Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

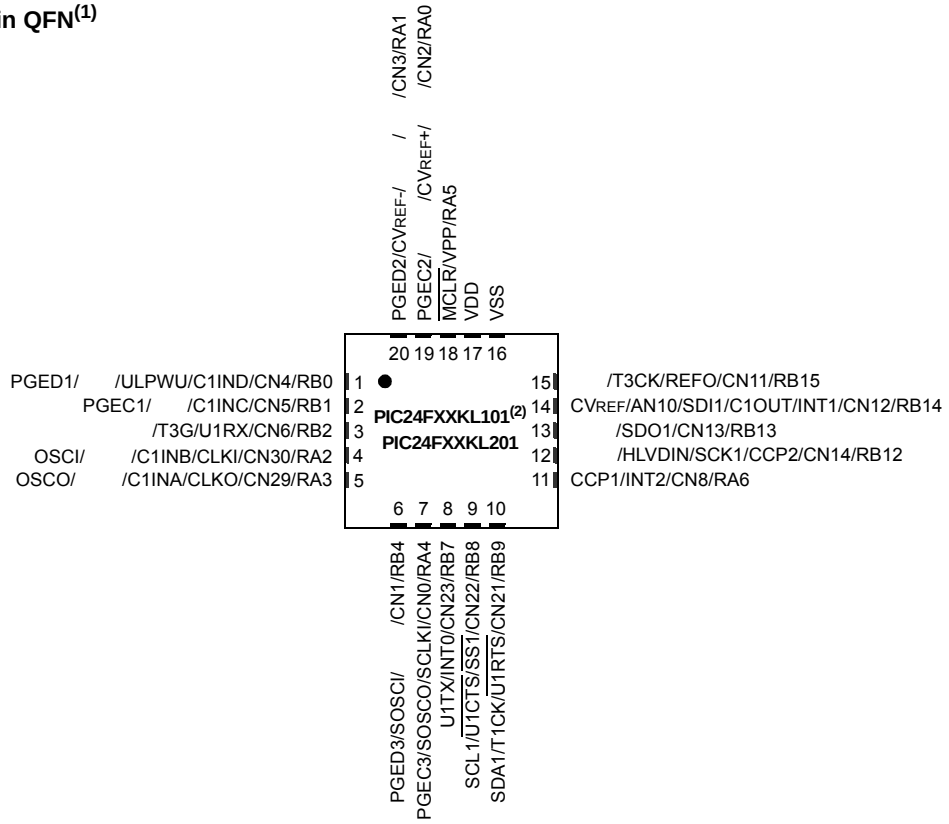
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

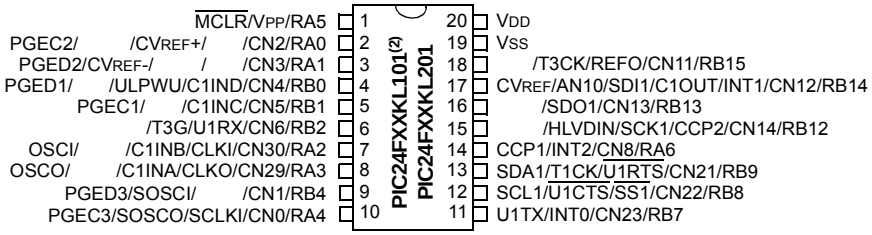
Details	
Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	8KB (2.75K x 24)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f08kl302-e-ss

Pin Diagrams: PIC24FXXKL10X/20X

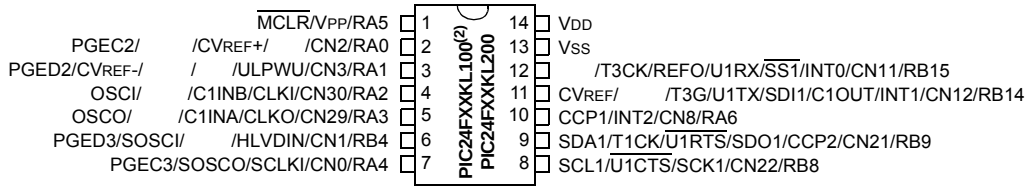
20-Pin QFN(1)



20-Pin PDIP/SSOP/SOIC(1)



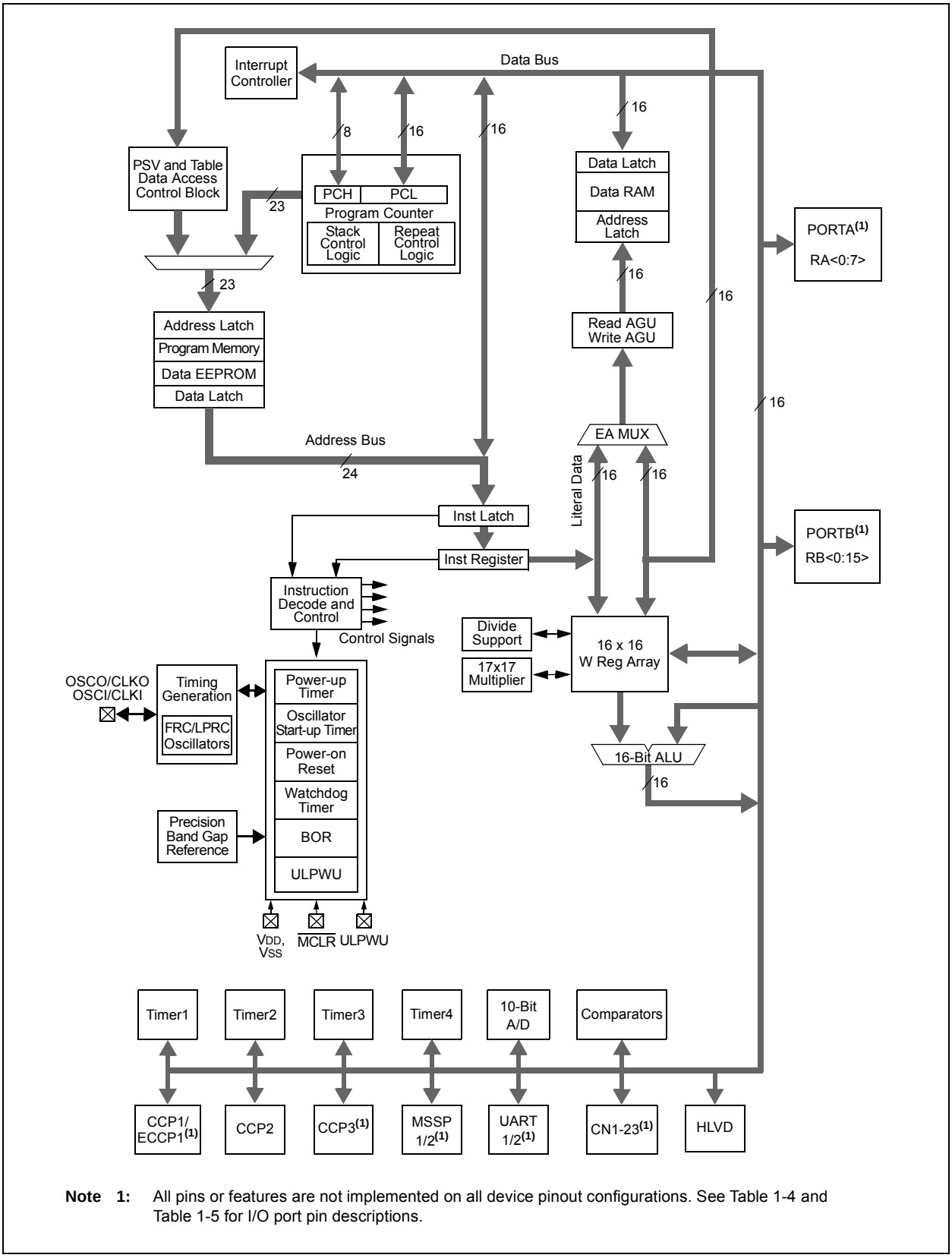
14-Pin PDIP/TSSOP(1)



Note 1: Analog features (indicated in) are not available on PIC24FXXKL100/101 devices.

2: Alternate location for I^2C™ functionality of MSSP1, as determined by the I2C1SEL Configuration bit.

FIGURE 1-1: PIC24F16KL402 FAMILY GENERAL BLOCK DIAGRAM



Note 1: All pins or features are not implemented on all device pinout configurations. See Table 1-4 and Table 1-5 for I/O port pin descriptions.

TABLE 1-4: PIC24F16KL40X/30X FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number				I/O	Buffer	Description
	20-Pin PDIP/ SSOP/ SOIC	20-Pin QFN	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN			
PGEC1	5	2	5	2	I/O	ST	ICSP™ Clock 1
PCED1	4	1	4	1	I/O	ST	ICSP Data 1
PGEC2	2	19	22	19	I/O	ST	ICSP Clock 2
PGED2	3	20	21	18	I/O	ST	ICSP Data 2
PGEC3	10	7	15	12	I/O	ST	ICSP Clock 3
PGED3	9	6	14	11	I/O	ST	ICSP Data 3
RA0	2	19	2	27	I/O	ST	PORTA Pins
RA1	3	20	3	28	I/O	ST	
RA2	7	4	9	6	I/O	ST	
RA3	8	5	10	7	I/O	ST	
RA4	10	7	12	9	I/O	ST	
RA5	1	18	1	26	I	ST	
RA6	14	11	20	17	I/O	ST	
RA7	—	—	19	16	I/O	ST	
RB0	4	1	4	1	I/O	ST	PORTB Pins
RB1	5	2	5	2	I/O	ST	
RB2	6	3	6	3	I/O	ST	
RB3	—	—	7	4	I/O	ST	
RB4	9	6	11	8	I/O	ST	
RB5	—	—	14	11	I/O	ST	
RB6	—	—	15	12	I/O	ST	
RB7	11	8	16	13	I/O	ST	
RB8	12	9	17	14	I/O	ST	
RB9	13	10	18	15	I/O	ST	
RB10	—	—	21	18	I/O	ST	
RB11	—	—	22	19	I/O	ST	
RB12	15	12	23	20	I/O	ST	
RB13	16	13	24	21	I/O	ST	
RB14	17	14	25	22	I/O	ST	
RB15	18	15	26	23	I/O	ST	
REFO	18	15	26	23	O	—	Reference Clock Output
SCK1	15	12	22	19	I/O	ST	MSSP1 SPI Serial Input/Output Clock
SCK2	18	15	14	11	I/O	ST	MSSP2 SPI Serial Input/Output Clock
SCL1	12	9	17	14	I/O	I^2C	MSSP1 I^2C Clock Input/Output
SCL2	18	15	7	4	I/O	I^2C	MSSP2 I^2C Clock Input/Output
SCLKI	10	7	12	9	I	ST	Digital Secondary Clock Input
SDA1	13	10	18	15	I/O	I^2C	MSSP1 I^2C Data Input/Output
SDA2	2	19	2	27	I/O	I^2C	MSSP2 I^2C Data Input/Output
SDI1	17	14	21	18	I	ST	MSSP1 SPI Serial Data Input
SDI2	2	19	19	16	I	ST	MSSP2 SPI Serial Data Input
SDO1	16	13	24	21	O	—	MSSP1 SPI Serial Data Output
SDO2	3	20	15	12	O	—	MSSP2 SPI Serial Data Output

Legend: TTL = TTL input buffer
ANA = Analog level input/output
ST = Schmitt Trigger input buffer
I^2C = I^2C™/SMBus input buffer

EXAMPLE 5-4: LOADING THE WRITE BUFFERS – 'C' LANGUAGE CODE

```
// C example using MPLAB C30

  #define NUM_INSTRUCTION_PER_ROW 64
   int __attribute__ ((space(auto_psv))) progAddr = &progAddr; // Global variable located in Pgm Memory
  unsigned int offset;
  unsigned int i;
  unsigned int progData[2*NUM_INSTRUCTION_PER_ROW];          // Buffer of data to write

  //Set up NVMCON for row programming
  NVMCON = 0x4004;                                          // Initialize NVMCON

  //Set up pointer to the first memory location to be written
  TBLPAG = __builtin_tblpage(&progAddr);                    // Initialize PM Page Boundary SFR
  offset = &progAddr & 0xFFFF;                              // Initialize lower word of address

  //Perform TBLWT instructions to write necessary number of latches
  for(i=0; i < 2*NUM_INSTRUCTION_PER_ROW; i++)
  {
      __builtin_tblwtl(offset, progData[i++]);              // Write to address low word
      __builtin_tblwth(offset, progData[i]);                // Write to upper byte
      offset = offset + 2;                                  // Increment address
   }
```

EXAMPLE 5-5: INITIATING A PROGRAMMING SEQUENCE – ASSEMBLY LANGUAGE CODE

```
        DISI    #5                      ; Block all interrupts
                                          for next 5 instructions
        MOV     #0x55, W0
        MOV     W0, NVMKEY              ; Write the 55 key
        MOV     #0xAA, W1               ;
        MOV     W1, NVMKEY              ; Write the AA key
        BSET    NVMCON, #WR             ; Start the erase sequence
        NOP                             ; 2 NOPs required after setting WR
        NOP                             ;
        BTSC    NVMCON, #15             ; Wait for the sequence to be completed
        BRA     $-2                     ;
```

EXAMPLE 5-6: INITIATING A PROGRAMMING SEQUENCE – 'C' LANGUAGE CODE

```
// C example using MPLAB C30

asm("DISI #5");              // Block all interrupts for next 5 instructions

__builtin_write_NVM();       // Perform unlock sequence and set WR
```

REGISTER 8-4: INTCON2: INTERRUPT CONTROL REGISTER2

R/W-0	R-0, HSC	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	INT2EP	INT1EP	INT0EP
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
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 15 **ALTIVT:** Enable Alternate Interrupt Vector Table bit
1 = Uses Alternate Interrupt Vector Table
0 = Uses standard (default) vector table

bit 14 **DISI:** DISI Instruction Status bit
1 = DISI instruction is active
0 = DISI instruction is not active

bit 13-3 **Unimplemented:** Read as '0'

bit 2 **INT2EP:** External Interrupt 2 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 1 **INT1EP:** External Interrupt 1 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 0 **INT0EP:** External Interrupt 0 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)

bit 7 **CLKLOCK:** Clock Selection Lock Enable bit
If FSCM is Enabled (FCKSM1 = 1):
1 = Clock and PLL selections are locked
0 = Clock and PLL selections are not locked and may be modified by setting the OSWEN bit
If FSCM is Disabled (FCKSM1 = 0):
Clock and PLL selections are never locked and may be modified by setting the OSWEN bit.

bit 6 **Unimplemented:** Read as '0'

bit 5 **LOCK:** PLL Lock Status bit(2)
1 = PLL module is in lock or the PLL module start-up timer is satisfied
0 = PLL module is out of lock, the PLL start-up timer is running or PLL is disabled

bit 4 **Unimplemented:** Read as '0'

bit 3 **CF:** Clock Fail Detect bit
1 = FSCM has detected a clock failure
0 = No clock failure has been detected

bit 2 **SOSCDRV**: Secondary Oscillator Drive Strength bit(3)
1 = High-power SOSC circuit is selected
0 = Low/high-power select is done via the SOSCSRC Configuration bit

bit 1 **SOSCEN:** 32 kHz Secondary Oscillator (SOSC) Enable bit
1 = Enables secondary oscillator
0 = Disables secondary oscillator

bit 0 **OSWEN:** Oscillator Switch Enable bit
1 = Initiates an oscillator switch to the clock source specified by the NOSC<2:0> bits
0 = Oscillator switch is complete

Note 1: Reset values for these bits are determined by the FNOSC<2:0> Configuration bits.
2: Also resets to '0' during any valid clock switch or whenever a non-PLL Clock mode is selected.
3: When SOSC is selected to run from a digital clock input rather than an external crystal (SOSCSRC = 0), this bit has no effect.

10.4 Doze Mode

Generally, changing clock speed and invoking one of the power-saving modes are the preferred strategies for reducing power consumption. There may be circumstances, however, where this is not practical. For example, it may be necessary for an application to maintain uninterrupted, synchronous communication, even while it is doing nothing else. Reducing system clock speed may introduce communication errors, while using a power-saving mode may stop communications completely.

Doze mode is a simple and effective alternative method to reduce power consumption while the device is still executing code. In this mode, the system clock continues to operate from the same source and at the same speed. Peripheral modules continue to be clocked at the same speed, while the CPU clock speed is reduced. Synchronization between the two clock domains is maintained, allowing the peripherals to access the SFRs while the CPU executes code at a slower rate.

Doze mode is enabled by setting the DOZEN bit (CLKDIV<11>). The ratio between peripheral and core clock speed is determined by the DOZE<2:0> bits (CLKDIV<14:12>). There are eight possible configurations, from 1:1 to 1:128, with 1:1 being the default.

It is also possible to use Doze mode to selectively reduce power consumption in event driven applications. This allows clock-sensitive functions, such as synchronous communications, to continue without interruption. Meanwhile, the CPU Idles, waiting for something to invoke an interrupt routine. Enabling the automatic return to full-speed CPU operation on interrupts is enabled by setting the ROI bit (CLKDIV<15>). By default, interrupt events have no effect on Doze mode operation.

10.5 Selective Peripheral Module Control

Idle and Doze modes allow users to substantially reduce power consumption by slowing or stopping the CPU clock. Even so, peripheral modules still remain clocked and thus, consume power. There may be cases where the application needs what these modes do not provide: the allocation of power resources to CPU processing, with minimal power consumption from the peripherals.

PIC24F devices address this requirement by allowing peripheral modules to be selectively disabled, reducing or eliminating their power consumption. This can be done with two control bits:

- The Peripheral Enable bit, generically named, “XXXEN”, located in the module’s main control SFR.
- The Peripheral Module Disable (PMD) bit, generically named, “XXXMD”, located in one of the PMD Control registers.

Both bits have similar functions in enabling or disabling its associated module. Setting the PMD bit for a module disables all clock sources to that module, reducing its power consumption to an absolute minimum. In this state, the control and status registers associated with the peripheral will also be disabled, so writes to those registers will have no effect, and read values will be invalid. Many peripheral modules have a corresponding PMD bit.

In contrast, disabling a module by clearing its XXXEN bit, disables its functionality, but leaves its registers available to be read and written to. Power consumption is reduced, but not by as much as when the PMD bits are used.

To achieve more selective power savings, peripheral modules can also be selectively disabled when the device enters Idle mode. This is done through the control bit of the generic name format, “XXXIDL”. By default, all modules that can operate during Idle mode will do so. Using the disable on Idle feature disables the module while in Idle mode, allowing further reduction of power consumption during Idle mode. This enhances power savings for extremely critical power applications.

11.3 Input Change Notification

The Input Change Notification (ICN) function of the I/O ports allows the PIC24F16KL402 family of devices to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins. This feature is capable of detecting input Change-of-States, even in Sleep mode, when the clocks are disabled. Depending on the device pin count, there are up to 23 external signals that may be selected (enabled) for generating an interrupt request on a Change-of-State.

There are six control registers associated with the Change Notification (CN) module. The CNEN1 and CNEN2 registers contain the interrupt enable control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CN pin also has a weak pull-up/pull-down connected to it. The pull-ups act as a current source that is connected to the pin. The pull-downs act as a current sink to eliminate the need for external resistors when push button or keypad devices are connected.

On any pin, only the pull-up resistor or the pull-down resistor should be enabled, but not both of them. If the push button or the keypad is connected to V_{DD}, enable the pull-down, or if they are connected to V_{SS}, enable the pull-up resistors. The pull-ups are enabled separately using the CNPU1 and CNPU2 registers, which contain the control bits for each of the CN pins.

Setting any of the control bits enables the weak pull-ups for the corresponding pins. The pull-downs are enabled separately, using the CNPD1 and CNPD2 registers, which contain the control bits for each of the CN pins. Setting any of the control bits enables the weak pull-downs for the corresponding pins.

When the internal pull-up is selected, the pin uses V_{DD} as the pull-up source voltage. When the internal pull-down is selected, the pins are pulled down to V_{SS} by an internal resistor. Make sure that there is no external pull-up source/pull-down sink when the internal pull-ups/pull-downs are enabled.

Note:	Pull-ups and pull-downs on Change Notification pins should always be disabled whenever the port pin is configured as a digital output.

EXAMPLE 11-1: PORT WRITE/READ EXAMPLE (ASSEMBLY LANGUAGE)

```
MOV     #0xFF00, W0        ; Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
MOV     W0, TRISB
MOV     #0x00FF, W0        ; Enable PORTB<15:8> digital input buffers
MOV     W0, ANSB
NOP                        ; Delay 1 cycle
BTSS    PORTB, #13         ; Next Instruction
```

EXAMPLE 11-2: PORT WRITE/READ EXAMPLE (C LANGUAGE)

```
TRISB = 0xFF00;            // Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
ANSB = 0x00FF;             // Enable PORTB<15:8> digital input buffers
NOP();                     // Delay 1 cycle
if(PORTBbits.RB13 == 1)    // execute following code if PORTB pin 13 is set.
{
}
```

REGISTER 16-3: ECCP1AS: ECCP1 AUTO-SHUTDOWN CONTROL REGISTER(1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0
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 15-8 **Unimplemented:** Read as '0'

bit 7 **ECCPASE:** ECCP1 Auto-Shutdown Event Status bit
1 = A shutdown event has occurred; ECCP outputs are in a shutdown state
0 = ECCP outputs are operating

bit 6-4 **ECCPAS<2:0>:** ECCP1 Auto-Shutdown Source Select bits
111 = VIL on $\overline{FLT0}$ pin, or either C1OUT or C2OUT is high
110 = VIL on $\overline{FLT0}$ pin or C2OUT comparator output is high
101 = VIL on $\overline{FLT0}$ pin or C1OUT comparator output is high
100 = VIL on $\overline{FLT0}$ pin
011 = Either C1OUT or C2OUT is high
010 = C2OUT comparator output is high
001 = C1OUT comparator output is high
000 = Auto-shutdown is disabled

bit 3-2 **PSSAC<1:0>:** P1A and P1C Pins Shutdown State Control bits
1x = P1A and P1C pins tri-state
01 = Drive pins, P1A and P1C, to '1'
00 = Drive pins, P1A and P1C, to '0'

bit 1-0 **PSSBD<1:0>:** P1B and P1D Pins Shutdown State Control bits
1x = P1B and P1D pins tri-state
01 = Drive pins, P1B and P1D, to '1'
00 = Drive pins, P1B and P1D, to '0'

Note 1: This register is implemented only on PIC24FXXKL40X/30X devices.

Note 1: The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.
2: Writing to the ECCPASE bit is disabled while an auto-shutdown condition persists.
3: Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart), the PWM signal will always restart at the beginning of the next PWM period.

REGISTER 17-10: PADCFG1: PAD CONFIGURATION CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	SDO2DIS(1)	SCK2DIS(1)	SDO1DIS	SCK1DIS
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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 15-12 **Unimplemented:** Read as '0'

bit 11 **SDO2DIS:** MSSP2 SDO2 Pin Disable bit(1)
- 1 = The SPI output data (SDO2) of MSSP2 to the pin is disabled
- 0 = The SPI output data (SDO2) of MSSP2 is output to the pin

bit 10 **SCK2DIS:** MSSP2 SCK2 Pin Disable bit(1)
- 1 = The SPI clock (SCK2) of MSSP2 to the pin is disabled
- 0 = The SPI clock (SCK2) of MSSP2 is output to the pin

bit 9 **SDO1DIS:** MSSP1 SDO1 Pin Disable bit
- 1 = The SPI output data (SDO1) of MSSP1 to the pin is disabled
- 0 = The SPI output data (SDO1) of MSSP1 is output to the pin

bit 8 **SCK1DIS:** MSSP1 SCK1 Pin Disable bit
- 1 = The SPI clock (SCK1) of MSSP1 to the pin is disabled
- 0 = The SPI clock (SCK1) of MSSP1 is output to the pin

bit 7-0 **Unimplemented:** Read as '0'

Note 1: These bits are implemented only on PIC24FXXKL40X/30X devices.

NOTES:

19.0 10-BIT HIGH-SPEED A/D CONVERTER

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the 10-Bit High-Speed A/D Converter, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"10-Bit A/D Converter"** (DS39705).

The 10-bit A/D Converter has the following key features:

- Successive Approximation (SAR) conversion
- Conversion speeds of up to 500 ksps
- Up to 12 analog input pins
- External voltage reference input pins
- Internal band gap reference input
- Automatic Channel Scan mode
- Selectable conversion trigger source
- Two-word conversion result buffer
- Selectable Buffer Fill modes
- Four result alignment options
- Operation during CPU Sleep and Idle modes

Depending on the particular device, PIC24F16KL402 family devices implement up to 12 analog input pins, designated AN0 through AN4 and AN9 through AN15. In addition, there are two analog input pins for external voltage reference connections (V_{REF}+ and V_{REF}-). These voltage reference inputs may be shared with other analog input pins.

A block diagram of the A/D Converter is displayed in Figure 19-1.

To perform an A/D conversion:

1. Configure the A/D module:
   a) Configure port pins as analog inputs and/or select band gap reference inputs (ANSA<3:0>, ANSB<15:12,4:0> and ANCFG<0>).
   b) Select the voltage reference source to match the expected range on analog inputs (AD1CON2<15:13>).
   c) Select the analog conversion clock to match the desired data rate with the processor clock (AD1CON3<7:0>).
   d) Select the appropriate sample/conversion sequence (AD1CON1<7:5> and AD1CON3<12:8>).
   e) Select how conversion results are presented in the buffer (AD1CON1<9:8>).
   f) Select interrupt rate (AD1CON2<5:2>).
   g) Turn on A/D module (AD1CON1<15>).
2. Configure A/D interrupt (if required):
   a) Clear the AD1IF bit.
   b) Select A/D interrupt priority.

NOTES:

REGISTER 23-3: FOSCSEL: OSCILLATOR SELECTION CONFIGURATION REGISTER

R/P-1	R/P-1	R/P-1	U-0	U-0	R/P-0	R/P-0	R/P-1
IESO	LPRCSEL	SOSCSRC	—	—	FNOSC2	FNOSC1	FNOSC0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable 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 **IESO:** Internal External Switchover bit
1 = Internal External Switchover mode is enabled (Two-Speed Start-up is enabled)
0 = Internal External Switchover mode is disabled (Two-Speed Start-up is disabled)

bit 6 **LPRCSEL:** Internal LPRC Oscillator Power Select bit
1 = High-Power/High-Accuracy mode
0 = Low-Power/Low-Accuracy mode

bit 5 **SOSCSRC:** Secondary Oscillator Clock Source Configuration bit
1 = SOSC analog crystal function is available on the SOSCI/SOSCO pins
0 = SOSC crystal is disabled; digital SCLKI function is selected on the SOSCO pin

bit 4-3 **Unimplemented:** Read as '0'

bit 2-0 **FNOSC<2:0>:** Oscillator Selection bits
111 = 8 MHz FRC Oscillator with Divide-by-N (FRCDIV)
110 = 500 kHz Low-Power FRC Oscillator with Divide-by-N (LPFRCDIV)
101 = Low-Power RC Oscillator (LPRC)
100 = Secondary Oscillator (SOSC)
011 = Primary Oscillator with PLL module (HS+PLL, EC+PLL)
010 = Primary Oscillator (XT, HS, EC)
001 = 8 MHz FRC Oscillator with Divide-by-N with PLL module (FRCDIV+PLL)
000 = 8 MHz FRC Oscillator (FRC)

REGISTER 23-9: DEVREV: DEVICE REVISION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R	R	R	R
—	—	—	—	REV3	REV2	REV1	REV0
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

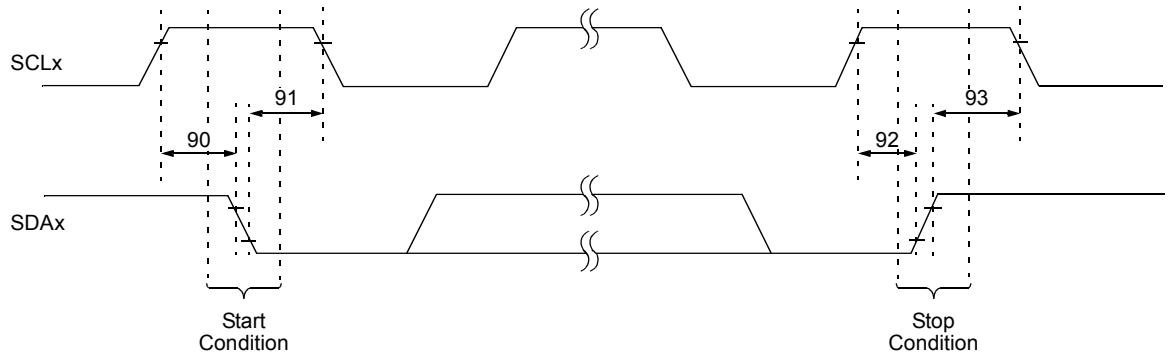
bit 23-4 **Unimplemented:** Read as ‘0’

bit 3-0 **REV<3:0>:** Revision Identifier bits

TABLE 25-1: SYMBOLS USED IN OPCODE DESCRIPTIONS

Field	Description
#text	Means literal defined by "text"
(text)	Means "content of text"
[text]	Means "the location addressed by text"
{ }	Optional field or operation
<n:m>	Register bit field
.b	Byte mode selection
.d	Double-Word mode selection
.S	Shadow register select
.w	Word mode selection (default)
bit4	4-bit bit selection field (used in word addressed instructions) ∈ {0...15}
C, DC, N, OV, Z	MCU Status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address ∈ {0000h...1FFFh}
lit1	1-bit unsigned literal ∈ {0,1}
lit4	4-bit unsigned literal ∈ {0...15}
lit5	5-bit unsigned literal ∈ {0...31}
lit8	8-bit unsigned literal ∈ {0...255}
lit10	10-bit unsigned literal ∈ {0...255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal ∈ {0...16384}
lit16	16-bit unsigned literal ∈ {0...65535}
lit23	23-bit unsigned literal ∈ {0...8388608}; LSB must be '0'
None	Field does not require an entry, may be blank
PC	Program Counter
Slit10	10-bit signed literal ∈ {-512...511}
Slit16	16-bit signed literal ∈ {-32768...32767}
Slit6	6-bit signed literal ∈ {-16...16}
Wb	Base W register ∈ {W0..W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor Working register pair (direct addressing)
Wn	One of 16 Working registers ∈ {W0..W15}
Wnd	One of 16 destination Working registers ∈ {W0..W15}
Wns	One of 16 source Working registers ∈ {W0..W15}
WREG	W0 (Working register used in File register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws--], [++Ws], [--Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns--], [++Wns], [--Wns], [Wns+Wb] }

FIGURE 26-13: MSSPx I²C™ BUS START/STOP BITS TIMING WAVEFORMS



Note: Refer to Figure 26-3 for load conditions.

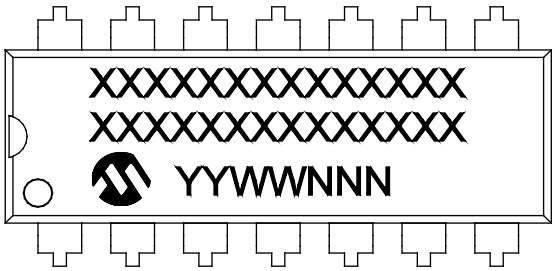
TABLE 26-33: I²C™ BUS START/STOP BITS REQUIREMENTS (MASTER MODE)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
90	TSU:STA	Start Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	2(TOSC)(BRG + 1)	—		
91	THD:STA	Start Condition Hold Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	2(TOSC)(BRG + 1)	—		
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	
			400 kHz mode	2(TOSC)(BRG + 1)	—		
93	THD:STO	Stop Condition Hold Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	
			400 kHz mode	2(TOSC)(BRG + 1)	—		

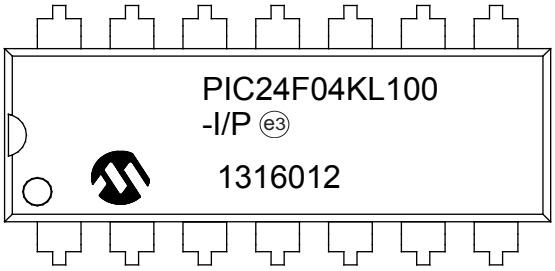
27.0 PACKAGING INFORMATION

27.1 Package Marking Information

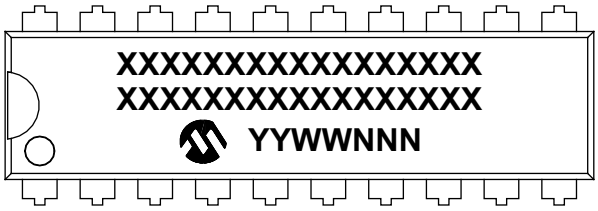
14-Lead PDIP (300 mil)



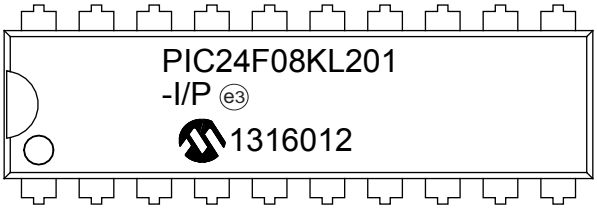
Example



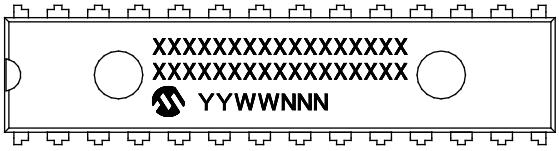
20-Lead PDIP (300 mil)



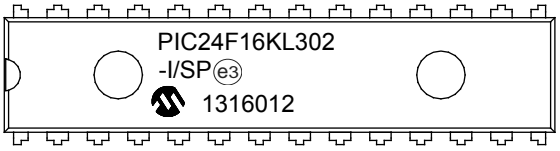
Example



28-Lead SPDIP (.300")



Example



Legend:	XX...X	Product-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

APPENDIX A: REVISION HISTORY

Revision A (September 2011)

Original data sheet for the PIC24F16KL402 family of devices.

Revision B (November 2011)

Updates DC Specifications in Tables 26-6 through 26-9 (all Typical and Maximum values).

Updates AC Specifications in Tables 26-7 through 26-30 (SPI Timing Requirements) with the addition of the FSCK specification.

Other minor typographic corrections throughout.

Revision C (October 2013)

Adds +125°C Extended Temperature information.

Updates several packaging drawings in **Section 27.0 "Packaging Information"**. Other minor typographic corrections throughout.

APPENDIX B: MIGRATING FROM PIC18/PIC24 TO PIC24F16KL402

The PIC24F16KL402 family combines traditional PIC18 peripherals with a faster PIC24 core to provide a low-cost, high-performance microcontroller with low-power consumption.

Code written for PIC18 devices can be migrated to the PIC24F16KL402 by using a C compiler that generates PIC24 machine level instructions. Assembly language code will need to be rewritten using PIC24 instructions. The PIC24 instruction set shares similarities to the PIC18 instruction set, which should ease porting of assembly code. Application code will require changes to support certain PIC24 peripherals.

Code written for PIC24 devices can be migrated to the PIC24F16KL402 without many code changes. Certain peripherals, however, will require application changes to support modules that were traditionally available only on PIC18 devices.

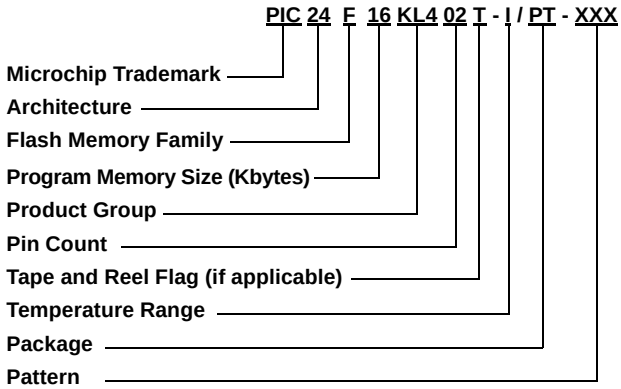
Refer to Table B-1 for a list of peripheral modules on the PIC24F16KL402 and where they originated from.

TABLE B-1: TABLE B-1: PIC24F16KL402 PERIPHERAL MODULE ORIGINATING ARCHITECTURE

Peripheral Module	PIC18	PIC24
ECCP/CCP	X	—
MSSP (I^2C™/SPI)	X	—
Timer2/4 (8-bit)	X	—
Timer3 (16-bit)	X	—
Timer1 (16-bit)	—	X
10-Bit A/D Converter	—	X
Comparator	—	X
Comparator Voltage Reference	—	X
UART	—	X
HLVD	—	X

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Architecture	24	= 16-bit modified Harvard without DSP
Flash Memory Family	F	= Standard voltage range Flash program memory
Product Group	KL4 KL3 KL2 KL1	= General purpose microcontrollers
Pin Count	00 01 02	= 14-pin = 20-pin = 28-pin
Temperature Range	I E	= -40°C to +85°C (Industrial) = -40°C to +125°C (Extended)
Package	SP SO SS ST ML, MQ P	= SPDIP = SOIC = SSOP = TSSOP = QFN = PDIP
Pattern	Three-digit QTP, SQTP, Code or Special Requirements (blank otherwise) ES	= Engineering Sample

Examples:

a) PIC24F16KL402-I/ML: General Purpose, 16-Kbyte Program Memory, 28-Pin, Industrial Temperature, QFN Package

b) PIC24F04KL101T-I/SS: General Purpose, 4-Kbyte Program Memory, 20-Pin, Industrial Temperature, SSOP Package, Tape-and-Reel