



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

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	ECANbus, I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	54
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-VQFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf65k80-i-mr

Email: info@E-XFL.COM

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

2.0 GUIDELINES FOR GETTING STARTED WITH PIC18FXXKXX MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC18F66K80 family family of 8-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and Vss pins (see Section 2.2 "Power Supply Pins")
- All AVDD and AVss pins, regardless of whether or not the analog device features are used (see Section 2.2 "Power Supply Pins")
- MCLR pin (see Section 2.3 "Master Clear (MCLR) Pin")
- ENVREG (if implemented) and VCAP/VDDCORE pins (see Section 2.4 "Voltage Regulator Pins (ENVREG and VCAP/VDDCORE)")

These pins must also be connected if they are being used in the end application:

- PGC/PGD pins used for In-Circuit Serial Programming[™] (ICSP[™]) and debugging purposes (see **Section 2.5 "ICSP Pins**")
- OSCI and OSCO pins when an external oscillator source is used

(see Section 2.6 "External Oscillator Pins")

Additionally, the following pins may be required:

• VREF+/VREF- pins are used when external voltage reference for analog modules is implemented

Note: The AVDD and AVss pins must always be connected, regardless of whether any of the analog modules are being used.

The minimum mandatory connections are shown in Figure 2-1.

FIGURE 2-1: RECOMMENDED



Key (all values are recommendations):

C1 through C6: 0.1 $\mu\text{F},$ 20V ceramic R1: 10 k Ω

R2: 100Ω to 470Ω

- Note 1: See Section 2.4 "Voltage Regulator Pins (ENVREG and VCAP/VDDCORE)" for explanation of ENVREG pin connections.
 - 2: The example shown is for a PIC18F device with five VDD/VSS and AVDD/AVSS pairs. Other devices may have more or less pairs; adjust the number of decoupling capacitors appropriately.

6.6.3 MAPPING THE ACCESS BANK IN INDEXED LITERAL OFFSET MODE

The use of Indexed Literal Offset Addressing mode effectively changes how the lower part of Access RAM (00h to 5Fh) is mapped. Rather than containing just the contents of the bottom part of Bank 0, this mode maps the contents from Bank 0 and a user-defined "window" that can be located anywhere in the data memory space.

The value of FSR2 establishes the lower boundary of the addresses mapped into the window, while the upper boundary is defined by FSR2 plus 95 (5Fh). Addresses in the Access RAM above 5Fh are mapped as previously described. (See **Section 6.3.2 "Access Bank"**.) An example of Access Bank remapping in this addressing mode is shown in Figure 6-10. Remapping the Access Bank applies *only* to operations using the Indexed Literal Offset mode. Operations that use the BSR (Access RAM bit = 1) will continue to use Direct Addressing as before. Any Indirect or Indexed Addressing operation that explicitly uses any of the indirect file operands (including FSR2) will continue to operate as standard Indirect Addressing. Any instruction that uses the Access Bank, but includes a register address of greater than 05Fh, will use Direct Addressing and the normal Access Bank map.

6.6.4 BSR IN INDEXED LITERAL OFFSET MODE

Although the Access Bank is remapped when the extended instruction set is enabled, the operation of the BSR remains unchanged. Direct Addressing, using the BSR to select the data memory bank, operates in the same manner as previously described.

FIGURE 6-10: REMAPPING THE ACCESS BANK WITH INDEXED LITERAL OFFSET ADDRESSING



10.0 INTERRUPTS

Members of the PIC18F66K80 family of devices have multiple interrupt sources and an interrupt priority feature that allows most interrupt sources to be assigned a high-priority level or a low-priority level. The high-priority interrupt vector is at 0008h and the low-priority interrupt vector is at 0018h. High-priority interrupt events will interrupt any low-priority interrupts that may be in progress.

The registers for controlling interrupt operation are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3, PIR4 and PIR5
- PIE1, PIE2, PIE3, PIE4 and PIE5
- IPR1, IPR2, IPR3, IPR4 and IPR5

It is recommended that the Microchip header files supplied with MPLAB[®] IDE be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

In general, interrupt sources have three bits to control their operation. They are:

- Flag bit Indicating that an interrupt event occurred
- Enable bit Enabling program execution to branch to the interrupt vector address when the flag bit is set
- **Priority bit** Specifying high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits that enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set (high priority). Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared (low priority). When the interrupt flag, enable bit and appropriate Global Interrupt Enable bit are set, the interrupt will vector immediately to address 0008h or 0018h, depending on the priority bit setting. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC[®] mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit that enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit that enables/disables all interrupt sources. All interrupts branch to address 0008h in Compatibility mode.

When an interrupt is responded to, the Global Interrupt Enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High-priority interrupt sources can interrupt a low-priority interrupt. Low-priority interrupts are not processed while high-priority interrupts are in progress.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (0008h or 0018h). Once in the Interrupt Service Routine (ISR), the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

The "return from interrupt" instruction, RETFIE, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used) that re-enables interrupts.

For external interrupt events, such as the INTx pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding enable bit or the GIE bit.

Note: Do not use the MOVFF instruction to modify any of the Interrupt Control registers while **any** interrupt is enabled. Doing so may cause erratic microcontroller behavior.

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIE		_	_	BCLIE	HLVDIE	TMR3IE	TMR3GIE
bit 7	·						bit 0
Legend:							
R = Readable	e bit	W = Writable	bit	U = Unimplen	nented bit, read	d as '0'	
-n = Value at	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	nown
bit 7	OSCFIE: Osc	illator Fail Inte	rupt Enable bi	t			
	1 = Enabled						
	0 = Disabled						
bit 6-4	Unimplemen	ted: Read as '	כ'				
bit 3	BCLIE: Bus C	Collision Interru	pt Enable bit				
	1 = Enabled						
	0 = Disabled						
bit 2	HLVDIE: High	n/Low-Voltage I	Detect Interrup	t Enable bit			
	1 = Enabled						
	0 = Disabled						
bit 1	TMR3IE: TMF	R3 Overflow Int	errupt Enable	bit			
	1 = Enabled						
	0 = Disabled						
bit 0	TMR3GIE: Tir	mer3 Gate Inte	rrupt Enable bi	it			
	1 = Enabled						
	0 = Disabled						

REGISTER 10-10: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

12.0 DATA SIGNAL MODULATOR

Note: The Data Signal Modulator is only available on 64-pin devices (PIC18F6XK80).

The Data Signal Modulator (DSM) is a peripheral which allows the user to mix a data stream, also known as a modulator signal, with a carrier signal to produce a modulated output.

Both the carrier and the modulator signals are supplied to the DSM module, either internally from the output of a peripheral, or externally through an input pin.

The modulated output signal is generated by performing a logical "AND" operation of both the carrier and modulator signals and then it is provided to the MDOUT pin.

The carrier signal is comprised of two distinct and separate signals: a carrier high (CARH) signal and a carrier low (CARL) signal. During the time in which the modulator (MOD) signal is in a logic high state, the DSM mixes the carrier high signal with the modulator signal. When the modulator signal is in a logic low state, the DSM mixes the carrier low signal with the modulator signal. Using this method, the DSM can generate the following types of key modulation schemes:

- Frequency-Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- On-Off Keying (OOK)

Additionally, the following features are provided within the DSM module:

- Carrier Synchronization
- · Carrier Source Polarity Select
- Carrier Source Pin Disable
- Programmable Modulator Data
- Modulator Source Pin Disable
- Modulated Output Polarity Select
- Slew Rate Control

Figure 12-1 shows a simplified block diagram of the Data Signal Modulator peripheral.

18.9 Measuring Temperature with the CTMU Module

The CTMU, along with an internal diode, can be used to measure the temperature. The A/D can be connected to the internal diode and the CTMU module can

source the current to the diode. The A/D reading will reflect the temperature. With the increase, the A/D readings will go low. This can be used for low-cost temperature measurement applications.

EXAMPLE 18-6: ROUTINE FOR TEMPERATURE MEASUREMENT USING INTERNAL DIODE

<pre>// Initialize CTMU CTMUICON = 0x03; CTMUCONHbits.CTMUEN = 1; CTMUCONLbits.EDG1STAT = 1;</pre>	
// Initialize ADC	
$ADCON0 = 0 \times E5;$ $ADCON1 = 0 \times 00;$	// Enable ADC and connect to Internal diode
$ADCON2 = 0 \times BE;$	//Right Justified
ADCON0bits.GO = 1; while(ADCON0bits.G0);	// Start conversion
Temp = ADRES;	// Read ADC results (inversely proportional to temperature)

Note: The temperature diode is not calibrated or standardized; the user must calibrate the diode to their application.

20.4.6 PROGRAMMABLE DEAD-BAND DELAY MODE

In half-bridge applications, where all power switches are modulated at the PWM frequency, the power switches normally require more time to turn off than to turn on. If both the upper and lower power switches are switched at the same time (one turned on and the other turned off), both switches may be on for a short period until one switch completely turns off. During this brief interval, a very high current (shoot-through current) will flow through both power switches, shorting the bridge supply. To avoid this potentially destructive shoot-through current from flowing during switching, turning on either of the power switches is normally delayed to allow the other switch to completely turn off.

In Half-Bridge mode, a digitally programmable dead-band delay is available to avoid shoot-through current from destroying the bridge power switches. The delay occurs at the signal transition from the non-active state to the active state. For an illustration, see Figure 20-14. The lower seven bits of the associated ECCP1DEL register (Register 20-4) set the delay period in terms of microcontroller instruction cycles (TcY or 4 Tosc).



FIGURE 20-15: EXAMPLE OF HALF-BRIDGE APPLICATIONS



21.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

21.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be devices such as serial EEPROMs, shift registers, display drivers and A/D Converters. The MSSP module can operate in either of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I²C[™])
 - Full Master mode
 - Slave mode (with general address call)

The $\mathrm{I}^2\mathrm{C}$ interface supports the following modes in hardware:

- Master mode
- · Multi-Master mode
- Slave mode with 5-bit and 7-bit address masking (with address masking for both 10-bit and 7-bit addressing)

21.2 Control Registers

The MSSP module has three associated control registers. These include a status register (SSPSTAT) and two control registers (SSPCON1 and SSPCON2). The use of these registers and their individual configuration bits differ significantly depending on whether the MSSP module is operated in SPI or I^2C mode.

Additional details are provided under the individual sections.

21.3 SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) RC5/SDO
- Serial Data In (SDI) RC4/SDA/SDI
- Serial Clock (SCK) RC3/REF0/SCL/SCK

Additionally, a fourth pin may be used when in a Slave mode of operation:

 Slave Select (SS) – RA5/AN4/C2INB/ HLVDIN/T1CKI/SS/CTMU1

Figure 21-1 shows the block diagram of the MSSP module when operating in SPI mode.

FIGURE 21-1:

MSSP BLOCK DIAGRAM (SPI MODE)



21.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- · Slave Select mode (Slave mode only)

The MSSP module consists of a Transmit/Receive Shift register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>), and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision Detect bit, WCOL (SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 21-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the SSPSTAT register indicates the various status conditions.

21.3.3 OPEN-DRAIN OUTPUT OPTION

The drivers for the SDO output and SCK clock pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor, and allows the output to communicate with external circuits without the need for additional level shifters. For more information, see **Section 11.1.3 "Open-Drain Outputs"**.

The open-drain output option is controlled by the SSPOD bit (ODCON<7>). Setting the SSPOD bit configures the SDO and SCK pins for open-drain operation.

EXAMPLE 21-1: LOADING THE SSPBUF (SSPSR) REGISTER

LOOP	BTFSS	SSPSTAT, BF	;Has data been received (transmit complete)?
	BRA	LOOP	;No
	MOVF	SSPBUF, W	;WREG reg = contents of SSPBUF
	MOVWF	RXDATA	;Save in user RAM, if data is meaningful
	MOVF	TXDATA, W	;W reg = contents of TXDATA
	MOVWF	SSPBUF	;New data to xmit

22.5 EUSARTx Synchronous Slave Mode

Synchronous Slave mode is entered by clearing bit, CSRC (TXSTAx<7>). This mode differs from the Synchronous Master mode in that the shift clock is supplied externally at the CKx pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in any low-power mode.

22.5.1 EUSARTx SYNCHRONOUS SLAVE TRANSMISSION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep mode.

If two words are written to the TXREGx and then the SLEEP instruction is executed, the following will occur:

- a) The first word will immediately transfer to the TSR register and transmit.
- b) The second word will remain in the TXREGx register.
- c) Flag bit, TXxIF, will not be set.
- d) When the first word has been shifted out of TSR, the TXREGx register will transfer the second word to the TSR and flag bit, TXxIF, will now be set.

e) If enable bit, TXxIE, is set, the interrupt will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Transmission:

- 1. Enable the synchronous slave serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
- 2. Clear bits, CREN and SREN.
- 3. If interrupts are desired, set enable bit, TXxIE.
- 4. If 9-bit transmission is desired, set bit, TX9.
- 5. Enable the transmission by setting enable bit, TXEN.
- 6. If 9-bit transmission is selected, the ninth bit should be loaded in bit, TX9D.
- 7. Start transmission by loading data to the TXREGx register.
- 8. If using interrupts, ensure that the GIE and PEIE bits (INTCON<7:6>) are set.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	TMR1GIF	TMR2IF	TMR1IF	
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	TMR1GIE	TMR2IE	TMR1IE	
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	TMR1GIP	TMR2IP	TMR1IP	
PIR3	_	_	RC2IF	TX2IF	CTMUIF	CCP2IF	CCP1IF		
PIE3	—	—	RC2IE	TX2IE	CTMUIE	CCP2IE	CCP1IE	_	
IPR3	—	—	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP		
RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	
TXREG1	EUSART1 Transmit Register								
TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	
BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	
SPBRGH1	EUSART1 B	Baud Rate Ger	nerator Regi	ster High Byt	e				
SPBRG1	EUSART1 B	Baud Rate Gei	nerator Regi	ster Low Byte	e				
RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	
TXREG2	EUSART2 T	ransmit Regis	ster						
TXSTA2	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	
BAUDCON2	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	_	WUE	ABDEN	
SPBRGH2	EUSART2 Baud Rate Generator Register High Byte								
SPBRG2	EUSART2 Baud Rate Generator Register Low Byte								
PMD0	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	UART2MD	UART1MD	SSPMD	
ODCON	SSPOD	CCP50D	CCP4OD	CCP3OD	CCP2OD	CCP10D	U2OD	U10D	

TABLE 22-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Legend: — = unimplemented, read as '0'. Shaded cells are not used for synchronous slave transmission.

26.2 HLVD Setup

To set up the HLVD module:

- 1. Select the desired HLVD trip point by writing the value to the HLVDL<3:0> bits.
- 2. Set the VDIRMAG bit to detect high voltage (VDIRMAG = 1) or low voltage (VDIRMAG = 0).
- 3. Enable the HLVD module by setting the HLVDEN bit.
- 4. Clear the HLVD interrupt flag (PIR2<2>), which may have been set from a previous interrupt.
- If interrupts are desired, enable the HLVD interrupt by setting the HLVDIE and GIE bits (PIE2<2> and INTCON<7>, respectively).

An interrupt will not be generated until the IRVST bit is set.

Note: Before changing any module settings (VDIRMAG, HLVDL<3:0>), first disable the module (HLVDEN = 0), make the changes and re-enable the module. This prevents the generation of false HLVD events.

26.3 Current Consumption

When the module is enabled, the HLVD comparator and voltage divider are enabled and consume static current. The total current consumption, when enabled, is specified in electrical specification Parameter D022B (Table 31-11).

Depending on the application, the HLVD module does not need to operate constantly. To reduce current requirements, the HLVD circuitry may only need to be enabled for short periods where the voltage is checked. After such a check, the module could be disabled.

26.4 HLVD Start-up Time

The internal reference voltage of the HLVD module, specified in electrical specification Parameter 37 (Section 31.0 "Electrical Characteristics"), may be used by other internal circuitry, such as the programmable Brown-out Reset. If the HLVD or other circuits using the voltage reference are disabled to lower the device's current consumption, the reference voltage circuit will require time to become stable before a low or high-voltage condition can be reliably detected. This start-up time, TIRVST, is an interval that is independent of device clock speed. It is specified in electrical specification Parameter 37 (Table 31-11).

The HLVD interrupt flag is not enabled until TIRVST has expired and a stable reference voltage is reached. For this reason, brief excursions beyond the set point may not be detected during this interval (see Figure 26-2 or Figure 26-3).

REGISTER 27-6: TXBnSIDH: TRANSMIT BUFFER 'n' STANDARD IDENTIFIER REGISTERS, HIGH BYTE $[0 \le n \le 2]$

			1				
R/W-x							
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
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 7-0 **SID<10:3>:** Standard Identifier bits (if EXIDE (TXBnSIDL<3>) = 0) Extended Identifier bits, EID<28:21> (if EXIDE = 1).

REGISTER 27-7: TXBnSIDL: TRANSMIT BUFFER 'n' STANDARD IDENTIFIER REGISTERS, LOW BYTE $[0 \le n \le 2]$

R/W-x	R/W-x	R/W-x	U-0	R/W-x	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDE	—	EID17	EID16
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 7-5	SID<2:0>: Standard Identifier bits (if EXIDE (TXBnSIDL<3>) = 0) Extended Identifier bits, EID<20:18> (if EXIDE = 1).
bit 4	Unimplemented: Read as '0'
bit 3	EXIDE: Extended Identifier Enable bit
	1 = Message will transmit extended ID, SID<10:0> become EID<28:18> 0 = Message will transmit standard ID, EID<17:0> are ignored
bit 2	Unimplemented: Read as '0'
bit 1-0	EID<17:16>: Extended Identifier bits

REGISTER 27-8: TXBnEIDH: TRANSMIT BUFFER 'n' EXTENDED IDENTIFIER REGISTERS, HIGH BYTE [0 \leq n \leq 2]

| R/W-x |
|-------|-------|-------|-------|-------|-------|-------|-------|
| EID15 | EID14 | EID13 | EID12 | EID11 | EID10 | EID9 | EID8 |
| 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 7-0 EID<15:8>: Extended Identifier bits (not used when transmitting standard identifier message)

PIC18F66K80 FAMILY

27.2.4 CAN BAUD RATE REGISTERS

This section describes the CAN Baud Rate registers.

Note:	These	registers	are	writable	in
	Configu	ration mode	only.		

REGISTER 27-52: BRGCON1: BAUD RATE CONTROL REGISTER 1

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| SJW1 | SJW0 | BRP5 | BRP4 | BRP3 | BRP2 | BRP1 | BRP0 |
| bit 7 | | | | | | | bit 0 |

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

bit 7-6	SJW<1:0>: Synchronized Jump Width bits
	11 = Synchronization jump width time = $4 \times TQ$
	10 = Synchronization jump width time = $3 \times TQ$
	01 = Synchronization jump width time = $2 \times TQ$
	00 = Synchronization jump width time = 1 x TQ
bit 5-0	BRP<5:0>: Baud Rate Prescaler bits
	111111 = Tq = (2 x 64)/Fosc
	111110 = Tq = (2 x 63)/Fosc
	:
	:
	000001 = Tq = (2 x 2)/Fosc
	000000 = Tq = (2 x 1)/Fosc

27.2.5 CAN MODULE I/O CONTROL REGISTER

This register controls the operation of the CAN module's I/O pins in relation to the rest of the microcontroller.

REGISTER 27-55: CIOCON: CAN I/O CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
TX2SRC	TX2EN	ENDRHI ⁽¹⁾	CANCAP	—	—	—	CLKSEL
bit 7							bit 0

Legend:				
R = Read	lable bit	W = Writable bit	U = Unimplemented bit,	read as '0'
-n = Value	e at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
bit 7	TX2SR0	: CANTX2 Pin Data Source	bit	
	1 = CAN 0 = CAN	ITX2 pin will output the CAN ITX2 pin will output CANTX	clock	
bit 6	TX2EN:	CANTX Pin Enable bit		
	1 = CAN 0 = CAN	ITX2 pin will output CANTX o ITX2 pin will have digital I/O t	or CAN clock as selected by th function	e TX2SRC bit
bit 5	ENDRH	I: Enable Drive High bit ⁽¹⁾		
	1 = CAN 0 = CAN	ITX pin will drive Voo when re ITX pin will be tri-state when	ecessive recessive	
bit 4	CANCA	P: CAN Message Receive Ca	apture Enable bit	
	1 = Ena 0 = Disa	ble CAN capture; CAN messa ble CAN capture; RC2/CCP1	age receive signal replaces in I input to CCP1 module	put on RC2/CCP1
bit 3-1	Unimple	emented: Read as '0'		
bit 0	CLKSE	.: CAN Clock Source Selection	on bit	
	1 = Use 0 = Use	the oscillator as the source of the PLL as the source of the	of the CAN system clock CAN system clock	
Note 1:	Always set th	is bit when using a differentia	I bus to avoid signal crosstalk	in CANTX from other nearby pir

		R/\/_0	R/W/_0	R/\//_0	R/M/ 0	R/\//_0	R/\//_0	R/\//_0
Mode 0								
	INAL	WARLE		TADZIE	TABIL: /	I ADVIE: /	RADIIE	RADULE
	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Mode 1	IRXIE	WAKIE	ERRIE	TXBnIE	TXB1IE ⁽¹⁾	TXB0IE ⁽¹⁾	RXBnIE	FIFOWMIE
	bit 7							bit 0
Legend:								
R = Read	able bit		W = Writabl	e bit	U = Unimple	emented bit, r	ead as '0'	
-n = Value	e at POR		'1' = Bit is s	et	'0' = Bit is cl	eared	x = Bit is un	known
			_					
bit 7	IRXIE: CAN	Bus Error Me	essage Rece	ived Interru	pt Enable bit			
	1 = Enable	invalid messa	ge received	interrupt				
hit 6		N bus Activity	Wake-un In	terrunt Enak	ole hit			
bit 0	1 = Enable	bus activity wa	ake-up interr	unt				
	0 = Disable	bus activity w	ake-up inter	rupt				
bit 5	ERRIE: CA	N bus Error In	terrupt Enab	le bit				
	1 = Enable	CAN module	error interrup	ot				
	0 = Disable	CAN module	error interru	ot				
bit 4	When CAN	<u>is in Mode 0:</u>	uffer O Intern	unt Enchlad	L:4			
	1 = Enable	AN Transmit B	er 2 interrunt	upt Enable I	DIL			
	0 = Disable	Transmit Buff	er 2 interrup	t				
	When CAN	is in Mode 1 d	or <u>2:</u>					
	TXBnIE: CA	AN Transmit B	Suffer Interrup	ots Enable b	oit			
	1 = Enable	transmit buffe	r interrupt; in	dividual inte	errupt is enable	ed by TXBIE	and BIE0	
hit 2		All transmit D	uffor 1 Interrup	is unt Enchlo I	⊢: + (1)			
DILS	1 = Enable	Transmit Ruffe	er 1 interrunt		UIL [®]			
	0 = Disable	Transmit Buff	er 1 interrup	t				
bit 2	TXB0IE: CA	AN Transmit B	Suffer 0 Interr	upt Enable	bit ⁽¹⁾			
	1 = Enable	Transmit Buffe	er 0 interrupt					
	0 = Disable	Transmit Buff	er 0 interrup	t				
bit 1	When CAN	is in Mode 0:	<i></i>					
	1 = Enable	AN Receive B	uffer 1 Interr	upt Enable t	DIT			
	0 = Disable	Receive Buffe	er 1 interrupt					
	When CAN	is in Mode 1 d	or 2:					
	RXBnIE: C	AN Receive B	uffer Interrup	ots Enable b	it			
	1 = Enable	receive buffer	interrupt; inc	dividual inter	rupt is enable	d by BIE0		
1.1.0		all receive bu	πer interrupt	S				
DIT U		<u>IS IN MODE U:</u> AN Receive B	uffer () Interr	unt Enable k	hit			
	1 = Enable	Receive Buffe	er 0 interrupt					
	0 = Disable	Receive Buffe	er 0 interrupt					
	When CAN	is in Mode 1:						
	Unimpleme	ented: Read a	IS '0'					
	When CAN	IS IN Mode 2:	nark Intorrun	t Enabla bit				
	1 = Enable	FIFO waterma	ark interrunt					
	0 = Disable	FIFO waterm	ark interrupt					
Note 1		1 and 2 there	o hito ara fa	read to 'a'				
NULE I.		z i anu z. ines	e uis ale 10	10 2010 0.				

© 2010-2012 Microchip Technology Inc.

27.10 Synchronization

To compensate for phase shifts between the oscillator frequencies of each of the nodes on the bus, each CAN controller must be able to synchronize to the relevant signal edge of the incoming signal. When an edge in the transmitted data is detected, the logic will compare the location of the edge to the expected time (Sync_Seg). The circuit will then adjust the values of Phase Segment 1 and Phase Segment 2 as necessary. There are two mechanisms used for synchronization.

27.10.1 HARD SYNCHRONIZATION

Hard synchronization is only done when there is a recessive to dominant edge during a bus Idle condition, indicating the start of a message. After hard synchronization, the bit time counters are restarted with Sync_Seg. Hard synchronization forces the edge, which has occurred to lie within the synchronization segment of the restarted bit time. Due to the rules of synchronization, if a hard synchronization occurs, there will not be a resynchronization within that bit time.

27.10.2 RESYNCHRONIZATION

As a result of resynchronization, Phase Segment 1 may be lengthened or Phase Segment 2 may be shortened. The amount of lengthening or shortening of the phase buffer segments has an upper bound given by the Synchronization Jump Width (SJW). The value of the SJW will be added to Phase Segment 1 (see Figure 27-6) or subtracted from Phase Segment 2 (see Figure 27-7). The SJW is programmable between 1 Tq and 4 Tq.

Clocking information will only be derived from recessive to dominant transitions. The property, that only a fixed maximum number of successive bits have the same value, ensures resynchronization to the bit stream during a frame. The phase error of an edge is given by the position of the edge relative to Sync_Seg, measured in Tq. The phase error is defined in magnitude of Tq as follows:

- e = 0 if the edge lies within Sync_Seg.
- e > 0 if the edge lies before the sample point.
- e < 0 if the edge lies after the sample point of the previous bit.

If the magnitude of the phase error is less than, or equal to, the programmed value of the Synchronization Jump Width, the effect of a resynchronization is the same as that of a hard synchronization.

If the magnitude of the phase error is larger than the Synchronization Jump Width and if the phase error is positive, then Phase Segment 1 is lengthened by an amount equal to the Synchronization Jump Width.

If the magnitude of the phase error is larger than the resynchronization jump width and if the phase error is negative, then Phase Segment 2 is shortened by an amount equal to the Synchronization Jump Width.

27.10.3 SYNCHRONIZATION RULES

- Only one synchronization within one bit time is allowed.
- An edge will be used for synchronization only if the value detected at the previous sample point (previously read bus value) differs from the bus value immediately after the edge.
- All other recessive to dominant edges fulfilling rules 1 and 2 will be used for resynchronization, with the exception that a node transmitting a dominant bit will not perform a resynchronization as a result of a recessive to dominant edge with a positive phase error.



FIGURE 27-6: LENGTHENING A BIT PERIOD (ADDING SJW TO PHASE SEGMENT 1)

27.15.6.1 Receiver Overflow

An overflow condition occurs when the MAB has assembled a valid received message (the message meets the criteria of the acceptance filters) and the receive buffer associated with the filter is not available for loading of a new message. The associated RXBnOVFL bit in the COMSTAT register will be set to indicate the overflow condition. This bit must be cleared by the MCU.

27.15.6.2 Receiver Warning

The receive error counter has reached the MCU warning limit of 96.

27.15.6.3 Transmitter Warning

The transmit error counter has reached the MCU warning limit of 96.

27.15.6.4 Receiver Bus Passive

This will occur when the device has gone to the errorpassive state because the receive error counter is greater or equal to 128.

27.15.6.5 Transmitter Bus Passive

This will occur when the device has gone to the errorpassive state because the transmit error counter is greater or equal to 128.

27.15.6.6 Bus-Off

The transmit error counter has exceeded 255 and the device has gone to bus-off state.

File	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300000h	CONFIG1L	_	XINST	—	SOSCSEL1	SOSCSEL0	INTOSCSEL	_	RETEN	-1-1 11-1
300001h	CONFIG1H	IESO	FCMEN	—	PLLCFG	FOSC3	FOSC2	FOSC1	FOSC0	00-0 1000
300002h	CONFIG2L		BORPWR1	BORWPR0	BORV1	BORV0	BOREN1	BOREN0	PWRTEN	-111 1111
300003h	CONFIG2H	_	WDTPS4	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN1	WDTEN0	-111 1111
300005h	CONFIG3H	MCLRE	—	—	—	MSSPMSK	T3CKMX ^(1,3)	T0CKMX ⁽¹⁾	CANMX	1 lqql
300006h	CONFIG4L	DEBUG	—	—	BBSIZ0	—	—	_	STVREN	111
300008h	CONFIG5L	_	—	—	_	CP3	CP2	CP1	CP0	1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—		—	11
30000Ah	CONFIG6L		_	—	_	WRT3	WRT2	WRT1	WRT0	1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—		—	111
30000Ch	CONFIG7L		—	—	—	EBTR3	EBTR2	EBTR1	EBTR0	1111
30000Dh	CONFIG7H	_	EBTRB	—	_	_	_	_	_	-1
3FFFFEh	DEVID1 ⁽²⁾	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx
3FFFFFh	DEVID2(2)	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	XXXX XXXX

TABLE 28-1: CONFIGURATION BITS AND DEVICE IDs

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

Note 1: Implemented only on the 64-pin devices (PIC18F6XK80).

2: See Register 28-13 for DEVID1 values. DEVID registers are read-only and cannot be programmed by the user.

3: Maintain as '0' on 28-pin, 40-pin and 44-pin devices.

REGISTER 28-7: CONFIG5L: CONFIGURATION REGISTER 5 LOW (BYTE ADDRESS 300008h)

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
—	—	—	—	CP3	CP2	CP1	CP0
bit 7							bit 0

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

bit 7-4	Unimplemented: Read as '0'
bit 3	CP3: Code Protection bit
	 1 = Block 3 is not code-protected⁽¹⁾ 0 = Block 3 is code-protected⁽¹⁾
bit 2	CP2: Code Protection bit
	 1 = Block 2 is not code-protected⁽¹⁾ 0 = Block 2 is code-protected⁽¹⁾
bit 1	CP1: Code Protection bit
	 1 = Block 1 is not code-protected⁽¹⁾ 0 = Block 1 is code-protected⁽¹⁾
bit 0	CP0: Code Protection bit 1 = Block 0 is not code-protected ⁽¹⁾ 0 = Block 0, is code-protected ⁽¹⁾

Note 1: For the memory size of the blocks, see Figure 28-6.

28-Lead Plastic Quad Flat, No Lead Package (MM) – 6x6x0.9 mm Body [QFN-S] with 0.40 mm Contact Length

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



	MILLIMETERS			
Dimension	MIN	NOM	MAX	
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W2			4.70
Optional Center Pad Length	T2			4.70
Contact Pad Spacing	C1		6.00	
Contact Pad Spacing	C2		6.00	
Contact Pad Width (X28)	X1			0.40
Contact Pad Length (X28)	Y1			0.85
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2124A