The dsPIC33FJ32MC202/204 and dsPIC33FJ16MC304 family devices that you have received conform functionally to the current Device Data Sheet (DS70283K), except for the anomalies described in this document.

The silicon issues discussed in the following pages are for silicon revisions with the Device and Revision IDs listed in Table 1. The silicon issues are summarized in Table 2.

The errata described in this document will be addressed in future revisions of the dsPIC33FJ32MC202/204 and dsPIC33FJ16MC304 silicon.

Note: This document summarizes all silicon errata issues from all revisions of silicon, previous as well as current. Only the issues indicated in the last column of Table 2 apply to the current silicon revision (A6).

Data Sheet clarifications and corrections start on page 12, following the discussion of silicon issues.

The silicon revision level can be identified using the current version of MPLAB® IDE and Microchip’s programmers, debuggers and emulation tools, which are available at the Microchip corporate web site (www.microchip.com).

For example, to identify the silicon revision level using MPLAB IDE in conjunction with MPLAB ICD 3 or PICkit™ 3:

1. Using the appropriate interface, connect the device to the MPLAB ICD 3 programmer/debugger or PICkit 3.
2. From the main menu in MPLAB IDE, select Configure>Select Device, and then select the target part number in the dialog box.
3. Select the MPLAB hardware tool (Debugger>Select Tool).
4. Perform a “Connect” operation to the device (Debugger>Connect). Depending on the development tool used, the part number and Device Revision ID value appear in the Output window.

Note: If you are unable to extract the silicon revision level, please contact your local Microchip sales office for assistance.

The Device and Revision ID values for the various dsPIC33FJ32MC202/204 and dsPIC33FJ16MC304 silicon revisions are shown in Table 1.

### TABLE 1: SILICON DEVREV VALUES

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Device ID(1)</th>
<th>Revision ID for Silicon Revision(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>A2</td>
</tr>
<tr>
<td>dsPIC33FJ32MC202</td>
<td>0x0F09</td>
<td>0x3001</td>
</tr>
<tr>
<td>dsPIC33FJ32MC204</td>
<td>0x0F0B</td>
<td></td>
</tr>
<tr>
<td>dsPIC33FJ16MC304</td>
<td>0x0F03</td>
<td></td>
</tr>
</tbody>
</table>

Note 1: The Device and Revision IDs (DEVID and DEVREV) are located at the last two implemented addresses in program memory.

2: Refer to the “dsPIC33F/PIC24H Flash Programming Specification” (DS70152) for detailed information on Device and Revision IDs for your specific device.
## TABLE 2: SILICON ISSUE SUMMARY

<table>
<thead>
<tr>
<th>Module</th>
<th>Feature</th>
<th>Item Number</th>
<th>Issue Summary</th>
<th>Affected Revisions(1)</th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>JTAG</td>
<td>Flash Programming</td>
<td>1</td>
<td>JTAG programming does not work.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UART</td>
<td>High-Speed Mode</td>
<td>2</td>
<td>The auto-baud feature may not calculate the correct baud rate when the Baud Rate Generator (BRG) is set up for 4x mode.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UART</td>
<td>Auto-Baud</td>
<td>3</td>
<td>With the auto-baud feature selected, the Sync Break character (0x55) may be loaded into the FIFO as data.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UART</td>
<td>Auto-Baud</td>
<td>4</td>
<td>The auto-baud feature measures baud rate inaccurately for certain baud rate and clock speed combinations.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UART</td>
<td>Auto-Baud</td>
<td>5</td>
<td>When an auto-baud is detected, the receive interrupt may occur twice.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UART</td>
<td>IR Mode</td>
<td>6</td>
<td>The 16x baud clock signal on the BCLK pin is present only when the module is transmitting.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UART</td>
<td>High-Speed Mode</td>
<td>7</td>
<td>When the UART is in 4x mode (BRGH = 1) and using two Stop bits (STSEL = 1), it may sample the first Stop bit instead of the second one.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPI</td>
<td>SCKx Pins</td>
<td>8</td>
<td>The SPIxCON1 DISSCK bit does not influence port functionality.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I²C™ SFR Writes</td>
<td></td>
<td>9</td>
<td>The BCL bit in I2CxSTAT can be cleared only with 16-bit operation and can be corrupted with 1-bit or 8-bit operations on I2CxSTAT.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I²C</td>
<td>10-Bit Addressing</td>
<td>10</td>
<td>When the I²C module is configured for 10-bit addressing using the same address bits (A10 and A9) as other I²C devices, A10 and A9 bits may not work as expected.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Identification</td>
<td>Extended Temperature</td>
<td>11</td>
<td>Revision A2 devices marked as Extended temperature range (E) devices support only Industrial temperature range (I).</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UART</td>
<td>Interrupts</td>
<td>12</td>
<td>The UART error interrupt may not occur, or may occur at an incorrect time, if multiple errors occur during a short period of time.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UART</td>
<td>IR Mode</td>
<td>13</td>
<td>When the UART module is operating in 8-bit mode (PDSEL&lt;1:0&gt; = 0x) and using the IrDA® encoder/decoder (IREN = 1), the module incorrectly transmits a data payload of 80h as 00h.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal Voltage Regulator</td>
<td>Sleep Mode</td>
<td>14</td>
<td>When the VREGS bit (RCON&lt;8&gt;) is set to a logic ‘0’, device may reset and higher Sleep current may be observed.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSV Operations</td>
<td>—</td>
<td>15</td>
<td>An address error trap occurs in certain addressing modes when accessing the first four bytes of any PSV page.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I²C</td>
<td>10-Bit Addressing</td>
<td>16</td>
<td>When the I²C module is configured as a 10-bit slave with an address of 0x02, the I2CxRCV register content for the lower address byte is 0x01 rather than 0x02.</td>
<td>X X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note 1:** Only those issues indicated in the last column apply to the current silicon revision.
### TABLE 2: SILICON ISSUE SUMMARY (CONTINUED)

<table>
<thead>
<tr>
<th>Module</th>
<th>Feature</th>
<th>Item Number</th>
<th>Issue Summary</th>
<th>Affected Revisions&lt;sup&gt;(1)&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>I²C</td>
<td>—</td>
<td>17.</td>
<td>With the I²C module enabled, the PORT bits and external interrupt input functions (if any) associated with SCLx and SDAx pins will not reflect the actual digital logic levels on the pins.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>I²C</td>
<td>10-Bit Addressing</td>
<td>18.</td>
<td>The 10-bit slave does not set the RBF flag or load the I2CxRCV register, on address match if the Least Significant bits (LSbs) of the address are the same as the 7-bit reserved addresses.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>I²C</td>
<td>—</td>
<td>19.</td>
<td>After the ACKSTAT bit is set when receiving a NACK, it may be cleared by the reception of a Start or Stop bit.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>CPU</td>
<td>EXCH instruction</td>
<td>20.</td>
<td>The EXCH instruction does not execute correctly.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>PWM</td>
<td>Debug Mode</td>
<td>21.</td>
<td>PTMR does not keep counting down after halting code execution in Debug mode.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>PWM</td>
<td>Doze Mode</td>
<td>22.</td>
<td>The Motor Control PWM module generates more interrupts than expected when Doze mode is used and the output postscaler value is different than 1:1.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>QEI</td>
<td>Interrupts</td>
<td>23.</td>
<td>The QEI module does not generate an interrupt in a particular overflow condition.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>PGEC3/PGED3 Programming Pins</td>
<td>Device Programming</td>
<td>24.</td>
<td>When using the PGEC3/PGED3 pins for device programming, the programming time may be slower as compared to other available PGECx/PGEDx pin pairs.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>UART</td>
<td>Break Character Generation</td>
<td>25.</td>
<td>The UART module will not generate back-to-back Break characters.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>QEI</td>
<td>Timer Gated Accumulation Mode</td>
<td>26.</td>
<td>When timer gated accumulation is enabled, the QEI does not generate an interrupt on every falling edge.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>QEI</td>
<td>Timer Gated Accumulation Mode</td>
<td>27.</td>
<td>When timer gated accumulation is enabled and an external signal is applied, the POSCNT increments and generates an interrupt after a match with MAXCNT.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>ADC</td>
<td>Current Consumption in Sleep Mode</td>
<td>28.</td>
<td>If the ADC module is in an enabled state when the device enters Sleep mode, the power-down current (IpD) of the device may exceed the device data sheet specifications.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>All</td>
<td>150ºC Operation</td>
<td>29.</td>
<td>These revisions of silicon only support 140ºC operation instead of 150ºC for high-temperature operation.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>CPU</td>
<td>Interrupt Disable</td>
<td>30.</td>
<td>When a previous DISI instruction is active (i.e., the DISICNT register is non-zero), and the value of the DISICNT register is updated manually, the DISICNT register freezes and disables interrupts permanently.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>CPU</td>
<td>div.sd</td>
<td>31.</td>
<td>When using the div.sd instruction, the Overflow bit is not getting set when an overflow occurs.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>UART</td>
<td>TX Interrupt</td>
<td>32.</td>
<td>A Transmit (TX) interrupt may occur before the data transmission is complete.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>JTAG</td>
<td>Flash Programming</td>
<td>33.</td>
<td>JTAG Flash programming is not supported.</td>
<td>X X X X X</td>
</tr>
<tr>
<td>UART</td>
<td>Transmit Mode</td>
<td>34.</td>
<td>TRMT bit is set before the Shift register is empty.</td>
<td>X X X X X</td>
</tr>
</tbody>
</table>

**Note 1:** Only those issues indicated in the last column apply to the current silicon revision.
Silicon Errata Issues

Note: This document summarizes all silicon errata issues from all revisions of silicon, previous as well as current. Only the issues indicated by the shaded column in the following tables apply to the current silicon revision (A6).

1. Module: JTAG
   JTAG programming does not work.
   **Work around**
   None.
   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

2. Module: UART
   The auto-baud feature may not calculate the correct baud rate when the High Baud Rate Enable bit, BRGH, is set. With the BRGH bit set, the baud rate calculation used is the same as BRG = 0.
   **Work around**
   If the auto-baud feature is needed, use the Low Baud Rate mode by clearing the BRGH bit.
   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

3. Module: UART
   With the auto-baud feature selected, the Sync Break character (0x55) may be loaded into the FIFO as data.
   **Work around**
   To prevent the Sync Break character from being loaded into the FIFO, load the UxBRG register with either 0x0000 or 0xFFFF prior to enabling the auto-baud feature (ABAUD = 1).
   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

4. Module: UART
   The auto-baud feature may miscalculate for certain baud rate and clock speed combinations, resulting in a BRG value that is greater than or less than the expected value by 1. This may result in reception or transmission failures.
   **Work around**
   Test the auto-baud rate at various clock speed and baud rate combinations that would be used in an application. If an inaccurate BRG value is generated, manually correct the baud rate in user software.
   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

5. Module: UART
   When an auto-baud is detected, the receive interrupt may occur twice. The first interrupt occurs at the beginning of the Start bit and the second after reception of the Sync field character.
   **Work around**
   If an extra interrupt is detected, ignore the additional interrupt.
   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

6. Module: UART
   When the UART is configured for IR interface operations (UxMODE<9:8> = 11), the 16x baud clock signal on the BCLK pin is present only when the module is transmitting. The pin is Idle at all other times.
   **Work around**
   Configure one of the output compare modules to generate the required baud clock signal when the UART is receiving data or in an Idle state.
   **Affected Silicon Revisions**
<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
7. Module: UART
When the UART is in 4x mode (BRGH = 1) and using two Stop bits (STSEL = 1), it may sample the first Stop bit instead of the second one. This issue does not affect the other UART configurations.

Work around
Use the 16x baud rate option (BRGH = 0) and adjust the baud rate accordingly.

Affected Silicon Revisions

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

8. Module: SPI
When the SPI module is enabled, setting the DISSCK bit in the SPIxCON1 register does not allow the user application to use the SCKx pin as a general purpose I/O pin.

Work around
None.

Affected Silicon Revisions

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

9. Module: I²C
The BCL bit in I2CxSTAT can be cleared only with 16-bit operation and can be corrupted with 1-bit or 8-bit operations on I2CxSTAT.

Work around
Use 16-bit operations to clear BCL.

Affected Silicon Revisions

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

10. Module: I²C
If there are two I²C devices on the bus, one of them is acting as the Master receiver and the other as the Slave transmitter. Suppose that both devices are configured for 10-Bit Addressing mode, and have the same value in the A10 and A9 bits of their addresses. When the Slave select address is sent from the Master, both the Master and Slave Acknowledges it. When the Master sends out the read operation, both the Master and the Slave enter into Read mode and both of them transmit the data. The resultant data will be the ANDing of the two transmissions.

Work around
Use different addresses including the higher two bits (A10 and A9) for different modules.

Affected Silicon Revisions

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

11. Module: Product Identification
Revision A2 devices marked as Extended temperature range (E) devices support only the Industrial temperature range (I).

Work around
Use Revision A3 or newer devices marked as Extended temperature range (E) devices.

Affected Silicon Revisions

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

12. Module: UART
The UART error interrupt may not occur, or may occur at an incorrect time, if multiple errors occur during a short period of time.

Work around
Read the error flags in the UxSTA register whenever a byte is received to verify the error status. In most cases, these bits will be correct, even if the UART error interrupt fails to occur.

Affected Silicon Revisions

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
13. Module: UART

When the UART is operating in 8-bit mode (PDSEL<1:0> = 0x) and using the IrDA encoder/decoder (IREN = 1), the module incorrectly transmits a data payload of 80h as 00h.

**Work around**
None.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th></th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

14. Module: Internal Voltage Regulator

When the VREGS bit (RCON<8>) is set to a logic '0', the device may reset and a higher Sleep current may be observed.

**Work around**
Ensure the VREGS bit (RCON<8>) is set to a logic '1' for device Sleep mode operation.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th></th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

15. Module: PSV Operations

An address error trap occurs in certain addressing modes when accessing the first four bytes of a PSV page. This occurs only when using the following addressing modes:

- MOV.D
- Register Indirect Addressing (Word or Byte mode) with pre/post-decrement

**Work around**
Do not perform PSV accesses to any of the first four bytes using the above addressing modes. For applications using the C language, MPLAB C30 Version 3.11 or higher, provides the following command-line switch that implements a work around for the erratum.

```
-merrata=psv_trap
```

Refer to the readme.txt file in the MPLAB C30 v3.11 toolsuite for further details.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th></th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

16. Module: I2C

When the I2C module is configured as a 10-bit slave with an address of 0x02, the I2CxRCV register content for the lower address byte is 0x01, rather than 0x02; however, the module Acknowledges both address bytes.

**Work around**
None.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th></th>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
17. Module: I\(^2\)C

With the I\(^2\)C module enabled, the PORT bits and external interrupt input functions (if any) associated with the SCLx and SDAx pins do not reflect the actual digital logic levels on the pins.

**Work around**

If the SDAx and/or SCLx pins need to be polled, these pins should be connected to other port pins in order to be read correctly. This issue does not affect the operation of the I\(^2\)C module.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

18. Module: I\(^2\)C

In 10-Bit Addressing mode, some address matches do not set the RBF flag or load the I2Cx Receive register, I2CxRCV, if the lower address byte matches the reserved addresses. In particular, these include all addresses with the form 'xx0000xxxx' and 'xx1111xxxx', with the following exceptions:

- '001111000x'
- '011111001x'
- '1011111010x'
- '1111111011x'

**Work around**

Ensure that the lower address byte in 10-Bit Addressing mode does not match any 7-bit reserved addresses.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

19. Module: I\(^2\)C

When the I\(^2\)C module is operating in either Master or Slave mode, after the ACKSTAT bit is set when receiving a NACK, it may be cleared by the reception of a Start or Stop bit.

**Work around**

Store the value of the ACKSTAT bit immediately after receiving a NACK.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

20. Module: CPU

The EXCH instruction does not execute correctly.

**Work around**

If writing source code in assembly, the recommended work around is to replace:

```
EXCH Wsource, Wdestination
```

with:

```
PUSH Wdestination
MOV Wsource, Wdestination
POP Wsource
```

If using the MPLAB C30 C compiler, specify the compiler option: `-merrata=exch` *(Project > Build Options > Projects > MPLAB C30 > Use Alternate Settings)*.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

21. Module: PWM

If the PTDIR bit is set (when PTMR is counting down), and the CPU execution is halted (after a breakpoint is reached), PTMR will start counting up as if PTDIR was zero.

**Work around**

None.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

22. Module: PWM

When the device is operated in Doze mode and the Motor Control PWM module has a postscaler set to any value different than 1:1 (PTOPS > 0 in the PxTCON register), the Motor Control PWM module generates more interrupts than expected.

**Work around**

Do not use Doze mode with the Motor Control PWM if the time base output postscaler is different than 1:1 (PTOPS > 0 in the PxTCON register).

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
23. Module: QEI

The Quadrature Encoder Interface (QEI) module does not generate an interrupt when MAXCNT is set to 0xFFFF and the following events occur:

1. POSCNT underflows from 0x0000 to 0xFFFF.
2. POSCNT stops.
3. POSCNT overflows from 0xFFFF to 0x0000.

This sequence of events occurs when the motor is running in one direction, which causes POSCNT to underflow to 0xFFFF. Then, if the motor stops and starts running in the opposite direction, an overflow from 0xFFFF to 0x0000 will be generated. The QEI module does not generate an interrupt when this condition occurs.

**Work around**

To prevent this condition from occurring, set MAXCNT to 0x7FFF, which will cause an interrupt to be generated by the QEI module.

In addition, a global variable could be used to keep track of bit 15, so that when an overflow or underflow condition is present on POSCNT, the variable will toggle bit 15. **Example 1** shows the code required for this global variable.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

**Example 1:**

```c
unsigned int POSCNT_b15 = 0;
unsigned int Motor_Position = 0;

int main(void)
{
    // ... User's code
    MAXCNT = 0x7FFF; // Instead of 0xFFFF
    Motor_Position = POSCNT_b15 + POSCNT;
    // ... User's code
}

void __attribute__((__interrupt__)) _QEIInterrupt(void)
{
    IFSxbits.QEIIF = 0; // Clear QEI interrupt flag
    // x=2 for dsPIC30F
    // x=3 for dsPIC33F
    POSCNT_b15 ^= 0x8000; // Overflow or Underflow
}
```


When using the PGEC3/PGED3 pins for device programming, the programming time may be slower as compared to other available PGECx/PGEDx pin pairs, because the Enhanced In-Circuit Serial Programming™ (ICSP™) algorithm cannot be executed on this pin pair.

Refer to the “dsPIC33F/PIC24H Flash Programming Specification” (DS70152) for additional information on this limitation.

**Work around**

Use alternate PGECx/PGEDx programming pin pairs.

**Affected Silicon Revisions**
25. Module: UART

The UART module will not generate consecutive Break characters. Trying to perform a back-to-back Break character transmission will cause the UART module to transmit the dummy character used to generate the first Break character instead of transmitting the second Break character. Break characters are generated correctly if they are followed by non-Break character transmission.

**Work around**

None.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

26. Module: QEI

When the TQCS and TQGATE bits in the QEIxCON register are set, a QEI interrupt should be generated after an input pulse on the QEA input. This interrupt is not generated in the affected silicon.

**Work around**

None.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

27. Module: QEI

When the TQCS and TQGATE bits in the QEIxCON register are set, the POSCNT counter should not increment, but erroneously does, and if allowed to increment to match MAXCNT, a QEI interrupt will be generated.

**Work around**

To prevent the erroneous increment of POSCNT while running the QEI in Timer Gated Accumulation mode, initialize MAXCNT = 0.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

28. Module: ADC

If the ADC module is in an enabled state when the device enters Sleep mode as a result of executing a PWRSAV #0 instruction, the device power-down current (IPD) may exceed the specifications listed in the device data sheet. This may happen even if the ADC module is disabled by clearing the ADON bit prior to entering Sleep mode.

**Work around 1:**

In order to remain within the IPD specifications listed in the device data sheet, the user software must completely disable the ADC module by setting the ADC module disable bit in the corresponding Peripheral Module Disable register (PMDx), prior to executing a PWRSAV #0 instruction.

**Note:** The ADC module must be re-initialized by the user application before resuming ADC operation.

**Work around 2:**

If the ADC module was previously initialized and enabled before entering Sleep, execute the lines of code provided in **Example 2**.

**Note:** Unlike **Work around 1**, the user application does not need to re-initialize the ADC module; however, it is necessary to re-enable the ADC module by setting the ADON bit after waking from Sleep.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

**EXAMPLE 2:**

```c
ADICON1bits.ADON = 0; //Disable the ADC module
__asm__ volatile("REPEAT #50"); //Wait 50 Tcy
__asm__ volatile("NOP"); //Repeat NOP 51 times
Sleep(); // Execute PWRSAV #0 and go to Sleep
```
29. Module: All

The affected silicon revisions listed below are not warranted for operation at +150ºC.

**Work around**

Only use the affected revisions of silicon for the high-temperature operating range, from -40ºC to +140ºC.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

30. Module: CPU

When a previous DISI instruction is active (i.e., the DISICNT register is non-zero), and the value of the DISICNT register is updated manually, the DISICNT register freezes and disables interrupts permanently.

**Work around**

Avoid updating the DISICNT register manually. Instead, use the DISI #n instruction with the required value for 'n'.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

31. Module: CPU

When using the Signed 32-by-16-bit Division instruction, div.sd, the Overflow bit does not always get set when an overflow occurs.

**Work around**

Test for and handle overflow conditions outside of the div.sd instruction.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

32. Module: UART

When using UTXISEL<1:0> = 01 (interrupt when last character is shifted out of the Transmit Shift Register) and the final character is being shifted out through the Transmit Shift Register, the Transmit (TX) interrupt may occur before the final bit is shifted out.

**Work around**

If it is critical that the interrupt processing occur only when all transmit operations are complete, hold off the interrupt routine processing by adding a loop at the beginning of the routine that polls the Transmit Shift Register Empty bit (TRMT) before processing the rest of the interrupt.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>A5</td>
<td>A6</td>
</tr>
</tbody>
</table>

33. Module: JTAG

JTAG Flash programming is not supported.

**Work around**

None.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>A5</td>
<td>A6</td>
</tr>
</tbody>
</table>
34. Module: UART

When the UART is in Transmit mode, the TRMT bit may be set before the Shift register is empty. In back-to-back transmission, if the data is loaded into the U1TXREG register when the TRMT bit is set, the new byte transmission starts immediately and the Stop bit may be abbreviated, as shown in the condition below:

- When BRGH (U1MODE<3>) = 1, the Stop bit will be shortened by 1/4th of a baud rate period.
- When BRGH (U1MODE<3>) = 0, the Stop bit will be shortened by 1/16th of a baud rate period.

**Work around**

When using the TRMT bit to load the U1TXREG, after the TRMT bit is set, insert a delay, as defined below, before loading the U1TXREG.

When the TRMT bit is set:

- If BRGH = 1, insert a delay of at least 1/4th of the baud rate period before loading U1TXREG.
- If BRGH = 0, insert a delay of at least 1/16th of the baud rate period before loading U1TXREG.

**Affected Silicon Revisions**

<table>
<thead>
<tr>
<th>A2</th>
<th>A3</th>
<th>A4</th>
<th>A5</th>
<th>A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Data Sheet Clarifications

The following typographic corrections and clarifications are to be noted for the latest version of the device data sheet (DS70283K):

**Note:** Corrections are shown in **bold**. Where possible, the original bold text formatting has been removed for clarity.

1. **Module: Electrical Characteristics**

   The specifications for F21a in Table 24-19 have been updated as shown in **bold** below.

   **TABLE 24-19: INTERNAL RC ACCURACY**

<table>
<thead>
<tr>
<th>AC CHARACTERISTICS</th>
<th>Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)</th>
<th>Operating temperature</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Param No.</td>
<td>Characteristic</td>
<td>Min</td>
<td>Typ</td>
</tr>
<tr>
<td>LPRC @ 32.768 kHz(^{(1,2)})</td>
<td>F21a LPRC</td>
<td>-20</td>
<td>±10</td>
</tr>
<tr>
<td>F21b LPRC</td>
<td>-40</td>
<td>—</td>
<td>+40</td>
</tr>
</tbody>
</table>

   **Note:**
   1. Change of LPRC frequency as VDD changes.
   2. LPRC impacts the Watchdog Timer Time-out Period (TWDT1). See Section 19.4 “Watchdog Timer (WDT)” for more information.

2. **Module: 19.0 Universal Asynchronous Receiver Transmitter (UART)**

   In Section 19.0, a note is added above Figure 19-1:

   **Note:** This note applies to the applications using the UART module for LIN/J2602 applications. The LIN/J2602 standard specifies that the inter-byte space should be a non-negative number. The inter-byte space is defined as the time between the end of the Stop bit of the preceding data and the start of the following data. It is recommended to load the data to the U1TXREG after the received Stop bit is completed. Due to the half-duplex nature of the LIN/J2602 transceiver, failing to provide non-negative inter-byte space will result in a truncated Stop bit. The loading of U1TXREG after the receive interrupt should be delayed ¾ of the Stop bit time.
APPENDIX A: REVISION HISTORY

Initial release of this document; issued for revision A2, A3, A4 and A5 silicon.
Includes silicon issues 1 (JTAG), 2-7 (UART), 8 (SPI), 9-10 (I²C), 11 (Product Identification), 12-13 (UART) 14 (Internal Voltage Regulator), 15 (PSV Operations), 16-19 (I²C), 20 (CPU), 21-22 (PWM) and 23 (QEI).
This document replaces the following errata document: DS80338, “dsPIC33FJ32MC202/204 and dsPIC33FJ16MC304 Rev. A2/A3/A4 Silicon Errata”


Updated silicon issue 20 (CPU).
Added silicon issue 28 (ADC) and data sheet clarification 1 (DC Characteristics: I/O Pin Input Specifications).

Updated the work around in silicon issue 28 (ADC).
Added silicon issue 29 (All).

Added silicon revision A6 references throughout the document.

Removed data sheet clarification issue 1.

Added silicon issues 30 (CPU), 31 (CPU), 32 (UART), and 33 (JTAG).

Rev H Document (12/2013)
Added data sheet clarification 1 (Electrical Characteristics).

Added silicon issue 34 (UART).
Added data sheet clarification 2 (19.0 Universal Asynchronous Receiver Transmitter (UART)).
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and Uni/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, ETHERSYNCH, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and QUIET-WIRE are registered trademarks of Microchip Technology Incorporated in the U.S.A.


SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.


ISBN: 978-1-5224-0516-0
Worldwide Sales and Service

**AMERICAS**

**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277

Technical Support:
http://www.microchip.com/support

Web Address:
www.microchip.com

**Atlanta**
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

**Austin, TX**
Tel: 512-257-3370

**Boston**
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

**Chicago**
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

**Cleveland**
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

**Dallas**
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

**Detroit**
Novi, MI
Tel: 248-848-4000

**Houston, TX**
Tel: 281-894-5983

**Indianapolis**
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

**Los Angeles**
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

**New York, NY**
Tel: 631-435-6000

**San Jose, CA**
Tel: 408-735-9110

**Canada - Toronto**
Tel: 905-673-0699
Fax: 905-673-6509

**ASIA/PACIFIC**

**Asia Pacific Office**
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

**Hong Kong**
Tel: 852-2943-5100
Fax: 852-2401-3431

**Australia - Sydney**
Tel: 61-2-9888-6733
Fax: 61-2-9888-6755

**China - Beijing**
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

**China - Chengdu**
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

**China - Dongguan**
Tel: 86-769-8702-9880

**China - Hangzhou**
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

**China - Hong Kong SAR**
Tel: 852-2943-5100
Fax: 852-2401-3431

**China - Nanjing**
Tel: 86-25-8473-2460
Fax: 86-25-8474-2470

**China - Qingdao**
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

**China - Shanghai**
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

**China - Shenyang**
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

**China - Shenzhen**
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

**China - Wuhan**
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

**China - Xian**
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

**ASIA/PACIFIC**

**China - Xiamen**
Tel: 86-592-2388138
Fax: 86-592-2388130

**China - Zhuhai**
Tel: 86-756-3210040
Fax: 86-756-3210049

**India - Bangalore**
Tel: 91-80-3090-4444
Fax: 91-80-3090-4143

**India - New Delhi**
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

**India - Pune**
Tel: 91-20-3019-1500

**Japan - Osaka**
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

**Japan - Tokyo**
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

**Korea - Daegu**
Tel: 82-53-744-4301
Fax: 82-53-744-4302

**Korea - Seoul**
Tel: 82-2-554-7200
Fax: 82-2-554-5932 or 82-2-558-5934

**Malaysia - Kuala Lumpur**
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

**Malaysia - Penang**
Tel: 60-4-227-8870
Fax: 60-4-227-4068

**Philippines - Manila**
Tel: 63-2-634-9065
Fax: 63-2-634-9069

**Singapore**
Tel: 65-6334-8870
Fax: 65-6334-8850

**Taiwan - Hsin Chu**
Tel: 886-3-5778-366
Fax: 886-3-5770-955

**Taiwan - Kaohsiung**
Tel: 886-7-213-7828

**Taiwan - Taipei**
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

**Thailand - Bangkok**
Tel: 66-2-694-1351
Fax: 66-2-694-1350

**EUROPE**

**Austria - Wels**
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

**Denmark - Copenhagen**
Tel: 45-4450-2826
Fax: 45-4485-2829

**France - Paris**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

**Germany - Dusseldorf**
Tel: 49-2129-3766400

**Germany - Karlsruhe**
Tel: 49-721-625370

**Germany - Munich**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

**Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

**Italy - Venice**
Tel: 39-049-7625286

**Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

**Poland - Warsaw**
Tel: 48-22-3325737

**Spain - Madrid**
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

**Sweden - Stockholm**
Tel: 46-8-5090-4654

**UK - Wokingham**
Tel: 44-118-921-5800
Fax: 44-118-921-5820

07/14/15