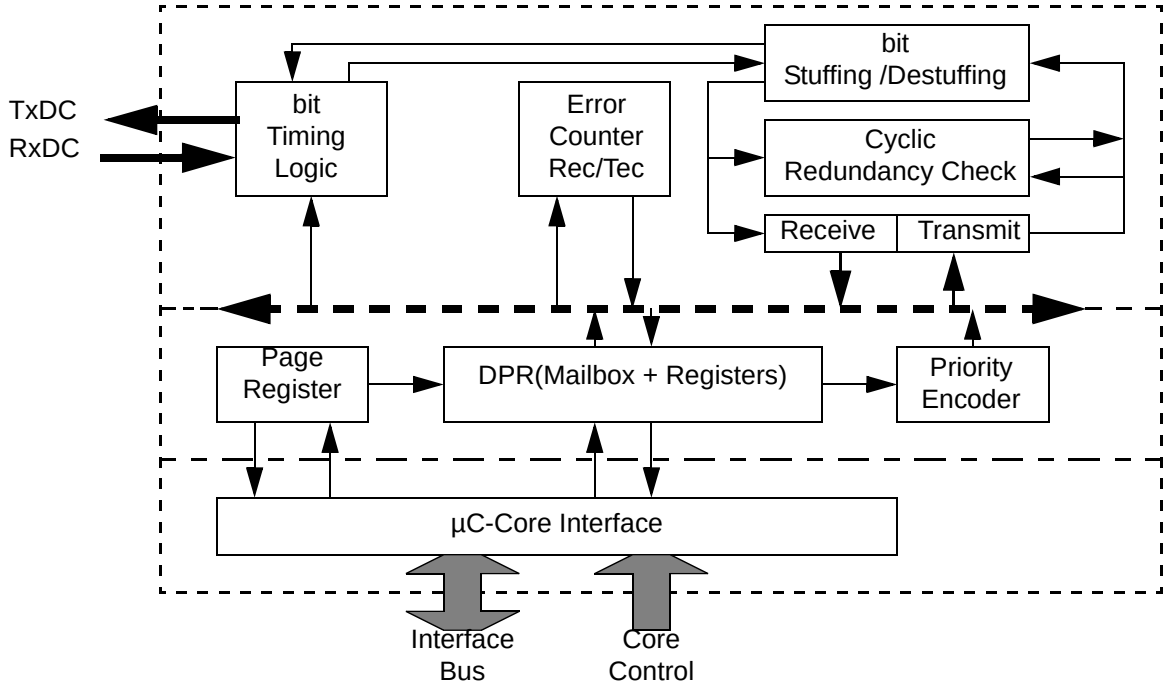
Bit Timing

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.

Figure 36. CAN Controller Block Diagram

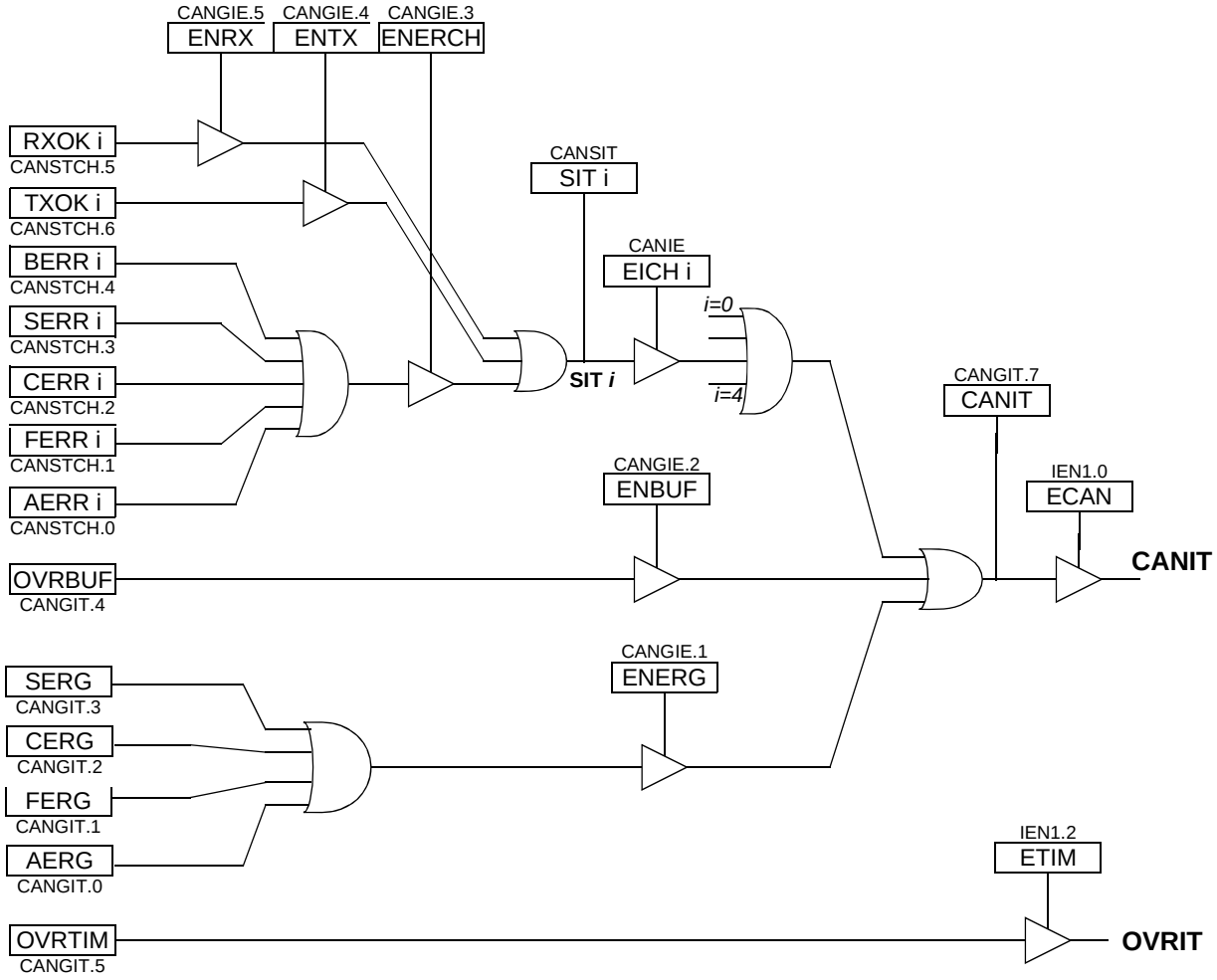


CAN Controller Mailbox and Registers Organization

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.

Figure 39. CAN Controller Interrupt Structure



To enable a transmission interrupt:

- Enable General CAN IT in the interrupt system register
- Enable interrupt by message object, EICHi
- Enable transmission interrupt, ENTX

To enable a reception interrupt:

- Enable General CAN IT in the interrupt system register
- Enable interrupt by message object, EICHi
- Enable reception interrupt, ENRX

To enable an interrupt on message object error:

- Enable General CAN IT in the interrupt system register
- Enable interrupt by message object, EICHi
- Enable interrupt on error, ENERCH

To enable an interrupt on general error:

- Enable General CAN IT in the interrupt system register
- Enable interrupt on error, ENERG

Table 68. CANPAGE Register

CANPAGE (S:B1h) – CAN Message Object Page Register

7	6	5	4	3	2	1	0
-	-	CHNB 1	CHNB 0	AINC	INDX2	INDX1	INDX0

Bit Number	Bit Mnemonic	Description
7 - 6	-	**Reserved** The values read from these bits are indeterminate. Do not set these bits.
5 - 4	CHNB3:0	**Selection of Message Object Number** The available numbers are: 0 to 3(See Figure 37).
3	AINC	**Auto Increment of the Index (Active Low)** 0 - auto-increment of the index (default value). 1 - non-auto-increment of the index.
2 - 0	INDX2:0	**Index** Byte location of the data field for the defined message object (See Figure 37).

Reset Value = xx00 0000b

Table 69. CANCONCH Register

CANCONCH (S:B3h) – CAN Message Object Control and DLC Register

7	6	5	4	3	2	1	0
CONCH 1	CONCH 0	RPLV	IDE	DLC 3	DLC 2	DLC 1	DLC 0

Bit Number	Bit Mnemonic	Description
7 - 6	CONCH1:0	**Configuration of Message Object** **CONCH1** **CONCH0** 0 0: disable 0 1: Launch transmission 1 0: Enable Reception 1 1: Enable Reception Buffer NOTE: The user must re-write the configuration to enable the corresponding bit in the CANEN1:2 registers.
5	RPLV	**Reply valid** Used in the automatic reply mode after receiving a remote frame 0 - reply not ready. 1 - reply ready & valid.
4	IDE	**Identifier Extension** 0 - CAN standard rev 2.0 A (ident = 11 bits). 1 - CAN standard rev 2.0 B (ident = 29 bits).
3 - 0	DLC3:0	**Data Length Code** Number of Bytes in the data field of the message. The range of DLC is from 0 up to 8. This value is updated when a frame is received (data or remote frame). If the expected DLC differs from the incoming DLC, a warning appears in the CANSTCH register.

No default value after reset

Table 84. CANIDM2 Register for V2.0 Part B
CANIDM2 for V2.0 Part B (S:C5h)
CAN Identifier Mask Registers 2

7	6	5	4	3	2	1	0
IDMSK 20	IDMSK 19	IDMSK 18	IDMSK 17	IDMSK 16	IDMSK 15	IDMSK 14	IDMSK 13

Bit Number	Bit Mnemonic	Description
7 - 0	IDMSK20:13	**IDentifier Mask Value**[1] 0 - comparison true forced. 1 - bit comparison enabled. See Figure 43.

Note: 1. The ID Mask is only used for reception.

No default value after reset.

Table 85. CANIDM3 Register for V2.0 Part B
CANIDM3 for V2.0 Part B (S:C6h)
CAN Identifier Mask Registers 3

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

Bit Number	Bit Mnemonic	Description
7 - 0	IDMSK12:5	**IDentifier Mask Value** 0 - comparison true forced. 1 - bit comparison enabled. See Figure 43.

Note: The ID Mask is only used for reception.

No default value after reset.

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

Table 113. IPH1 Register
IPH1 (S:F7h)
Interrupt high priority Register 1

7	6	5	4	3	2	1	0
-	-	-	-		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	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
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 X000b

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.

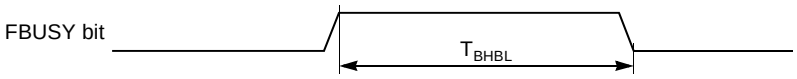
Flash/EEPROM Memory

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

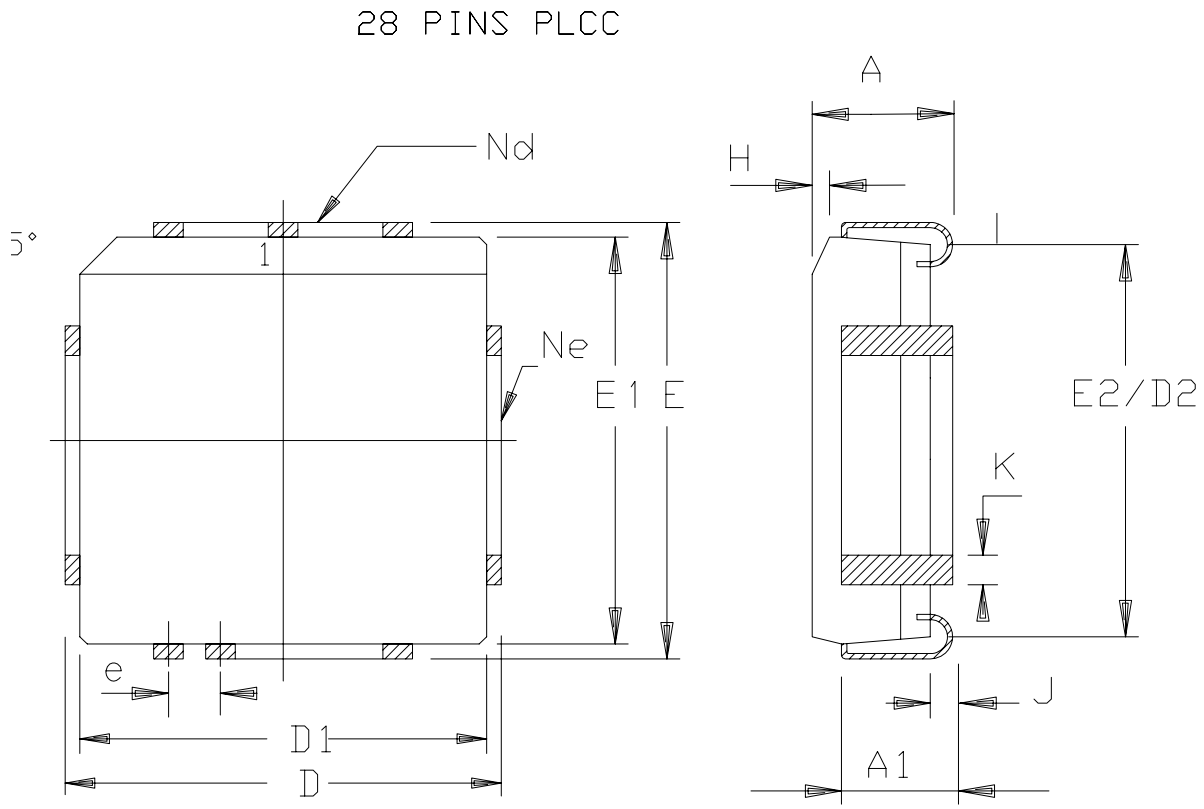


A/D Converter

Table 121. AC Parameters for A/D Conversion

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

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
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			

Table of Contents