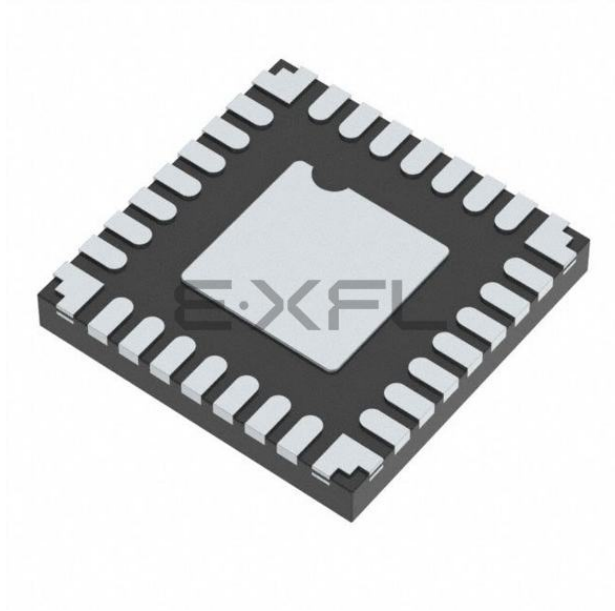
# Microchip Technology - DSPIC33EP16GS202-I/M6 Datasheet



**Welcome to E-XFL.COM**

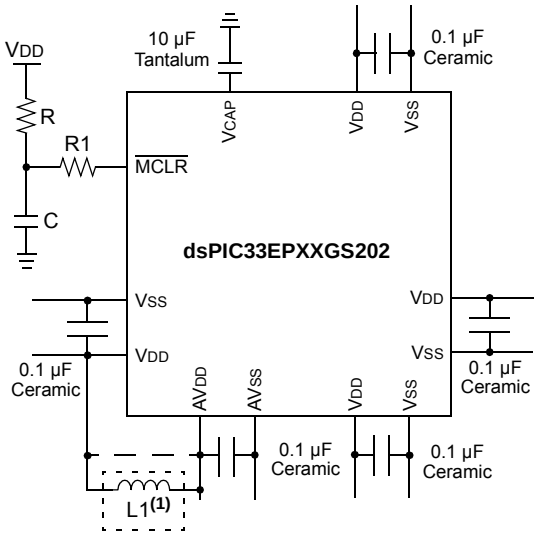
## What is "Embedded - Microcontrollers"?

"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 | dsPIC |
| Core Size | 16-Bit |
| Speed | 70 MIPs |
| Connectivity | I²C, IrDA, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 21 |
| Program Memory Size | 16KB (16K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 3.6V |
| Data Converters | A/D 12x12b; D/A 2x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-UFQFN Exposed Pad |
| Supplier Device Package | 28-UQFN (4x4) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/dspic33ep16gs202-i-m6 |

**FIGURE 2-1: RECOMMENDED MINIMUM CONNECTION**




**Note 1:** As an option, instead of a hard-wired connection, an inductor (L1) can be substituted between VDD and AVDD to improve ADC noise rejection. The inductor impedance should be less than 1Ω and the inductor capacity greater than 10 mA.

Where:

$$f = \frac{F_{CNV}}{2}$$ (i.e., A/D Conversion Rate/2)

$$f = \frac{1}{(2\pi\sqrt{LC})}$$

$$L = \left(\frac{1}{(2\pi f\sqrt{C})}\right)^2$$

### 2.2.1 TANK CAPACITORS

On boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits including DSCs to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 µF to 47 µF.

## 2.3 CPU Logic Filter Capacitor Connection (VCAP)

A low-ESR (<1 Ohms) capacitor is required on the VCAP pin, which is used to stabilize the voltage regulator output voltage. The VCAP pin must not be connected to VDD and must have a capacitor greater than 4.7 µF (10 µF is recommended), 16V connected to ground. The type can be ceramic or tantalum. See **Section 25.0 "Electrical Characteristics"** for additional information.

The placement of this capacitor should be close to the VCAP pin. It is recommended that the trace length not exceeds one-quarter inch (6 mm). See **Section 22.4 "On-Chip Voltage Regulator"** for details.

## 2.4 Master Clear (MCLR) Pin

The MCLR pin provides two specific device functions:

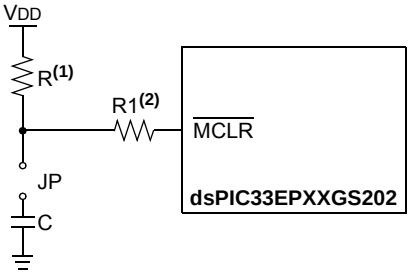
- Device Reset
- Device Programming and Debugging.

During device programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the MCLR pin. Consequently, specific voltage levels (VIH and VIL) and fast signal transitions must not be adversely affected. Therefore, specific values of R and C will need to be adjusted based on the application and PCB requirements.

For example, as shown in Figure 2-2, it is recommended that the capacitor, C, be isolated from the MCLR pin during programming and debugging operations.

Place the components as shown in Figure 2-2 within one-quarter inch (6 mm) from the MCLR pin.

**FIGURE 2-2: EXAMPLE OF MCLR PIN CONNECTIONS**




**Note 1:** R ≤ 10 kΩ is recommended. A suggested starting value is 10 kΩ. Ensure that the MCLR pin VIH and VIL specifications are met.

**2:** R1 ≤ 470Ω will limit any current flowing into MCLR from the external capacitor, C, in the event of MCLR pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the MCLR pin VIH and VIL specifications are met.

## 2.5 ICSP Pins

The PGECx and PGEDx pins are used for ICSP and debugging purposes. It is recommended to keep the trace length between the ICSP connector and the ICSP pins on the device as short as possible. If the ICSP connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of Ohms, not to exceed 100 Ohms.

Pull-up resistors, series diodes and capacitors on the PGECx and PGEDx pins are not recommended as they will interfere with the programmer/debugger communications to the device. If such discrete components are an application requirement, they should be removed from the circuit during programming and debugging. Alternatively, refer to the AC/DC characteristics and timing requirements information in the respective device Flash programming specification for information on capacitive loading limits and pin Voltage Input High ($V_{IH}$) and Voltage Input Low ($V_{IL}$) requirements.

Ensure that the "Communication Channel Select" (i.e., PGECx/PGEDx pins) programmed into the device matches the physical connections for the ICSP to MPLAB® PICkit™ 3, MPLAB ICD 3 or MPLAB REAL ICE™.

For more information on MPLAB ICD 2, MPLAB ICD 3 and REAL ICE connection requirements, refer to the following documents that are available on the Microchip web site.

- *"Using MPLAB® ICD 3"* (poster) DS51765
- *"Multi-Tool Design Advisory"* DS51764
- *"MPLAB® REAL ICE™ In-Circuit Emulator User's Guide"* DS51616
- *"Using MPLAB® REAL ICE™ In-Circuit Emulator"* (poster) DS51749

## 2.6 External Oscillator Pins

Many DSCs have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator. For details, see **Section 8.0 "Oscillator Configuration"** for details.

The oscillator circuit should be placed on the same side of the board as the device. Also, place the oscillator circuit close to the respective oscillator pins, not exceeding one-half inch (12 mm) distance between them. The load capacitors should be placed next to the oscillator itself, on the same side of the board. Use a grounded copper pour around the oscillator circuit to isolate them from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed. A suggested layout is shown in Figure 2-3.

**FIGURE 2-3: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT**

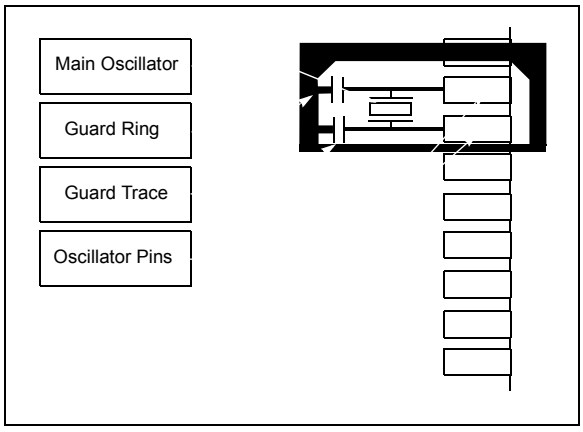


**TABLE 4-17: NVM REGISTER MAP**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NVMCON | 0728 | WR | WREN | WRERR | NVMSIDL | — | — | RPDF | URERR | — | — | — | — | NVMOP3 | NVMOP2 | NVMOP1 | NVMOP0 | 0000 |
| NVMADR | 072A | NVMADR<15:0> | | | | | | | | | | | | | | | | 0000 |
| NVMADRU | 072C | — | — | — | — | — | — | — | — | NVMADR<23:16> | | | | | | | | 0000 |
| NVMKEY | 072E | — | — | — | — | — | — | — | — | NVMKEY<7:0> | | | | | | | | 0000 |
| NVMSRCADRL | 0730 | NVMSRCADR<15:0> | | | | | | | | | | | | | | | | 0000 |
| NVMSRCADRH | 0732 | — | — | — | — | — | — | — | — | NVMSRCADR<23:16> | | | | | | | | 0000 |

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**TABLE 4-18: SYSTEM CONTROL 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RCON | 0740 | TRAPR | IOPUWR | — | — | VREGSF | — | CM | VREGS | EXTR | SWR | SWDTEN | WDTO | SLEEP | IDLE | BOR | POR | **Note 1** |
| OSCCON | 0742 | — | COSC2 | COSC1 | COSC0 | — | NOSC2 | NOSC1 | NOSC0 | CLKLOCK | IOLOCK | LOCK | — | CF | — | — | OSWEN | **Note 2** |
| CLKDIV | 0744 | ROI | DOZE2 | DOZE1 | DOZE0 | DOZEN | FRCDIV2 | FRCDIV1 | FRCDIV0 | PLLPOST1 | PLLPOST0 | — | PLLPRE4 | PLLPRE3 | PLLPRE2 | PLLPRE1 | PLLPRE0 | 3040 |
| PLLFBD | 0746 | — | — | — | — | — | — | — | PLLDIV<8:0> | | | | | | | | | 0030 |
| OSCTUN | 0748 | — | — | — | — | — | — | — | — | — | — | TUN<5:0> | | | | | | 0000 |
| LFSR | 074C | — | LFSR<14:0> | | | | | | | | | | | | | | | 0000 |
| ACLKCON | 0750 | ENAPLL | APLLCK | SELACLK | — | — | APSTSCLR2 | APSTSCLR1 | APSTSCLR0 | ASRCSEL | FRCSEL | — | — | — | — | — | — | 2740 |

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
**Note 1:** RCON register Reset values are dependent on the type of Reset.
**2:** OSCCON register Reset values are dependent on the Configuration fuses.

**TABLE 4-19: PMD 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PMD1 | 0760 | — | — | T3MD | T2MD | T1MD | — | PWMMD | — | I2C1MD | — | U1MD | — | SPI1MD | — | — | ADCMD | 0000 |
| PMD2 | 0762 | — | — | — | — | — | — | — | IC1MD | — | — | — | — | — | — | — | OC1MD | 0000 |
| PMD3 | 0764 | — | — | — | — | — | CMPMD | — | — | — | — | — | — | — | — | — | — | 0000 |
| PMD6 | 076A | — | — | — | — | — | PWM3MD | PWM2MD | PWM1MD | — | — | — | — | — | — | — | — | 0000 |
| PMD7 | 076C | — | — | — | — | — | — | CMP2MD | CMP1MD | — | — | — | — | — | — | PGA1MD | — | 0000 |
| PMD8 | 076E | — | — | — | — | — | PGA2MD | — | — | — | — | — | — | — | — | — | — | 0000 |

**Legend:** — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

### 4.6.3 MOVE AND ACCUMULATOR INSTRUCTIONS

Move instructions, and the DSP accumulator class of instructions, provide a greater degree of addressing flexibility than other instructions. In addition to the addressing modes supported by most MCU instructions, move and accumulator instructions also support Register Indirect with Register Offset Addressing mode, also referred to as Register Indexed mode.

| | |
|---|---|
| **Note:** | For the MOV instructions, the addressing mode specified in the instruction can differ for the source and destination EA. However, the 4-bit Wb (Register Offset) field is shared by both source and destination (but typically, only used by one). |

In summary, the following addressing modes are supported by move and accumulator instructions:

- Register Direct
- Register Indirect
- Register Indirect Post-modified
- Register Indirect Pre-modified
- Register Indirect with Register Offset (Indexed)
- Register Indirect with Literal Offset
- 8-Bit Literal
- 16-Bit Literal

| | |
|---|---|
| **Note:** | Not all instructions support all the addressing modes given above. Individual instructions may support different subsets of these addressing modes. |

### 4.6.4 MAC INSTRUCTIONS

The dual source operand DSP instructions (CLR, ED, EDAC, MAC, MPY, MPY.N, MOVSAC and MSC), also referred to as MAC instructions, use a simplified set of addressing modes to allow the user application to effectively manipulate the Data Pointers through register indirect tables.

The two-source operand prefetch registers must be members of the set {W8, W9, W10, W11}. For data reads, W8 and W9 are always directed to the X RAGU, and W10 and W11 are always directed to the Y AGU. The Effective Addresses generated (before and after modification) must therefore, be valid addresses within X Data Space for W8 and W9, and Y Data Space for W10 and W11.

| | |
|---|---|
| **Note:** | Register Indirect with Register Offset Addressing mode is available only for W9 (in X space) and W11 (in Y space). |

In summary, the following addressing modes are supported by the MAC class of instructions:

- Register Indirect
- Register Indirect Post-Modified by 2
- Register Indirect Post-Modified by 4
- Register Indirect Post-Modified by 6
- Register Indirect with Register Offset (Indexed)

### 4.6.5 OTHER INSTRUCTIONS

Besides the addressing modes outlined previously, some instructions use literal constants of various sizes. For example, BRA (branch) instructions use 16-bit signed literals to specify the branch destination directly, whereas the DISI instruction uses a 14-bit unsigned literal field. In some instructions, such as ULNK, the source of an operand or result is implied by the opcode itself. Certain operations, such as a NOP, do not have any operands.

## 5.0 FLASH PROGRAM MEMORY

**Note 1:** This data sheet summarizes the features of the dsPIC33EPXXGS202 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Flash Programming"** (DS70609) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

**2:** Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

The dsPIC33EPXXGS202 family devices contain internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable during normal operation over the entire VDD range.

Flash memory can be programmed in three ways:

- In-Circuit Serial Programming™ (ICSP™) programming capability
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)
- Run-Time Self-Programming (RTSP)

ICSP allows for a dsPIC33EPXXGS202 family device to be serially programmed while in the end application circuit. This is done with a programming clock and programming data (PGECx/PGEDx) line, and three other lines for power (VDD), ground (VSS) and Master Clear ($\overline{MCLR}$). This allows customers to manufacture boards with unprogrammed devices and then program the device just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

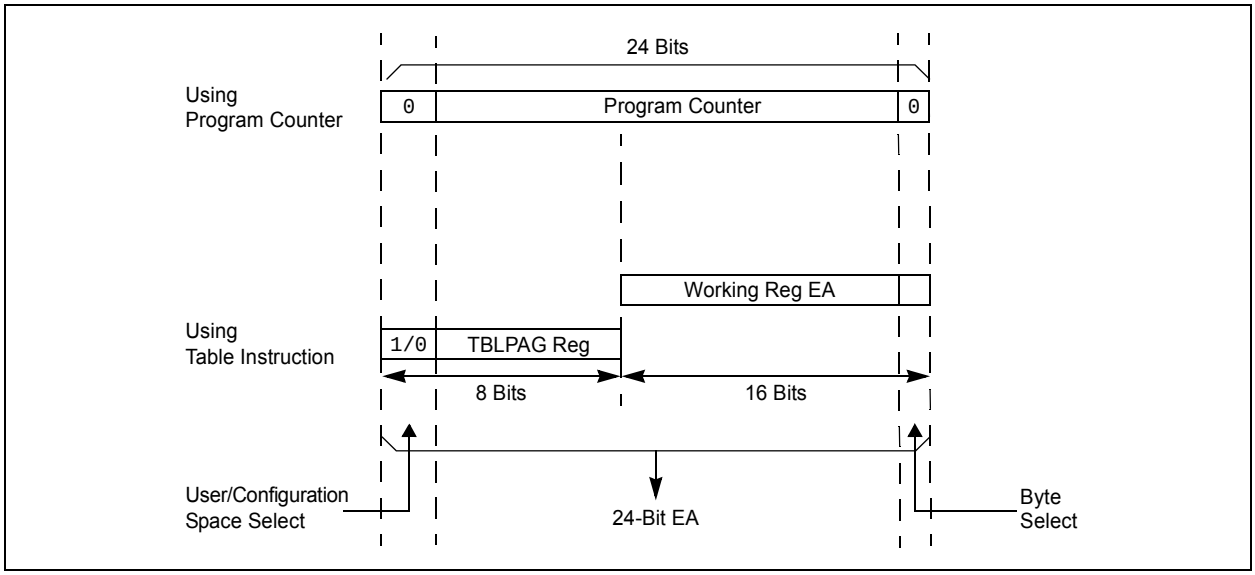
Enhanced In-Circuit Serial Programming uses an on-board bootloader, known as the Program Executive, to manage the programming process. Using an SPI data frame format, the Program Executive can erase, program and verify program memory. For more information on Enhanced ICSP, see the device programming specification.

RTSP is accomplished using TBLRD (Table Read) and TBLWT (Table Write) instructions. With RTSP, the user application can write program memory data with a single program memory word and erase program memory in blocks or 'pages' of 512 instructions (1536 bytes) at a time.

### 5.1 Table Instructions and Flash Programming

Regardless of the method used, all programming of Flash memory is done with the Table Read and Table Write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using bits<7:0> of the TBLPAG register and the Effective Address (EA) from a W register, specified in the table instruction, as shown in Figure 5-1. The TBLRDL and the TBLWTL instructions are used to read or write to bits<15:0> of program memory. TBLRDL and TBLWTL can access program memory in both Word and Byte modes. The TBLRDH and TBLWTH instructions are used to read or write to bits<23:16> of program memory. TBLRDH and TBLWTH can also access program memory in Word or Byte mode.

**FIGURE 5-1: ADDRESSING FOR TABLE REGISTERS**



**REGISTER 5-1: NVMCON: NONVOLATILE MEMORY (NVM) CONTROL REGISTER (CONTINUED)**

bit 3-0 **NVMOP<3:0>:** NVM Operation Select bits(1,3,4)
1111 = Reserved
•
•
•
0101 = Reserved
0100 = Reserved
0011 = Memory page erase operation
0010 = Memory row program operation
0001 = Memory double-word program operation(5)
0000 = Reserved

**Note 1:** These bits can only be reset on a POR.
**2:** If this bit is set, power consumption will be further reduced (IIDLE), and upon exiting Idle mode, there is a delay (TVREG) before Flash memory becomes operational.
**3:** All other combinations of NVMOP<3:0> are unimplemented.
**4:** Execution of the PWRSAV instruction is ignored while any of the NVM operations are in progress.
**5:** Two adjacent words on a 4-word boundary are programmed during execution of this operation.

**REGISTER 5-2: NVMADR: NONVOLATILE MEMORY LOWER ADDRESS REGISTER**

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| NVMADR<15:8> | | | | | | | |
| bit 15 | | | | | | | bit 8 |

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| NVMADR<7:0> | | | | | | | |
| 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 15-0 **NVMADR<15:0>:** Nonvolatile Memory Lower Write Address bits
Selects the lower 16 bits of the location to program or erase in Program Flash Memory. This register may be read or written to by the user application.

## 6.1 Reset Resources

Many useful resources are provided on the main product page of the Microchip web site for the devices listed in this data sheet. This product page contains the latest updates and additional information.

### 6.1.1 KEY RESOURCES

- **"Reset"** (DS70602) in the *"dsPIC33/PIC24 Family Reference Manual"*
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All Related *"dsPIC33/PIC24 Family Reference Manual"* Sections
- Development Tools

**REGISTER 10-5: RPINR7: PERIPHERAL PIN SELECT INPUT REGISTER 7**

| 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 |
|---|---|---|---|---|---|---|---|
| IC1R<7:0> | | | | | | | |
| 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 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **IC1R<7:0>:** Assign Input Capture 1 (IC1) to the Corresponding RPn Pin bits

10110101 = Input tied to RP181
10110100 = Input tied to RP180
•
•
•
00000001 = Input tied to RP1
00000000 = Input tied to Vss

**REGISTER 10-6: RPINR11: PERIPHERAL PIN SELECT INPUT REGISTER 11**

| 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 |
|---|---|---|---|---|---|---|---|
| OCFAR<7:0> | | | | | | | |
| 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 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **OCFAR<7:0>:** Assign Output Compare Fault A (OCFA) to the Corresponding RPn Pin bits

10110101 = Input tied to RP181
10110100 = Input tied to RP180
•
•
•
00000001 = Input tied to RP1
00000000 = Input tied to Vss

**REGISTER 10-13: RPINR38: PERIPHERAL PIN SELECT INPUT REGISTER 38**

| 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 |
|---|---|---|---|---|---|---|---|
| SYNCI2R<7:0> | | | | | | | |
| 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 15-8 **Unimplemented:** Read as ‘0’

bit 7-0 **SYNCI2R<7:0>:** Assign PWM Synchronization Input 2 to the Corresponding RPn Pin bits

10110101 = Input tied to RP181
10110100 = Input tied to RP180
•
•
•
00000001 = Input tied to RP1
00000000 = Input tied to Vss

**REGISTER 10-20: RPOR4: PERIPHERAL PIN SELECT OUTPUT REGISTER 4**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | RP41R5 | RP41R4 | RP41R3 | RP41R2 | RP41R1 | RP41R0 |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | RP40R5 | RP40R4 | RP40R3 | RP40R2 | RP40R1 | RP40R0 |
| 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 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP41R<5:0>:** Peripheral Output Function is Assigned to RP41 Output Pin bits (see Table 10-2 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP40R<5:0>:** Peripheral Output Function is Assigned to RP40 Output Pin bits (see Table 10-2 for peripheral function numbers)

**REGISTER 10-21: RPOR5: PERIPHERAL PIN SELECT OUTPUT REGISTER 5**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | RP43R5 | RP43R4 | RP43R3 | RP43R2 | RP43R1 | RP43R0 |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | RP42R5 | RP42R4 | RP42R3 | RP42R2 | RP42R1 | RP42R0 |
| 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 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP43R<5:0>:** Peripheral Output Function is Assigned to RP43 Output Pin bits (see Table 10-2 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP42R<5:0>:** Peripheral Output Function is Assigned to RP42 Output Pin bits (see Table 10-2 for peripheral function numbers)

**REGISTER 10-24: RPOR8: PERIPHERAL PIN SELECT OUTPUT REGISTER 8**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | RP177R5 | RP177R4 | RP177R3 | RP177R2 | RP177R1 | RP177R0 |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | RP176R5 | RP176R4 | RP176R3 | RP176R2 | RP176R1 | RP176R0 |
| 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 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP177R<5:0>:** Peripheral Output Function is Assigned to RP177 Output Pin bits (see Table 10-2 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP176R<5:0>:** Peripheral Output Function is Assigned to RP176 Output Pin bits (see Table 10-2 for peripheral function numbers)

**REGISTER 10-25: RPOR9: PERIPHERAL PIN SELECT OUTPUT REGISTER 9**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | RP179R5 | RP179R4 | RP179R3 | RP179R2 | RP179R1 | RP179R0 |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | RP178R5 | RP178R4 | RP178R3 | RP178R2 | RP178R1 | RP178R0 |
| 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 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP179R<5:0>:** Peripheral Output Function is Assigned to RP179 Output Pin bits (see Table 10-2 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP178R<5:0>:** Peripheral Output Function is Assigned to RP178 Output Pin bits (see Table 10-2 for peripheral function numbers)

**REGISTER 15-20: IOCONx: PWMx I/O CONTROL REGISTER**

| R/W-1 | R/W-1 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| PENH | PENL | POLH | POLL | PMOD1(1) | PMOD0(1) | OVRENH | OVRENL |
| 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 |
|---|---|---|---|---|---|---|---|
| OVRDAT1 | OVRDAT0 | FLTDAT1(2) | FLTDAT0(2) | CLDAT1(2) | CLDAT0(2) | SWAP | OSYNC |
| 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 **PENH:** PWMxH Output Pin Ownership bit
1 = PWM module controls the PWMxH pin
0 = GPIO module controls the PWMxH pin

bit 14 **PENL:** PWMxL Output Pin Ownership bit
1 = PWM module controls the PWMxL pin
0 = GPIO module controls the PWMxL pin

bit 13 **POLH:** PWMxH Output Pin Polarity bit
1 = PWMxH pin is active-low
0 = PWMxH pin is active-high

bit 12 **POLL:** PWMxL Output Pin Polarity bit
1 = PWMxL pin is active-low
0 = PWMxL pin is active-high

bit 11-10 **PMOD<1:0>:** PWMx I/O Pin Mode bits(1)
11 = PWMx I/O pin pair is in the True Independent Output mode
10 = PWMx I/O pin pair is in the Push-Pull Output mode
01 = PWMx I/O pin pair is in the Redundant Output mode
00 = PWMx I/O pin pair is in the Complementary Output mode

bit 9 **OVRENH:** Override Enable for PWMxH Pin bit
1 = OVRDAT1 provides data for output on the PWMxH pin
0 = PWMx generator provides data for the PWMxH pin

bit 8 **OVRENL:** Override Enable for PWMxL Pin bit
1 = OVRDAT0 provides data for output on the PWMxL pin
0 = PWMx generator provides data for the PWMxL pin

bit 7-6 **OVRDAT<1:0>:** Data for PWMxH, PWMxL Pins if Override is Enabled bits
If OVERENH = 1, OVRDAT1 provides the data for the PWMxH pin.
If OVERENL = 1, OVRDAT0 provides the data for the PWMxL pin.

bit 5-4 **FLTDAT<1:0>:** State for PWMxH and PWMxL Pins if FLTMOD<1:0> are Enabled bits(2)
IFLTMOD (FCLCONx<15>) = 0: Normal Fault mode:
If Fault is active, then FLTDAT1 provides the state for the PWMxH pin.
If Fault is active, then FLTDAT0 provides the state for the PWMxL pin.
IFLTMOD (FCLCONx<15>) = 1: Independent Fault mode:
If current-limit is active, then FLTDAT1 provides the state for the PWMxH pin.
If Fault is active, then FLTDAT0 provides the state for the PWMxL pin.

**Note 1:** These bits should not be changed after the PWM module is enabled (PTEN = 1).
**2:** State represents the active/inactive state of the PWMx depending on the POLH and POLL bits settings.

**REGISTER 19-2: ADCON1H: ADC CONTROL REGISTER 1 HIGH**

| 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-1 | R/W-1 | U-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| FORM | SHRRES1 | SHRRES0 | — | — | — | — | — |
| 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 **FORM:** Fractional Data Output Format bit
1 = Fractional
0 = Integer

bit 6-5 **SHRRES<1:0>:** Shared ADC Core Resolution Selection bits
11 = 12-bit resolution
10 = 10-bit resolution
01 = 8-bit resolution
00 = 6-bit resolution

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

**REGISTER 19-3: ADCON2L: ADC CONTROL REGISTER 2 LOW**

| R/W-0 | R/W-0 | U-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| REFCIE | REFERCIE(2) | — | EIEN | — | SHREISEL2(1) | SHREISEL1(1) | SHREISEL0(1) |
| bit 15 | | | | | | | bit 8 |

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | SHRADCS6 | SHRADCS5 | SHRADCS4 | SHRADCS3 | SHRADCS2 | SHRADCS1 | SHRADCS0 |
| 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 **REFCIE:** Band Gap and Reference Voltage Ready Common Interrupt Enable bit
1 = Common interrupt will be generated when the band gap will become ready
0 = Common interrupt is disabled for the band gap ready event

bit 14 **REFERCIE:** Band Gap or Reference Voltage Error Common Interrupt Enable bit(2)
1 = Common interrupt will be generated when the band gap or reference voltage error is detected
0 = Common interrupt is disabled for the band gap and reference voltage error event

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

bit 12 **EIEN:** Early Interrupts Enable bit
1 = The early interrupt feature is enabled for the input channels interrupts (when EISTATx flag is set)
0 = The individual interrupts are generated when conversion is done (when ANxRDY flag is set)

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

bit 10-8 **SHREISEL<2:0>:** Shared Core Early Interrupt Time Selection bits(1)
111 = Early interrupt is set and interrupt is generated 8 TADCORE clocks prior to when the data is ready
110 = Early interrupt is set and interrupt is generated 7 TADCORE clocks prior to when the data is ready
101 = Early interrupt is set and interrupt is generated 6 TADCORE clocks prior to when the data is ready
100 = Early interrupt is set and interrupt is generated 5 TADCORE clocks prior to when the data is ready
011 = Early interrupt is set and interrupt is generated 4 TADCORE clocks prior to when the data is ready
010 = Early interrupt is set and interrupt is generated 3 TADCORE clocks prior to when the data is ready
001 = Early interrupt is set and interrupt is generated 2 TADCORE clocks prior to when the data is ready
000 = Early interrupt is set and interrupt is generated 1 TADCORE clock prior to when the data is ready

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

bit 6-0 **SHRADCS<6:0>:** Shared ADC Core Input Clock Divider bits
These bits determine the number of TCORESRC (Core Source Clock) periods for one shared TADCORE (ADC Core Clock) period.
1111111 = 254 Core Source Clock periods
•
•
•
0000011 = 6 Core Source Clock periods
0000010 = 4 Core Source Clock periods
0000001 = 2 Core Source Clock periods
0000000 = 2 Core Source Clock periods

**Note 1:** For the 6-bit shared ADC core resolution (SHRRES<1:0> = 00), the SHREISEL<2:0> settings, from '100' to '111', are not valid and should not be used. For the 8-bit shared ADC core resolution (SHRRES<1:0> = 01), the SHREISEL<2:0> settings, '110' and '111', are not valid and should not be used.

**2:** To avoid false interrupts, the REFERCIE bit must be set only after the module is enabled (ADON = 1).

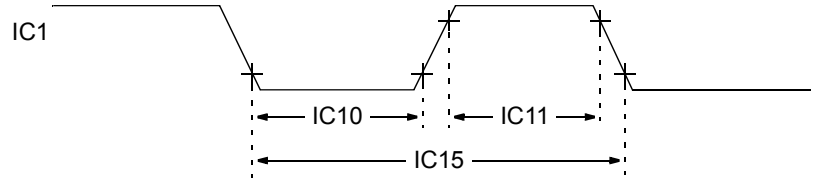
**REGISTER 19-20: ADTRIGxL: ADC CHANNEL TRIGGER x SELECTION REGISTER LOW (x = 0 to 3) (CONTINUED)**

bit 4-0 **TRGSRC(4x)<4:0>:** Trigger Source Selection for Corresponding Analog Inputs bits

11111 = ADTRG31
11110 = Reserved
11101 = Reserved
11100 = Reserved
11011 = Reserved
11010 = PWM Generator 3 current-limit trigger
11001 = PWM Generator 2 current-limit trigger
11000 = PWM Generator 1 current-limit trigger
10111 = Reserved
10110 = Output Compare 1 trigger
10101 = Reserved
10100 = Reserved
10011 = Reserved
10010 = Reserved
10001 = PWM Generator 3 secondary trigger
10000 = PWM Generator 2 secondary trigger
01111 = PWM Generator 1 secondary trigger
01110 = PWM secondary Special Event Trigger
01101 = Timer2 period match
01100 = Timer1 period match
01011 = Reserved
01010 = Reserved
01001 = Reserved
01000 = Reserved
00111 = PWM Generator 3 primary trigger
00110 = PWM Generator 2 primary trigger
00101 = PWM Generator 1 primary trigger
00100 = PWM Special Event Trigger
00011 = Reserved
00010 = Level software trigger
00001 = Common software trigger
00000 = No trigger is enabled

**NOTES:**

**FIGURE 25-6: INPUT CAPTURE 1 (IC1) TIMING CHARACTERISTICS**




**Note:** Refer to Figure 25-1 for load conditions.

**TABLE 25-27: INPUT CAPTURE 1 MODULE TIMING REQUIREMENTS**

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for Industrial<br>-40°C ≤ TA ≤ +125°C for Extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics(1)** | **Min.** | **Max.** | **Units** | **Conditions** | |
| IC10 | TccL | IC1 Input Low Time | Greater of:<br>12.5 + 25 or<br>(0.5 TCY/N) + 25 | — | ns | Must also meet Parameter IC15 | N = Prescale Value (1, 4, 16) |
| IC11 | TccH | IC1 Input High Time | Greater of:<br>12.5 + 25 or<br>(0.5 TCY/N) + 25 | — | ns | Must also meet Parameter IC15 | |
| IC15 | TccP | IC1 Input Period | Greater of:<br>25 + 50 or<br>(1 TCY/N) + 50 | — | ns | | |

**Note 1:** These parameters are characterized but not tested in manufacturing.

**FIGURE 25-16: SPI1 SLAVE MODE (FULL-DUPLEX, CKE = 1, CKP = 1, SMP = 0) TIMING CHARACTERISTICS**

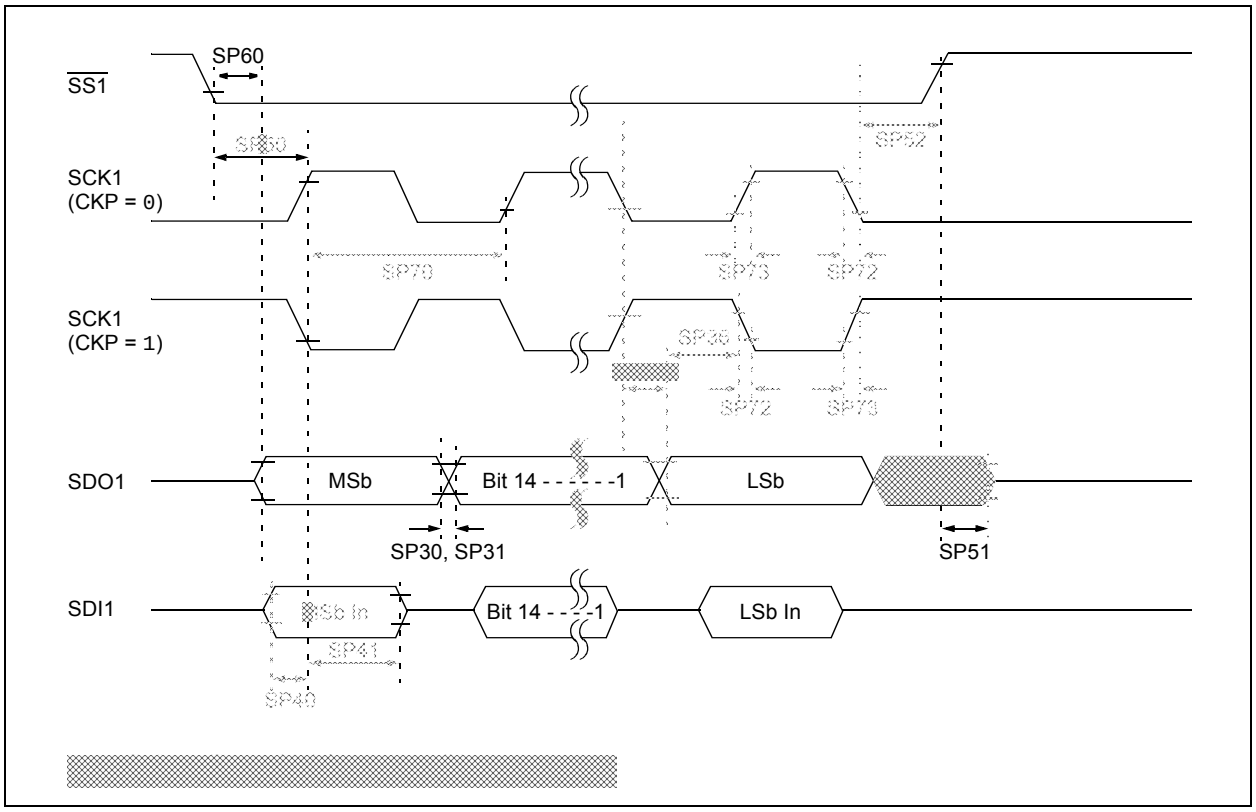


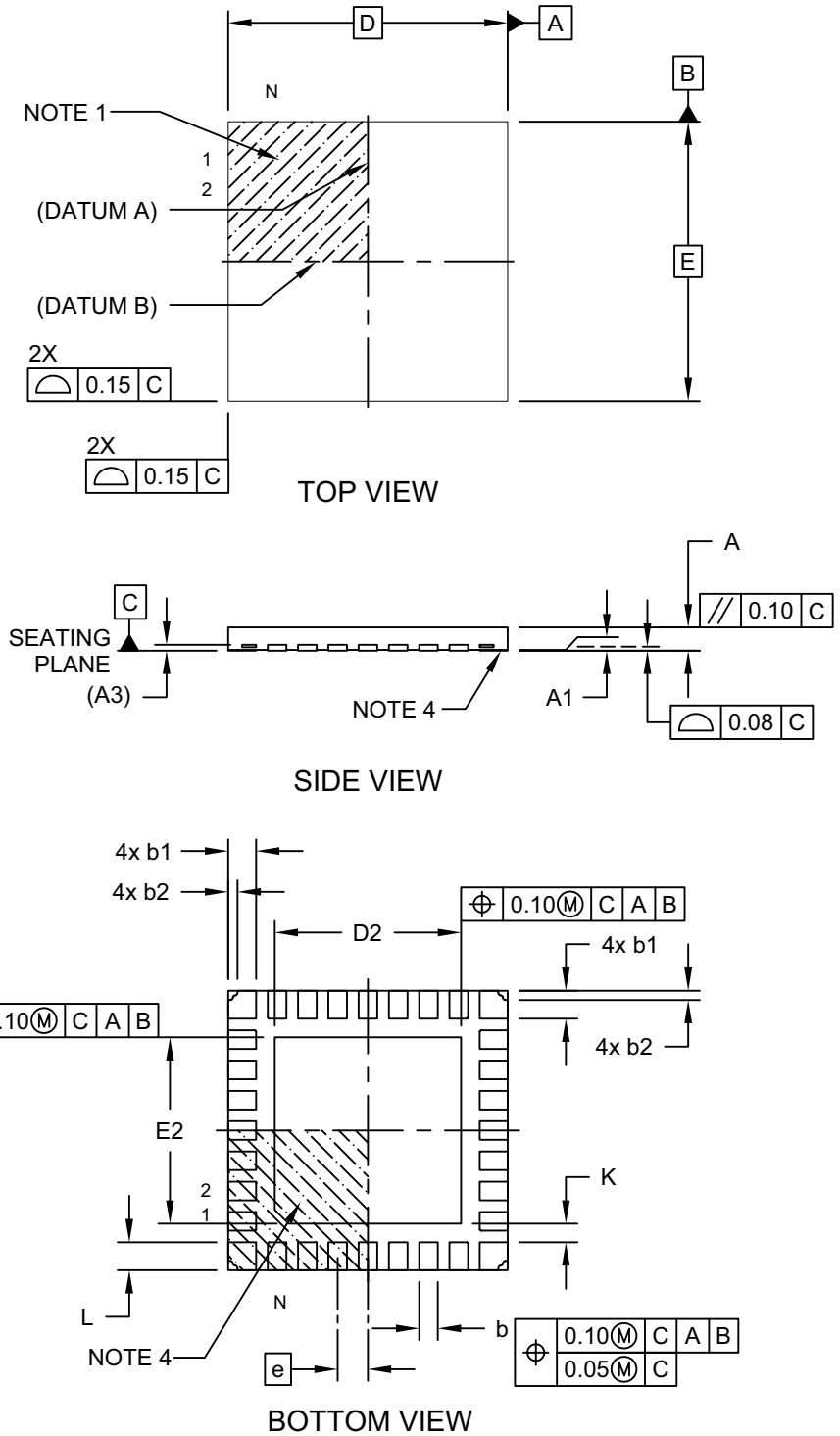
**TABLE 25-46: PGAx MODULE SPECIFICATIONS**

| AC/DC CHARACTERISTICS(1) | | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for Industrial<br>-40°C ≤ TA ≤ +125°C for Extended | | | | |
|---|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristic** | | **Min.** | **Typ.** | **Max.** | **Units** | **Comments** |
| PA01 | VIN | Input Voltage Range | | AVSS – 0.3 | — | AVDD + 0.3 | V | |
| PA02 | VCM | Common-Mode Input Voltage Range | | AVSS | — | AVDD – 1.6 | V | |
| PA03 | VOS | Input Offset Voltage | | -20 | — | +20 | mV | |
| PA04 | VOS | Input Offset Voltage Drift with Temperature | | — | ±15 | — | µV/°C | |
| PA05 | RIN+ | Input Impedance of Positive Input | | — | >1M \|\| 7 pf | — | Ω\|\| pF | |
| PA06 | RIN- | Input Impedance of Negative Input | | — | 10K \|\| 7 pf | — | Ω\|\| pF | |
| PA07 | GERR | Gain Error | | -2 | — | +2 | % | Gain = 4x and 8x |
| | | | | -3 | — | +3 | % | Gain = 16x |
| | | | | -4 | — | +4 | % | Gain = 32x and 64x |
| PA08 | LERR | Gain Nonlinearity Error | | — | — | 0.5 | % | % of full scale, Gain = 16x |
| PA09 | IDD | Current Consumption | | — | 2.0 | — | mA | Module is enabled with a 2-volt P-P output voltage swing |
| PA10a | BW | Small Signal Bandwidth (-3 dB) | G = 4x | — | 10 | — | MHz | |
| PA10b | | | G = 8x | — | 5 | — | MHz | |
| PA10c | | | G = 16x | — | 2.5 | — | MHz | |
| PA10d | | | G = 32x | — | 1.25 | — | MHz | |
| PA10e | | | G = 64x | — | 0.625 | — | MHz | |
| PA11 | OST | Output Settling Time to 1% of Final Value | | — | 0.4 | — | µs | Gain = 16x, 100 mV input step change |
| PA12 | SR | Output Slew Rate | | — | 40 | — | V/µs | Gain = 16x |
| PA13 | TGSEL | Gain Selection Time | | — | 1 | — | µs | |
| PA14 | TON | Module Turn On/Setting Time | | — | — | 10 | µs | |

**Note 1:** The PGAx module is functional at VBORMIN < VDD < VDDMIN, but with degraded performance. Unless otherwise stated, module functionality is tested, but not characterized.

**28-Lead Plastic Quad Flat, No Lead Package (MX) - 6x6x0.5mm Body [UQFN]**
**Ultra-Thin with 0.40 x 0.60 mm Terminal Width/Length and Corner Anchors**

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




Microchip Technology Drawing C04-0209 Rev C Sheet 1 of 2