dsPIC30F1010/202X Rev. A2 Silicon Errata

The dsPIC30F1010/202X (Rev. A2) devices that you received were found to conform to the specifications and functionality described in the following documents:

- *dsPIC30F1010/202X Family Data Sheet* (DS70178)
- *dsPIC30F/33F Programmer's Reference Manual* (DS70157)
- *dsPIC30F Family Reference Manual* (DS70046)

The exceptions to the specifications in the documents listed above are described in this section. These exceptions are described for the devices listed below:

- dsPIC30F1010
- dsPIC30F2020
- dsPIC33F2023

dsPIC30F1010/202X Rev. A2 silicon is identified by performing a "Reset and Connect" operation to the device using MPLAB® ICD 2 with MPLAB IDE v7.41.03 or later. The output window will show a successful connection to the device specified in Configure>Select Device.

The errata described in this section will be addressed in future revisions of silicon.

Silicon Errata Summary

The following list summarizes the errata described in this document:

1. PWM Dead Time

   If a value less than 0x0010 is written to the DTRx and ALTDTRx registers, either or both of the PWMHx and PWMLx outputs will not function.

2. PWM Duty Cycle

   Duty cycle resolution is not 1.1 ns over the entire duty cycle range.

3. PWM Triggers

   The PWM Special Event Trigger and PWM Individual Trigger do not function near the beginning of the PWM period.

4. PWM Override Enable

   The PWM override feature does not work correctly.

5. PWM Duty Cycle

   When the PWM module is operated with Immediate Duty Cycle updates enabled, any duty cycle value less than or equal to 0x0010 causes the PWM outputs to flip to the inverted state.

6. PWM Override Priority

   The PWM Fault, Current-Limit, and Output Override priorities do not work correctly.

7. PWM Jitter

   The PWM output may exhibit an occasional jitter proportional to the operating speed of the dsPIC30F1010/202X device.

8. ADC Global Software Trigger

   The Global Software Trigger bit (GSWTRG in the ADCON register) is not reset unless the PxRDY bits in the ADSTAT register are reset.

9. ADC Sample and Hold Timing

   The resolution of the PWM to ADC sample and hold trigger timing is 41.6 ns instead of the 8 ns specified in the device data sheet.

10. ADC Interrupts

    Individual ADC Interrupts for the ADC pin pairs do not work.

11. ADC Conversion Rate

    The maximum conversion rate for the ADC module is 1.5 Msps.

12. ADC Shared Sample and Hold Circuit

    Depending on conversion configuration, ADC inputs that do not have dedicated sample and hold circuits may produce inaccurate conversion results.

13. Current Reset Mode

    Setting the XPRES bit in the PWMCONx register should enable a current-limit source to reset the PWM period when the PWM generated is configured in Independent Time Base mode. This functionality is not working correctly.

14. Output Compare Module

    The output compare module will produce a glitch on the output when an I/O pin is initially set high and the module is configured to drive the pin low at a specified time.

15. Output Compare Module in PWM Mode

    The output compare module will miss one compare event when the duty cycle register value is updated from 0x0000 to 0x0001.

16. Output Compare Module

    In Dual Compare Match mode, the OCx output is not reset when the OCxR and OCxRS registers are loaded with values having a difference of ‘1’.

17. SPI Module in Slave Select Mode

    The SPI module Slave Select functionality will not work correctly.

18. SPI Module in Frame Master Mode

    The SPI module will fail to generate frame synchronization pulses in Frame Master mode if FRMDLY = 1.

19. SPI Module

    The SMP bit does not have any effect when the SPI module is configured for a 1:1 prescale factor in Master mode.

20. UART Module

    If the Baud Rate Generator (BRG) register contains an odd value and the parity option is enabled, the module may falsely indicate parity errors.

21. UART Module

    The Receive Buffer Overrun Error Status bit may be set prematurely.

22. UART Module

    UART receptions may be corrupted in high baud rate mode (BRGH = 1).

23. UART Module

    UTXISEL0 bit in the UxSTA register is always read as zero regardless of the value written to it.

24. UART Module

    The auto-baud feature does not work properly in high baud rate mode (BRGH = 1).

25. UART Module

    When the auto-baud feature is enabled, the Sync Break character (0x55) may be loaded into the FIFO as data.

26. UART Module (IrDA® Reception)

    The operation of the RXINV bit in the UxMODE register is inverted.

27. UART Module

    The auto-baud feature measures baud rate inaccurately for certain baud rate and clock speed combinations.

28. I^{2}C™ Module

    The bus collision status bit does not get set when a bus collision occurs during a Restart or Stop event.

29. I^{2}C Module

    The I2CxTRN register can be written to even if a write collision is detected.

30. I^{2}C Module

    The ACKSTAT bit does not reflect the status of a transmission received from an I^{2}C Slave device.

31. I^{2}C Module

    The D_A status bit in the I2CxSTAT register does not get set on a write to the I2CxTRN register by an I^{2}C Slave device.

32. $\overline{\text{MCLR}}$ pin

    When the dsPIC® DSC is operated with the PLL enabled, the $\overline{\text{MCLR}}$ pin does not operate correctly in the event of a brown-out condition.

33. Decimal Adjust Instruction

    The decimal adjust instruction, `DAW.b`, may improperly clear the Carry bit, C (SR<0>).

34. Sleep Mode

    Execution of the Sleep instruction (`PWRSAV #0`) may cause incorrect program operation after the device wakes up from Sleep. The current consumption during Sleep may also increase beyond the specifications listed in the device data sheet.

35. PWM Module

    The PWM module may not operate at temperatures below -20ºC.

36. PWM Module

    In Push-Pull mode, with immediate updates enabled, the PWM pins may become swapped.

37. Power Supply PWM: “On-the-fly” dead time adjustment

    The dead time registers (DTRx/ALTDTRx) must be modified only when the PWM is not running and should not be modified “on-the-fly”.

38. UART Module

    The 16x baud clock signal on the BCLK pin is present only when the module is transmitting.

39. UART Module

    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.

40. SPI Module

    The SPIxCON1 DISSCK bit does not influence port functionality.

41. I^{2}C Module

    The BCL bit in I2CSTAT can be cleared only with 16-bit operation and can be corrupted with 1-bit or 8-bit operations on I2CSTAT.

42. I^{2}C Module: 10-bit addressing mode

    When the I^{2}C module is configured for 10-bit addressing using the same address bits (A10 and A9) as other I^{2}C device A10 and A9 bits may not work as expected.

43. I^{2}C Module: 10-bit Addressing Mode

    The 10-bit slave does not set the RBF flag or load the I2CxRCV register on an address match if the Least Significant bits of the address are the same as the 7-bit reserved addresses.

44. I^{2}C Module: 10-bit Addressing Mode

    If the I^{2}C module is configured for a 10-bit slave with an address of 0x102, the I2CxRCV register content for the lower address byte is 0x01 rather than 0x02.

45. UART (FIFO Error)

    Under certain circumstances, the PERR and FERR error bits may not be correct for all bytes in the receive FIFO.

46. PSV Operations

    An address error trap occurs in certain addressing modes when accessing the first four bytes of any PSV page.

The following sections describe the errata and work around to these errata, where they may apply.

1. Module: Power Supply PWM: Dead Time

If dead time functionality is enabled (DTC<1:0> = 0 or 1 in the PWMCONx register), the minimum usable value that can be written to the dead time registers, DTRx and ALTDTRx, is 0x0010. Writing a value less than 0x0010 will cause either or both the PWMxH and PWMxL outputs not to function. The minimum acceptable dead time value is 0x0010. As a result of this errata, the minimum usable dead time is 16 ns. Dead time resolution is 4 ns for dead times greater than 16 ns.

Work around

The dead time must either be disabled (DTC<1:0> = 2) or DTRx and ALTDRx must have a value of 0x0010 or greater. If zero dead time is required, configure the DTC<1:0> bits in the PWMCONx register to specify no dead time.

2. Module: Power Supply PWM: Duty Cycle

The data sheet indicates that the power supply PWM module has a 1.1 ns duty cycle resolution. This is true for all values of PDCx except the following:

1. 0x0010 < PDCx < 0x0040
2. (Period – 0x0040) < PDCx < (Period – 0x0010)

In these ranges, duty cycle resolution is 16 ns. The PWM Period is either the Master Period, PTPER, or the individual PWM generator period, PHASEx.

Work around

If possible, the system should be designed so that the PWM generator will operate in the duty cycle range where the 1.1 ns resolution is possible. For operation outside this range, the design must take into account the reduced resolution.

3. Module: Power Supply PWM: Special Event Trigger and Individual Trigger

Each PWM generator can be configured to generate a trigger for the ADC module or a trigger interrupt at any point during the PWM period. The point in time during the PWM period that the trigger is set is specified in the TRIGx register for PWM Individual Trigger, or in the SEVTCMP register for the Special Event Trigger. The minimum trigger value in TRIGx or SEVTCMP is 0x0008. Values below 0x0008 result in a PWM trigger not being initiated at all. As a result, no ADC sampling or trigger interrupt will occur.

Work around

If the Special Event Trigger or the Individual Trigger is implemented, the user should perform a check in firmware to make sure that TRIGx and/or SEVTCMP is always greater than 0x0008 and less than the PWM period.

4. Module: PWM Override Enable

The OVRDAT<1:0> bits in the IOCONx register should determine the state of the PWMx output pins when the OVRENH and OVRENL bits (IOCONx<9:8>) are set. However, the PWM override feature does not work correctly. The PWMxH and PWMxL pins do not exhibit the state specified by the OVRDAT<1:0> bits when only one of the override bits (OVRENH or OVRENL) is set. If both bits are set, the override state is exhibited correctly on the PWMxL and PWMxH pins.

Work around

If override capability is desired on only one of the PWM pin pairs, use the GPIO module to override the PWM outputs. This can be done using the PENH and PENL bits in the IOCONx register. When the PENH/PENL bits in the IOCONx register. When the PENH/PENL bits are cleared, the GPIO module assumes control of the PWMxH/L output pin. The GPIO module must be setup in advance for the desired override output states, and the pins must be configured as digital outputs. This includes setting the PORTx and TRISx registers correctly, which correspond to the PWMxH and PWMxL pins.

5. Module: PWM Duty Cycle

The Power Supply PWM module has a feature to enable immediate duty cycle updates. This feature is enabled by setting IUE = `1` in the PWMCONx register. The dsPIC30F1010/202X Device Data Sheet states that the minimum PWM duty cycle value is 0x0010. Duty cycle values less than 0x0010 should cause the PWM outputs to display states corresponding to a duty cycle value of 0x0000.

When the immediate duty cycle updates are enabled, and a value of 0x0010 or less is loaded into the selected duty cycle register, the outputs of the PWM generator (PWMxH and PWMxL) will exhibit a state opposite to the expected state. For example, if the expected state of the PWM output is a continuous ‘`0`’, then a continuous ‘`1`’ will be observed, and vice versa.

The above behavior applies when the Master Duty Cycle (MDC register) or Individual Duty Cycle (PDCx register) provides the duty cycle value.

Work around

If immediate duty cycle updates are enabled, do not load the duty cycle register with a value less than or equal to 0x0010. If immediate duty cycle updates are not enabled, no action is required because the correct PWM state will be exhibited for all duty cycle values.

6. Module: PWM Override Priority

The “*dsPIC30F1010/202X Family Data Sheet*” (DS70178) states the priority of PWMx pin ownership as:

- PWM Generator (lowest priority)
- Output Override
- Current-Limit Override
- Fault Override
- PENx (GPIO/PWM) ownership (highest priority)

Instead of following the above priority scheme, the PWMx pin ownership is determined by ANDing the Output Override Data bits (OVRDAT<1:0>), Current-Limit Override Data bits (CLDAT<1:0>), and Fault Override Data bits (FLTDAT<1:0>) in the IOCONx register.

For example, the override data may be set as follows:

- OVRDAT<1:0> = `00`
- CLDAT<1:0> = `01`
- FLTDAT<1:0> = `10`

If all three overrides occur simultaneously, the following operations shown in Equation 1 will determine the state of the PWMx pin.

Therefore, when multiple overrides occur simultaneously, only the override data for the active override sources will be ANDed together while the inactive override sources will be ignored.

If only one override is active, override priorities do not apply and operation of the PWM overrides is normal.

Work around

None.

EQUATION 1:

$$PWMxH = (OVRDAT<1>)\ AND\ (CLDAT<1>)\ AND\ (FLTDAT<1>) = 0\ AND\ 0\ AND\ 1 = 0$$

$$PWMxL = (OVRDAT<0>)\ AND\ (CLDAT<0>)\ AND\ (FLTDAT<0>) = 0\ AND\ 1\ AND\ 0 = 0$$

7. Module: PWM Jitter

The outputs of the PWM module may exhibit a jitter proportional to the speed of operation of the device. The jitter may be observed as a deviation in the PWM Period, Duty Cycle or Phase, and may be affected independent of each other. As a result, the maximum deviation exhibited on the PWM output pin at 30 MIPS is 8.4 nsec.

The jitter is caused by silicon process variations, noise on the VDD rail and the operating temperature of the dsPIC DSC. However, for a given set of operating conditions, the maximum jitter will be the same for all three parameters, and independent of each other.

Table 1 shows the maximum jitter that may be exhibited at various operating speeds.

TABLE 1:

Speed of Operation	Maximum Jitter on PWM Output
30 MIPS	8.4 nsec
20 MIPS	12.6 nsec
15 MIPS	16.8 nsec

The maximum jitter at any operating speed can be determined using Equation 2.

EQUATION 2:

$$Maximum\ jitter\ observed\ (nsec) = \frac{252}{(S)}$$

Where:

- S is the speed of operation in MIPS.

The maximum percentage error observed on the PWM output can be calculated using Equation 3.

EQUATION 3:

$$Error\ (\%) = \pm\left[\frac{(x_{programmed} - x_{observed})}{x_{programmed}}\right] \cdot 100$$

Where:

- $x_{observed}$ is the observed value of parameter of interest (PWM period, Duty Cycle or Phase).
- $x_{programmed}$ is the programmed value of parameter of interest (PWM period, Duty Cycle or Phase).

Work around

Operate the Power Supply PWM module so that the percentage error in the parameter of interest (from Equation 3) is within permissible limits of the application.

8. Module: ADC Module: Global Software Trigger

In order to perform multiple analog-to-digital conversions using the global software trigger, the PxRDY bits in the ADSTAT register must be cleared. The data sheet indicates that the user can configure the ADC pin pairs to perform a conversion when the GSWTRG bit in the ADCON register is set. When the conversion is available, the user must then clear the GSWTRG bit and set it again to perform another conversion. Contrary to what the data sheet indicates, this will not initiate another conversion unless the PxRDY bits are cleared. Clearing the PxRDY bits automatically clears the GSWTRG bit.

This only applies to a polling based approach. If an interrupt based approach is used, the user is required to clear the PxRDY bits in the ADC Interrupt Service Routine (ISR).

Work around

The following sequence should be followed to manually trigger ADC conversions using the global software trigger (polling based only.)

1. Set the GSWTRG bit in ADCON to initiate a conversion on channels which have the trigger source as the global software trigger (via the TRGSRCx<5:0> bits in the ADCPCx registers).
2. Check the PxRDY bits to determine when the conversion(s) is completed.
3. Clear the PxRDY bits. The GSWTRG bit will be cleared as a result of this operation.
4. Repeat steps 1 to 3 to perform additional conversions.

Alternatively, the individual software trigger can be selected by setting the TRGSRCx<5:0> bits in the ADCPCx register equal to 0x01. Instead of using the global software trigger, the individual software trigger (ADCPCx<SWTRGx>) bits can be used to trigger a conversion on a given analog pin pair. In a bit polling approach, the PENDx in the ADCPCx register should be used to determine when a conversion is completed. In an interrupt based approach, the PxRDY bits get set when the conversion is complete. This bit must be cleared in the ADC Interrupt Service Routine in order to enable future interrupts.

9. Module: ADC Sample and Hold Timing

The dedicated ADC sample and hold circuits can be triggered by signals from the PWM module. The dsPIC30F1010/202X data sheet indicates that the resolution of the PWM-ADC sample and hold trigger timing is 8 ns. The existing implementation has a 41.6 ns resolution. In other words, when the PWM-ADC trigger is fired, an ADC sample may occur 1 ns to 41.6 ns later.

Work around

None.

10. Module: ADC Interrupts

The dsPIC30F1010/202X data sheet specifies that each ADC pin pair has its own interrupt vector. These interrupts do not work on the dsPIC30F1010/202X Rev. A2 devices.

Work around

Each ADC pin pair can be configured to initiate a global ADC interrupt by setting the corresponding IRQENx bit in the ADCPCx register. The ADBASE register can be used to create a jump table in the global ADC interrupt which will execute the appropriate ADC service routine for a particular ADC pin pair. There is an ADBASE register code example in the dsPIC30F1010/202X data sheet which illustrates using the ADBASE register in this way.

11. Module: ADC Module: Conversion Rate

The data sheet indicates that the conversion rate for the ADC module is 2.0 Msps. The ADC module on the dsPIC30F1010/202X Rev. A2 silicon has a maximum conversion rate of 1.5 Msps.

Work around

None.

12. Module: ADC Module: Shared Sample and Hold Circuit

The ADC inputs that do not have a dedicated sample and hold circuit will yield inaccurate conversion results unless the work around is implemented. For dsPIC30F1010/202X devices, the affected channels are AN1, AN3, AN5, AN7, AN8, AN9, AN10 and AN11 (depending on package). Additionally, this also applies to AN4 and AN6 on the dsPIC30F1010.

Work around

In the ADCON register, configure the ADC with Order = `0` and SEQSAMP = `1`. This configuration allows for accurate conversion of the analog channels which use the shared sample and hold circuit.

13. Module: Current Reset Mode

Setting the XPRES bit in the PWMCONx register should enable a current-limit source to reset the PWM period in Independent Time Base mode. This mode is not functioning correctly.

If the selected current-limit signal (either an analog comparator or external signal) triggers after the falling edge of PWMH, then the XPRES operation functions correctly. The PWM deasserted time is truncated and the PWM period is terminated early, and a new PWM cycle begins.

If the selected current-limit signal (either an analog comparator or external signal) triggers before the falling edge of PWMH, the PWMH asserted time is truncated, and the inactive time after the falling edge PWMH remains constant.

The proper XPRES behavior is to ignore the current-limit signal until the falling edge of the PWM period.

This issue may not be a problem in applications that control inductor current above a specified minimum current level. When the inductor current falls below the specified minimum value during the PWMH off-time, the PWM period is truncated and a new cycle begins to increase the inductor current.

Work around

None.

14. Module: Output Compare Module

A glitch will be produced on an output compare pin under the following conditions:

- The user software initially drives the I/O pin high using the output compare module or a write to the associated PORT register.
- The output compare module is configured and enabled to drive the pin low at some later time (OCxCON = 0x0002 or OCxCON = 0x0003).

When these events occur, the output compare module will drive the pin low for one instruction cycle (TCY) after the module is enabled.

Work around

None. However, the user may use a timer interrupt and write to the associated PORT register to control the pin manually.

15. Module: Output Compare Module in PWM Mode

The output compare module will miss a compare event when the current duty cycle register (OCxRS) value is 0x0000 (0% duty cycle) and the OCxRS register is updated with a value of 0x0001. The compare event is missed only the first time a value of 0x0001 is written to OCxRS, and the PWM output remains low for one PWM period. Subsequent PWM high and low times occur as expected.

Work around

None. If the current OCxRS register value is 0x0000, avoid writing a value of 0x0001 to OCxRS. Instead, write a value of 0x0002; however, in this case the duty cycle will be slightly different from the desired value.

16. Module: Output Compare Module

When the Output Compare Module is operated in the Dual Compare Match mode, a timer compare match with the value in the OCxR register sets the OCx output producing a rising edge on the OCx pin. Then, when a timer compare match with the value in the OCxRS register occurs, the OCx output is reset producing a falling edge on the OCx pin.

The above statement applies to all conditions except when the difference between OCxR and OCxRS is ‘1’. In this case, the Output Compare module may miss the reset compare event, and cause the OCx pin to remain continuously high. This condition will remain until the difference between values in the OCxR and OCxRS registers is made greater than ‘1’.

Work around

Ensure in software that the difference between values in OCxR and OCxRS registers is maintained greater than ‘1’.

17. Module: SPI Module in Slave Select Mode

The SPI module Slave Select functionality (enabled by setting SSEN = 1) will not function correctly. Whether the $\overline{\text{SSx}}$ pin (x = 1 or 2) is high or low, the SPI data transfer will be completed and an interrupt will be generated. This applies to the dsPIC30F2023 device only.

Note: The dsPIC30F1010/202X devices have only one SPI. All references to x = 2 are intended for software compatibility with other dsPIC DSC devices.

Work around

Manually poll the $\overline{\text{SSx}}$ pin state in the SPI interrupt by reading the associated PORT bit:

- If the PORT bit is ‘0’, then perform the required data read/write.
- If the PORT bit is ‘1’, then clear the SPI interrupt flag (SPIxIF), perform a dummy read of the SPIxBUF register, and return from the Interrupt Service Routine.

18. Module: SPI Module in Frame Master Mode

The SPI module will fail to generate frame synchronization pulses when configured in the Frame Master mode if the start of data is selected to coincide with the start of the frame synchronization pulse (FRMEN = 1, SPIFSD = 0). However, the module functions correctly in Frame Slave mode, and also in Frame Master mode if FRMDLY = 0. This applies to the dsPIC30F2023 device only.

Work around

Manually drive the $\overline{\text{SSx}}$ pin (x = 1 or 2) high using the associated PORT register, and then drive it low after the required 1 bit-time pulse-width. This operation needs to be performed when the transmit buffer is written.

If FRMDLY = 0, no work around is needed.

Note: The dsPIC30F1010/202X devices have only one SPI. All references to x = 2 are intended for software compatibility with other dsPIC DSC devices.

19. Module: SPI Module

The SMP bit (SPIxCON1<9>, where x = 1 or 2) does not have any effect when the SPI module is configured for a 1:1 prescale factor in Master mode. In this mode, whether the SMP bit is set or cleared, the data is always sampled at the end of data output time.

Note:	The dsPIC30F1010/202X devices have only one SPI. All references to x = 2 are intended for software compatibility with other dsPIC DSC devices.

Work around

If sampling at the middle of data output time is required, then configure the SPI module to use a clock prescale factor other than 1:1 using the PPRE<1:0> and SPRE<2:0> bits in the SPIxCON1 register.

20. Module: UART Module

With the parity option enabled, a parity error, indicated by the PERR bit (UxSTA<3>) being set, may occur if the Baud Rate Generator contains an odd value. This affects both even and odd parity options.

Work around

Load the Baud Rate Generator register, UxBRG, with an even value, or disable the peripheral's parity option by loading either 0b00 or 0b11 into the Parity and Data Selection bits, PDSEL<1:0> (UxMODE<2:1>).

21. Module: UART Module

The Receive Buffer Overrun Error Status bit, OERR (UxSTA<1>), may set before the UART FIFO has overflowed. After the fourth byte is received by the UART, the FIFO is full. The OERR bit should set after the fifth byte has been received in the UART Shift register. Instead, the OERR bit may set after the fourth received byte with the UART Shift register empty.

Work around

After four bytes have been received by the UART, the UART Receiver Interrupt Flag bit, U1RXIF (IFS0<11>), will be set, indicating the UART FIFO is full. The OERR bit may also be set. After reading the UART receive buffer, UxRXREG, four times to clear the FIFO, clear both the OERR and UxRXIF bits in software.

22. Module: UART Module

UART receptions may be corrupted if the Baud Rate Generator (BRGH) is set up for 4x mode (BRGH = 1).

Work around

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

23. Module: UART Module

The UTXISEL0 bit (UxSTA<13>) is always read as zero regardless of the value written to it. The bit can be written to either a '0' or '1', but will always be read as zero. This will affect read-modify-write operations such as bitwise or shift operations. Using a read-modify-write instruction on the UxSTA register (e.g., BSET, BLCR) will always write the UTXISEL0 bit to zero.

Work around

If a UTXISEL0 value of '1' is needed, avoid using read-modify-write instructions on the UxSTA register.

Copy the UxSTA register to a temporary variable and set UxSTA<13> prior to performing read-modify-write operations. Copy the new value back to the UxSTA register.

24. Module: UART Module

The auto-baud feature may not calculate the correct baud rate when the High Baud Rate Enable bit, BRGH, is set. With the BRGH 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.

25. Module: UART Module

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).

26. Module: UART Module

The UART module can be used to transmit and receive IrDA® signals with the use of an IrDA transceiver by setting the IREN bit in the UxMODE register. In this mode, the operation of the RXINV bit enables reception of signals with an Idle state of either ‘1’ or ‘0’. The operation of this bit is the inverse of the stated operation in the dsPIC30F1010/202X Device Data Sheet (DS70178).

The signal received from an IrDA transceiver can have an idle state of ‘1’ or ‘0’. The following table summarizes how UART receptions will occur when used with the IrDA decoder.

TABLE 2:

Type of Signal Used for Transmission	State of RXINV bit	UART reception
Idle State = ‘1’	RXINV = 0	May be erroneous
	RXINV = 1	Error-free
Idle State = ‘0’	RXINV = 0	Error-free
	RXINV = 1	May be erroneous

Work around

Invert the state of RXINV bit in the UxMODE register.

If the idle state of the received signal is ‘1’, configure RXINV = 1. If the idle state of the received signal is ‘0’, configure RXINV = 0.

27. Module: UART Module

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.

28. Module: I^{2}C Module

The Bus Collision Status bit (BCL) does not get set when a bus collision occurs during a Restart or Stop event. However, the BCL bit gets set when a bus collision occurs during a Start event.

Work around

None.

29. Module: I^{2}C Module

Writing to I2CxTRN during a Start bit transmission generates a write collision, indicated by the IWCOL (I2CxSTAT<7>) bit being set. In this state, additional writes to the I2CxTRN register should be blocked. However, in this condition, the I2CxTRN register can be written, although transmissions will not occur until the IWCOL bit is cleared in software.

Work around

After each write to the I2CxTRN register, read the IWCOL bit to ensure a collision has not occurred.

If the IWCOL bit is set, it must be cleared in software and I2CxTRN must be rewritten.

30. Module: I^{2}C Module

The ACKSTAT bit (I2CxSTAT<15>) only reflects the received ACK/NACK status for Master transmissions, but not for Slave transmissions. As a result, a Slave cannot use this bit to determine if it received an ACK or a NACK from a Master. In future silicon revisions, the ACKSTAT bit will reflect received ACK/NACK status for both Master and Slave transmissions.

Work around

After transmitting a byte, the Slave should poll the SDA line (subject to a time out period dependent on the application) to determine if an ACK (0) or a NACK (1) was received.

31. Module: I^{2}C Module

The D_A Status bit (I2CxSTAT<5>) gets set on a slave data reception in the I2CxRCV register, but does not get set on a slave write to the I2CxTRN register. In future silicon revisions, the D_A bit will get set on a slave write to I2CxTRN.

Work around

Use the D_A status bit only for determining slave reception status and not slave transmission status.

32. Module: $\overline{\text{MCLR}}$ Pin

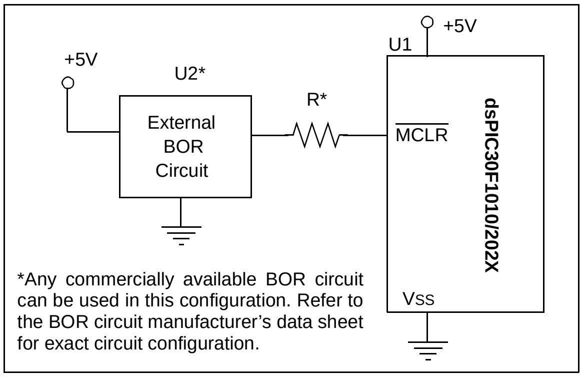
A brown-out event occurs when VDD drops below the minimum operating voltage for the device but not all the way down to VSS. When the dsPIC DSC SMPS device is running with the PLL enabled and a brown-out event occurs, the device may stop running and the $\overline{\text{MCLR}}$ pin will not reset the device. If this occurs, the device can only be reset by cycling power to the VDD pins.

It is recommended that an external Brown-out Reset (BOR) circuit be used to hold the device in reset during a brown-out event, to overcome this problem. The external BOR circuit will use the $\overline{\text{MCLR}}$ pin to hold the device in reset. The following work around, in combination with the external BOR circuit, will ensure that the device is cleanly reset after a brown-out event occurs.

Work around

The dsPIC DSC SMPS device must be powered up with the PLL disabled, the Fail-Safe Clock Monitor enabled and Clock Switching enabled. The PLL should be enabled in software via a clock switch after the device is reset (refer to **Section 29. "Oscillator"** in the "*dsPIC30F Family Reference Manual*" (DS70268) for details on clock switching). This ensures that the $\overline{\text{MCLR}}$ pin is functional and that the device can be reset by an external BOR circuit (see Figure 1).

FIGURE 1:



*Any commercially available BOR circuit can be used in this configuration. Refer to the BOR circuit manufacturer's data sheet for exact circuit configuration.

Use one of the following methods to achieve the work around.

Method 1: Insert the code shown in Example 1 at the start of the program.

Method 2: Call the code shown in Example 1 in the beginning of code execution by including the `ClockSwitch.s` file in the project and adding the following code:

- For assembly programming, add the following instruction at the beginning of the program:

```
.global __reset
…
…
__reset:
rcall ClockSwitch
…
…
```

- For C programming, add the following instruction at the beginning of the program:

```
int main(void)
{
        ClockSwitch;
        …
        …
}
```

EXAMPLE 1: CLOCK SWITCHING EXAMPLE

```
; This function performs a clock-switch from FRC to FRC+PLL. All other oscillator
; settings remain unchanged.
; Filename: ClockSwitch.s

_ClockSwitch:
    mov    #OSCCON+1,w4   ; Get address of high OSCCON byte
    mov    #0x0078, w0    ; 1st password for high byte access to OSCCON
    mov    #0x009A, w1    ; 2nd password for low byte access to OSCCON
    mov    #0x0001, w2    ; NOSC value for FRC+PLL
    mov.b  w0, [w4]       ; Write 1st password
    mov.b  w1, [w4]       ; Write 2nd password
    mov.b  w2, [w4]       ; Write NOSC value
    mov    #OSCCON,w4     ; Get address of low OSCCON byte
    mov    #0x0046, w0    ; 1st password for high byte access to OSCCON
    mov    #0x0057, w1    ; 2nd password for low byte access to OSCCON
    mov    #0x0001, w2    ; Set OSWEN bit
    mov.b  w0, [w4]       ; Write 1st password
    mov.b  w1, [w4]       ; Write 2nd password
    mov.b  w2, [w4]       ; Write OSWEN bit
    return
```

33. Module: CPU – DAW.b instruction

The Decimal Adjust instruction, DAW.b, may improperly clear the Carry bit, C (SR<0>), when executed.

Work around

Check the state of the Carry bit prior to executing the DAW.b instruction. If the Carry bit is set, set the Carry bit again after executing the DAW.b instruction. Example 2 shows how the application should process the Carry bit during a BCD addition operation.

EXAMPLE 2: CHECK CARRY BIT BEFORE DAW.b

```
    .include “p30fxxxx.inc”
    .......
    mov.b   #0x80, w0  ;First BCD number
    mov.b   #0x80, w1  ;Second BCD number
    add.b   w0, w1, w2 ;Perform addition
    bra     NC, L0     ;If C set go to L0
    daw.b   w2         ;If not, do DAW and
    bset.b  SR, #C     ;set the carry bit
    bra     L1         ;and exit
L0:daw.b    w2
L1: ....
```

34. Module: Sleep Mode

Execution of the Sleep instruction (PWRSAV #0) may cause incorrect program operation after the device wakes up from Sleep. The current consumption during Sleep may also increase beyond the specifications listed in the device data sheet.

Work arounds

To avoid this issue, any of the following three work arounds can be implemented, depending on the application requirements.

Work around 1:

Ensure that the PWRSAV #0 instruction is located at the end of the last row of Program Flash Memory available on the target device and fill the remainder of the row with NOP instructions.

This can be accomplished by replacing all occurrences of the PWRSAV #0 instruction with a function call to a suitably aligned subroutine. The address( ) attribute provided by the MPLAB ASM30 assembler can be utilized to correctly align the instructions in the subroutine. For an application written in C, the function call would be GotoSleep( ), while for an assembly language application, the function call would be CALL _GotoSleep.

The Address Error Trap Service Routine software can then replace the invalid return address saved on the stack with the address of the instruction immediately following the _GotoSleep or GotoSleep( ) function call. This ensures that the device continues executing the correct code sequence after waking up from Sleep mode.

Example 3 demonstrates the work around described above, as it would apply to a dsPIC30F2023 device.

EXAMPLE 3:

```
; ---------------------------------------------------------------------------------------------
.global __reset
.global _main
.global _GotoSleep
.global __AddressError
.global __INT1Interrupt
; ---------------------------------------------------------------------------------------------
    .section *, code
_main:
    BSET    INTCON2, #INT1EP  ; Set up INT pins to detect falling edge
    BCLR    IFS1, #INT1IF     ; Clear interrupt pin interrupt flag bits
    BSET    IEC1, #INT1IE     ; Enable ISR processing for INT pins
    CALL    _GotoSleep        ; Call function to enter SLEEP mode
_continue:
    BRA  _continue
; ---------------------------------------------------------------------------------------------
; Address Error Trap
__AddressError:
    BCLR    INTCON1, #ADDRERR
    ; Set program memory return address to _continue
    POP.D  W0
    MOV.B  #tblpage (_continue), W1
    MOV    #tbloffset (_continue), W0
    PUSH.D W0
    RETFIE
; ---------------------------------------------------------------------------------------------
__INT1Interrupt:
    BCLR   IFS1, #INT1IF          ; Ensure flag is reset
    RETFIE                        ; Return from Interrupt Service Routine
; ---------------------------------------------------------------------------------------------
    .section *, code, address (0x1FC0)
_GotoSleep:
; fill remainder of the last row with NOP instructions
    .rept 31
        NOP
    .endr
; Place SLEEP instruction in the last word of program memory
    PWRSAV #0
```

Work around 2:

Instead of executing a `PWRSAV #0` instruction to put the device into Sleep mode, perform a clock switch to the 512 kHz Low-Power RC (LPRC) Oscillator with a 64:1 postscaler mode. This enables the device to operate at 0.002 MIPS, thereby significantly reducing the current consumption of the device. Similarly, instead of using an interrupt to wake-up the device from Sleep mode, perform another clock switch back to the original oscillator source to resume normal operation. Depending on the device, refer to **Section 7. "Oscillator"** (DS70054) or **Section 29. "Oscillator"** (DS70268) in the "*dsPIC30F Family Reference Manual*" (DS70046) for more details on performing a clock switch operation.

Note: The above work around is recommended for users for whom application hardware changes are not possible.

Work around 3:

Instead of executing a `PWRSAV #0` instruction to put the device into Sleep mode, perform a clock switch to the 32 kHz Low-Power (LP) Oscillator with a 64:1 postscaler mode. This enables the device to operate at 0.000125 MIPS, thereby significantly reducing the current consumption of the device. Similarly, instead of using an interrupt to wake-up the device from Sleep mode, perform another clock switch back to the original oscillator source to resume normal operation. Depending on the device, refer to **Section 7. "Oscillator"** (DS70054) or **Section 29. "Oscillator"** (DS70268) in the "*dsPIC30F Family Reference Manual*" (DS70046) for more details on performing a clock switch operation.

Note: The above work around is recommended for users for whom application hardware changes are possible, and also for users whose application hardware already includes a 32 kHz LP Oscillator crystal.

35. Module: PWM Module

The PWM module may not operate at temperatures below -20ºC. During this condition the PWM module will relinquish control of the associated PWM pin and the Port I/O will determine the state of the pin. In addition, the PWM module will stop generating the ADC trigger before the module relinquishes control of the PWM pins.

Work around

None.

36. Module: PWM Module

In Push-Pull mode, with immediate updates enabled, the PWM pins may become swapped.

Work around

If using the PWM module in Push-Pull mode, immediate updates must be disabled.

37. Module: Power Supply PWM

The dead time registers (DTRx/ALTDTRx) must be modified only when the PWM is not running. Adjusting the dead time “on-the-fly” can result in an unpredictable glitch on the PWM output, which may cause shoot-through.

Work around

None.

38. Module: UART Module

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.

39. Module: UART Module

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.

40. Module: SPI Module

Setting the DISSCK bit in the SPIxCON1 register does not allow the user application to use the SCK pin as a general purpose I/O pin.

Work around

None.

41. Module: I^{2}C Module

The BCL bit in I2CSTAT can be cleared only with 16-bit operation and can be corrupted with 1-bit or 8-bit operations on I2CSTAT.

Work around

Use 16-bit operations to clear BCL.

42. Module: I^{2}C Module

If there are two I^{2}C 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.

43. Module: I^{2}C Module

In 10-bit Addressing mode, some address matches do not set the RBF flag or load the 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
- 101111010X
- 111111011X

Work around

The lower address byte in 10-bit Addressing mode should not be a reserved address.

44. Module: I^{2}C Module

If the I^{2}C module is configured for a 10-bit slave with an address of 0x102, the I2CxRCV register content for the lower address byte is 0x01 rather than 0x02. However, the I^{2}C module acknowledges both address bytes.

Work around

None.

45. Module: UART (FIFO Error Flags)

Under certain circumstances, the PERR and FERR error bits may not be correct for all bytes in the receive FIFO. This has only been observed when both of the following conditions are met:

- The UART receive interrupt is set to occur when the FIFO is full or three-quarters full (U1STA<7:6> = `1x`), and
- More than two bytes with an error are received.

In these two circumstances, only the first two bytes with a parity or framing error will have the corresponding bits indicated correctly. The error bits will not be set after this occurs.

Work around

None.

46. Module: Module: PSV Operations

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

APPENDIX A: REVISION HISTORY

Revision A (6/2007)

Initial release of this document.

Revision B (8/2007)

Updated issue 12 (ADC Module: Shared Sample and Hold Circuit). Added issue 34 (Sleep Mode).

Revision C (5/2008)

Summary items 5 and 6 were listed in the incorrect order and have been corrected.

Updated silicon issue 5 (PWM Duty Cycle) and changed the title of silicon issue 6 (PWM Override Priority) to be consistent with the summary title.

Added silicon issues 35 (PWM Module), 36 (PWM Module), 37 (Power Supply PWM), 38 (UART Module), 39 (UART Module), 40 (SPI Module), 41 (I^2C Module), and 42 (I^2C Module).

Revision D (7/2008)

Updated issue 35 (PWM Module) and added silicon issues 43 (I^2C Module), 44 (I^2C Module), 45 (UART (FIFO Error Flags) and 46 (I^2C Module).

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. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICtail, PIC^{32} logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

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://support.microchip.com
Web Address:
www.microchip.com

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

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

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

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

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

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

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
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: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

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

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

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

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: 86-755-8203-2660
Fax: 86-755-8203-1760

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

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

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

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

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

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

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

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

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 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: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
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: 45-4450-2828
Fax: 45-4485-2829

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

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

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

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

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

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

01/02/08