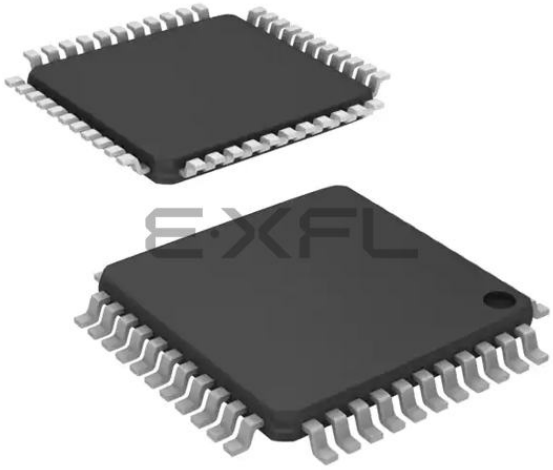


Welcome to E-XFL.COM

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, LVD, POR, PWM, WDT
Number of I/O	38
Program Memory Size	16KB (5.5K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 22x10b/12b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f16km204-e-pt

TABLE 1-5: PIC24FV16KM204 FAMILY PINOUT DESCRIPTION (CONTINUED)

Function	F					FV					I/O	Buffer	Description
	Pin Number					Pin Number							
	20-Pin PDIP/ SSOP/ SOIC	28-Pin PDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN	20-Pin PDIP/ SSOP/ SOIC	28-Pin PDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN			
RB9	13	18	15	1	1	13	18	15	1	1	I/O	ST	PORTB Pins
RB10	—	21	18	8	9	—	21	18	8	9	I/O	ST	PORTB Pins
RB11	—	22	19	9	10	—	22	19	9	10	I/O	ST	PORTB Pins
RB12	15	23	20	10	11	15	23	20	10	11	I/O	ST	PORTB Pins
RB13	16	24	21	11	12	16	24	21	11	12	I/O	ST	PORTB Pins
RB14	17	25	22	14	15	17	25	22	14	15	I/O	ST	PORTB Pins
RB15	18	26	23	15	16	18	26	23	15	16	I/O	ST	PORTB Pins
RC0	—	—	—	25	27	—	—	—	25	27	I/O	ST	PORTC Pins
RC1	—	—	—	26	28	—	—	—	26	28	I/O	ST	PORTC Pins
RC2	—	—	—	27	29	—	—	—	27	29	I/O	ST	PORTC Pins
RC3	—	—	—	36	39	—	—	—	36	39	I/O	ST	PORTC Pins
RC4	—	—	—	37	40	—	—	—	37	40	I/O	ST	PORTC Pins
RC5	—	—	—	38	41	—	—	—	38	41	I/O	ST	PORTC Pins
RC6	—	—	—	2	2	—	—	—	2	2	I/O	ST	PORTC Pins
RC7	—	—	—	3	3	—	—	—	3	3	I/O	ST	PORTC Pins
RC8	—	—	—	4	4	—	—	—	4	4	I/O	ST	PORTC Pins
RC9	—	—	—	5	5	—	—	—	5	5	I/O	ST	PORTC Pins
REFO	18	26	23	15	16	18	26	23	15	16	O	—	Reference Clock Output
RTCC	—	25	22	14	15	—	25	22	14	15	O	—	Real-Time Clock/Calendar Output
SCK1	15	22	19	9	10	15	22	19	9	10	I/O	ST	MSSP1 SPI Clock
SDI1	17	21	18	8	9	17	21	18	8	9	I	ST	MSSP1 SPI Data Input
SDO1	16	24	21	11	12	16	24	21	11	12	O	—	MSSP1 SPI Data Output
$\overline{SS1}$	18	26	23	15	16	18	26	23	15	16	I	ST	MSSP1 SPI Slave Select Input
SCK2	—	14	11	38	41	—	14	11	38	41	I/O	ST	MSSP2 SPI Clock
SDI2	—	19	16	36	39	—	19	16	36	39	I	ST	MSSP2 SPI Data Input
SDO2	—	15	12	37	40	—	15	12	37	40	O	—	MSSP2 SPI Data Output
$\overline{SS2}$	—	23	20	35	38	—	23	20	35	38	I	ST	MSSP2 SPI Slave Select Input

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

TABLE 4-9: MCCP2 REGISTER MAP

File Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CCP2CON1L	164h	CCPON	—	CCPSIDL	r	TMRSYNC	CLKSEL2	CLKSEL1	CLKSEL0	TMRPS1	TMRPS0	T32	CCSEL	MOD3	MOD2	MOD1	MOD0	0000
CCP2CON1H	166h	OPSSRC	RTRGEN	—	—	IOPS3	IOPS2	IOPS1	IOPS0	TRIGEN	ONESHOT	ALTSYNC	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	0000
CCP2CON2L	168h	PWMRSEN	ASDGM	—	SSDG	—	—	—	—	ASDG7	ASDG6	ASDG5	ASDG4	ASDG3	ASDG2	ASDG1	ASDG0	0000
CCP2CON2H	16Ah	OENSYNC	—	OCFEN(1)	OCEEN(1)	OCDEN(1)	OCCEN(1)	OCBEN(1)	OCAEN	ICGSM1	ICGSM0	—	AUXOUT1	AUXOUT0	ICSEL2	ICSEL1	ICSEL0	0100
CCP2CON3L	16Ch	—	—	—	—	—	—	—	—	—		DT5	DT4	DT3	DT2	DT1	DT0	0000
CCP2CON3H	16Eh	OETRIG	OSCNT2	OSCNT1	OSCNT0	—	OUTM2(1)	OUTM1(1)	OUTM0(1)	—	—	POLACE	POLBDF(1)	PSSACE1	PSSACE0	PSSBDF1(1)	PSSBDF0(1)	0000
CCP2STATL	170h	—	—	—	—	—	—	—	—	CCPTRIG	TRSET	TRCLR	ASEVT	SCEVT	ICDIS	ICOV	ICBNE	0000
CCP2TMRL	174h	MCCP2 Time Base Register Low Word																0000
CCP2TMRH	176h	MCCP2 Time Base Register High Word																0000
CCP2PRL	178h	MCCP2 Time Base Period Register Low Word																FFFF
CCP2PRH	17Ah	MCCP2 Time Base Period Register High Word																FFFF
CCP2RAL	17Ch	Output Compare 2 Data Word A																0000
CCP2RBL	180h	Output Compare 2 Data Word B																0000
CCP2BUFL	184h	Input Capture 2 Data Buffer Low Word																0000
CCP2BUFH	186h	Input Capture 2 Data Buffer High Word																0000

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition, r = reserved.

Note 1: These bits are available only on PIC24F(V)16KM2XX devices.

5.5.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

The user can program one row of Flash program memory at a time by erasing the programmable row. The general process is:

1. Read a row of program memory (32 instructions) and store in data RAM.
2. Update the program data in RAM with the desired new data.
3. Erase a row (see Example 5-1):
   a) Set the NVMOPx bits (NVMCON<5:0>) to '011000' to configure for row erase. Set the ERASE (NVMCON<6>) and WREN (NVMCON<14>) bits.
   b) Write the starting address of the block to be erased into the TBLPAG and W registers.
   c) Write 55h to NVMKEY.
   d) Write AAh to NVMKEY.
   e) Set the WR bit (NVMCON<15>). The erase cycle begins and the CPU stalls for the duration of the erase cycle. When the erase is done, the WR bit is cleared automatically.
4. Write the first 32 instructions from data RAM into the program memory buffers (see Example 5-1).
5. Write the program block to Flash memory:
   a) Set the NVMOPx bits to '000100' to configure for row programming. Clear the ERASE bit and set the WREN bit.
   b) Write 55h to NVMKEY.
   c) Write AAh to NVMKEY.
   d) Set the WR bit. The programming cycle begins and the CPU stalls for the duration of the write cycle. When the write to Flash memory is done, the WR bit is cleared automatically.

For protection against accidental operations, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be NOPs, as displayed in Example 5-5.

EXAMPLE 5-1: ERASING A PROGRAM MEMORY ROW – ASSEMBLY LANGUAGE CODE

```
; Set up NVMCON for row erase operation
        MOV     #0x4058, W0                 ;
        MOV     W0, NVMCON                  ; Initialize NVMCON
; Init pointer to row to be ERASED
        MOV     #tblpage(PROG_ADDR), W0     ;
        MOV     W0, TBLPAG                  ; Initialize PM Page Boundary SFR
        MOV     #tbloffset(PROG_ADDR), W0   ; Initialize in-page EA[15:0] pointer
        TBLWTL  W0, [W0]                    ; Set base address of erase block
        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                                 ; Insert two NOPs after the erase
        NOP                                 ; command is asserted
```

EXAMPLE 5-2: ERASING A PROGRAM MEMORY ROW – 'C' LANGUAGE CODE

```
// C example using MPLAB C30

int __attribute__ ((space(auto_psv))) progAddr = 0x1234;    // Variable located in Pgm Memory, declared as a
                                                            // global variable
    unsigned int offset;

//Set up pointer to the first memory location to be written

    TBLPAG = __builtin_tblpage(&progAddr);                  // Initialize PM Page Boundary SFR
    offset = __builtin_tbloffset(&progAddr);                // Initialize lower word of address

    __builtin_tblwtl(offset, 0x0000);                       // Set base address of erase block
                                                            // with dummy latch write

    NVMCON = 0x4058;                                        // Initialize NVMCON

    asm("DISI #5");                                         // Block all interrupts for next 5 instructions
    __builtin_write_NVM();                                  // C30 function to perform unlock
                                                            // sequence and set WR
```

EXAMPLE 5-3: LOADING THE WRITE BUFFERS – ASSEMBLY LANGUAGE CODE

```
; Set up NVMCON for row programming operations
        MOV     #0x4004, W0                 ;
        MOV     W0, NVMCON                  ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
        MOV     #0x0000, W0                 ;
        MOV     W0, TBLPAG                  ; Initialize PM Page Boundary SFR
        MOV     #0x1500, W0                 ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
        MOV     #LOW_WORD_0, W2             ;
        MOV     #HIGH_BYTE_0, W3            ;
        TBLWTL  W2, [W0]                    ; Write PM low word into program latch
        TBLWTH  W3, [W0++]                  ; Write PM high byte into program latch
; 1st_program_word
        MOV     #LOW_WORD_1, W2             ;
        MOV     #HIGH_BYTE_1, W3            ;
        TBLWTL  W2, [W0]                    ; Write PM low word into program latch
        TBLWTH  W3, [W0++]                  ; Write PM high byte into program latch
; 2nd_program_word
        MOV     #LOW_WORD_2, W2             ;
        MOV     #HIGH_BYTE_2, W3            ;
        TBLWTL  W2, [W0]                    ; Write PM low word into program latch
        TBLWTH  W3, [W0++]                  ; Write PM high byte into program latch
        •
        •
        •
; 32nd_program_word
        MOV     #LOW_WORD_31, W2            ;
        MOV     #HIGH_BYTE_31, W3           ;
        TBLWTL  W2, [W0]                    ; Write PM low word into program latch
        TBLWTH  W3, [W0]                    ; Write PM high byte into program latch
```

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 = 0x1234     // 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 = __builtin_tbloffset(&progAddr);                 // 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
   }
```

6.4.3 READING THE DATA EEPROM

To read a word from data EEPROM, the Table Read instruction is used. Since the EEPROM array is only 16 bits wide, only the TBLRDL instruction is needed. The read operation is performed by loading TBLPAG and WREG with the address of the EEPROM location, followed by a TBLRDL instruction.

A typical read sequence, using the Table Pointer management (builtin_tblpage and builtin_tbloffset) and Table Read (builtin_tblrdl) procedures from the C30 compiler library, is provided in Example 6-5.

Program Space Visibility (PSV) can also be used to read locations in the data EEPROM.

EXAMPLE 6-5: READING THE DATA EEPROM USING THE TBLRD COMMAND

```
int __attribute__ ((space(eedata))) eeData = 0x1234;
int data;                                               // Data read from EEPROM
/*-----------------------------------------------------------------------------------------------
The variable eeData must be a Global variable declared outside of any method

the code following this comment can be written inside the method that will execute the read
-----------------------------------------------------------------------------------------------
*/
    unsigned int offset;

    // Set up a pointer to the EEPROM location to be erased
    TBLPAG = __builtin_tblpage(&eeData);                // Initialize EE Data page pointer
    offset = __builtin_tbloffset(&eeData);              // Initizlize lower word of address
    data = __builtin_tblrdl(offset);                    // Write EEPROM data to write latch
```

TABLE 8-1: TRAP VECTOR DETAILS

Vector Number	IVT Address	AIVT Address	Trap Source
0	000004h	000104h	Reserved
1	000006h	000106h	Oscillator Failure
2	000008h	000108h	Address Error
3	00000Ah	00010Ah	Stack Error
4	00000Ch	00010Ch	Math Error
5	00000Eh	00010Eh	Reserved
6	000010h	000110h	Reserved
7	000012h	000112h	Reserved

TABLE 8-2: IMPLEMENTED INTERRUPT VECTORS

Interrupt Source	Vector Number	IVT Address	AIVT Address	Interrupt Bit Locations		
				Flag	Enable	Priority
ADC1 – ADC1 Convert Done	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
CLC1	96	0000D4h	0001D4h	IFS6<0>	IEC6<0>	IPC24<2:0>
CLC2	97	0000D6h	0001D6h	IFS6<1>	IEC6<1>	IPC24<6:4>
Comparator Interrupt	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CTMU	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
DAC1 – Buffer Update	78	0000B0h	0001B0h	IFS4<14>	IEC4<14>	IPC19<10:8>
DAC2 – Buffer Update	79	0000B2h	0001B2h	IFS4<15>	IEC4<15>	IPC19<14:12>
HLVD – High/Low-Voltage Detect	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC18<2:0>
ICN – Input Change Notification	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
INT0 – External Interrupt 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
INT1 – External Interrupt 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
INT2 – External Interrupt 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
MCCP1 – Capture/Compare	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
MCCP1 – Time Base	7	000022h	000122h	IFS0<7>	IEC0<7>	IPC1<14:12>
MCCP2 – Capture/Compare	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
MCCP2 – Time Base	8	000024h	000124h	IFS0<8>	IEC0<8>	IPC2<2:0>
MCCP3 – Capture/Compare	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
MCCP3 – Time Base	27	00004Ah	00014Ah	IFS1<11>	IEC1<11>	IPC6<14:12>
MSSP1 – Bus Collision Interrupt	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
MSSP1 – I^2C™/SPI Interrupt	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
MSSP2 – Bus Collision Interrupt	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
MSSP2 – I^2C/SPI Interrupt	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
NVM – NVM Write Complete	15	000032h	000132h	IFS0<15>	IEC0<15>	IPC3<14:12>
RTCC – Real-Time Clock/Calendar	62	000090h	000190h	IFS3<14>	IEC3<14>	IPC15<10:8>
SCCP4 – Capture/Compare	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
SCCP4 – Time Base	28	00004Ch	00014Ch	IFS1<12>	IEC1<12>	IPC7<2:0>
SCCP5 – Capture/Compare	22	000040h	000140h	IFS1<6>	IEC1<6>	IPC5<10:8>
SCCP5 – Time Base	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
TMR1 – Timer1	3	00001Ah	00011Ah	IFS0<3>	IEC0<3>	IPC0<14:12>
UART1 Error	65	000096h	000196h	IFS4<1>	IEC4<1>	IPC16<6:4>
UART2 Error	66	000098h	000198h	IFS4<2>	IEC4<2>	IPC16<10:8>
UART1RX – UART1 Receiver	11	00002Ah	00012Ah	IFS0<11>	IEC0<11>	IPC2<14:12>
UART1TX – UART1 Transmitter	12	00002Ch	00012Ch	IFS0<12>	IEC0<12>	IPC3<2:0>
UART2RX – UART2 Receiver	30	000050h	000150h	IFS1<14>	IEC1<14>	IPC7<10:8>
UART2TX – UART2 Transmitter	31	000052h	000152h	IFS1<15>	IEC1<15>	IPC7<14:12>
ULPWU – Ultra Low-Power Wake-up	80	0000B4h	0001B4h	IFS5<0>	IEC5<0>	IPC20<2:0>

REGISTER 8-13: IEC1: INTERRUPT ENABLE CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
U2TXIE	U2RXIE	INT2IE	CCT4IE	CCT3IE	—	—	—
bit 15							bit 8

U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CCP5IE	—	INT1IE	CNIE	CMIE	BCL1IE	SSP1IE
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 **U2TXIE:** UART2 Transmitter Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 14 **U2RXIE:** UART2 Receiver Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 13 **INT2IE:** External Interrupt 2 Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 12 **CCT4IE:** Capture/Compare 4 Timer Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 11 **CCT3IE:** Capture/Compare 3 Timer Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 10-7 **Unimplemented:** Read as '0'

bit 6 **CCP5IE:** Capture/Compare 5 Event Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 5 **Unimplemented:** Read as '0'

bit 4 **INT1IE:** External Interrupt 1 Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 3 **CNIE:** Input Change Notification Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 2 **CMIE:** Comparator Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 1 **BCL1IE:** MSSP1 I^2C™ Bus Collision Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 0 **SSP1IE:** MSSP1 SPI/I^2C Event Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

REGISTER 8-20: IPC1: INTERRUPT PRIORITY CONTROL REGISTER 1

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CCT1IP2	CCT1IP1	CCT1IP0	—	CCP4IP2	CCP4IP1	CCP4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	CCP3IP2	CCP3IP1	CCP3IP0	—	—	—	—
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 **Unimplemented:** Read as ‘0’

bit 14-12 **CCT1IP<2:0>:** Capture/Compare 1 Timer Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **CCP4IP<2:0>:** Capture/Compare 4 Event Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **CCP3IP<2:0>:** Capture/Compare 3 Event Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as ‘0’

9.3 Control Registers

The operation of the oscillator is controlled by three Special Function Registers (SFRs):

- OSCCON
- CLKDIV
- OSCTUN

The OSCCON register (Register 9-1) is the main control register for the oscillator. It controls clock source switching and allows the monitoring of clock sources.

The Clock Divider register (Register 9-2) controls the features associated with Doze mode, as well as the postscaler for the FRC Oscillator.

The FRC Oscillator Tune register (Register 9-3) allows the user to fine-tune the FRC Oscillator over a range of approximately ±5.25%. Each bit increment or decrement changes the factory calibrated frequency of the FRC Oscillator by a fixed amount.

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R-0, HSC	R-0, HSC	R-0, HSC	U-0	R/W-x(1)	R/W-x(1)	R/W-x(1)
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15							bit 8

R/SO-0, HSC	U-0	R-0, HSC(2)	U-0	R/CO-0, HS	R/W-0(3)	R/W-0	R/W-0
CLKLOCK	—	LOCK	—	CF	SOSCDRV	SOSCEN	OSWEN
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
HS = Hardware Settable bit	CO = Clearable Only bit	SO = Settable Only 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 **Unimplemented:** Read as '0'

bit 14-12 **COSC<2:0>:** Current Oscillator Selection bits
111 = 8 MHz Fast RC Oscillator with Postscaler (FRCDIV)
110 = 500 kHz Low-Power Fast RC Oscillator (FRC) with Postscaler (LPFRCDIV)
101 = Low-Power RC Oscillator (LPRC)
100 = Secondary Oscillator (SOSC)
011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
010 = Primary Oscillator (XT, HS, EC)
001 = 8 MHz FRC Oscillator with Postscaler and PLL module (FRCPLL)
000 = 8 MHz FRC Oscillator (FRC)

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits(1)
111 = 8 MHz Fast RC Oscillator with Postscaler (FRCDIV)
110 = 500 kHz Low-Power Fast RC Oscillator (FRC) with Postscaler (LPFRCDIV)
101 = Low-Power RC Oscillator (LPRC)
100 = Secondary Oscillator (SOSC)
011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
010 = Primary Oscillator (XT, HS, EC)
001 = 8 MHz FRC Oscillator with Postscaler and PLL module (FRCPLL)
000 = 8 MHz FRC Oscillator (FRC)

Note 1: Reset values for these bits are determined by the FNOSCx Configuration bits.
2: This bit 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.

REGISTER 14-10: 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 PIC24FXXKM20X devices.

REGISTER 16-1: RCFGCAL: RTCC CALIBRATION AND CONFIGURATION REGISTER(1) (CONTINUED)

bit 7-0 **CAL<7:0>:** RTC Drift Calibration bits

01111111 = Maximum positive adjustment; adds 508 RTC clock pulses every one minute
.
.
.
00000001 = Minimum positive adjustment; adds 4 RTC clock pulses every one minute
00000000 = No adjustment
11111111 = Minimum negative adjustment; subtracts 4 RTC clock pulses every one minute
.
.
.
10000000 = Maximum negative adjustment; subtracts 512 RTC clock pulses every one minute

Note 1: The RCFGCAL register is only affected by a POR.

2: A write to the RTCEN bit is only allowed when RTCWREN = 1.

3: This bit is read-only; it is cleared to '0' on a write to the lower half of the MINSEC register.

REGISTER 18-1: HLVDCON: HIGH/LOW-VOLTAGE DETECT CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
HLVDEN	—	HLSIDL	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VDIR	BGVST	IRVST	—	HLVDL3	HLVDL2	HLVDL1	HLVDL0
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 **HLVDEN:** High/Low-Voltage Detect Power Enable bit
1 = HLVD is enabled
0 = HLVD is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **HLSIDL:** HLVD Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-8 **Unimplemented:** Read as '0'

bit 7 **VDIR:** Voltage Change Direction Select bit
1 = Event occurs when voltage equals or exceeds trip point (HLVDL<3:0>)
0 = Event occurs when voltage equals or falls below trip point (HLVDL<3:0>)

bit 6 **BGVST:** Band Gap Voltage Stable Flag bit
1 = Indicates that the band gap voltage is stable
0 = Indicates that the band gap voltage is unstable

bit 5 **IRVST:** Internal Reference Voltage Stable Flag bit
1 = Indicates that the internal reference voltage is stable and the High-Voltage Detect logic generates the interrupt flag at the specified voltage range
0 = Indicates that the internal reference voltage is unstable and the High-Voltage Detect logic will not generate the interrupt flag at the specified voltage range, and the HLVD interrupt should not be enabled

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

bit 3-0 **HLVDL<3:0>:** High/Low-Voltage Detection Limit bits
1111 = External analog input is used (input comes from the HLVDIN pin)
1110 = Trip Point 1(1)
1101 = Trip Point 2(1)
1100 = Trip Point 3(1)
.
.
.
0000 = Trip Point 15(1)

Note 1: For the actual trip point, see **Section 27.0 "Electrical Characteristics"**.

REGISTER 19-8: AD1CSSH: A/D INPUT SCAN SELECT REGISTER (HIGH WORD)(1)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	CSS30	CSS29	CSS28	CSS27	CSS26	—	—
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
CSS23	CSS22	CSS21	CSS20(2)	CSS19(2)	CSS18	CSS17	CSS16
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 **Unimplemented:** Read as '0'

bit 14-10 **CSS<30:26>:** A/D Input Scan Selection bits
1 = Includes the corresponding channel for input scan
0 = Skips the channel for input scan

bit 9-8 **Unimplemented:** Read as '0'

bit 7-0 **CSS<23:16>:** A/D Input Scan Selection bits(2)
1 = Includes the corresponding channel for input scan
0 = Skips the channel for input scan

Note 1: Unimplemented channels are read as '0'. Do not select unimplemented channels for sampling as indeterminate results may be produced.
2: The CSS<20:19> bits are not implemented in 20-pin devices.

REGISTER 19-9: AD1CSSL: A/D INPUT SCAN SELECT REGISTER (LOW WORD)(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
CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8(2,3)
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
CSS7(2,3)	CSS6(2,3)	CSS5(2)	CSS4	CSS3	CSS2	CSS1	CSS0
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 15-0 **CSS<15:0>:** A/D Input Scan Selection bits(2,3)
1 = Includes the corresponding ANx input for scan
0 = Skips the channel for input scan

Note 1: Unimplemented channels are read as '0'. Do not select unimplemented channels for sampling as indeterminate results may be produced.
2: The CSS<8:5> bits are not implemented in 20-pin devices.
3: The CSS<8:6> bits are not implemented in 28-pin devices.

REGISTER 21-1: AMPxCON: OP AMP x CONTROL REGISTER(1)

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
AMPEN	—	AMPSIDL	AMPSLP	—	—	—	—
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPDSEL	—	NINSEL2	NINSEL1	NINSEL0	PINSEL2	PINSEL1	PINSEL0
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 **AMPEN:** Op Amp x Control Module Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **AMPSIDL:** Op Amp x Peripheral Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **AMPSLP:** Op Amp x Peripheral Enabled in Sleep Mode bit
1 = Continues module operation when device enters Sleep mode
0 = Discontinues module operation in Sleep mode

bit 11-8 **Unimplemented:** Read as '0'

bit 7 **SPDSEL:** Op Amp x Power/Speed Select bit
1 = Higher power and bandwidth (faster response time)
0 = Lower power and bandwidth (slower response time)

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

bit 5-3 **NINSEL<2:0>:** Negative Op Amp Input Select bits
111 = Reserved; do not use
110 = Reserved; do not use
101 = Op amp negative input is connected to the op amp output (voltage follower)
100 = Reserved; do not use
011 = Reserved; do not use
010 = Op amp negative input is connected to the OAxIND pin
001 = Op amp negative input is connected to the OAxINB pin
000 = Op amp negative input is connected to AVSS

bit 2-0 **PINSEL<2:0>:** Positive Op Amp Input Select bits
111 = Op amp positive input is connected to the output of the A/D input multiplexer
110 = Reserved; do not use
101 = Op amp positive input is connected to the DAC1 output for OA1 (DAC2 output for OA2)
100 = Reserved; do not use
011 = Reserved; do not use
010 = Op amp positive input is connected to the OAxINC pin
001 = Op amp positive input is connected to the OAxINA pin
000 = Op amp positive input is connected to AVSS

Note 1: This register is available only on PIC24F(V)16KM2XX devices.

26.11 Demonstration/Development Boards, Evaluation Kits and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

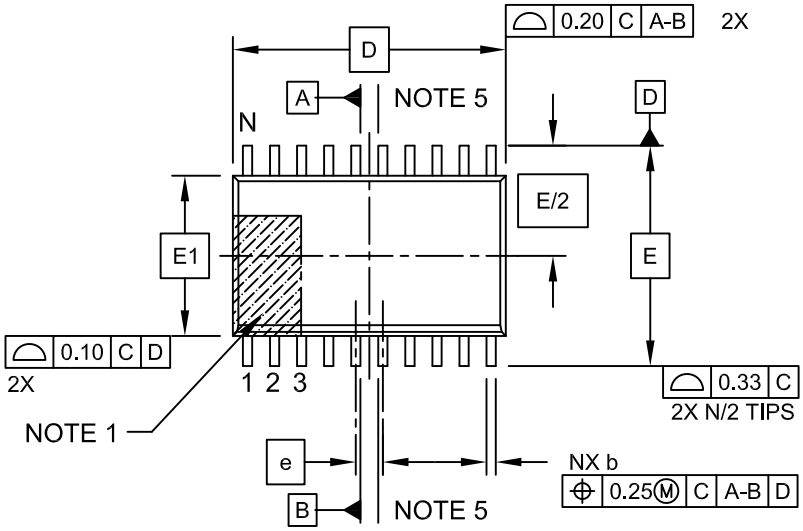
26.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

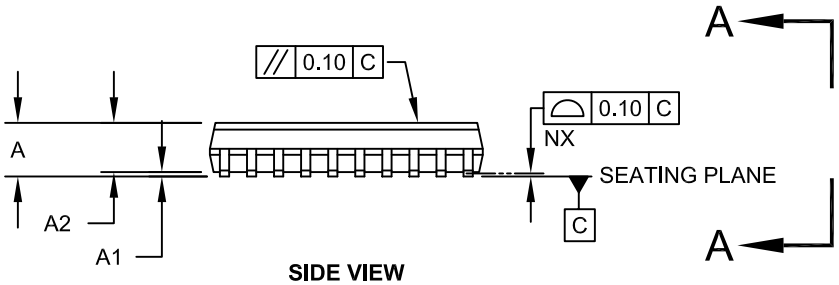
- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

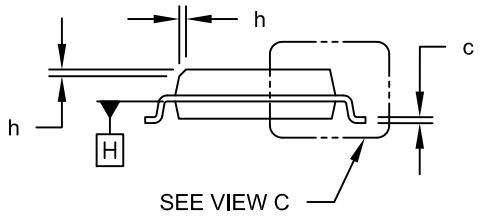
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW

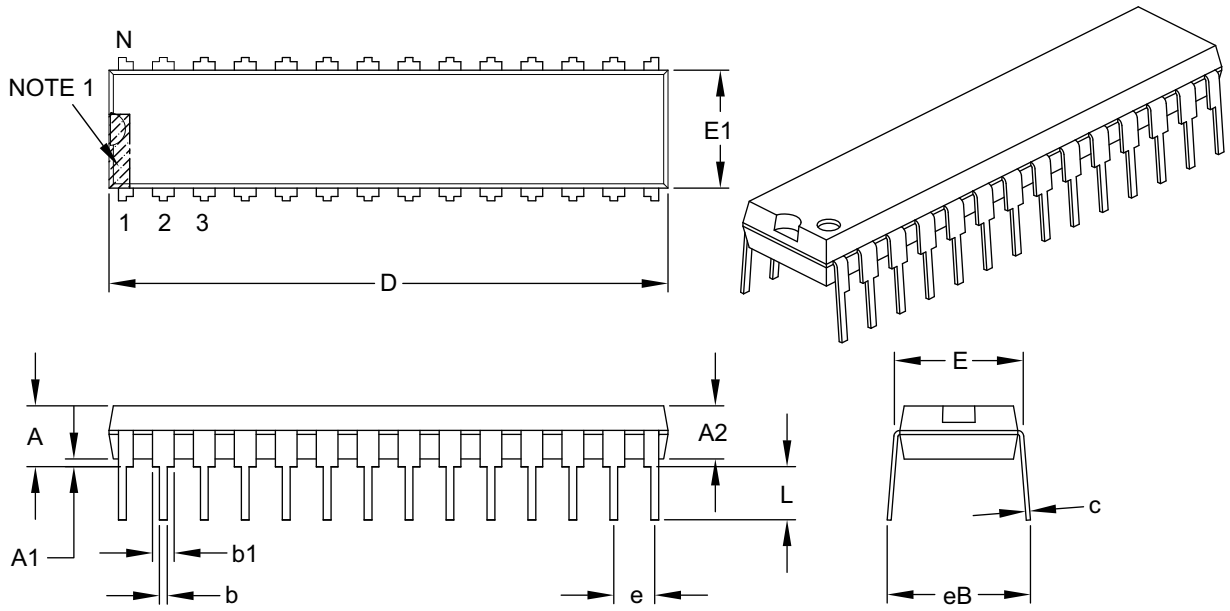


VIEW A-A

Microchip Technology Drawing C04-094C Sheet 1 of 2

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

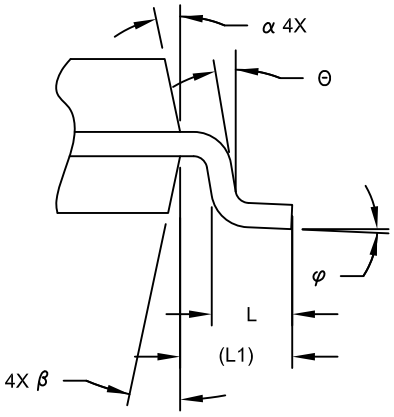
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

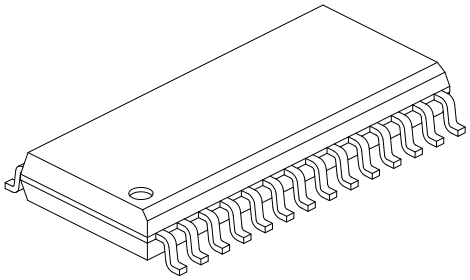
Microchip Technology Drawing C04-070B

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		17.90 BSC	
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1		1.40 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

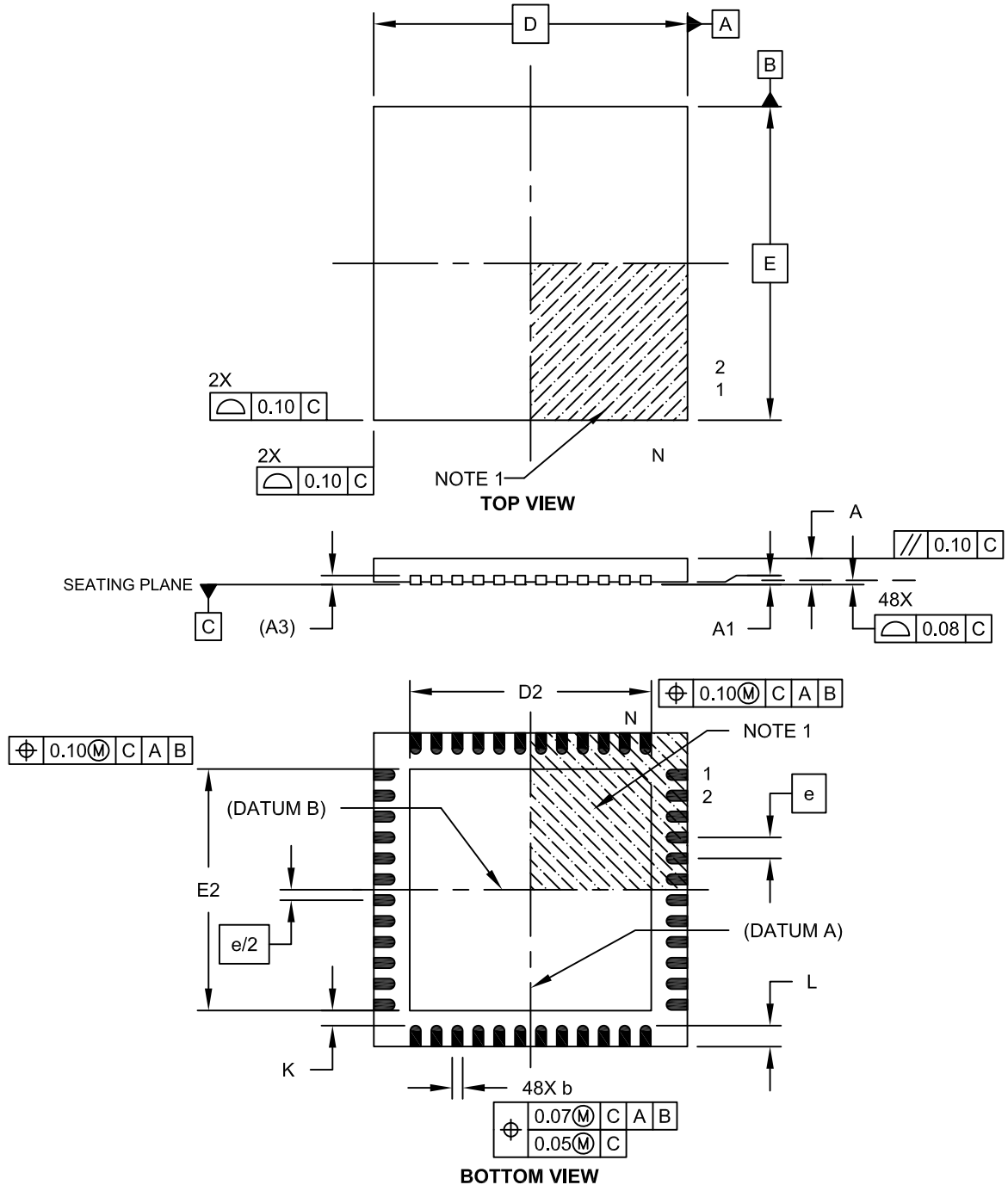
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

48-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 6x6x0.5 mm Body [UQFN]

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



Microchip Technology Drawing C04-153A Sheet 1 of 2

NOTES: