

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·XF

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I <sup>2</sup> C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f67k22-i-ptrsl

Email: info@E-XFL.COM

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

Din Marra	Pin Number	Pin	Buffer	Description
Pin Name	TQFP	Туре	Туре	Description
				PORTG is a bidirectional I/O port.
RG0/ECCP3/P3A RG0 ECCP3 P3A	5	I/O I/O O	ST ST	Digital I/O. Capture 3 input/Compare 3 output/PWM3 output. ECCP3 PWM Output A.
RG1/TX2/CK2/AN19/ C3OUT RG1 TX2 CK2 AN19 C3OUT	6	I/O O I/O I	ST — ST Analog	Digital I/O. EUSART asynchronous transmit. EUSART synchronous clock (see related RX2/DT2). Analog Input 19.
RG2/RX2/DT2/AN18/ C3INA RG2 RX2 DT2 AN18 C3INA	7	1/0 1 1/0 1	ST ST ST Analog Analog	Digital I/O. EUSART asynchronous receive. EUSART synchronous data (see related TX2/CK2). Analog Input 18. Comparator 3 Input A.
RG3/CCP4/AN17/P3D/ C3INB RG3 CCP4 AN17 P3D C3INB	8	I/O I/O I O I	ST ST Analog — Analog	Digital I/O. Capture 4 input/Compare 4 output/PWM4 output. Analog Input 17. ECCP3 PWM Output D. Comparator 3 Input B.
RG4/RTCC/T7CKI/T5G/ CCP5/AN16/P1D/C3INC RG4 RTCC T7CKI <sup>(3)</sup> T5G CCP5 AN16 P1D C3INC	10	I/O 0 1 1/O 1 0 1	ST — ST ST Analog — Analog	Digital I/O. RTCC output. Timer7 clock input. Timer5 external clock gate input. Capture 5 input/Compare 5 output/PWM5 output. Analog Input 16. ECCP1 PWM Output D. Comparator 3 Input C.
RG5	9 patible input			See the MCLR/RG5 pin.
ST = Schmitt T I = Input P = Power	rigger input wit	h CMC	)S levels	Analog = Analog input O = Output OD = Open-Drain (no P diode to VDD)

### TABLE 1-4: PIC18F8XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)

P = Power $I^{2}C = I^{2}C^{TM}/SMBus$ 

**Note 1:** Default assignment for ECCP2 when the CCP2MX Configuration bit is set.

**2:** Alternate assignment for ECCP2 when the CCP2MX Configuration bit is cleared.

- **3:** Not available on PIC18F65K22 and PIC18F85K22 devices.
- 4: PSP is available only in Microcontroller mode.

5: The CC6, CCP7, CCP8 and CCP9 pin placement depends on the setting of the ECCPMX Configuration bit (CONFIG3H<1>).

### 6.2 PIC18 Instruction Cycle

#### 6.2.1 CLOCKING SCHEME

The microcontroller clock input, whether from an internal or external source, is internally divided by four to generate four non-overlapping, quadrature clocks (Q1, Q2, Q3 and Q4). Internally, the Program Counter is incremented on every Q1, with the instruction fetched from the program memory and latched into the Instruction Register (IR) during Q4.

The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 6-4.

### 6.2.2 INSTRUCTION FLOW/PIPELINING

An "Instruction Cycle" consists of four Q cycles, Q1 through Q4. The instruction fetch and execute are pipelined in such a manner that a fetch takes one instruction cycle, while the decode and execute take another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction (such as GOTO) causes the Program Counter to change, two cycles are required to complete the instruction. (See Example 6-3.)

A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).



#### FIGURE 6-4: CLOCK/INSTRUCTION CYCLE

#### EXAMPLE 6-3: INSTRUCTION PIPELINE FLOW

	TCY0	TCY1	Tcy2	TCY3	TcY4	TCY5
1. MOVLW 55h	Fetch 1	Execute 1		_		
2. MOVWF PORTB		Fetch 2	Execute 2		_	
3. BRA SUB_1			Fetch 3	Execute 3		
4. BSF PORTA, BIT3 (1	Forced NOP)			Fetch 4	Flush (NOP)	
5. Instruction @ addres	ss SUB_1				Fetch SUB_1	Execute SUB_1

All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

### 6.4 Data Addressing Modes

Note:	The execution of some instructions in the
	core PIC18 instruction set are changed
	when the PIC18 extended instruction set is
	enabled. For more information, see
	Section 6.6 "Data Memory and the
	Extended Instruction Set"

While the program memory can be addressed in only one way, through the Program Counter, information in the data memory space can be addressed in several ways. For most instructions, the addressing mode is fixed. Other instructions may use up to three modes, depending on which operands are used and whether or not the extended instruction set is enabled.

The addressing modes are:

- Inherent
- Literal
- Direct
- Indirect

An additional addressing mode, Indexed Literal Offset, is available when the extended instruction set is enabled (XINST Configuration bit = 1). For details on this mode's operation, see **Section 6.6.1 "Indexed Addressing with Literal Offset**".

# 6.4.1 INHERENT AND LITERAL ADDRESSING

Many PIC18 control instructions do not need any argument at all. They either perform an operation that globally affects the device or they operate implicitly on one register. This addressing mode is known as Inherent Addressing. Examples of this mode include SLEEP, RESET and DAW.

Other instructions work in a similar way, but require an additional explicit argument in the opcode. This method is known as the Literal Addressing mode because the instructions require some literal value as an argument. Examples of this include ADDLW and MOVLW, which respectively, add or move a literal value to the W register. Other examples include CALL and GOTO, which include a 20-bit program memory address.

#### 6.4.2 DIRECT ADDRESSING

Direct Addressing specifies all or part of the source and/or destination address of the operation within the opcode itself. The options are specified by the arguments accompanying the instruction.

In the core PIC18 instruction set, bit-oriented and byte-oriented instructions use some version of Direct Addressing by default. All of these instructions include some 8-bit literal address as their Least Significant Byte. This address specifies the instruction's data source as either a register address in one of the banks of data RAM (see Section 6.3.3 "General Purpose Register File") or a location in the Access Bank (see Section 6.3.2 "Access Bank").

The Access RAM bit, 'a', determines how the address is interpreted. When 'a' is '1', the contents of the BSR (Section 6.3.1 "Bank Select Register") are used with the address to determine the complete 12-bit address of the register. When 'a' is '0', the address is interpreted as being a register in the Access Bank. Addressing that uses the Access RAM is sometimes also known as Direct Forced Addressing mode.

A few instructions, such as MOVFF, include the entire 12-bit address (either source or destination) in their opcodes. In these cases, the BSR is ignored entirely.

The destination of the operation's results is determined by the destination bit, 'd'. When 'd' is '1', the results are stored back in the source register, overwriting its original contents. When 'd' is '0', the results are stored in the W register. Instructions without the 'd' argument have a destination that is implicit in the instruction, either the target register being operated on or the W register.

### 6.4.3 INDIRECT ADDRESSING

Indirect Addressing allows the user to access a location in data memory without giving a fixed address in the instruction. This is done by using File Select Registers (FSRs) as pointers to the locations to be read or written to. Since the FSRs are themselves located in RAM as Special Function Registers, they can also be directly manipulated under program control. This makes FSRs very useful in implementing data structures such as tables and arrays in data memory.

The registers for Indirect Addressing are also implemented with Indirect File Operands (INDFs) that permit automatic manipulation of the pointer value with auto-incrementing, auto-decrementing or offsetting with another value. This allows for efficient code using loops, such as the example of clearing an entire RAM bank in Example 6-5. It also enables users to perform Indexed Addressing and other Stack Pointer operations for program memory in data memory.

#### EXAMPLE 6-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

	LFSR	FSR0, 100h	;	
NEXT	CLRF	POSTINC0	;	Clear INDF
			;	register then
			;	inc pointer
	BTFSS	FSROH, 1	;	All done with
			;	Bank1?
	BRA	NEXT	;	NO, clear next
CONTINU	JE		;	YES, continue

# 8.3 Wait States

While it may be assumed that external memory devices will operate at the microcontroller clock rate, this is often not the case. In fact, many devices require longer times to write or retrieve data than the time allowed by the execution of table read or table write operations.

To compensate for this, the External Memory Bus can be configured to add a fixed delay to each table operation using the bus. Wait states are enabled by setting the WAIT Configuration bit. When enabled, the amount of delay is set by the WAIT<1:0> bits (MEMCON<5:4>). The delay is based on multiples of microcontroller instruction cycle time and is added following the instruction cycle when the table operation is executed. The range is from no delay to 3 Tcy (default value).

# 8.4 Port Pin Weak Pull-ups

With the exception of the upper address lines, A<19:16>, the pins associated with the External Memory Bus are equipped with weak pull-ups. The pull-ups are controlled by the upper three bits of the PADCFG1 register (PADCFG1<7:5>). They are named RDPU, REPU and RJPU, and control pull-ups on PORTD, PORTE and PORTJ, respectively. Setting one of these bits enables the corresponding pull-ups for that port. All pull-ups are disabled by default on all device Resets.

In Extended Microcontroller mode, the port pull-ups can be useful in preserving the memory state on the external bus while the bus is temporarily disabled (EBDIS = 1).

# 8.5 Program Memory Modes and the External Memory Bus

The PIC18F87K22 family of devices is capable of operating in one of two program memory modes, using combinations of on-chip and external program memory. The functions of the multiplexed port pins depend on the program memory mode selected, as well as the setting of the EBDIS bit.

In **Microcontroller Mode**, the bus is not active and the pins have their port functions only. Writes to the MEMCOM register are not permitted. The Reset value of EBDIS ('0') is ignored and the ABW pins behave as I/O ports.

In **Extended Microcontroller Mode**, the external program memory bus shares I/O port functions on the pins. When the device is fetching or doing table read/table write operations on the external program memory space, the pins will have the external bus function.

If the device is fetching and accessing internal program memory locations only, the EBDIS control bit will change the pins from external memory to I/O port functions. When EBDIS = 0, the pins function as the external bus. When EBDIS = 1, the pins function as I/O ports.

If the device fetches or accesses external memory while EBDIS = 1, the pins will switch to the external bus. If the EBDIS bit is set by a program executing from external memory, the action of setting the bit will be delayed until the program branches into the internal memory. At that time, the pins will change from external bus to I/O ports.

If the device is executing out of internal memory when EBDIS = 0, the memory bus address/data and control pins will not be active. They will go to a state where the active address/data pins are tri-state, the  $\overline{CE}$ ,  $\overline{OE}$ , WRH, WRL, UB and LB signals are '1', and ALE and BA0 are '0'. Note that only those pins associated with the current address width are forced to tri-state; the other pins continue to function as I/O. In the case of 16-bit address width, for example, only AD<15:0> (PORTD and PORTE) are affected; A<19:16> (PORTH<3:0>) continue to function as I/O.

In all external memory modes, the bus takes priority over any other peripherals that may share pins with it. This includes the Parallel Master Port (PMP) and serial communication modules which would otherwise take priority over the I/O port.

# 8.6 16-Bit Data Width Modes

In 16-Bit Data Width mode, the external memory interface can be connected to external memories in three different configurations:

- 16-Bit Byte Write
- 16-Bit Word Write
- 16-Bit Byte Select

The configuration to be used is determined by the WM<1:0> bits in the MEMCON register (MEMCON<1:0>). These three different configurations allow the designer maximum flexibility in using both 8-bit and 16-bit devices with 16-bit data.

For all 16-bit modes, the Address Latch Enable (ALE) pin indicates that the Address bits, AD<15:0>, are available on the external memory interface bus. Following the address latch, the Output Enable (OE) signal will enable both bytes of program memory at once to form a 16-bit instruction word. The Chip Enable (CE signal) is active at any time that the microcontroller accesses external memory, whether reading or writing; it is inactive (asserted high) whenever the device is in Sleep mode.

In Byte Select mode, JEDEC standard Flash memories will require BA0 for the byte address line and one I/O line to select between Byte and Word mode. The other 16-bit modes do not need BA0. JEDEC standard static RAM memories will use the UB or LB signals for byte selection.

### 8.8 Operation in Power-Managed Modes

In alternate, power-managed Run modes, the external bus continues to operate normally. If a clock source with a lower speed is selected, bus operations will run at that speed. In these cases, excessive access times for the external memory may result if Wait states have been enabled and added to external memory operations. If operations in a lower power Run mode are anticipated, users should provide in their applications for adjusting memory access times at the lower clock speeds. In Sleep and Idle modes, the microcontroller core does not need to access data; bus operations are suspended. The state of the external bus is frozen, with the address/data pins, and most of the control pins, holding at the same state they were in when the mode was invoked. The only potential changes are to the  $\overline{CE}$ ,  $\overline{LB}$  and  $\overline{UB}$  pins, which are held at logic high.

TABLE 8-3:	REGISTERS ASSOCIATED WITH THE EXTERNAL MEMORY BUS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MEMCON <sup>(1)</sup>	EBDIS	_	WAIT1	WAIT0		—	WM1	WM0
PADCFG1	RDPU	REPU	RJPU <sup>(1)</sup>	_	—	RTSECSEL1	RTSECSEL0	—
PMD1	PSPMD	CTMUMD	RTCCMD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	EMBMD

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used during External Memory Bus access.

Note 1: Unimplemented in 64-pin devices (PIC18F6XK22), read as '0'.

### 11.3 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are six Peripheral Interrupt Enable registers (PIE1 through PIE6). When IPEN (RCON<7>) = 0, the PEIE bit must be set to enable any of these peripheral interrupts.

### REGISTER 11-10: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	TMR1GIE	TMR2IE	TMR1IE
bit 7							bit 0
Legend:							
R = Readabl	le 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
h:+ 7		lal Clava Dart F		www.mt Enchla hi			
DIC 7	1 - Enchlos t	he DSD read/w	rite interrunt	errupt Enable bi	L		
	0 = Disables 1	the PSP read/w	/rite interrupt				
bit 6	ADIE: A/D Co	onverter Interru	pt Enable bit				
	1 = Enables t	he A/D interrup	t				
	0 = Disables f	the A/D interrup	ot				
bit 5	RC1IE: EUSA	ART Receive In	terrupt Enable	bit			
	1 = Enables t	he EUSART re	ceive interrupt	•			
hit 4		RT Transmit In	terrunt Enable	, hit			
	1 = Enables t	he EUSART tra	ansmit interrup	t			
0 = Disables the EUSART transmit interrup				ot			
bit 3	SSP1IE: Master Synchronous Serial Port Interrupt Enable bit						
	1 = Enables the MSSP interrupt						
	0 = Disables the MSSP interrupt						
bit 2	TMR1GIE: TMR1 Gate Interrupt Enable bit						
1 = Enables the gate							
hit 1	v = Disableu (i)e yale						
1 = Enables the TMR2 to PR2 match interrupt							
	0 = Disables t	the TMR2 to PF	R2 match inter	rupt			
bit 0	TMR1IE: TMF	R1 Overflow Int	errupt Enable	bit			
	1 = Enables t	he TMR1 overf	low interrupt				
	0 = Disables f	the TMR1 over	flow interrupt				

# 12.2 PORTA, TRISA and LATA Registers

PORTA is an 8-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISA and LATA.

RA4/T0CKI is a Schmitt Trigger input. All other PORTA pins have TTL input levels and full CMOS output drivers.

RA5 and RA<3:0> are multiplexed with analog inputs for the A/D Converter.

The operation of the analog inputs as A/D Converter inputs is selected by clearing or setting the ANSEL control bits in the ANCON1 register. The corresponding TRISA bits control the direction of these pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

Note:	RA5 and RA<3:0> are configured as
	analog inputs on any Reset and are read
	as '0'. RA4 is configured as a digital input.

OSC2/CLKO/RA6 and OSC1/CLKI/RA7 normally serve as the external circuit connections for the external (primary) oscillator circuit (HS Oscillator modes), or the external clock input and output (EC Oscillator modes). In these cases, RA6 and RA7 are not available as digital I/O and their corresponding TRIS and LAT bits are read as '0'. When the device is configured to use HF-INTOSC, MF-INTOSC or LF-INTOSC as the default oscillator mode, RA6 and RA7 are automatically configured as digital I/O; the oscillator and clock in/clock out functions are disabled.

RA5 has additional functionality for Timer1 and Timer3. It can be configured as the Timer1 clock input or the Timer3 external clock gate input.

EXAMPLE 12-1:	<b>INITIALIZING PORTA</b>

CLRF	PORTA	; Initialize PORTA by
		; clearing output latches
CLRF	LATA	; Alternate method to
		; clear output data latches
BANKSEL	ANCON1	; Select bank with ANCON1 register
MOVLW	00h	; Configure A/D
MOVWF	ANCON1	; for digital inputs
BANKSEL	TRISA	; Select bank with TRISA register
MOVLW	0BFh	; Value used to initialize
		; data direction
MOVWF	TRISA	; Set RA<7, 5:0> as inputs,
		; RA<6> as output

# PIC18F87K22 FAMILY

# 12.11 Parallel Slave Port

PORTD can function as an 8-bit-wide Parallel Slave Port (PSP), or microprocessor port, when control bit, PSPMODE (PSPCON<4>), is set. The port is asynchronously readable and writable by the external world through the RD control input pin (RE0/P2D/RD/AD8) and WR control input pin (RE1/P2C/WR/AD9).

Note:	The Parallel Slave Port is available only in
	Microcontroller mode.

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an eight-bit latch.

Setting bit, PSPMODE, enables port pin, RE0/P2D/RD/AD8, to be the RD input, RE1/P2C/WR/AD9 to be the WR input and RE2/P2B/CCP10/CS/AD10 to be the CS (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (= 111).

A write to the PSP occurs when both the  $\overline{CS}$  and  $\overline{WR}$  lines are first detected low and ends when either are detected high. The PSPIF and IBF flag bits (PIR1<7> and PSPCON<7>, respectively) are set when the write ends.

A read from the PSP occurs when both the  $\overline{CS}$  and  $\overline{RD}$  lines are first detected low. The data in PORTD is read out and the OBF bit (PSPCON<6>) is set. If the user writes new data to PORTD to set OBF, the data is immediately read out, but the OBF bit is not set.

When either the  $\overline{CS}$  or  $\overline{RD}$  line is detected high, the PORTD pins return to the input state and the PSPIF bit is set. User applications should wait for PSPIF to be set before servicing the PSP. When this happens, the IBF and OBF bits can be polled and the appropriate action taken.

The timing for the control signals in Write and Read modes is shown in Figure 12-4 and Figure 12-5, respectively.

### FIGURE 12-3: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)



# 14.0 TIMER1 MODULE

The Timer1 timer/counter module incorporates these features:

- Software-selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR1H and TMR1L)
- Selectable clock source (internal or external) with device clock or SOSC oscillator internal options
- Interrupt-on-overflow
- Reset on ECCP Special Event Trigger
- · Timer with gated control

Figure 14-1 displays a simplified block diagram of the Timer1 module.

The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power-managed operation. The Timer1 can also work on the SOSC oscillator.

Timer1 is controlled through the T1CON Control register (Register 14-1). It also contains the Secondary Oscillator Enable bit (SOSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR10N (T1CON<0>).

The FOSC clock source should not be used with the ECCP capture/compare features. If the timer will be used with the capture or compare features, always select one of the other timer clocking options.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	SOSCEN	T1SYNC	RD16	TMR10N
bit 7							bit 0

### REGISTER 14-1: T1CON: TIMER1 CONTROL REGISTER

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	TMR1CS<1:0>: Timer1 Clock Source Select bits10 = Timer1 clock source is either from a pin or oscillator, depending on the SOSCEN bit:SOSCEN = 0:External clock from the T1CKI pin (on the rising edge).SOSCEN = 1:Depending on the SOSCSEL Configuration bit, the clock source is either a crystal oscillator on theSOSCI/SOSCO pins or an internal clock from the SCLKI pin.01 = Timer1 clock source is the system clock (Fosc) <sup>(1)</sup> 00 = Timer1 clock source is the instruction clock (Fosc/4)
bit 5-4	<b>T1CKPS&lt;1:0&gt;:</b> Timer1 Input Clock Prescale Select bits 11 = 1:8 Prescale value 10 = 1:4 Prescale value 01 = 1:2 Prescale value 00 = 1:1 Prescale value
bit 3	<b>SOSCEN:</b> SOSC Oscillator Enable bit 1 = SOSC is enabled and available for Timer1 0 = SOSC is disabled for Timer1 The oscillator inverter and feedback resistor are turned off to eliminate power drain.
bit 2	T1SYNC: Timer1 External Clock Input Synchronization Select bit         TMR1CS<1:0> = 10:         1 = Do not synchronize external clock input         0 = Synchronize external clock input         TMR1CS<1:0> = 0x:         This bit is ignored. Timer1 uses the internal clock when TMR1CS<1:0> = 1x.
bit 1	<b>RD16:</b> 16-Bit Read/Write Mode Enable bit 1 = Enables register read/write of Timer1 in one 16-bit operation 0 = Enables register read/write of Timer1 in two 8-bit operations
bit 0	TMR1ON: Timer1 On bit 1 = Enables Timer1 0 = Stops Timer1

Note 1: The Fosc clock source should not be selected if the timer will be used with the ECCP capture/compare features.

# PIC18F87K22 FAMILY

NOTES:

# 18.1 RTCC MODULE REGISTERS

The RTCC module registers are divided into the following categories:

# **RTCC Control Registers**

- RTCCFG
- RTCCAL
- PADCFG1
- ALRMCFG
- ALRMRPT

# **RTCC Value Registers**

- RTCVALH
- RTCVALL

Both registers access the following registers:

- YEAR
- MONTH
- DAY
- WEEKDAY
- HOUR
- MINUTE
- SECOND

# **Alarm Value Registers**

- ALRMVALH
- ALRMVALL
   Both registers access the following registers:
  - ALRMMNTH
  - ALRMDAY
  - ALRMWD
  - ALRMHR
  - ALRMMIN
  - ALRMSEC
- Note: The RTCVALH and RTCVALL registers can be accessed through RTCRPT<1:0> (RTCCFG<1:0>). ALRMVALH and ALRMVALL can be accessed through ALRMPTR<1:0> (ALRMCFG<1:0>).

# 19.4 PWM Mode

In Pulse-Width Modulation (PWM) mode, the CCP4 pin produces up to a 10-bit resolution PWM output. Since the CCP4 pin is multiplexed with a PORTC or PORTE data latch, the appropriate TRIS bit must be cleared to make the CCP4 pin an output.

Note:	Clearing the CCP4CON register will force
	the RC1 or RE7 output latch (depending
	on device configuration) to the default low
	level. This is not the PORTC or PORTE
	I/O data latch.

Figure 19-3 shows a simplified block diagram of the ECCP1 module in PWM mode.

For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 19.4.3** "Setup for PWM Operation".





A PWM output (Figure 19-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

### FIGURE 19-4: PWM OUTPUT



# 19.4.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

### EQUATION 19-1:



PWM frequency is defined as 1/[PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- · The CCP4 pin is set

(An exception: If PWM duty cycle = 0%, the CCP4 pin will not be set)

 The PWM duty cycle is latched from CCPR4L into CCPR4H

Note:	The	Timer2	postscalers	(see			
	Sectior	n 15.0 "Tin	ner2 Module") a	are not			
	used in	the deter	mination of the	PWM			
	frequen	quency. The postscaler could be used					
	to have	a servo up	date rate at a di	fferent			
	frequen	cy than the	PWM output.				

# PIC18F87K22 FAMILY

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	
PIR2	OSCFIF	—	SSP2IF	BCL2IF	BCL1IF	HLVDIF	TMR3IF	TMR3GIF	
PIE2	OSCFIE	—	SSP2IE	BCL2IE	BCL1IE	HLVDIE	TMR3IE	TMR3GIE	
IPR2	OSCFIP	—	SSP2IP	BCL2IP	BCL1IP	HLVDIP	TMR3IP	TMR3GIP	
PIR3	TMR5GIF	—	RC2IF	TX2IF	CTMUIF	CCP2IF	CCP1IF	RTCCIF	
PIE3	TMR5GIE	—	RC2IE	TX2IE	CTMUIE	CCP2IE	CCP1IE	RTCCIE	
IPR3	TMR5GIP	—	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP	RTCCIP	
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	
SSP1BUF	MSSP1 Red	ceive Buffer/T	ransmit Reg	ister					
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	
SSP1STAT	SMP	CKE	D/A	Р	S	R/W	UA	BF	
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	
SSP2STAT	SMP	CKE	D/A	Р	S	R/W	UA	BF	
SSP2BUF	MSSP2 Receive Buffer/Transmit Register								
ODCON1	SSP10D	CCP2OD	CCP10D					SSP2OD	
PMD0	CCP3MD	CCP2MD	CCP1MD	UART2MD	UART1MD	SSP2MD	SSP1MD	ADCMD	

### TABLE 21-2: REGISTERS ASSOCIATED WITH SPI OPERATION

**Legend:** Shaded cells are not used by the MSSP module in SPI mode.

	SYNC = 0, BRGH = 0, BRG16 = 0											
BAUD	Fosc	= 40.000	) MHz	Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	_								_		_	
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	_	_	_
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	_	_
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	_	_	_

TABLE 22-3:	<b>BAUD RATES F</b>	FOR ASYNCHRONOUS M	ODES

	SYNC = 0, BRGH = 0, BRG16 = 0											
BAUD	Fosc = 4.000 MHz			Fos	c = 2.000	MHz	Fosc = 1.000 MHz					
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)			
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51			
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12			
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—			
9.6	8.929	-6.99	6	—	_	_	—	_	_			
19.2	20.833	8.51	2	—	_	_	—	_	_			
57.6	62.500	8.51	0	—	_	_	—	_	_			
115.2	62.500	-45.75	0	_	—		_	—	—			

	SYNC = 0, BRGH = 1, BRG16 = 0											
BAUD	Fosc	= 40.000	) MHz	Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	_	_	_	_	_	_	_	_	_	_	_	_
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	_	_

	<b>SYNC</b> = 0, <b>BRGH</b> = 1, <b>BRG16</b> = 0										
BAUD	Fost	c = 4.000	MHz	Fos	c = 2.000	MHz	Fosc = 1.000 MHz				
(K)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)		
0.3		_	_	_	_	_	0.300	-0.16	207		
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51		
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25		
9.6	9.615	0.16	25	9.615	-0.16	12	_	_	_		
19.2	19.231	0.16	12	—	_	_	_	_	_		
57.6	62.500	8.51	3	—	—	—	—	_	_		
115.2	125.000	8.51	1	—	—	—	—	—	_		

 $\ensuremath{\textcircled{}^{\circ}}$  2009-2011 Microchip Technology Inc.

### 22.1.3 AUTO-BAUD RATE DETECT

The Enhanced USART module supports the automatic detection and calibration of baud rate. This feature is active only in Asynchronous mode and while the WUE bit is clear.

The automatic baud rate measurement sequence (Figure 22-1) begins whenever a Start bit is received and the ABDEN bit is set. The calculation is self-averaging.

In the Auto-Baud Rate Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RXx signal, the RXx signal is timing the BRG. In ABD mode, the internal Baud Rate Generator is used as a counter to time the bit period of the incoming serial byte stream.

Once the ABDEN bit is set, the state machine will clear the BRG and look for a Start bit. The Auto-Baud Rate Detect must receive a byte with the value, 55h (ASCII "U", which is also the LIN/J2602 bus Sync character), in order to calculate the proper bit rate. The measurement is taken over both a low and a high bit time in order to minimize any effects caused by asymmetry of the incoming signal. After a Start bit, the SPBRGx begins counting up, using the preselected clock source on the first rising edge of RXx. After eight bits on the RXx pin or the fifth rising edge, an accumulated value totalling the proper BRG period is left in the SPBRGHx:SPBRGx register pair. Once the 5th edge is seen (this should correspond to the Stop bit), the ABDEN bit is automatically cleared.

If a rollover of the BRG occurs (an overflow from FFFFh to 0000h), the event is trapped by the ABDOVF status bit (BAUDCONx<7>). It is set in hardware by BRG rollovers and can be set or cleared by the user in software. ABD mode remains active after rollover events and the ABDEN bit remains set (Figure 22-2).

While calibrating the baud rate period, the BRG registers are clocked at 1/8th the preconfigured clock rate. The BRG clock will be configured by the BRG16 and BRGH bits. The BRG16 bit must be set to use both SPBRG1 and SPBRGH1 as a 16-bit counter. This allows the user to verify that no carry occurred for 8-bit modes by checking for 00h in the SPBRGHx register. Refer to Table 22-4 for counter clock rates to the BRG.

While the ABD sequence takes place, the EUSART state machine is held in Idle. The RCxIF interrupt is set once the fifth rising edge on RXx is detected. The value in the RCREGx needs to be read to clear the RCxIF interrupt. The contents of RCREGx should be discarded.

- Note 1: If the WUE bit is set with the ABDEN bit, Auto-Baud Rate Detection will occur on the byte *following* the Break character.
  - 2: It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible due to bit error rates. Overall system timing and communication baud rates must be taken into consideration when using the Auto-Baud Rate Detection feature.
  - To maximize baud rate range, if that feature is used, it is recommended that the BRG16 bit (BAUDCONx<3>) be set.

# TABLE 22-4:BRG COUNTER<br/>CLOCK RATES

BRG16	BRGH	BRG Counter Clock
0	0	Fosc/512
0	1	Fosc/128
1	0	Fosc/128
1	1	Fosc/32

### 22.1.3.1 ABD and EUSART Transmission

Since the BRG clock is reversed during ABD acquisition, the EUSART transmitter cannot be used during ABD. This means that whenever the ABDEN bit is set, TXREGx cannot be written to. Users should also ensure that ABDEN does not become set during a transmit sequence. Failing to do this may result in unpredictable EUSART operation.

# 22.2.5 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN/J2602 bus standard. The Break character transmit consists of a Start bit, followed by twelve '0' bits and a Stop bit. The Frame Break character is sent whenever the SENDB and TXEN bits (TXSTAx<3> and TXSTAx<5>, respectively) are set while the Transmit Shift Register is loaded with data. Note that the value of data written to TXREGx will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN/J2602 specification).

Note that the data value written to the TXREGx for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 22-10 for the timing of the Break character sequence.

### 22.2.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN/J2602 bus master.

- 1. Configure the EUSART for the desired mode.
- 2. Set the TXEN and SENDB bits to set up the Break character.
- 3. Load the TXREGx with a dummy character to initiate transmission (the value is ignored).
- 4. Write '55h' to TXREGx to load the Sync character into the transmit FIFO buffer.
- 5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREGx becomes empty, as indicated by the TXxIF, the next data byte can be written to TXREGx.

### 22.2.6 RECEIVING A BREAK CHARACTER

The Enhanced USART module can receive a Break character in two ways.

The first method forces configuration of the baud rate at a frequency of 9/13 the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in **Section 22.2.4** "**Auto-Wake-up on Sync Break Character**". By enabling this feature, the EUSART will sample the next two transitions on RXx/DTx, cause an RCxIF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABDEN bit once the TXxIF interrupt is observed.



# FIGURE 22-10: SEND BREAK CHARACTER SEQUENCE

# REGISTER 28-5: CONFIG3L: CONFIGURATION REGISTER 3 LOW (BYTE ADDRESS 300004h)

U-1	U-1	U-1	U-1	U-1	U-0	U-0	R/P-1
WAIT <sup>(1)</sup>	BW <sup>(1)</sup>	ABW1 <sup>(1)</sup>	ABW0 <sup>(1)</sup>	EASHFT <sup>(1)</sup>	—	—	RTCOSC
bit 7							bit 0

Legend:	P = Programmable bit		
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	WAIT: External Bus Wait Enable bit <sup>(1)</sup>						
	<ul><li>1 = Wait states on the external bus are disabled</li><li>0 = Wait states on the external bus are enabled and selected by MEMCON &lt;5:4&gt;</li></ul>						
bit 6	BW: Data Bus Width Select bit <sup>(1)</sup>						
	1 = 16-Bit Data Width modes 0 = 8-Bit Data Width modes						
bit 5-4	ABW<1:0>: External Memory Bus Configuration bits <sup>(1)</sup>						
	11 = 8-Bit Address mode (Microcontroller mode)						
	10 = 12-Bit Address mode						
	01 = 16-Bit Address mode						
	00 = 20-Bit Address mode (4)						
bit 3	EASHFT: External Address Bus Shift Enable bit <sup>(1)</sup>						
	<ul> <li>1 = Address shifting is enabled; external address is shifted to start at 000000h</li> <li>0 = Address shifting is disabled; external address bus reflects the PC value</li> </ul>						
bit 2-1	Unimplemented: Read as '0'						
bit 0	RTCOSC: RTCC Reference Clock Select bit						
	1 = RTCC uses SOSC as the reference clock						
	0 = RTCC uses LF-INTOSC as the reference clock						

Note 1: Unimplemented on 64-pin devices (PIC18F6XK22), read as '0'.

# PIC18F87K22 FAMILY

DEC	FSZ	SZ Decrement f, Skip if 0		DCFS	SNZ	Decrement f, Skip if Not 0					
Syn	ax:	DECFSZ f	{,d {,a}}		Synta	IX:	DCFSNZ	f {,d {,a}}			
Ope	rands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$			Opera	ands:	$\begin{array}{l} 0 \leq f \leq 255 \\ d  \in  [0,1] \\ a  \in  [0,1] \end{array}$				
Operation: $(f) - 1 \rightarrow dest,$ skip if result = 0		Opera	ation:	(f) – $1 \rightarrow d$ skip if resu	est, It ≠ 0						
Stat	us Affected:	None			Status	s Affected:	None				
Enc	oding:	0010	11da ff:	ff ffff	Enco	ding:	0100	11da fff	f ffff		
Description:		The conten decremente placed in W placed back	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'.			ription:	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'.				
		If the result which is alr and a NOP i it a two-cyc	is '0', the nex eady fetched s executed in le instruction.	t instruction is discarded stead, making			If the result instruction discarded a instead, ma instruction.	t is not '0', the which is alread and a NOP is e aking it a two-c	next ly fetched is kecuted ycle		
		If 'a' is '1', th GPR bank.	ne BSR is use	d to select the				If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.			
		set is enabl in Indexed I mode when Section 29 Bit-Oriente Literal Offs	Literal Offset $\lambda$ ever f $\leq$ 95 (5 .2.3 "Byte-Or d Instruction set Mode" for	ction operates Addressing Fh). See iented and is in Indexed details.				If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed			
Wor	ds:	1					Literal Off	set Mode" for	details.		
Cycles:		1(2) <b>Note:</b> 3 cy by a	1(2) Note: 3 cycles if skip and followed by a 2-word instruction.			s: s:	1 1(2) <b>Note:</b> 3 cycles if skip and followed				
QC	Cycle Activity:				0.0		by	a 2-word instri	uction.		
	Q1	Q2	Q3	Q4	ູ ບູບ <sub>ູ</sub>		02	03	04		
	Decode	register 'f'	Data	destination	ſ	Decode	Read	Process	Write to		
lf sl	kip:	- 0					register 'f'	Data	destination		
	Q1	Q2	Q3	Q4	If ski	p:					
	No	No	No	No	Г	Q1	Q2	Q3	Q4		
lf o	operation	operation	operation	operation		N0 operation	N0 operation	NO	N0 operation		
11 51			$\Omega^2 \qquad \Omega^3 \qquad \Omega^4$			If skip and followed by 2-word instruction:					
	No	No	No	No	1	Q1	, Q2	Q3	Q4		
	operation	operation	operation	operation		No	No	No	No		
	No	No	No	No		operation	operation	operation	operation		
	operation	operation	operation	operation		No operation	No operation	No operation	No operation		
<u>Exa</u>	<u>mple:</u>	HERE CONTINUE	DECFSZ GOTO	CNT, 1, 1 LOOP	Exam	iple:	HERE ZERO	DCFSNZ TEM	IP, 1, 0		
Before Instruction						NZERO	:				
PC = Address (HERE) After Instruction		1	Before Instruc TEMP After Instructio	tion =	?						
	If CNT	= 0;	I		,	TEMP	=	TEMP – 1,			
		= Address	(CONTINUE	])		If TEMP	=	0; Address (r			
If CNT PC		= Address	$\neq 0;$ = Address (HERE + 2)			If TEMP $\neq$ 0; PC = Address (ZERO)			NZERO)		

# 31.5.3 TIMING DIAGRAMS AND SPECIFICATIONS



### TABLE 31-6: EXTERNAL CLOCK TIMING REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Мах	Units	Conditions
1A	Fosc	External CLKIN Frequency <sup>(1)</sup>	DC	64	MHz	EC, ECIO Oscillator mode -40°C ≤ TA ≤ +85°C
			DC	48	MHz	-40°C ≤ TA ≤ +125°C
		Oscillator Frequency <sup>(1)</sup>	DC	4	MHz	RC Oscillator mode
			0.1	4	MHz	XT Oscillator mode
			4	16	MHz	HS Oscillator mode
			4	16	MHz	HS + PLL Oscillator mode
			5	33	kHz	LP Oscillator mode
1	Tosc	External CLKIN Period <sup>(1)</sup>	15.6	—	ns	EC, ECIO Oscillator mode
		Oscillator Period <sup>(1)</sup>	250	—	ns	RC Oscillator mode
			250	10,000	ns	XT Oscillator mode
			40 62.5	250 250	ns ns	HS Oscillator mode HS + PLL Oscillator mode
			5	200	μS	LP Oscillator mode
2	Тсү	Instruction Cycle Time <sup>(1)</sup>	62.5	_	ns	Tcy = 4/Fosc
3	TosL,	External Clock in (OSC1)	30		ns	XT Oscillator mode
	TosH	High or Low Time	2.5	—	μS	LP Oscillator mode
			10	—	ns	HS Oscillator mode
4	TosR,	External Clock in (OSC1)		20	ns	XT Oscillator mode
	TosF	Rise or Fall Time	—	50	ns	LP Oscillator mode
			—	7.5	ns	HS Oscillator mode

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time base period for all configurations except PLL. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin. When an external clock input is used, the "max." cycle time limit is "DC" (no clock) for all devices.

# 64-Lead Plastic Quad Flat, No Lead Package (MR) – 9x9x0.9 mm Body [QFN]

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



Microchip Technology Drawing C04-149B Sheet 1 of 2