

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFI

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	96KB (48K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.25K × 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf4685-i-pt

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



PIC18F2682/2685 (28-PIN) BLOCK DIAGRAM FIGURE 1-1:

2: OSC1/CLKI and OSC2/CLKO are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to Section 2.0 "Oscillator Configurations" for additional information.

Din Nomo	Pi	n Num	ber	Pin	Buffer	Description
Pin Name	PDIP	QFN	TQFP	Туре	Туре	Description
						PORTA is a bidirectional I/O port.
RA0/AN0/CVREF	2	19	19			
RA0				I/O	TTL	Digital I/O.
AN0				I	Analog	Analog input 0.
CVREF				0	Analog	Analog comparator reference output.
RA1/AN1	3	20	20			
RA1				I/O	TTL	Digital I/O.
AN1				I	Analog	Analog input 1.
RA2/AN2/VREF-	4	21	21			
RA2				I/O	TTL	Digital I/O.
AN2				I	Analog	Analog input 2.
VREF-				I	Analog	A/D reference voltage (low) input.
RA3/AN3/VREF+	5	22	22			
RA3				I/O	TTL	Digital I/O.
AN3				I	Analog	Analog input 3.
VREF+				I	Analog	A/D reference voltage (high) input.
RA4/T0CKI	6	23	23			
RA4				I/O	TTL	Digital I/O.
TOCKI				I	ST	Timer0 external clock input.
RA5/AN4/SS/HLVDIN	7	24	24			
RA5				I/O	TTL	Digital I/O.
AN4				I	Analog	Analog input 4.
SS				I	TTL	SPI slave select input.
HLVDIN				I	Analog	High/Low-Voltage Detect input.
RA6						See the OSC2/CLKO/RA6 pin.
RA7						See the OSC1/CLKI/RA7 pin.
Legend: TTL = TTL	compat	ible inpu	ut			CMOS = CMOS compatible input or output
ST = Schr	nitt Trig	ger inpu	ut with Cl	MOS le	vels	I = Input
O = Outp	out					P = Power

TABLE 1-3:	PIC18F4682/4685 PINOUT I/O DESCRIPTIONS (CONTINUED)
------------	---

= Output 0

= Power

5.1.2.4 Stack Full and Underflow Resets

Device Resets on stack overflow and stack underflow conditions are enabled by setting the STVREN bit in Configuration Register 4L. When STVREN is set, a full or underflow condition will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. When STVREN is cleared, a full or underflow condition will set the appropriate STKFUL or STKUNF bit, but not cause a device Reset. The STKFUL or STKUNF bit is cleared by user software or a Power-on Reset.

5.1.3 FAST REGISTER STACK

A Fast Register Stack is provided for the STATUS, WREG and BSR registers to provide a "fast return" option for interrupts. Each stack is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources will push values into the stack registers. The values in the registers are then loaded back into their associated registers, if the RETFIE, FAST instruction is used to return from the interrupt.

If both low and high priority interrupts are enabled, the stack registers cannot be used reliably to return from low priority interrupts. If a high priority interrupt occurs while servicing a low priority interrupt, the stack register values stored by the low priority interrupt will be overwritten. In these cases, users must save the key registers in software during a low priority interrupt.

If interrupt priority is not used, all interrupts may use the Fast Register Stack for returns from interrupt. If no interrupts are used, the Fast Register Stack can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call. To use the Fast Register Stack for a subroutine call, a CALL label, FAST instruction must be executed to save the STATUS, WREG and BSR registers to the Fast Register Stack. A RETURN, FAST instruction is then executed to restore these registers from the Fast Register Stack.

Example 5-1 shows a source code example that uses the Fast Register Stack during a subroutine call and return.

EXAMPLE 5-1: FAST REGISTER STACK CODE EXAMPLE

CALL	SUB1, FA	AST	;STATUS, WREG, BSR ;SAVED IN FAST REGISTER ;STACK
SUB1	•		
	• RETURN,	FAST	;RESTORE VALUES SAVED ;IN FAST REGISTER STACK

5.1.4 LOOK-UP TABLES IN PROGRAM MEMORY

There may be programming situations that require the creation of data structures, or look-up tables, in program memory. For PIC18 devices, look-up tables can be implemented in two ways:

- Computed GOTO
- Table Reads

5.1.4.1 Computed GOTO

A computed GOTO is accomplished by adding an offset to the program counter. An example is shown in Example 5-2.

A look-up table can be formed with an ADDWF PCL instruction and a group of RETLW nn instructions. The W register is loaded with an offset into the table before executing a CALL to that table. The first instruction of the called routine is the ADDWF PCL instruction. The next instruction executed will be one of the RETLW nn instructions, that returns the value 'nn' to the calling function.

The offset value (in WREG) specifies the number of bytes that the program counter should advance and should be multiples of 2 (LSb = 0).

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

EXAMPLE 5-2: COMPUTED GOTO USING AN OFFSET VALUE

	MOVF	OFFSET,	W
	CALL	TABLE	
ORG	nn00h		
TABLE	ADDWF	PCL	
	RETLW	nnh	
	RETLW	nnh	
	RETLW	nnh	
	•		
	•		

5.1.4.2 Table Reads and Table Writes

A better method of storing data in program memory allows two bytes of data to be stored in each instruction location.

Look-up table data may be stored two bytes per program word by using table reads and writes. The Table Pointer (TBLPTR) register specifies the byte address and the Table Latch (TABLAT) register contains the data that is read from or written to program memory. Data is transferred to or from program memory one byte at a time.

Table read and table write operations are discussed further in Section 6.1 "Table Reads and Table Writes".

-	-			,					/	
File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
B0DLC ⁽⁸⁾ Transmit mode	-	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	-x xxxx	58, 304
B0EIDL ⁽⁸⁾	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	XXXX XXXX	61, 301
B0EIDH ⁽⁸⁾	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	XXXX XXXX	61, 301
B0SIDL ⁽⁸⁾ Receive mode	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	XXXX X-XX	58, 300
B0SIDL ⁽⁸⁾ Transmit mode	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xxx- x-xx	58, 300
B0SIDH(8)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	XXXX XXXX	61, 299
B0CON ⁽⁸⁾ Receive mode	RXFUL	RXM1	RXRTRRO	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0	0000 0000	60, 298
B0CON ⁽⁸⁾ Transmit mode	TXBIF	TXABT	TXLARB	TXERR	TXREQ	RTREN	TXPRI1	TXPRI0	0000 0000	60, 298
TXBIE		—	—	TXB2IE	TXB1IE	TXB0IE	_		0 00	61, 321
BIE0	B5IE	B4IE	B3IE	B2IE	B1IE	B0IE	RXB1IE	RXB0IE	0000 0000	61, 321
BSEL0	B5TXEN	B4TXEN	B3TXEN	B2TXEN	B1TXEN	B0TXEN	—		0000 00	61, 304
MSEL3	FIL15_1	FIL15_0	FIL14_1	FIL14_0	FIL13_1	FIL13_0	FIL12_1	FIL12_0	0000 0000	61, 313
MSEL2	FIL11_1	FIL11_0	FIL10_1	FIL10_0	FIL9_1	FIL9_0	FIL8_1	FIL8_0	0000 0000	61, 312
MSEL1	FIL7_1	FIL7_0	FIL6_1	FIL6_0	FIL5_1	FIL5_0	FIL4_1	FIL4_0	0000 0101	61, 311
MSEL0	FIL3_1	FIL3_0	FIL2_1	FIL2_0	FIL1_1	FIL1_0	FIL0_1	FIL0_0	0101 0000	61, 310
RXFBCON7	F15BP_3	F15BP_2	F15BP_1	F15BP_0	F14BP_3	F14BP_2	F14BP_1	F14BP_0	0000 0000	61, 309
RXFBCON6	F13BP_3	F13BP_2	F13BP_1	F13BP_0	F12BP_3	F12BP_2	F12BP_1	F12BP_0	0000 0000	61, 309
RXFBCON5	F11BP_3	F11BP_2	F11BP_1	F11BP_0	F10BP_3	F10BP_2	F10BP_1	F10BP_0	0000 0000	61, 309
RXFBCON4	F9BP_3	F9BP_2	F9BP_1	F9BP_0	F8BP_3	F8BP_2	F8BP_1	F8BP_0	0000 0000	61, 309
RXFBCON3	F7BP_3	F7BP_2	F7BP_1	F7BP_0	F6BP_3	F6BP_2	F6BP_1	F6BP_0	0000 0000	61, 309
RXFBCON2	F5BP_3	F5BP_2	F5BP_1	F5BP_0	F4BP_3	F4BP_2	F4BP_1	F4BP_0	0001 0001	61, 309
RXFBCON1	F3BP_3	F3BP_2	F3BP_1	F3BP_0	F2BP_3	F2BP_2	F2BP_1	F2BP_0	0001 0001	61, 309
RXFBCON0	F1BP_3	F1BP_2	F1BP_1	F1BP_0	F0BP_3	F0BP_2	F0BP_1	F0BP_0	0000 0000	61, 309
SDFLC	_	_	_	FLC4	FLC3	FLC2	FLC1	FLC0	0 0000	61, 308
RXFCON1	RXF15EN	RXF14EN	RXF13EN	RXF12EN	RXF11EN	RXF10EN	RXF9EN	RXF8EN	0000 0000	61, 308
RXFCON0	RXF7EN	RXF6EN	RXF5EN	RXF4EN	RXF3EN	RXF2EN	RXF1EN	RXF0EN	0000 0000	61, 308
RXF15EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	XXXX XXXX	61, 306
RXF15EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	XXXX XXXX	61, 306
RXF15SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	61, 305
RXF15SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	XXXX XXXX	61, 305
RXF14EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	XXXX XXXX	61, 306
RXF14EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	XXXX XXXX	61, 306
RXF14SIDL	SID2	SID1	SID0	—	EXIDEN		EID17	EID16	xxx- x-xx	61, 305
RXF14SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	XXXX XXXX	61, 305
RXF13EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	XXXX XXXX	62, 306
RXF13EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	XXXX XXXX	62, 306
RXF13SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	62, 305
RXF13SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	XXXX XXXX	62, 305

TABLE 5-2:REGISTER FILE SUMMARY (PIC18F2682/2685/4682/4685) (CONTINUED)

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition. Shaded cells are unimplemented, read as '0'.

Note 1: Bit 21 of the PC is only available in Test mode and Serial Programming modes.

2: The SBOREN bit is only available when CONFIG2L<1:0> = 01; otherwise, it is disabled and reads as '0'. See Section 4.4 "Brown-out Reset (BOR)".

3: These registers and/or bits are not implemented on PIC18F2682/2685 devices and are read as '0'. Reset values are shown for PIC18F4682/4685 devices; individual unimplemented bits should be interpreted as '---'.

4: The PLLEN bit is only available in specific oscillator configurations; otherwise, it is disabled and reads as '0'. See Section 2.6.4 "PLL in INTOSC Modes".

5: The RE3 bit is only available when Master Clear Reset is disabled (CONFIG3H<7> = 0); otherwise, RE3 reads as '0'. This bit is read-only.

6: RA6/RA7 and their associated latch and direction bits are individually configured as port pins based on various primary oscillator modes. When disabled, these bits read as '0'.

7: CAN bits have multiple functions depending on the selected mode of the CAN module.

8: This register reads all '0's until the ECAN[™] technology is set up in Mode 1 or Mode 2.

9: These registers and/or bits are available on PIC18F4682/4685 devices only.

7.0 DATA EEPROM MEMORY

The data EEPROM is a nonvolatile memory array, separate from the data RAM and program memory, that is used for long-term storage of program data. It is not directly mapped in either the register file or program memory space but is indirectly addressed through the Special Function Registers (SFRs). The EEPROM is readable and writable during normal operation over the entire VDD range.

Five SFRs are used to read and write to the data EEPROM as well as the program memory. They are:

- EECON1
- EECON2
- EEDATA
- EEADR
- EEADRH

The data EEPROM allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and the EEADRH:EEADR register pair holds the address of the EEPROM location being accessed.

The EEPROM data memory is rated for high erase/write cycle endurance. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer; it will vary with voltage and temperature, as well as from chip to chip. Please refer to parameter D122 (Table 27-1 in **Section 27.0 "Electrical Characteristics**") for exact limits.

7.1 EEADR and EEADRH Registers

The EEADRH:EEADR register pair is used to address the data EEPROM for read and write operations. EEADRH holds the two Most Significant bits of the address; the upper 6 bits are ignored. The 10-bit range of the pair can address a memory range of 1024 bytes (00h to 3FFh).

7.2 EECON1 and EECON2 Registers

Access to the data EEPROM is controlled by two registers: EECON1 and EECON2. These are the same registers which control access to the program memory and are used in a similar manner for the data EEPROM.

The EECON1 register (Register 7-1) is the control register for data and program memory access. Control bit EEPGD determines if the access will be to program or data EEPROM memory. When clear, operations will access the data EEPROM memory. When set, program memory is accessed.

Control bit CFGS determines if the access will be to the Configuration registers or to program memory/data EEPROM memory. When set, subsequent operations access Configuration registers. When CFGS is clear, the EEPGD bit selects either program Flash or data EEPROM memory.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set in hardware when the WREN bit is set and cleared when the internal programming timer expires and the write operation is complete.

Note:	During normal operation, the WRERR is
	read as '1'. This can indicate that a write
	operation was prematurely terminated by
	a Reset, or a write operation was
	attempted improperly.

The WR control bit initiates write operations. The bit cannot be cleared, only set, in software; it is cleared in hardware at the completion of the write operation.

Note:	The EEIF interrupt flag bit (PIR2<4>) is set						
	when the write is complete. It must be						
	cleared in software.						

Control bits, RD and WR, start read and erase/write operations, respectively. These bits are set by firmware and cleared by hardware at the completion of the operation.

The RD bit cannot be set when accessing program memory (EEPGD = 1). Program memory is read using table read instructions. See **Section 6.1 "Table Reads and Table Writes"** regarding table reads.

The EECON2 register is not a physical register. It is used exclusively in the memory write and erase sequences. Reading EECON2 will read all '0's.

14.1 **Timer3 Operation**

Timer3 can operate in one of three modes:

- Timer
- · Synchronous Counter
- Asynchronous Counter

The operating mode is determined by the clock select bit, TMR3CS (T3CON<1>). When TMR3CS is cleared (= 0), Timer3 increments on every internal instruction cycle (Fosc/4). When the bit is set, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator if enabled.

As with Timer1, the RC1/T1OSI and RC0/T1OSO/ T13CKI pins become inputs when the Timer1 oscillator is enabled. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.



FIGURE 14-2: TIMER3 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)



17.4.9 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I²C logic module is in the Idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded with the contents of SSPADD<5:0> and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. Following this, the RSEN bit (SSPCON2<1>) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

- Note 1: If RSEN is programmed while any other event is in progress, it will not take effect.
 - **2:** A bus collision during the Repeated Start condition occurs if:
 - SDA is sampled low when SCL goes from low-to-high.
 - SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode, or the default first address in 10-bit mode. After the first eight bits are transmitted and an \overline{ACK} is received, the user may then transmit an additional eight bits of address (10-bit mode) or eight bits of data (7-bit mode).

17.4.9.1 WCOL Status Flag

If the user writes the SSPBUF when a Repeated Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

Note: Because queueing of events is not allowed, writing of the lower 5 bits of SSPCON2 is disabled until the Repeated Start condition is complete.

FIGURE 17-20: REPEATED START CONDITION WAVEFORM





FIGURE 17-28: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION

17.4.17.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

- a) A low level is sampled on SDA when SCL goes from low level to high level.
- b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user deasserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0> and counts down to 0. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 17-29). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high to low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high to low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 17-30.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 17-29: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

FIGURE 17-30: BUS COLLISION DURING A REPEATED START CONDITION (CASE 2)

EXAMPLE 23-2: WIN AND ICODE BITS USAGE IN INTERRUPT SERVICE ROUTINE TO ACCESS TX/RX BUFFERS (CONTINUED)

ErrorInter	rupt	
BCF	PIR3, ERRIF	; Clear the interrupt flag
		; Handle error.
RETFIE		
TXB2Interr	upt	
BCF	PIR3, TXB2IF	; Clear the interrupt flag
GOTO	AccessBuffer	
TXB1Interr	upt	
BCF	PIR3, TXB1IF	; Clear the interrupt flag
GOTO	AccessBuffer	
TXB0Interr	upt	
BCF	PIR3, TXBOIF	; Clear the interrupt flag
GOTO	AccessBuffer	
RXB1Interr	upt	
BCF	PIR3, RXB1IF	; Clear the interrupt flag
GOTO	Accessbuffer	
RXB0Interr	upt	
BCF	PIR3, RXB0IF	; Clear the interrupt flag
GOTO	AccessBuffer	
AccessBuff	er	; This is either TX or RX interrupt
; Copy	CANSTAT.ICODE bits to CANC	CON.WIN bits
MOVF	TempCANCON, W	; Clear CANCON.WIN bits before copying
		; new ones.
ANDLW	B'11110001'	; Use previously saved CANCON value to
		; make sure same value.
MOVWF	TempCANCON	; Copy masked value back to TempCANCON
MOVF	TempCANSTAT, W	; Retrieve ICODE bits
ANDLW	B'00001110'	; Use previously saved CANSTAT value
		; to make sure same value.
IORWF	TempCANCON	; Copy ICODE bits to WIN bits.
MOVFF	TempCANCON, CANCON	; Copy the result to actual CANCON
; Acce	ss current buffer…	
; User	code	
; Rest	ore CANCON.WIN bits	
MOVF	CANCON, W	; Preserve current non WIN bits
ANDLW	B'11110001'	
IORWF	TempCANCON	; Restore original WIN bits
; Do n	ot need to restore CANSTAT	- it is read-only register.
; Retu	rn from interrupt or check	for another module interrupt source

23.2.3.1 Programmable TX/RX and Auto-RTR Buffers

The ECAN module contains 6 message buffers that can be programmed as transmit or receive buffers. Any of these buffers can also be programmed to automatically handle RTR messages.

Note: These registers are not used in Mode 0.

REGISTER 23-22: BnCON: TX/RX BUFFER n CONTROL REGISTERS IN RECEIVE MODE $[0 \le n \le 5, TXnEN (BSEL0 \le n) = 0]^{(1)}$

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
RXFUL ⁽²⁾	RXM1	RXRTRRO	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7	RXFUL: Receive Full Status bit ⁽²⁾
	1 = Receive buffer contains a received message
	0 = Receive buffer is open to receive a new message
bit 6	RXM1: Receive Buffer Mode bit
	 1 = Receive all messages including partial and invalid (acceptance filters are ignored) 0 = Receive all valid messages as per acceptance filters
bit 5	RXRTRRO: Read-Only Remote Transmission Request for Received Message bit
	1 = Received message is a remote transmission request
	0 = Received message is not a remote transmission request
bit 4-0	FILHIT4:FILHIT0: Filter Hit bits
	These bits indicate which acceptance filter enabled the last message reception into this buffer.
	01111 = Acceptance Filter 15 (RXF15)
	01110 = Acceptance Filter 14 (RXF14)
	00001 = Acceptance Filter 1 (RXF1)
	00000 = Acceptance Filter U (RXFU)

- **Note 1:** These registers are available in Mode 1 and 2 only.
 - 2: This bit is set by the CAN module upon receiving a message and must be cleared by software after the buffer is read. As long as RXFUL is set, no new message will be loaded and the buffer will be considered full.

TABLE 23-1: CAN CONTROLLER REGISTER MAP

Address ⁽¹⁾	Name	Address	Name	Address	Name	Address	Name
F7Fh	SPBRGH ⁽³⁾	F5Fh	CANCON_RO0	F3Fh	CANCON_RO2	F1Fh	RXM1EIDL
F7Eh	BAUDCON ⁽³⁾	F5Eh	CANSTAT_RO0	F3Eh	CANSTAT_RO2	F1Eh	RXM1EIDH
F7Dh	(4)	F5Dh	RXB1D7	F3Dh	TXB1D7	F1Dh	RXM1SIDL
F7Ch	(4)	F5Ch	RXB1D6	F3Ch	TXB1D6	F1Ch	RXM1SIDH
F7Bh	(4)	F5Bh	RXB1D5	F3Bh	TXB1D5	F1Bh	RXM0EIDL
F7Ah	(4)	F5Ah	RXB1D4	F3Ah	TXB1D4	F1Ah	RXM0EIDH
F79h	ECCP1DEL ⁽³⁾	F59h	RXB1D3	F39h	TXB1D3	F19h	RXM0SIDL
F78h	(4)	F58h	RXB1D2	F38h	TXB1D2	F18h	RXM0SIDH
F77h	ECANCON	F57h	RXB1D1	F37h	TXB1D1	F17h	RXF5EIDL
F76h	TXERRCNT	F56h	RXB1D0	F36h	TXB1D0	F16h	RXF5EIDH
F75h	RXERRCNT	F55h	RXB1DLC	F35h	TXB1DLC	F15h	RXF5SIDL
F74h	COMSTAT	F54h	RXB1EIDL	F34h	TXB1EIDL	F14h	RXF5SIDH
F73h	CIOCON	F53h	RXB1EIDH	F33h	TXB1EIDH	F13h	RXF4EIDL
F72h	BRGCON3	F52h	RXB1SIDL	F32h	TXB1SIDL	F12h	RXF4EIDH
F71h	BRGCON2	F51h	RXB1SIDH	F31h	TXB1SIDH	F11h	RXF4SIDL
F70h	BRGCON1	F50h	RXB1CON	F30h	TXB1CON	F10h	RXF4SIDH
F6Fh	CANCON	F4Fh	CANCON_RO1 ⁽²⁾	F2Fh	CANCON_RO3 ⁽²⁾	F0Fh	RXF3EIDL
F6Eh	CANSTAT	F4Eh	CANSTAT_RO1 ⁽²⁾	F2Eh	CANSTAT_RO3 ⁽²⁾	F0Eh	RXF3EIDH
F6Dh	RXB0D7	F4Dh	TXB0D7	F2Dh	TXB2D7	F0Dh	RXF3SIDL
F6Ch	RXB0D6	F4Ch	TXB0D6	F2Ch	TXB2D6	F0Ch	RXF3SIDH
F6Bh	RXB0D5	F4Bh	TXB0D5	F2Bh	TXB2D5	F0Bh	RXF2EIDL
F6Ah	RXB0D4	F4Ah	TXB0D4	F2Ah	TXB2D4	F0Ah	RXF2EIDH
F69h	RXB0D3	F49h	TXB0D3	F29h	TXB2D3	F09h	RXF2SIDL
F68h	RXB0D2	F48h	TXB0D2	F28h	TXB2D2	F08h	RXF2SIDH
F67h	RXB0D1	F47h	TXB0D1	F27h	TXB2D1	F07h	RXF1EIDL
F66h	RXB0D0	F46h	TXB0D0	F26h	TXB2D0	F06h	RXF1EIDH
F65h	RXB0DLC	F45h	TXB0DLC	F25h	TXB2DLC	F05h	RXF1SIDL
F64h	RXB0EIDL	F44h	TXB0EIDL	F24h	TXB2EIDL	F04h	RXF1SIDH
F63h	RXB0EIDH	F43h	TXB0EIDH	F23h	TXB2EIDH	F03h	RXF0EIDL
F62h	RXB0SIDL	F42h	TXB0SIDL	F22h	TXB2SIDL	F02h	RXF0EIDH
F61h	RXB0SIDH	F41h	TXB0SIDH	F21h	TXB2SIDH	F01h	RXF0SIDL
F60h	RXB0CON	F40h	TXB0CON	F20h	TXB2CON	F00h	RXF0SIDH

Note 1: Shaded registers are available in Access Bank low area, while the rest are available in Bank 15.

2: CANSTAT register is repeated in these locations to simplify application firmware. Unique names are given for each instance of the controller register due to the Microchip header file requirement.

3: These registers are not CAN registers.

4: Unimplemented registers are read as '0'.

PIC18F2682/2685/4682/4685

In Mode 1 and 2, there are an additional 10 acceptance filters, RXF6-RXF15, creating a total of 16 available filters. RXF15 can be used either as an acceptance filter or acceptance mask register. Each of these acceptance filters can be individually enabled or disabled by setting or clearing the RXFENn bit in the RXFCONn register. Any of these 16 acceptance filters can be dynamically associated with any of the receive buffers. Actual association is made by setting appropriate bits in the RXFBCONn register. Each RXFBCONn register contains a nibble for each filter. This nibble can be used to associate a specific filter to any of available receive buffers. User firmware may associate more than one filter to any one specific receive buffer.

In addition to dynamic filter to buffer association, in Mode 1 and 2, each filter can also be dynamically associated to available acceptance mask registers. The FILn_m bits in the MSELn register can be used to link a specific acceptance filter to an acceptance mask register. As with filter to buffer association, one can also associate more than one mask to a specific acceptance filter.

When a filter matches and a message is loaded into the receive buffer, the filter number that enabled the message reception is loaded into the FILHIT bit(s). In Mode 0 for RXB1, the RXB1CON register contains the FILHIT<2:0> bits. They are coded as follows:

- 101 = Acceptance Filter 5 (RXF5)
- 100 = Acceptance Filter 4 (RXF4)
- 011 = Acceptance Filter 3 (RXF3)
- 010 = Acceptance Filter 2 (RXF2)
- 001 = Acceptance Filter 1 (RXF1)
- 000 = Acceptance Filter 0 (RXF0)

Note: '000' and '001' can only occur if the RXB0DBEN bit is set in the RXB0CON register, allowing RXB0 messages to rollover into RXB1.

The coding of the RXB0DBEN bit enables these three bits to be used similarly to the FILHIT bits and to distinguish a hit on filter RXF0 and RXF1, in either RXB0 or after a rollover into RXB1.

- 111 = Acceptance Filter 1 (RXF1)
- 110 = Acceptance Filter 0 (RXF0)
- 001 = Acceptance Filter 1 (RXF1)
- 000 = Acceptance Filter 0 (RXF0)

If the RXB0DBEN bit is clear, there are six codes corresponding to the six filters. If the RXB0DBEN bit is set, there are six codes corresponding to the six filters, plus two additional codes corresponding to RXF0 and RXF1 filters, that rollover into RXB1.

In Mode 1 and 2, each buffer control register contains 5 bits of filter hit bits (FILHIT<4:0>). A binary value of '0' indicates a hit from RXF0 and 15 indicates RXF15.

If more than one acceptance filter matches, the FILHIT bits will encode the binary value of the lowest numbered filter that matched. In other words, if filter RXF2 and filter RXF4 match, FILHIT will be loaded with the value for RXF2. This essentially prioritizes the acceptance filters with a lower number filter having higher priority. Messages are compared to filters in ascending order of filter number.

The mask and filter registers can only be modified when the PIC18F2682/2685/4682/4685 devices are in Configuration mode.

FIGURE 23-3: MESSAGE ACCEPTANCE MASK AND FILTER OPERATION

24.0 SPECIAL FEATURES OF THE CPU

PIC18F2682/2685/4682/4685 devices include several features intended to maximize reliability and minimize cost through elimination of external components. These are:

- Oscillator Selection
- Resets:
 - Power-on Reset (POR)
 - Power-up Timer (PWRT)
 - Oscillator Start-up Timer (OST)
 - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor
- · Two-Speed Start-up
- Code Protection
- ID Locations
- · In-Circuit Serial Programming

The oscillator can be configured for the application depending on frequency, power, accuracy and cost. All of the options are discussed in detail in **Section 2.0 "Oscillator Configurations"**.

A complete discussion of device Resets and interrupts is available in previous sections of this data sheet.

In addition to their Power-up and Oscillator Start-up Timers provided for Resets, PIC18F2682/2685/4682/ 4685 devices have a Watchdog Timer, which is either permanently enabled via the Configuration bits or software controlled (if configured as disabled). The inclusion of an internal RC oscillator also provides the additional benefits of a Fail-Safe Clock Monitor (FSCM) and Two-Speed Start-up. FSCM provides for background monitoring of the peripheral clock and automatic switchover in the event of its failure. Two-Speed Start-up enables code to be executed almost immediately on start-up, while the primary clock source completes its start-up delays.

All of these features are enabled and configured by setting the appropriate Configuration register bits.

24.1 Configuration Bits

The Configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped starting at program memory location 300000h.

The user will note that address 300000h is beyond the user program memory space. In fact, it belongs to the configuration memory space (300000h-3FFFFh), which can only be accessed using table reads and table writes.

Programming the Configuration registers is done in a manner similar to programming the Flash memory. The WR bit in the EECON1 register starts a self-timed write to the Configuration register. In normal operation mode, a TBLWT instruction with the TBLPTR pointing to the Configuration register sets up the address and the data for the Configuration register write. Setting the WR bit starts a long write to the Configuration registers are written a byte at a time. To write or erase a configuration cell, a TBLWT instruction can write a '1' or a '0' into the cell. For additional details on Flash programming, refer to Section 6.5 "Writing to Flash Program Memory".

 TABLE 24-1:
 CONFIGURATION BITS AND DEVICE IDs

File	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300001h	CONFIG1H	IESO	FCMEN	_		FOSC3	FOSC2	FOSC1	FOSC0	00 0111
300002h	CONFIG2L	_		_	BORV1	BORV0	BOREN1	BOREN0	PWRTEN	1 1111
300003h	CONFIG2H	_		_	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN	1 1111
300005h	CONFIG3H	MCLRE		_		_	LPT10SC	PBADEN	_	101-
300006h	CONFIG4L	DEBUG	XINST	BBSIZ1	BBSIZ2	_	LVP		STVREN	1000 -1-1
300008h	CONFIG5L	_	-	CP5 ⁽¹⁾	CP4	CP3	CP2	CP1	CP0	11 1111
300009h	CONFIG5H	CPD	CPB	_	_	—	_	_	—	11
30000Ah	CONFIG6L	_		WRT5 ⁽¹⁾	WRT4	WRT3	WRT2	WRT1	WRT0	11 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC		_			_	111
30000Ch	CONFIG7L	_	-	EBTR5 ⁽¹⁾	EBTR4	EBTR3	EBTR2	EBTR1	EBTR0	11 1111
30000Dh	CONFIG7H	_	EBTRB	_	_	_	_		_	-1
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	×××× ×××××(2)
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	×××× ×××××(2)

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition. Shaded cells are unimplemented, read as '0'. **Note 1:** Unimplemented in PIC18F2682/4682 devices; maintain this blt set.

2: See Register 24-12 and Register 24-13 for DEVID1 and DEVID2 values. DEVID registers are read-only and cannot be programmed by the user.

PIC18F2682/2685/4682/4685

TABLE 25-2: PIC18FXXXX INSTRUCTION SET

Mnemonic,		Description	Cycles	16-Bit Instruction Word				Status Bits	Notae	
Opera	nds	Description	Cycles	MSb			LSb	Affected	Notes	
BYTE-ORI	ENTED	OPERATIONS								
ADDWF	f, d, a	Add WREG and f	1	0010	01da	ffff	ffff	C, DC, Z, OV, N	1, 2	
ADDWFC	f, d, a	Add WREG and Carry bit to f	1	0010	00da	ffff	ffff	C, DC, Z, OV, N	1, 2	
ANDWF	f, d, a	AND WREG with f	1	0001	01da	ffff	ffff	Z, N	1,2	
CLRF	f, a	Clear f	1	0110	101a	ffff	ffff	Z	2	
COMF	f, d, a	Complement f	1	0001	11da	ffff	ffff	Z, N	1, 2	
CPFSEQ	f, a	Compare f with WREG, skip =	1 (2 or 3)	0110	001a	ffff	ffff	None	4	
CPFSGT	f, a	Compare f with WREG, skip >	1 (2 or 3)	0110	010a	ffff	ffff	None	4	
CPFSLT	f, a	Compare f with WREG, skip <	1 (2 or 3)	0110	000a	ffff	ffff	None	1, 2	
DECF	f, d, a	Decrement f	1	0000	01da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4	
DECFSZ	f, d, a	Decrement f, Skip if 0	1 (2 or 3)	0010	11da	ffff	ffff	None	1, 2, 3, 4	
DCFSNZ	f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	0100	11da	ffff	ffff	None	1, 2	
INCF	f, d, a	Increment f	1	0010	10da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4	
INCFSZ	f, d, a	Increment f, Skip if 0	1 (2 or 3)	0011	11da	ffff	ffff	None	4	
INFSNZ	f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	0100	10da	ffff	ffff	None	1, 2	
IORWF	f, d, a	Inclusive OR WREG with f	1	0001	00da	ffff	ffff	Z, N	1, 2	
MOVF	f, d, a	Move f	1	0101	00da	ffff	ffff	Z, N	1	
MOVFF	f_s, f_d	Move f _s (source) to 1st word	2	1100	ffff	ffff	ffff	None		
	5. u	f _d (destination)2nd word		1111	ffff	ffff	ffff			
MOVWF	f, a	Move WREG to f	1	0110	111a	ffff	ffff	None		
MULWF	f, a	Multiply WREG with f	1	0000	001a	ffff	ffff	None	1, 2	
NEGF	f, a	Negate f	1	0110	110a	ffff	ffff	C, DC, Z, OV, N		
RLCF	f, d, a	Rotate Left f through Carry	1	0011	01da	ffff	ffff	C, Z, N	1, 2	
RLNCF	f, d, a	Rotate Left f (No Carry)	1	0100	01da	ffff	ffff	Z, N	,	
RRCF	f, d, a	Rotate Right f through Carry	1	0011	00da	ffff	ffff	C, Z, N		
RRNCF	f, d, a	Rotate Right f (No Carry)	1	0100	00da	ffff	ffff	Z, N		
SETF	f, a	Set f	1	0110	100a	ffff	ffff	None	1, 2	
SUBFWB	f, d, a	Subtract f from WREG with	1	0101	01da	ffff	ffff	C, DC, Z, OV, N		
		borrow								
SUBWF	f, d, a	Subtract WREG from f	1	0101	11da	ffff	ffff	C, DC, Z, OV, N	1, 2	
SUBWFB	f, d, a	Subtract WREG from f with	1	0101	10da	ffff	ffff	C, DC, Z, OV, N		
		borrow								
SWAPF	f, d, a	Swap nibbles in f	1	0011	10da	ffff	ffff	None	4	
TSTFSZ	f, a	Test f, skip if 0	1 (2 or 3)	0110	011a	ffff	ffff	None	1, 2	
XORWF	f, d, a	Exclusive OR WREG with f	1 ′	0001	10da	ffff	ffff	Z, N		

Note 1: When a PORT register is modified as a function of itself (e.g., MOVF PORTB, 1, 0), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and where applicable, 'd' = 1), the prescaler will be cleared if assigned.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

4: Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

5: If the table write starts the write cycle to internal memory, the write will continue until terminated.

PIC18F2682/2685/4682/4685

BNC	;	Branch if	Not Carry		BNN		Branch if	Not Negativ	/e
Synt	ax:	BNC n			Synta	ax:	BNN n		
Oper	ands:	-128 ≤ n ≤ 1	127		Oper	ands:	-128 ≤ n ≤ 1	127	
Oper	ation:	if Carry bit i (PC) + 2 + 2	s '0' 2n → PC		Oper	ation:	if Negative (PC) + 2 + 2	bit is '0' 2n → PC	
Statu	s Affected:	None			Statu	s Affected:	None		
Enco	oding:	1110	0011 nn:	nn nnnn	Enco	ding:	1110	0111 nn	nn nnnn
Desc	cription:	If the Carry will branch.	bit is '0', then	the program	Desc	ription:	If the Negat program wi	tive bit is '0', t Il branch.	hen the
		The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.				The 2's con added to the incrementer instruction, PC + 2 + 2r two-cycle in	nplement num e PC. Since th d to fetch the the new addro n. This instruction.	ber '2n' is le PC will have next ess will be tion is then a	
Word	ls:	1			Word	Words: 1			
Cycle	es:	1(2)			Cycle	Cycles: 1(2)			
QC	ycle Activity:				QC	cle Activity:			
lf Ju	imp:				lf Ju	mp:			
	Q1	Q2	Q3	Q4	1	Q1	Q2	Q3	Q4
	Decode	Read literal 'n'	Process Data	Write to PC		Decode	Read literal 'n'	Process Data	Write to PC
	No	No	No	No		No	No	No	No
	operation	operation	operation	operation	ļ	operation	operation	operation	operation
lf No	o Jump:				lf No	Jump:			
	Q1	Q2	Q3	Q4	1	Q1	Q2	Q3	Q4
	Decode	Read literal 'n'	Process Data	No operation		Decode	Read literal 'n'	Process Data	No operation
<u>Exar</u>	Example: HERE BNC Jump Before Instruction		Exan	n <u>ple:</u> Before Instruc	HERE	BNN Jump	,		
After Instruction If Carry = 0; PC = address (Jump) If Carry = 1; PC = address (HERE + 2)					After Instruction If Negative = 0; PC = address (Jump) If Negative = 1; PC = address (HERE + 2)				

27.2 DC Characteristics: Power-Down and Supply Current PIC18F2682/2685/4682/4685 (Industrial) PIC18LF2682/2685/4682/4685 (Industrial) (Continued)

PIC18LF2682/2685/4682/4685 (Industrial)		$\begin{array}{llllllllllllllllllllllllllllllllllll$							
PIC18F2682/2685/4682/4685 (Industrial, Extended)		$ \begin{array}{ll} \mbox{Standard Operating Conditions (unless otherwise stated)} \\ \mbox{Operating temperature} & -40^{\circ}C \leq Ta \leq +85^{\circ}C \mbox{ for industrial} \\ -40^{\circ}C \leq Ta \leq +125^{\circ}C \mbox{ for extended} \end{array} $							
Param No.	Param Device Supply Current (IDD) ^(2,3)		Max	Units	Conditions				
	PIC18LF268X/468X	65	220	μA	-40°C				
		65	220	μA	+25°C	VDD = 2.0V			
		70	220	μA	+85°C				
	PIC18LF268X/468X	120	330	μA	-40°C				
		120	330	μA	+25°C	VDD = 3.0V	FOSC = 1 MHz		
		130	330	μA	+85°C		EC oscillator)		
	All devices	300	600	μA	-40°C	-			
		240	600	μA	+25°C	$V_{DD} = 5.0V$			
		300	600	μA	+85°C	VDD - 3.0V			
	Extended devices only	320	600	μA	+125°C				
	PIC18LF268X/468X	260	760	μA	-40°C	-			
		255	760	μA	+25°C	VDD = 2.0V			
		270	760	μA	+85°C				
	PIC18LF268X/468X	420	1.4	μA	-40°C	-			
		430	1.4	μA	+25°C	VDD = 3.0V	FOSC = 4 MHZ (PRI_IDLE mode		
		450	1.4	μA	+85°C		EC oscillator)		
	All devices	0.9	2.2	mA	-40°C	-			
		0.9	2.2	mA	+25°C	VDD = 5.0V			
		0.9	2.2	mA	+85°C	100 0.01			
	Extended devices only	1	3	mA	+125°C				
	Extended devices only	2.8	7	mA	+125°C	VDD = 4.2V	Fosc = 25 MHz		
		4.3	11	mA	+125°C	VDD = 5.0V	(PRI_IDLE mode, EC oscillator)		
	All devices	6	18	mA	-40°C				
		6.2	18	mA	+25°C	VDD = 4.2 V			
		6.6	18	mA	+85°C		Fosc = 40 MHz		
	All devices	8.1	22	mA	-40°C		EC oscillator)		
		9.1	22	mA	+25°C	VDD = 5.0V	,		
		8.3	22	mA	+85°C				

Legend: Shading of rows is to assist in readability of the table.

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSs and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.

The test conditions for all IDD measurements in active operation mode are:

OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;

 \overline{MCLR} = VDD; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through RExT is not included. The current through the resistor can be estimated by the formula Ir = VDD/2REXT (mA) with REXT in kΩ.

4: Standard low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

40-Lead Plastic Dual In-Line (P) – 600 mil Body [PDIP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

	Units		INCHES	
Dimensio	n Limits	MIN	NOM	MAX
Number of Pins	Ν		40	
Pitch	е		.100 BSC	
Top to Seating Plane	Α	—	-	.250
Molded Package Thickness	A2	.125	-	.195
Base to Seating Plane	A1	.015	-	-
Shoulder to Shoulder Width	E	.590	-	.625
Molded Package Width	E1	.485	-	.580
Overall Length	D	1.980	-	2.095
Tip to Seating Plane	L	.115	-	.200
Lead Thickness	С	.008	-	.015
Upper Lead Width	b1	.030	-	.070
Lower Lead Width	b	.014	-	.023
Overall Row Spacing §	eВ	_	_	.700

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. § Significant Characteristic.

3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.

4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-016B

Data EEPROM Memory	107
Associated Registers	111
EEADR and EEADRH Registers	107
EECON1 and EECON2 Registers	107
Operation During Code-Protect	110
Protection Against Spurious Write	110
Reading	109
Use	110
Write Verify	109
Writing	109
Data Memory	69
Access Bank	71
Bank Select Register (BSR)	69
Extended Instruction Set	93
General Purpose Register File	71
Map for PIC18F268X/468X	70
Special Function Registers	72
DAW	384
DC and AC Characteristics	455
DC Characteristics	430
Power-Down and Supply Current	422
Supply Voltage	421
DCFSNZ	385
DECF	384
DECFSZ	385
Development Support	415
Device Differences	465
Device Overview	9
Features (table)	11
New Core Features	9
Device Reset Timers	47
Oscillator Start-up Timer (OST)	47
PLL Lock Time-out	47
Power-up Timer (PWRT)	47
Direct Addressing	91
Disable Mode	327

E EC

AN Module	
Baud Rate Setting	335
Bit Time Partitioning	335
Bit Timing Configuration Registers	340
Calculating TQ, Nominal Bit Rate and Nominal Bit	Time
338	
CAN Interrupts	342
Acknowledge	344
Bus Activity Wake-up	344
Bus-Off	344
Code Bits	343
Error	344
Message Error	343
Receive	343
Receiver Bus Passive	344
Receiver Overflow	344
Receiver Warning	344
Transmit	343
Transmitter Bus Passive	344
Transmitter Warning	344
CAN Message Buffers	329
Dedicated Receive	329
Dedicated Transmit	329
Programmable Auto-RTR	330
Programmable Transmit/Receive	329

CAN Message Transmission	. 330
Aborting	330
Initiating	330
Priority	331
CAN Modes of Operation	227
CAN Desisters	. 321
CAN REGISTERS	. 211
	. 314
Control and Status	. 277
Controller Map	. 322
Dedicated Receive Buffers	. 290
Dedicated Transmit Buffers	. 284
I/O Control	. 317
Interrupt	. 318
Configuration Mode	. 327
Disable Mode	. 327
Error Detection	341
Acknowledge	3/1
	244
	. 341
	. 341
Error States	. 341
Form	. 341
Modes and Counters	. 341
Stuff Bit	. 341
Error Modes State (diagram)	. 342
Error Recognition Mode	. 328
Filter-Mask Truth (table)	333
Functional Modes	328
	328
Mode 1 (Enhanced Logacy Mode)	2020
Mode 1 (Enhanced Legacy Mode)	. 320
Mode 2 (Ennanced FIFO Mode)	. 329
Information Processing Time (IPT)	. 338
Lengthening a Bit Period	. 339
Listen Only Mode	. 328
Loopback Mode	. 328
Message Acceptance Filters and Masks 305	, 333
Message Acceptance Mask and	
Filter Operation	. 334
Message Recention	332
Enhanced EIEO Mode	333
Priority	. 330
Timo Stomping	202
Name Made	. 333
	. 321
	. 340
Overview	. 275
Phase Buffer Segments	. 338
Programmable TX/RX and Auto-RTR Buffers	. 297
Programming Time Segments	. 340
Propagation Segment	. 338
Sample Point	. 338
Shortening a Bit Period	. 340
Synchronization	339
Hard	339
Resynchronization	330
Resylicitionization	220
	. ააყ
	. 338
Time Quanta	. 338
Values for ICODE (table)	. 343
Effect on Standard PIC Instructions	. 412
Electrical Characteristics	. 419

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To:	Technical Publications Manager	Total Pages Sent
RE:	Reader Response	
From	: Name	
	Company	
	Address	
	City / State / ZIP / Country	
	Telephone: ()	FAX: ()
Appli	cation (optional):	
Woul	d you like a reply?YN	
Devic	ce: PIC18F2682/2685/4682/4685	Literature Number: DS39761C
Ques	tions:	
1. V	Vhat are the best features of this docu	ument?
_		
_		
2. ⊦	low does this document meet your ha	ardware and software development needs?
_		
-		
3. C	o you find the organization of this do	cument easy to follow? If not, why?
—		
	What additions to the document do vo	u think would enhance the structure and subject?
ч . v		
5. V	Vhat deletions from the document cou	uld be made without affecting the overall usefulness?
		Ū
_		
6. Is	s there any incorrect or misleading inf	formation (what and where)?
_		
_		
7. ⊦	low would you improve this documen	t?
_		
_		