Product Change Notification: SYST-17HGRB251

Date:

27-Jul-2026

Product Category:

Digital Signal Controllers

Notification Subject:

dsPIC33AK256MPS306 Family Silicon Errata and Data Sheet Clarification

Affected CPNs:

SYST-17HGRB251_Affected_CPN_07272026.pdf
SYST-17HGRB251_Affected_CPN_07272026.csv

Notification Text:

SYST-17HGRB251

Microchip has released a new Document for the dsPIC33AK256MPS306 Family Silicon Errata and Data Sheet Clarification of devices. If you are using one of these devices please read the document located at **dsPIC33AK256MPS306 Family Silicon Errata and Data Sheet Clarification**.

Notification Status: Final

Description of Change:

Added data sheet clarifications 1. Module: 40 MSPS Analog-to-Digital Converter (ADC) and 2. Module: I/O Ports with Edge Detect. Updates to workaround code in 10. Module: CPU.

Impacts to Data Sheet: Refer to **DS70005629C**

Reason for Change: None

Change Implementation Status: Complete

Date Document Changes Effective: 27 Jul 2026

NOTE: Please be advised that this is a change to the document only the product has not been

changed.

Markings to Distinguish Revised from Unrevised Devices: N/A

Attachments:

dsPIC33AK256MPS306 Family Silicon Errata and Data Sheet Clarification

Please contact your local **Microchip sales office** with questions or concerns regarding this notification.

Terms and Conditions:

If you wish to receive Microchip PCNs via email please register for our PCN email service at our **PCN home page** select register then fill in the required fields. You will find instructions about registering for Microchips PCN email service in the **PCN FAQ** section.

If you wish to change your PCN profile, including opt out, please go to the **PCN home page** select login and sign into your myMicrochip account. Select a profile option from

Affected Catalog Part Numbers (CPN)

DSPIC33AK256MPS306-H/PT

DSPIC33AK256MPS306-I/M7

DSPIC33AK256MPS306-I/PT

DSPIC33AK256MPS306T-E/M7

DSPIC33AK256MPS306T-E/PT

DSPIC33AK256MPS306T-I/M7

DSPIC33AK256MPS306T-I/PT

DSPIC33AK128MPS103-E/M7

DSPIC33AK128MPS103-H/M7

DSPIC33AK128MPS103-I/M7

DSPIC33AK128MPS103T-E/M7

DSPIC33AK128MPS103T-I/M7

DSPIC33AK128MPS105-E/M7

DSPIC33AK128MPS105-E/PT

DSPIC33AK128MPS105-H/M7

DSPIC33AK128MPS105-H/PT

DSPIC33AK128MPS105-I/M7

DSPIC33AK128MPS105-I/PT

DSPIC33AK128MPS105T-E/M7

DSPIC33AK128MPS105T-E/PT

DSPIC33AK128MPS105T-I/M7

DSPIC33AK128MPS105T-I/PT

DSPIC33AK128MPS106-E/M7

DSPIC33AK128MPS106-E/PT

DSPIC33AK128MPS106-H/M7

DSPIC33AK128MPS106-H/PT

DSPIC33AK128MPS106-I/M7

DSPIC33AK128MPS106-I/PT

DSPIC33AK128MPS106T-E/M7

DSPIC33AK128MPS106T-E/PT

DSPIC33AK128MPS106T-I/M7

DSPIC33AK128MPS106T-I/PT

DSPIC33AK128MPS303-E/M7

DSPIC33AK128MPS303-H/M7

DSPIC33AK128MPS303-I/M7

DSPIC33AK128MPS303T-E/M7

DSPIC33AK128MPS303T-I/M7

DSPIC33AK128MPS305-E/M7

DSPIC33AK128MPS305-E/PT

DSPIC33AK128MPS305-H/M7

DSPIC33AK128MPS305-H/PT

DSPIC33AK128MPS305-I/M7

DSPIC33AK128MPS305-I/PT

DSPIC33AK128MPS305T-E/M7

DSPIC33AK128MPS305T-E/PT

DSPIC33AK128MPS305T-I/M7

DSPIC33AK128MPS305T-I/PT

DSPIC33AK128MPS306-E/M7

DSPIC33AK128MPS306-E/PT

DSPIC33AK128MPS306-H/M7

DSPIC33AK128MPS306-H/PT

DSPIC33AK128MPS306-I/M7

DSPIC33AK128MPS306-I/PT

DSPIC33AK128MPS306T-E/M7

DSPIC33AK128MPS306T-E/PT

DSPIC33AK128MPS306T-I/M7

DSPIC33AK128MPS306T-I/PT

DSPIC33AK256MPS103-E/M7

DSPIC33AK256MPS103-H/M7

DSPIC33AK256MPS103-I/M7

DSPIC33AK256MPS103T-E/M7

DSPIC33AK256MPS103T-I/M7

DSPIC33AK256MPS105-E/M7

DSPIC33AK256MPS105-E/PT

DSPIC33AK256MPS105-H/M7

DSPIC33AK256MPS105-H/PT

DSPIC33AK256MPS105-I/M7

DSPIC33AK256MPS105-I/PT

DSPIC33AK256MPS105T-E/M7

DSPIC33AK256MPS105T-E/PT

DSPIC33AK256MPS105T-I/M7

DSPIC33AK256MPS105T-I/PT

DSPIC33AK256MPS106-E/M7

DSPIC33AK256MPS106-E/PT

DSPIC33AK256MPS106-H/M7

DSPIC33AK256MPS106-H/PT

DSPIC33AK256MPS106-I/M7

DSPIC33AK256MPS106-I/PT

DSPIC33AK256MPS106T-E/M7

DSPIC33AK256MPS106T-E/PT

DSPIC33AK256MPS106T-I/M7

DSPIC33AK256MPS106T-I/PT

DSPIC33AK256MPS303-E/M7

DSPIC33AK256MPS303-H/M7

DSPIC33AK256MPS303-I/M7

DSPIC33AK256MPS303T-E/M7

DSPIC33AK256MPS303T-I/M7

DSPIC33AK256MPS305-E/M7

DSPIC33AK256MPS305-E/PT

DSPIC33AK256MPS305-H/M7

DSPIC33AK256MPS305-H/PT

DSPIC33AK256MPS305-I/M7

DSPIC33AK256MPS305-I/PT

DSPIC33AK256MPS305T-E/M7

DSPIC33AK256MPS305T-E/PT

DSPIC33AK256MPS305T-I/M7

DSPIC33AK256MPS305T-I/PT

DSPIC33AK256MPS306-E/M7

DSPIC33AK256MPS306-E/PT

DSPIC33AK256MPS306-H/M7

dsPIC33AK256MPS306 Family Silicon Errata and Data Sheet Clarification

dsPIC33AK256MPS306 Family

Introduction

The dsPIC33AK256MPS306 family devices that you have received conform functionally to the current device data sheet (DS70005629**C**), 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 dsPIC33AK256MPS306 family 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 (**A1**).

Data sheet clarifications and corrections start on Data Sheet Clarifications, 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 website (www.microchip.com).

To determine the silicon revision level using MPLAB IDE with a hardware debugger, follow these steps:

1. Connect the device to the hardware debugger using the appropriate interface.
2. Open an MPLAB® X IDE project.
3. Configure the MPLAB X IDE project for the correct device and hardware debugger.
4. In MPLAB X IDE, navigate to **Window** > **Dashboard**, and then click the **Refresh Debug Tool Status** icon ( ).
5. The part number and Device Revision ID value appear in the Output window, depending on the development tool used.
   Note: If you cannot retrieve the silicon revision level, contact your local Microchip sales office for support.

The following table lists the DEVREV values for the various silicon revisions of the dsPIC33AK256MPS306 family.

Table 1. Silicon DEVREV Values

Part Number	Device ID (1)	Revision ID for Silicon Revision (1)	
		A0	A1
dsPIC33AK128MPS303	0xB514	0x0000	0x0001
dsPIC33AK128MPS305	0xB515		
dsPIC33AK128MPS306	0xB516		
dsPIC33AK256MPS303	0xB51C		
dsPIC33AK256MPS305	0xB51D		
dsPIC33AK256MPS306	0xB51E		
dsPIC33AK128MPS103	0xB504		
dsPIC33AK128MPS105	0xB505		
dsPIC33AK128MPS106	0xB506		
dsPIC33AK256MPS103	0xB50C		
dsPIC33AK256MPS105	0xB50D		
dsPIC33AK256MPS106	0xB50E		

Note:

1. Refer to the *"dsPIC33AK256MPS306 Family Flash Programming Specification"* for detailed information on Device and Revision IDs for your specific device.

Table 2. Silicon Issue Summary

Module	Feature	Item Number	Issue Summary	Affected Revisions (1,2)	
				A0	A1
PWM	PCI	1	In Complementary Output mode, when PSYNC is enabled in PCI Edge Detect mode, PCI override can be applied twice.	X	X
PWM	ADC Trigger Events	2	In LLC mode, when TRIGy is equal to EOC and with CAPTREN set, ADC triggers are not generated when the ADC trigger source is TRIGy.	X	X
PWM	Override	3	When PWML is configured for active override, the PWM outputs are not overridden.	X	X
I3C	IBI Data Threshold	4	IBI data threshold (I3C1QUETHLDCON [IBIDATTHLD]) default value of 0 can cause the device to halt code execution.	X	X
I3C	Target Mode	5	In Target mode, during data reception with DMA enabled, when the data buffer threshold I3C1BUFTHLD[RXTHLD] is configured greater than 0, data might be incorrectly stored in the RAM.	X	X
PWRM	Enable	6	The ON bit in VM1CON[31] will not enable the module.	X	
PWRM	Event Status	7	On device start up, the power monitor module can set the UVBG bit in the VM1EVENT[0] status bit erroneously.	X	
PWRM	Scan	8	If the power monitor is enabled during programming, the device ID might fail to be read.	X	
GPIO	IO Current Limit	9	The source/sink capability of IO pins is less than the electrical specifications.	X	
CPU	Interrupt Vector	10	Interrupt vectoring and simultaneous PBU/cache invalidation may stall the CPU indefinitely.	X	X
SILICON	Silicon Temperature Variants	11	Silicon only supports Industrial temperature (-40°C to 85°C) and not Extended temperature (-40°C to 125°C).	X	
DMA	Descriptor Write-Back	12	With descriptor write-back enabled, the DMA always increments/decrements the payload pointer by 4 bytes, regardless of the descriptor's configured element size.	X	X

Table 2. Silicon Issue Summary (continued)

Module	Feature	Item Number	Issue Summary	Affected Revisions (1,2)	
				A0	A1
DMA	Descriptor Mode	13	If descriptor mode is enabled on only the source or only the destination, the DMA may use the previous transfer's element size for alignment checking, potentially causing a false ADRERR/bus trap.	X	X
SPI	Host Mode	14	SPI Host mode may exhibit unreliable operation when using peripheral bus clock.	X	X
In-Circuit Debugger	Breakpoints	15	If a hardware breakpoint is placed immediately after a conditional branch instruction, the breakpoint can be missed and will result in no debugger halt.	X	X

Notes:

1. Only the issues marked with an "X" in the last column apply to the current silicon revision.
2. If the column cell is unmarked or blank, it indicates the issue is resolved in the latest revised silicon.

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 (**A1**).

1. Module: PWM

In Complementary Output mode, when PSYNC is enabled in PCI Edge Detect mode, PCI override can be applied twice.

Workaround

1. Ensure that the PCI active signal does not have a deassert period of less than 4 PWM clock cycles.
2. Use EOC synchronization or Latched modes to control the timing of PCI deassertion.

Affected Silicon Revisions

A0	A1						
X	X						

2. Module: PWM

In LLC mode, when TRIGy is equal to EOC and with CAPTREN set, ADC triggers are not generated when the ADC trigger source is TRIGy.

Workaround

Program the initial PGxTRIGy register value (where y = C, D, E, or F as selected by CAPTRSEL) to 50% of the EOC value.

Affected Silicon Revisions

A0	A1						
X	X						

3. Module: PWM

When the PWML output is configured for active high override (PGxIOCON2bits.OVRENL = 1 and PGxIOCON2bits.OVRDAT = 1), it is expected that PWMH will be forced to the inactive state and PWML to the active state. However, both output pins are not overridden as expected, and normal PWM output continues to appear on both pins.

Workaround

None. Only PCI inputs may override PWML to active.

Affected Silicon Revisions

A0	A1						
X	X						

4. Module: I3C

IBI data threshold (I3C1QUETHLDCON [IBIDATTHLD]) default value of 0 can cause the device to halt code execution.

Workaround

The user must set I3C1QUETHLDCON [IBIDATTHLD] = 1 or greater to ensure the device does not halt code execution.

Affected Silicon Revisions

A0	A1						
X	X						

5. Module: I3C

In Target mode, during data reception with DMA enabled, when the data buffer threshold I3C1BUFTHLD[RXTHLD] is configured greater than 0, data might be incorrectly stored in the RAM.

Workaround

Set I3C1BUFTHLD[RXTHLD] to a value of 0 to avoid data corruption.

Affected Silicon Revisions

A0	A1						
X	X						

6. Module: PWRM

The ON bit in VM1CON[31] does not enable the power monitor module as intended and should not be set.

Workaround

Only enable the power monitor module via the configuration fuse bit FPWRM.

Affected Silicon Revisions

A0	A1						
X							

7. Module: PWRM

The UVBG bit in VM1EVENT[0] can be set erroneously on device start up. This can be safely ignored if the device is not actively resetting.

Workaround

If the power monitor is enabled, clear VM1EVENT on device start up as part of the device and module initialization.

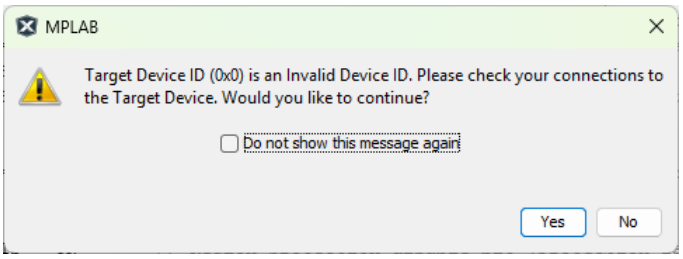
Affected Silicon Revisions

A0	A1						
X							

8. Module: PWRM

If the power monitor module is enabled via the FPWRM configuration bit, during device programming/ICSP entry, the power monitor may trigger a reset event during the loading of the configuration fuses, and as a result, an incorrect DEVID message may appear.

Figure 1. Example of Incorrect DEVID from MPLAB X IDE



Workaround

Select **Yes** in the message prompt or the equivalent of 'Yes' in the programming environment, and the device will program as expected.

Affected Silicon Revisions

A0	A1						
X							

9. Module: GPIO

When a GPIO is configured as an output and driven low, if an external signal sinks more than 8.7 mA of current into a 4x pin, or 11.6 mA on an 8x pin, the pin may become damaged. Similarly, when a GPIO is configured as an output and driven high, if an external signal sources more than 5.8 mA of current from a 4x pin, the pin may become damaged. 8x pins are not affected by this condition.

Workaround

If an I/O is configured as an output and driven high or low, limit the current to or from the pin with a series resistor, or avoid direct pin contention scenarios.

Affected Silicon Revisions

A0	A1						
X							

10. Module: CPU

If a Prefetch Branch Unit (PBU) cache invalidation request is issued and immediately followed by the CPU reading an IVT entry from Flash for exception processing, the CPU may enter a perpetually stalled state. The CPU will only recover in response to a hardware reset (Watchdog Timeout, $\overline{\text{MCLR}}$, POR, etc.). Cache invalidation requests capable of triggering this erratum have three possible sources:

1. Manual cache invalidation: Firmware or DMA writes a '1' to the CHECON[CHEINV] bit.
2. RTSP operations: Firmware or DMA writes a '1' to the NVMCON[WR] bit while CHECON[CHECOH] == 1 and NVMADR < 0xC00000.
3. BOOTSWP instruction executed.

Workaround

Either of the following workarounds can be used to overcome the issue.

Workaround 1

Place the IVT in RAM instead of Flash. This option is recommended for the best performance, as all interrupts will have 2-5 clocks less latency, and up to 3 clocks of latency jitter will be removed each time the CPU fetches an IVT entry.

Example 1

```
// Copies IVT from flash into RAM and begins using the RAM copy.
// Avoids cache invalidation errata. Call once at startup/initialization.
void SetIVTInRAM(void)
{
    static __attribute__ ((aligned(0x40))) void * ram_ivt[__IVT_NUM];
//  __IVT_NUM requires XC-DSC v3.20 or newer

// If undefined, change to highest implemented vector number

    if (IVTBASE != (uint32_t)ram_ivt)
    {
          __builtin_memcpy(ram_ivt, (const void*)IVTBASE, sizeof(ram_ivt));
          uint32_t PACCON1Save = PACCON1;
          PACCON1 |= 0x000D0000UL;
// BMXIRAMLWR/BMXIRAMHWR/IVTBASEWR must be set
          while((PACCON1 & 0x000D000DUL) != 0x000D0000UL);
// Infinite loop if lock bits set.
// SetIVTInRAM() MUST be called before any code that locks BMXIRAMLWR/BMXIRAMHWR/IVTBASE.
          int originalIPLMask = __builtin_write_DISICTL(7);
          BMXIRAML = (uint32_t)ram_ivt;
// Specify ram_ivt[]'s address range as read-only + executable
          BMXIRAMH = ((uint32_t)ram_ivt) + sizeof(ram_ivt);
          IVTBASE = (uint32_t)ram_ivt;
          PACCON1 = PACCON1Save;
          __builtin_write_DISICTL(originalIPLMask);
    }
}
```

Workaround 2

Temporarily block all IVT vector fetches while triggering cache invalidation events. This option suspends interrupts but continues to allow traps to be detected and immediately serviced by a common handler routine.

Example 2

```
void  __attribute__ ((interrupt, nocontext)) IVTReadFailException(void)
{
     // Common trap handler code - will be reentrant if multiple dissimilar traps
     // of increasing priority take place.
     //
     // Individual flag bits in INTCON1 and INTCON3 to INTCON5 may be polled to
     // determine the nature of this trap. Only flags that are serviced should be
     // cleared if this function is allowed to return so as to preserve status
     // for recursive instances.
}

// Cache invalidation errata workaround
void CacheInvalidationEvent(void)
{
    VFA = (uint32_t)IVTReadFailException;
// Fallback address to vector to when the IVT can't be read via the CPU-I bus
    uint32_t PACCON1Save = PACCON1;
    PACCON1bits.IVTBASEWR = 1;            // Must have IVTBASE write access
    while(PACCON1bits.IVTBASELK);         // Infinite loop if locked. IVTBASELK must never be set
                                          // when calling CacheInvalidationEvent()
    uint32_t IVTBASESave = IVTBASE;
    int DISICTLSave =  __builtin_write_DISICTL(7); // Mask interrupts <= IPL7
    IVTBASE = 0;
// Set illegal IVT base address in SFR space: CPU will jump to [VFA] if a trap occurs
     NVMCONbits.WR = 1; /* and/or */ CHECONbits.CHEINV = 1; /* and/or BOOTSWP */
    (void)CHECON;                         // Dummy SFR read to ensure the prior SFR write
                                          // instruction has exited CPU pipeline
    IVTBASE = IVTBASESave;                // Reenable IVT usage
    PACCON1 = PACCON1Save;
    __builtin_write_DISICTL(DISICTLSave); // Reenable interrupts
}
```

Since this workaround blocks interrupts, and RTSP code often sets NVMCON[WR] many times before executing from any of the erased/reprogrammed address ranges, it is suggested that the CHECON[CHECOH] bit be cleared at Reset/code initialization. In such cases, this workaround would only need to be applied to manual cache invalidation and BOOTSWP events, not to NVMCONbits.WR = 1 operations. After all RTSP operations are complete, one manually triggered cache invalidation can be performed if there is any chance that stale instructions remain in the cache from before starting the RTSP session.

Example 3

```
int main(void)
{
    CHECONbits.CHECOH = 0;
// Disable automatic cache invalidation each time NVMCONbits.WR is set - do manual invalidation
// only
    // Do all RTSP/bootloader operations
    CacheInvalidationEvent(); // Set CHECONbits.CHEINV = 1 in here, not NVMCONbits.WR
    // Safe to begin executing from the reprogrammed flash regions now
```

Affected Silicon Revisions

A0	A1						
X	X						

11. Module: SILICON

Silicon only supports Industrial temperature (-40°C to 85°C) and not Extended temperature (-40°C to 125°C).

Workaround

None.

Affected Silicon Revisions

A0	A1						
X							

12. Module: DMA

In the DMA controller's descriptor write-back feature, after a transfer, the payload pointer is always incremented or decremented by 4 bytes, regardless of the configured transfer element size (SIZEn, e.g., byte, half-word, or word) in the descriptor.

Workaround

Configure the descriptor table to use SIZEn = 2 (32-bit transfer size) for all transfers.

Affected Silicon Revisions

A0	A1						
X	X						

13. Module: DMA

When DMA descriptor mode is enabled on either the source or the destination, but not both, the DMA checks for address alignment against the transfer size of the previously completed transfer rather than the current channel's programmed transfer size in the descriptor. Consequently, if the previous transfer used a 32-bit (word) size and the current transfer is configured for 8-bit (byte), the DMA may incorrectly flag an Address Alignment Error (ADRERR) and generate a bus trap if the address is not word-aligned.

Workaround

Enable descriptor mode on both the source and destination sides (SDTEN = 1 and DDTEN = 1), or ensure all DMA transfers use a 32-bit transfer size with word-aligned addresses.

Affected Silicon Revisions

A0	A1						
X	X						

14. Module: SPI

When the SPI module operates in Host mode (MSTEN = 1) with the peripheral bus selected as the clock source (MCLKEN = 0), the module may exhibit unreliable operation. Internal synchronization timing cannot be guaranteed across all operating conditions. This issue occurs regardless of SPI clock polarity/phase settings (CKP, CKE, SMP bits) and affects both dedicated SPI pins and remappable (PPS) pins. Client mode operation is not affected.

Workaround

Configure the SPI module to use the dedicated clock generator (CLKGENx) as the clock source by setting SPIxCON1.MCLKEN = 1 when the module is used in Host mode (MSTEN = 1).

Affected Silicon Revisions

A0	A1						
X	X						

15. Module: In-Circuit Debugger

When the Program Counter matches certain hardware breakpoint addresses, a halt in debug mode may be missed. Breakpoint addresses subject to being missed are the non-speculative execution addresses associated with a conditional branch, as well as the targets of dynamic program flow change instructions, such as computed branch offsets in C switch statements, function return addresses and function pointer destinations. The breakpoint will be missed if the NVM is busy at the time of the address match, resulting in breakpoints at uncached addresses being susceptible, while the same breakpoint address reached via a looping construct will normally trigger a successful debug halt.

Workaround 1

Enable software breakpoints.

Workaround 2

Add a second hardware breakpoint or relocate program breakpoints suspected of being missed to the next address or line of code.

Affected Silicon Revisions

A0	A1						
X	X						

Data Sheet Clarifications

The following typographic corrections and clarifications are to be noted for the latest version of the device data sheet (DS70005629**C**).

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

None.

1. Module: 40 MSPS Analog-to-Digital Converter (ADC)

Replace **Table 16-3. TRGnSRC Trigger Source Selection Bits** with the following table:

Value	Description
011001-011110	Reserved
011111	ADTRG31 (PPS)
11000	**QEI1 Capture Event Output**
10111	**RDC ADC Trigger**
10110	**SCCP4 Trigger**
10101	**SCCP3 Trigger**
10100	**SCCP2 Trigger**
10011	**SCCP1 Trigger**
10010	**PTG Trigger 12**
10001	**SCCP5 (MCCP) Trigger**
10000	**CLC4 Out**
1111	**CLC3 Out**
1110	**CLC2 Out**
1101	**CLC1 Out**
1100	ITC
1011	**PWM4 ADC Trigger 2**
1010	**PWM4 ADC Trigger 1**
1001	**PWM3 ADC Trigger 2**
1000	**PWM3 ADC Trigger 1**
111	**PWM2 ADC Trigger 2**
110	**PWM2 ADC Trigger 1**
101	**PWM1 ADC Trigger 2**
100	**PWM1 ADC Trigger 1**
11	Conversion repeat timer trigger defined by RPTCNT[5:0] (ADnCON[23:18]) bits
10	Immediate re-trigger request
01	Software trigger initiated by using the ADnSWTRG register
0	Triggers are disabled

2. Module: I/O Ports with Edge Detect

Replace **Table 11-14. IOM 1-8 Group A Reference Pins** with the following table:

REFSEL	Pin Function Name	Pin
1011	**IOMON1D11**	**RPV3**
1010	**IOMON1D10**	**RPV2**
1001	**IOMON1D9**	**RPV1**
1000	**IOMON1D8**	**RPV0**
0111	IOMAD7	RC2
0110	IOMAD6	RC5
0101	IOMAD5	RC3
0100	IOMAD4	RC4
0011	IOMAD3	RD0
0010	IOMAD2	RD1
0001	IOMAD1	RD2
0000	IOMAD0	RD3

Replace **Table 11-15. IOM 1-8 Group A Feedback Pins** with the following table:

FBKSEL	Pin Function Name	Pad
1011	**IOMON1F11**	**IOIM3R[7:0]**
1010	**IOMON1F10**	**IOIM2R[7:0]**
1001	**IOMON1F9**	**IOIM1R[7:0]**
1000	**IOMON1F8**	**IOIM0R[7:0]**
0111	IOMAF7	RC7
0110	IOMAF6	RC1
0101	IOMAF5	RC0
0100	IOMAF4	RD7
0011	IOMAF3	RD8
0010	IOMAF2	RA0
0001	IOMAF1	RA7
0000	IOMAF0	RB7

Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision.

Revision	Date	Description
A	3/2026	Initial revision; issued for silicon revision **A0**.
B	7/2026	Added silicon issues 14 (14. Module: SPI) and 15 (15. Module: In-Circuit Debugger).
C	7/2026	Added data sheet clarifications 1. Module: 40 MSPS Analog-to-Digital Converter (ADC) and 2. Module: I/O Ports with Edge Detect. Updates to workaround code in 10. Module: CPU.

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.

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.