PIC16(L)F1934/1936/1937 Family Silicon Errata and Data Sheet Clarifications

PIC16(L)F1934/1936/1937

Product Page Links

Introduction

The PIC16(L)F1934/1936/1937 devices that you have received conform functionally to the current device data sheet (DS41364**E**), 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 the table below.

The errata described in this document will be addressed in future revisions of the PIC16(L)F1934/1936/1937 silicon.

Note: This document summarizes all silicon errata issues from all revisions of silicon, previous as well as current.

Table 1. Silicon Device Identification

Part Number	DEVICE ID[13:0]							
	DEV[8:0]	REV[4:0]						
		A2	A3	A5	A6	A7	A8	A9
PIC16F1934	10 0011 010	0 0010	0 0011	0 0101	0 0110	0 0111	0 1000	0 1001
PIC16LF1934	10 0100 010	0 0010	0 0011	0 0101	0 0110	0 0111	0 1000	0 1001
PIC16F1936	10 0011 011	0 0010	0 0011	0 0101	0 0110	0 0111	0 1000	0 1001
PIC16LF1936	10 0100 011	0 0010	0 0011	0 0101	0 0110	0 0111	0 1000	0 1001
PIC16F1937	10 0011 100	0 0010	0 0011	0 0101	0 0110	0 0111	0 1000	0 1001
PIC16LF1937	10 0100 100	0 0010	0 0011	0 0101	0 0110	0 0111	0 1000	0 1001

Important: Refer to the **Device/Revision ID** section in the device data sheet for more detailed information on Device Identification and Revision IDs for your specific device.

Silicon Issue Summary

Table 2. Silicon Issue Summary

Module	Feature	Item No.	Issue Summary	Affected Revisions						
				A2	A3	A5	A6	A7	A8	A9
Data EE Memory	Memory Endurance	1.1.1	Erase/Write Endurance Limited	X	X	X				
	Writes	1.1.2	Minimum V_{DD} for Writes	X	X	X	X	X	X	X
Program Flash Memory (PFM)	Memory Endurance	1.2.1	Erase/Write Endurance Limited	X	X	X				
	Writes	1.2.2	Minimum V_{DD} for Writes	X	X	X	X	X	X	X
Capture Compare PWM (CCP)	PWM Dead-Band Delay	1.3.1	Dead-Band Delay Results in Unpredictable Waveforms	X	X	X	X	X	X	X
	ECCP2 Switching	1.3.2	ECCP2 Switching Between Single, Half-Bridge, and Full-Bridge PWM Modes	X	X	X	X	X	X	X
	ECCP2 Changing Direction	1.3.3	ECCP2 PWM Outputs Will Improperly Go Active	X	X	X	X	X	X	X
	Capture Mode	1.3.4	Capture Triggered While CCPx Pin is Held High	X	X	X	X	X	X	X
	ECCPx Dead-Band Delay	1.3.5	ECCPx Dead-Band Delay in Half-Bridge Mode	X	X	X	X	X	X	X
	PWM with Pulse Steering	1.3.6	Disabling a PWM Output During a PWM Cycle May Stop the Output Earlier than Expected	X	X	X	X	X	X	X
	Capture Mode	1.3.7	Capture Triggered While CCPx Pin is Held Low	X	X	X	X	X	X	X
Brown-Out Reset (BOR)	Threshold	1.4.1	BOR Threshold Voltage Level	X						
Analog-to-Digital Converter (ADC)	ADC Conversion	1.5.1	ADC Conversion May Not Complete	X	X	X				
Oscillator (OSC)	HS Oscillator	1.6.1	HS Oscillator Minimum V_{DD}	X	X	X				
	Oscillator Start-Up Timer (OSTS) Bit	1.6.2	OSTS Bit Remains Set	X	X	X	X	X	X	
	MFINTOSC	1.6.3	The Device May not Wake from Sleep when Using the MF Internal Oscillator	X	X	X	X	X	X	X
Enhanced Capture Compare PWM (ECCP)	Enhanced PWM	1.7.1	PWM 0% Duty Cycle Direction Change	X	X	X	X	X	X	X
	Enhanced PWM	1.7.2	PWM 0% Duty Cycle Port Steering	X	X	X	X	X	X	X
Timer1	Timer0 Gate Source	1.8.1	Gate Toggle Mode Works Improperly	X	X	X	X	X	X	X
	Timer1 Gate Toggle Mode	1.8.2	T1 Gate Flip-Flop Does Not Clear	X	X	X	X	X	X	X
LDO	Minimum V_{DD} above 85°C	1.9.1	Minimum Operating V_{DD} for the PIC16F193x devices at T_A > 85°C	X	X	X	X	X	X	X

Table 2. Silicon Issue Summary (continued)

Module	Feature	Item No.	Issue Summary	Affected Revisions						
				A2	A3	A5	A6	A7	A8	A9
Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)	Auto-Baud Detect	1.10.1	Auto-Baud Detect May Store Incorrect Count Value in the SPBRG Registers	X	X	X	X	X		
Resets	RESET Instruction	1.11.1	Extended Reset if Clock is MFINTOSC or HFINTOSC	X	X	X	X	X	X	
Master Synchronous Serial Port (MSSP)	SPI Master Mode	1.12.1	The Buffer Full (BF) Bit or MSSP Interrupt Flag (SSPIF) Bit Becomes Set Half of a SCK Cycle Early	X	X	X	X	X	X	X

Note: Only those issues indicated in the last column apply to the current silicon revision.

1. Silicon Errata Issues

NOTICE This document summarizes all silicon errata issues from all revisions of silicon, previous and current. Only the issues indicated by the bold font in the following tables apply to the current silicon revision.

1.1. Module: Data EE Memory

1.1.1. Erase/Write Endurance Limited

The typical write/erase endurance of the Data EE Memory is limited to 10k cycles.

Work around

Use an error correction method that stores data in multiple locations.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X					

1.1.2. Minimum V_{DD} for Writes

The minimum voltage required for a Data EE write operation is 2.0 volts.

Work around

None.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.2. Module: Program Flash Memory (PFM)

1.2.1. Erase/Write Endurance Limited

The typical write/erase endurance of the PFM is limited to 1k cycles when V_{DD} is above 3.0 volts. Endurance degrades when V_{DD} is below 3.0 volts.

Work around

Use an error correction method that stores data in multiple locations.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X					

1.2.2. Minimum V_{DD} for Writes

The minimum voltage required for a PFM write operation is 2.0 volts.

Work around

None.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.3. Module: Capture Compare PWM (CCP)

1.3.1. Dead-Band Delay Results in Unpredictable Waveforms

When the ECCP is configured for PWM Half-Bridge mode with a dead-band delay greater than or equal to the PWM duty cycle, unpredictable waveforms may result.

Work around

Make sure the dead-band delay is always less than the PWM duty cycle.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.3.2. ECCP2 Switching Between Single, Half-Bridge, and Full-Bridge PWM Modes

Switching PWM modes during the current PWM cycle by modifying the P2M[1:0] bits in the CCP2CON register will cause the PWM outputs to switch immediately and not on the start of the next PWM cycle.

Work around

None.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.3.3. ECCP2 PWM Outputs Will Improperly Go Active

When changing directions in Full-Bridge PWM mode, the active and modulated outputs will improperly go active at the same time, and the dead-band delay does not occur, which can lead to large shoot-through currents.

Work around

None.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.3.4. Capture Triggered While CCPx Pin is Held High

If the module is configured to capture on the first rising edge and the CCPx pin is high at this time, a capture will be triggered.

Work around

Clear the CCP Interrupt Flag (`CCPxIF = 0`) immediately after configuring the module for a capture event.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.3.5. ECCPx Dead-Band Delay in Half-Bridge Mode

In Half-Bridge mode, the dead-band delay is 1 T_{OSC} longer than calculated for the first PWM cycle and 1.5 T_{OSC} for the following cycles.

Work around

None.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.3.6. Disabling a PWM Output During a PWM Cycle May Stop the Output Earlier Than Expected

When the PWM is in Steering mode, disabling a PWM output during a PWM cycle will cause the output to end 1 T_{OSC} earlier than expected.

Work around

None.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.3.7. Capture Triggered While CCPx Pin is Held Low

If the module is configured to capture on the first falling edge and the CCPx pin is low at this time, a capture will be triggered.

Work around

Clear the CCP Interrupt Flag (`CCPxIF = 0`) immediately after configuring the module for a capture event.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.4. Module: Brown-Out Reset (BOR)

1.4.1. BOR Threshold Voltage Level

When the BOR is configured for 2.5 volt operation, the BOR Reset will typically occur at 2.7 volts.

Work around

None.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X							

1.5. Module: Analog-to-Digital Converter (ADC)

1.5.1. ADC Conversion May Not Complete

An ADC conversion may not complete under these conditions:

1. When F_{OSC} is greater than 8 MHz and it is the clock source used for the ADC converter.
2. The ADC is operating from its dedicated ADCRC oscillator and the device is not in Sleep mode (at any F_{OSC} frequency). When this occurs, the ADC Interrupt Flag (ADIF) does not get set, the GO/$\overline{\text{DONE}}$ bit does not get cleared, and the conversion result does not get loaded into the ADRESH and ADRESL result registers.

Work around

Method 1:

Select the system clock, F_{OSC}, as the ADC clock source and reduce the F_{OSC} frequency to 8 MHz or less when performing ADC conversions.

Method 2:

Select the dedicated ADCRC oscillator as the ADC conversion clock source and perform all conversions with the device in Sleep.

Method 3:

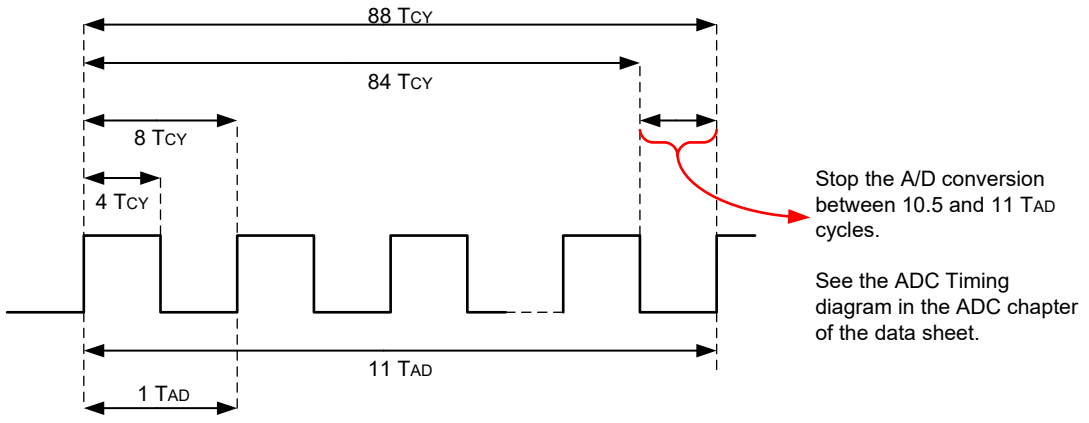
This method is provided if the application cannot use Sleep mode and requires continuous operation at frequencies above 8 MHz. This method requires early termination of an ADC conversion. Provide a fixed time delay in software to stop the ADC conversion manually, after all 10 bits are converted, but before the conversion would complete automatically. The conversion is stopped by clearing the GO/$\overline{\text{DONE}}$ bit in software, and must be cleared during the last ½ T_{AD} cycle, before the conversion would have completed automatically.

In Figure 1-1, 88 instruction cycles will be required to complete the full conversion. Each T_{AD} cycle consists of 8 T_{CY} periods. A fixed delay is provided to stop the A/D conversion after 86 instruction cycles and terminate the conversion at the correct time.

Note: The exact delay time will depend on the T_{AD} divisor (ADCS) selection. The T_{CY} counts shown in Figure 1-1 apply to this example only. Refer to Table 1-1 for examples of the required delay counts for other configurations.

Figure 1-1. Instruction Cycle Delay Calculation Example

FOSC = 32 MHz
TCY = 4/32 MHz = 125 ns
TAD = 1 µs, ADCS = FOSC/32



See the '**ADC Clock Period (TAD) vs Device Operating Frequency**' table in the ADC section of the data sheet.

Table 1-1. Instruction Cycle Delay Counts by T_{AD} Selection

T_{AD}	Instruction Cycle Delay Counts by T_{AD} Selection
$F_{OSC}/64$	172
$F_{OSC}/32$	86
$F_{OSC}/16$	43

Example 1-1. Code Example of Instruction Cycle Delay

```
BSF     ADCON0, ADGO     ; Start ADC Conversion
                         ; Provide 86 instruction cycle delay here
BCF     ADCON0, ADGO     ; Terminate the conversion manually
MOVF    ADRESH, W        ; Read the conversion result
```

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X					

1.6. Module: Oscillator (OSC)

1.6.1. HS Oscillator Minimum V_{DD}

The HS oscillator requires a minimum voltage of 3.0 volts (at 65°C or less) to operate at 20 MHz.

Work around

None.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X					

1.6.2. OSTS Bit Remains Set

During the Two-Speed Start-up sequence, the Oscillator Start-up Timer (OST) is enabled to count 1024 clock cycles. After the count is reached, the OSTS bit is set, the system clock is held low until the next falling edge of the external crystal (LP, XT, or HS mode), before switching to the external clock source.

When an external oscillator is configured as the primary clock and Fail-Safe Clock mode is enabled (FCMEN = 1), any of the following conditions will result in the OST failing to restart:

- $\overline{\text{MCLR}}$ Reset
- Wake from Sleep
- Clock change from INTOSC to Primary Clock

This anomaly will manifest itself as a clock failure condition for external oscillators which take longer than the clock failure time-out period to start.

Work around

None.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X		

1.6.3. The Device May Not Wake From Sleep When Using the MF Internal Oscillator

The device may not wake up from Sleep mode when using the Medium-Frequency Internal Oscillator (MFINTOSC) as the system clock.

Work around

To ensure the device wakes up from Sleep, Use one of the following work arounds:

1. Use the Low-Frequency Internal Oscillator (LFINTOSC) instead of the MFINTOSC.
2. If the MFINTOSC must be used as the system clock:
   a. Switch to the LFINTOSC before executing the `SLEEP` command
   b. Upon wake-up, switch to the 500 kHz MFINTOSC
   c. Wait for the oscillator start-up time to expire (approximately 2 µs)
   d. Check the MFIOFR status bit to ensure the MFINTOSC started up
   e. If the MFIOFR bit is clear, switch to the 500 kHz HFINTOSC
   f. Wait for the oscillator start-up time to expire (approximately 2 µs)
   g. When the HFIOFR bit is set, switch back to the 500 kHz MFINTOSC

```
OSCCONbits.IRCF = 0x0;          // Select the LFINTOSC
SLEEP();
NOP();
NOP();
OSCCONbits.IRCF = 0x7;          // Select the 500 kHz MFINTOSC
__delay_us(2);                  // Oscillator start-up delay
if(OSCSTATbits.MFIOFR == 0)
{
    OSCCONbits.IRCF = 0xA;      // Switch to the 500 kHz HFINTOSC
}
__delay_us(2);                  // Oscillator start-up delay
if(OSCSTATbits.HFIOFR == 1)
{
    OSCCONbits.IRCF = 0x7;      // Switch back to the 500 kHz MFINTOSC
}
```

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.7. Module: Enhanced Capture Compare PWM (ECCP)

1.7.1. PWM 0% Duty Cycle Direction Change

When the PWM is configured for Full-Bridge mode and the duty cycle is set to 0%, writing the PxM[1:0] bits to change the direction has no effect on PxA and PxC outputs.

Work around

Increase the duty cycle to a value greater than 0% before changing directions.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.7.2. PWM 0% Duty Cycle Port Steering

In PWM mode, when the duty cycle is set to 0% and the STRxSYNC bit is set, writing the STRxA, STRxB, STRxC, and the STRxD bits to enable/disable steering to port pins has no effect on the outputs.

Work around

Increase the duty cycle to a value greater than 0% before enabling/disabling steering to port pins.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.8. Module: Timer1

1.8.1. Gate Toggle Mode Works Improperly

Timer1 Gate Toggle mode provides unexpected results when Timer0 overflow is selected as the Timer1 Gate source. We do not recommend using Timer0 overflow as the Timer1 Gate source while in Timer1 Gate Toggle mode or when Toggle mode is used in conjunction with Timer1 Gate Single-Pulse mode.

Work around

None.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.8.2. T1 Gate Flip-Flop Does Not Clear

When Timer1 Gate Toggle mode is enabled, clearing the TMR1ON bit does not clear the output value of the flip-flop and hold it clear.

When Timer1 Gate Toggle mode is enabled, it is possible to measure the full-cycle length of a Timer1 gate signal. To perform this function, the Timer1 gate source is routed through a flip-flop that changes state on every incrementing edge of the gate signal. Timer1 Gate Toggle mode is enabled by setting the T1GTM bit of the T1GCON register. When working properly, clearing either the T1GTM bit or the TMR1ON bit would also clear the output value of this flip-flop, and hold it clear. This is done in order to control which edge is being measured.

Work around

Clear the T1GTM bit in the T1GCON register to clear, and hold, clear the output value of the flip-flop.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.9. Module: Low Dropout (LDO) Voltage Regulator

1.9.1. Minimum Operating V_{DD} for the PIC16F193x Devices at TA > 85°C

The minimum voltage required for the PIC16F193x devices is 3.5 volts for temperatures above 85°C.
Note: This issue only applies to the PIC16F193x devices operating in the Extended temperature range. The PIC16LF193x devices are not affected.

Work around

None.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

1.10. Module: Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART)

1.10.1. Auto-Baud Detect May Store Incorrect Count Value in the SPBRG Registers

When using automatic baud detection (ABDEN), on occasion, an incorrect count value can be stored at the end of the auto-baud detection in the SPBRGH:SPBRGL (SPBRG) registers. The SPBRG value may be off by several counts. This condition happens sporadically when the device clock frequency drifts to a frequency where the SPBRG value oscillates between two different values. The issue is present regardless of the baud rate Configuration bit settings.

Work around

When using auto-baud, it is good practice to always verify the obtained value of SPBRG, to ensure it remains within the application specifications. Two recommended methods are shown below.

For additional auto-baud information, see Technical Brief TB3069, "*Use of Auto-Baud for Reception of LIN Serial Communications Devices: Mid-Range and Enhanced Mid-Range*".

Example 1-2. Method 1 - EUSART Auto-Baud Detect Work Around

```
#define SPBRG_16BIT  *((*int) &SPBRG; // Define location for 16-bit SPBRG value
const int DEFAULT_BAUD = 0x0067;          // Default Auto-Baud value
const int TOL = 0x05;                     // Baud Rate % tolerance
const int MIN_BAUD = DEFAULT_BAUD - TOL;  // Minimum Auto-Baud limit
const int MAX_BAUD = DEFAULT_BAUD + TOL;  // Maximum Auto-Baud limit

ABDEN = 1;                                 // Start Auto-Baud
while(ABDEN);                              // Wait until Auto-Baud completes
if((SPBRG_16BIT > MAX_BAUD) || (SPBRG_16BIT < MIN_BAUD))
{                                         // Compare if value is within limits
    SPBRG_16BIT = DEFAULT_BAUD;           // If out of spec, use DEFAULT_BAUD
}                                         // If in spec, continue using the
                                          // Auto-Baud value in SPBRG
```

Note: In firmware, define default, minimum, and maximum auto-baud (SPBRG) values according to the application requirements. For example, if the application runs at 9600 baud at 16 MHz, then the default SPBRG value would be (assuming 16-bit Asynchronous mode) `0x67`. The minimum and maximum allowed values can be calculated based on the application. In this example, a ±5% tolerance is required, so tolerance is `0x67` * 5% = `0x05`.

Example 1-3. Method 2 - EUSART Auto-Baud Detect Work Around

```
#define SPBRG_16BIT  *((*int)&SPBRG; // Define location for 16-bit SPBRG value
const int DEFAULT_BAUD = 0x0067;          // Default Auto-Baud value
const int TOL = 0x05;                     // Baud rate % tolerance
const int MIN_BAUD = DEFAULT_BAUD - TOL; // Minimum Auto-Baud limit
const int MAX_BAUD = DEFAULT_BAUD + TOL; // Maximum Auto-Baud limit
int Average_Baud;                        // Define Average_Baud variable
int Integrator;                          // Define Integrator variable
Average_Baud = DEFAULT_BAUD;             // Set initial average baud rate
Integrator = DEFAULT_BAUD * 15;          // The running 16 count average

ABDEN = 1;                               // Start Auto-Baud
while(ABDEN);                            // Wait until Auto-Baud completes

Integrator+ = SPGRB_16BIT;
Average_Baud = Integrator/16;
if((SPBRG_16BIT > MAX_BAUD) || (SPBRG_16BIT < MIN_BAUD))
{                                       // Check if value is within limits
    SPBRG_16BIT = Average_Baud;         // If out of spec, use previous average
}
else                                    // If in spec, calculate the running
{                                       // average but continue using the
    Integrator+ = SPBRG_16BIT;          // Auto-Baud value in SPBRG
    Average_Baud = Integrator/16;
```

```
    Integrator- = Average_Baud;
}
```

Note: Similar to Method 1, define default, minimum, and maximum auto-baud (SPBRG) values. In firmware, compute a running average of SPBRG. If the new SPBRG value falls outside the minimum or maximum limits, then use the current running average. For example, if the application runs at 9600 baud at 16 MHz, then the default SPBRG value would be (assuming 16-bit Asynchronous mode) 0x67. The minimum and maximum allowed values can be calculated based on the application. In this example, a ±5% tolerance is required, so tolerance is `0x67` * 5% = `0x05`.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X			

1.11. Module: Resets

1.11.1. Extended Reset if Clock Selection is MFINTOSC or HFINTOSC

When using either the MFINTOSC or HFINTOSC as the primary clock source, and after executing a `RESET` instruction or when the device executes a Stack Overflow/Underflow Reset, the device may remain in Reset.

Work around

Method 1: Use the Watchdog Timer (WDT) to recover from this issue.

Method 2: In place of the `RESET` instruction, use `GOTO 0x0000`.

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X		

1.12. Module: Master Synchronous Serial Port (MSSP)

1.12.1. The Buffer Full (BF) Bit or MSSP Interrupt Flag (SSPIF) Bit Becomes Set Half of a SCK Cycle Early

When the MSSP is used in SPI Master mode and the CKE bit is clear (CKE = `0`), the Buffer Full (BF) bit and the MSSP Interrupt Flag (SSPIF) bit becomes set half an SCK cycle early. If the user software immediately reacts to either of the bits being set, a write collision may occur as indicated by the WCOL bit being set.

Work around

To avoid a write collision, one of the following methods should be used:

Method 1:

Add a software delay of one SCK period after detecting the completed transfer (the BF bit or SSPIF bit becomes set) and prior to writing to the SSPBUF register. Verify the WCOL bit is clear after writing to SSPBUF. If the WCOL bit is set, clear the bit in software and rewrite the SSPBUF register.

Method 2:

As part of the MSSP initialization procedure, set the CKE bit (CKE = `1`).

Affected Silicon Revisions

A2	A3	A5	A6	A7	A8	**A9**	
X	X	X	X	X	X	**X**	

2. Data Sheet Clarifications

The following typographic corrections and clarifications are to be noted for the latest version of the device data sheet (DS41364**E**):

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

2.1. V_{OL}/V_{OH} Note 4 Correction

In **Section 30.4: "DC Characteristics: PIC16(L)F1934/6/7-I/E"**, Note 4 has been corrected as shown in the table below (in **bold**):

Param No.	Sym.	Characteristics	Min.	Typ†	Max.	Units	Conditions
D080	V_{OL}	Output Low Voltage(4)					
		All I/O Pins	—	—	0.6	V	I_{OL} = 8.0 mA, V_{DD} = 5.0V I_{OL} = 6.0 mA, V_{DD} = 3.3V I_{OL} = 1.8 mA, V_{DD} = 1.8V
D090	V_{OH}	Output High Voltage(4)					
		All I/O Pins	V_{DD} - 0.7	—	—	V	I_{OH} = 3.5 mA, V_{DD} = 5.0V I_{OH} = 3.0 mA, V_{DD} = 3.3V I_{OH} = 1.0 mA, V_{DD} = 1.8

* These parameters are characterized but not tested.

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Notes:

1. In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an external clock in RC mode.
2. Negative current is defined as current sourced by the pin.
3. The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
4. **Including the OSC2 pin with CLKOUT mode disabled.**

3. Appendix A: Revision History

Doc. Rev.	Date	Comments
N	05/2025	Updated the document format to the current Microchip publication standard; module numbers under silicon errata have been renumbered accordingly. Added silicon issue 1.6.3; added data sheet clarification 2.1. Minor typographic corrections to existing issues.
M	11/2014	Added silicon issue 1.12.1; other minor corrections.
L	11/2013	Added silicon revision A9; added silicon issues 1.6.2 and 1.11.1; other minor corrections.
K	07/2012	Added silicon revision A8; removed errata issues 1.6.2, 1.6.3, and 1.6.4.
J	07/2012	Removed DS Clarifications module 1.
H	02/2012	Added silicon issues 1.6.2, 1.6.3, 1.6.4, and 1.10.1; added DS Clarifications module 1; other minor corrections.
G	09/2011	Added silicon revision A7; Removed DS Clarifications 1 and 2.
F	09/2010	Added silicon issue 1.9.1.
E	07/2010	Revised item 1.5.1; added issue 1.8.2; other minor corrections.
D	06/2010	Added silicon revision A6.
C	05/2010	Added errata issues 1.5.1, 1.6.1, 1.6.2, 1.7.1, 1.7.2, 1.8.1; added DS Clarifications modules 1 and 2.
B	01/2010	Added silicon revision A5.
A	09/2009	Initial release of this document.

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-1051-6

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications 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. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. 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 ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

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 unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Product Page Links

PIC16F1934, PIC16F1936, PIC16F1937