Migration Guide From AVR® to PIC32CM MCUs

AVR to PIC32CM MCUs

Introduction

PIC32CM devices are a family of 32-bit microcontrollers (MCUs) from Microchip Technology, based on the Arm® Cortex®-M0+ and M23 cores. These devices offer higher performance, more memory, and advanced peripherals compared to traditional 8-bit microcontrollers, such as the AVR® family. This document provides a comprehensive comparison of various features across the AVR and PIC32CM device families, helping users decide whether a PIC32CM device family can serve as an alternative solution to AVR.

If users choose a PIC32CM MCU, this document helps ease the migration process by associating concepts between AVR and PIC32CM. Users can then refer to the Migration Guide from AVR and PIC16/18F to PIC32CM Development Tools Ecosystem for a comprehensive comparison of the tools. These documents also provide links to other PIC32CM “Getting Started” documents for additional reference.

Table of Contents

1. Migrating From AVR® to PIC32CM

In cases where an AVR application requires a higher-performing device, users can upgrade to a PIC32CM MCU. The MPLAB® tools ecosystem provides a variety of free drivers and libraries through MPLAB Code Configurator (MCC) Harmony to help users get started with developing PIC32CM applications. This section outlines some key benefits and challenges of migrating from AVR to PIC32CM.

1.1. Key Benefits

Migrating to PIC32CM MCUs enables AVR customers to leverage the following benefits:

1. Enhanced Performance and Processing Power.
   - 32-bit Arm Cortex-M0+ and M23 cores deliver higher computational performance and efficiency
   - Higher clock frequencies enable faster execution of complex algorithms and real-time tasks
2. Increased memory resources support larger and more sophisticated applications.
3. Low power consumption through reduced minimum supply voltages and advanced low-power modes, making PIC32CM MCUs ideal for battery-powered and energy-sensitive applications.
4. Expansion of applications (device family-dependent) with:
   - More advanced analog and digital peripheral sets
   - Enhanced connectivity and integration with Controller Area Network with Flexible Data-Rate (CAN-FD)
   - Integrated security features including hardware cryptography, secure boot, TrustZone® technology, and tamper detection
   - Functional safety compliance (ISO 26262, IEC 61508, and IEC 60730 standards), suitable for automotive and industrial safety applications
5. Future-proof platform with the Arm ecosystem.
   - Access to a broad range of development tools, middleware, and third-party support within the Arm ecosystem
6. Familiar development experience using the MPLAB toolchain.
   - MPLAB Tools for Visual Studio Code (VS Code®), MPLAB Discover, and MPLAB MCC Harmony provide streamlined development
   - Comprehensive documentation and support, with extensive migration guides, application notes, and community resources

1.2. Challenges

Migrating to PIC32CM MCUs may come with challenges, including:

1. Core architectural differences that impact instruction sets, data handling, and code optimization.
2. Memory management strategies may require changes because PIC32CM MCUs have larger and more complex memory maps.
3. Differences between similar peripherals, such as their names, features and configuration methods, will require additional learning.
4. Code written for AVR architectures (especially low-level or assembly code) will not be directly compatible and will require adaptation. This also applies to drivers generated by MCC Melody and Harmony.
5. Some third-party libraries or code examples for AVR may not be available or directly compatible with PIC32CM, requiring fresh solutions or adaptations.

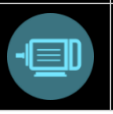
6. Documentation structures and terminologies in PIC32CM data sheets and reference manuals may differ from those for AVR devices.

2. Target Applications

All AVR and PIC32CM devices are general-purpose MCUs suitable for consumer electronics, industrial control, and IoT applications. These devices can operate with low power for energy-efficient applications. Some device families have additional features that make them suitable for specific applications.

This section aims to ease the migration path by helping users identify which PIC32CM MCU can serve as an alternative solution based on the AVR application.

Table 2-1. AVR® and PIC32CM Target Applications

Device	Ultra-Low Power	Connectivity (3)	Analog Sensing(4)	Touch	Display	Motor Control	Automotive	Functional Safety(1,2)	Security
									
AVR®									
tinyAVR®	tinyAVR® tinyAVR® A tinyAVR® 0/1/2	—	tinyAVR® 1	tinyAVR® 1	—	—	tinyAVR® 0/1/2	tinyAVR® 0/1/2(1) ATtiny25/45/85 (1)	—
megaAVR®	megaAVR® A/P megaAVR® 0	megaAVR® U AT90CAN AT90USB	—	—	megaAVR® A/P 329/3290 649/6490	AT90PWM	megaAVR® 0 megaAVR® M/C/P AT90CAN	megaAVR® 0(1) megaAVR® PB(1)	—
AVR® XMEGA®	AVR® XMEGA®AU/B /C/D	AVR® XMEGA® A/B/C	—	—	AVR® XMEGA® B	AVR® XMEGA® AU	—	—	—
AVR®	—	AVR® DU	AVR® DB/EA/EB	AVR® DA	—	AVR® EB	AVR® DA/DB/DD	AVR® SD(2) AVR® DA/DB/DD/ EA(1)	—
PIC32CM									
MC00	—	—	✓	—	—	✓	—	✓ (1)	—
LE00 LS00/60	✓	✓	✓	✓	—	—	—	—	✓
JH00/01	—	✓	—	✓	—	✓	✓	✓ (2)	—
GV00	—	—	—	✓	—	—	—	—	—
GC00/SG00	—	✓	—	✓	—	—	✓	✓ (2)	✓
PL10(5)	✓	—	—	✓	—	—	✓	✓ (2)	—

Notes:

1. The PIC32CM MC and most AVR devices have Functional Safety support available, including Failure Modes, Effects, and Diagnostic Analysis (FMEDA), Functional Safety manuals, and self-test libraries.
2. AVR SD, PIC32CM GC/SG, JH00/01, and PL10 are designed in compliance with Functional Safety standards (ISO 26262, IEC 61508).
3. Most devices support only the common serial communication peripherals (UART, SPI, and I^2C). This column lists the family of devices that support other protocols, such as USB, CAN and I3C®.
4. Most devices have ADCs and comparators. This column lists device families that contain other peripherals, such as op amps and advanced ADCs.
5. PIC32CM PL10 is package- and pin-compatible with AVR Dx/Ex, with familiar Core Independent Peripherals (CIPs) from AVR.

3. Feature Summary

This section summarizes the system and peripheral features for each device family: AVR® and PIC32CM.

Table 3-1. AVR® and PIC32CM Device Feature Summaries

Features	Core	AVR®				PIC32CM					
		AVR®-RISC				Cortex®-M0+				Cortex®-M23	
	Family	tinyAVR®	megaAVR®	AVR® XMEGA	AVR®	MC00	JH00/01	GV00	PL10	LE00, LS00/60	GC00, SG00
Operating Conditions	Max. CPU Clock (MHz)	20	20	32	24	48	48	48	24	48	72
	Power Supply (V)	1.8-5.5	1.8-5.5	1.6-3.6	1.8-5.5	2.7-5.5	2.7-5.5	1.62-3.63	1.8-5.5	1.62-3.63	1.71-3.63
	Temperature (Celsius)	-40 to 125	-40 to 125	-40 to 125	-40 to 125	-40 to 125	-40 to 125	-40 to 125	-40 to 125	-40 to 125	-40 to 125
Memory	Flash (Bytes)	512-32K	8K-256K	8K-384K	16K-128K	64K-128K	32K-512K	16K-32K	32K-128K	128K-512K	512K
	SRAM (Bytes)	32-3K	1K-32K	1K-32K	2K-16K	8K-16K	4K-64K	2K-4K	4K-16K	16K-64K	128K
	EEPROM (Bytes)	Up to 256 (tinyAVR® 0/1/2)	Up to 4K	Up to 4K	Up to 512	N/A	Up to 16K via Flash (64/32 JH00 only)	Up to 16K via Flash	N/A	N/A	N/A
System Flexibility	Power-on-Reset (POR)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Direct Memory Access (DMA)	N/A	N/A	AVR® XMEGA A, B, E	For USB only	Yes	Yes	N/A	Yes	Yes	Yes
	Event System (EVSYS)	tinyAVR® 0/1/2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Reset Control (RSTC)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Low Power	Idle	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Standby	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Sleepwalking	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Power-Down/Off	Yes	Yes	Yes	Yes	N/A	N/A	N/A	N/A	Yes	Yes
User Interface	Pin Count	4-24	28-100	32-100	16-64	32-48	32-100	32-64	20-64	32-100	48-100
	Programmable I/O	Up to 22	Up to 69	Up to 53	Up to 55	Up to 38	Up to 84	Up to 52	Up to 55	Up to 80	Up to 80
	External Interrupt	All GPIO	All GPIO	All GPIO	All GPIO	Up to 16	Up to 16	Up to 16	Up to 16	Up to 16	Up to 16
	Multi-Voltage I/O (MVIO)	N/A	N/A	N/A	AVR® DB/DD	N/A	N/A	N/A	Yes	N/A	N/A

Table 3-1. AVR® and PIC32CM Device Feature Summaries (continued)

Features	Core	AVR®				PIC32CM					
		AVR®-RISC				Cortex®-M0+				Cortex®-M23	
	Family	tinyAVR®	megaAVR®	AVR® XMEGA	AVR®	MC00	JH00/01	GV00	PL10	LE00, LS00/60	GC00, SG00
Intelligent Analog	ADCs	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	DAC	tinyAVR® 1 only	N/A	AVR® XMEGA A,E	AVR® DA/DB/ DD/EA	Yes	512/256K only	Yes	N/A	Yes	N/A
	Analog Comparators (AC)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Zero Cross Detect (ZCD)	N/A	N/A	N/A	AVR® DA/DB/ DD	N/A	N/A	N/A	N/A	N/A	N/A
	Internal V_{REF} (V)	0.55-4.096	0.55-4.03	1V	1.024-4.096	1.024-4.096	1.024-4.096	1	1.024-4.096	1-2.4	1.20326
	Op Amp	N/A	N/A	N/A	AVR® DB only	N/A	N/A	N/A	N/A	Yes	N/A
Touch	Peripheral Touch Controller (PTC)	tinyAVR® 1 only	N/A	N/A	AVR® DA only	N/A	Yes	Yes	Yes	Yes	Yes
LCD	LCD Controller	N/A	megaAVR® A/P, 329/3290, 649/6490	AVR® XMEGA B	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Timing and Measurement	TC (Timer/ Counters)	Up to 16-bit	Up to 16-bit	Up to 32-bit	Up to 24-bit	Up to 32-bit	Up to 32-bit	Up to 32-bit	Up to 32-bit	Up to 32-bit	Up to 32-bit
	Capture/ Compare	via TCx	via TCx	via TCx	via TCx	via TC/TCC	via TC/TCC	via TC/TCC	via TC/TCC	via TC/TCC	via TC/TCC
	Real Time Counter (RTC)	16-bit (tinyAVR® 0/1/2)	16-bit	Up to 32-bit	16-bit	32-bit	32-bit	32-bit	32-bit	32-bit	32-bit
	Frequency Meter (FREQM)/ Measurement	via TCB	via TCB	via TCx	via TCB	Yes	Yes	N/A	N/A	Yes	Yes
Waveform Control/ Generation	PWM	via TCx	via TCx	via TCx	via TCx	via TC/TCC	via TC/TCC	via TC	via TC/TCC	via TC/TCC	via TC/TCC
	Waveform Extensions (WEX)	N/A	N/A	Yes	AVR® EB only	via TCC	via TCC	N/A	via TCC	via TCC	via TCC
	Numerical Controller Oscillator (NCO)	N/A	N/A	N/A	AVR® EB only	N/A	N/A	N/A	N/A	N/A	N/A
	Position Decoder (PDEC)	N/A	N/A	N/A	N/A	Yes	512/256/ 128K only	N/A	N/A	N/A	N/A

Table 3-1. AVR® and PIC32CM Device Feature Summaries (continued)

Features	Core	AVR®				PIC32CM					
		AVR®-RISC				Cortex®-M0+				Cortex®-M23	
	Family	tinyAVR®	megaAVR®	AVR® XMEGA	AVR®	MC00	JH00/01	GV00	PL10	LE00, LS00/60	GC00, SG00
Communication/ Connectivity	USART	tinyAVR® 0/1/2	Yes	Yes	Yes	via SERCOM	via SERCOM	via SERCOM	via SERCOM	via SERCOM	via SERCOM
	SPI	tinyAVR® 0/1/2	Yes	Yes	Yes						
	I²C/TWI	tinyAVR® 0/1/2	Yes	Yes	Yes						
	RS-485	tinyAVR® 0/1/2	via USART	via USART	via USART	via USART	via USART	via USART	via USART	via USART	via USART
	LIN	tinyAVR® 0/1/2	via USART	via USART	via USART	via USART	via USART	via USART	via USART	via USART	via USART
	SMBus Compatible	tinyAVR® 0/1/2	via TWI	via TWI	via TWI	via I²C	via I²C	via I²C	via I²C	via I²C	via I²C
	USB	N/A	megaAVR® U AT90USB	AVR® XMEGA A, B, C	AVR® DU only	N/A	N/A	N/A	N/A	Yes	Yes
	CAN	N/A	AT90CAN	N/A	N/A	N/A	JH01 only	N/A	N/A	N/A	Yes
	I2S	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Yes	N/A
Logic and Math	Divide and Square Root Accelerator (DIVAS)	N/A	N/A	N/A	N/A	Yes	Yes	N/A	N/A	N/A	N/A
	Configurable Custom Logic (CCL)	tinyAVR® 0/1/2	Yes	AVR® XMEGA E	Yes	Yes	Yes	N/A	Yes	Yes	Yes
Safety and Monitoring	Brown-out Detection (BOD)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Voltage Level Monitor	Yes	Yes	N/A	Yes	N/A	N/A	N/A	Yes	Yes	Yes
	Watchdog Timer (WDT)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Synchronous WDT (SWDT)	N/A	N/A	N/A	AVR® SD only	N/A	N/A	N/A	N/A	N/A	N/A
	Temperature Sensor (TSENS)	via ADC	via ADC	via ADC	via ADC	Yes	N/A	via ADC	via ADC	N/A	via ADC
	Device Service Unit (DSU)	N/A	N/A	N/A	N/A	Yes	Yes	Yes	Yes	Yes	Yes
	CRC	tinyAVR® 0/1/2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Functional Safety (FuSa)	Functional Safety Compliant	tinyAVR® 0/1/2 ATtiny25/ 45/85 have FuSa Support Available	megaAVR® 0/PB have FuSa Support available	N/A	AVR® SD only AVR® DA/DB/ DD/EA have FuSa Support available	FuSa Support available	Yes	N/A	Yes	N/A	Yes

Table 3-1. AVR® and PIC32CM Device Feature Summaries (continued)

Features	Core	AVR®				PIC32CM					
		AVR®-RISC				Cortex®-M0+				Cortex®-M23	
	Family	tinyAVR®	megaAVR®	AVR® XMEGA	AVR®	MC00	JH00/01	GV00	PL10	LE00, LS00/60	GC00, SG00
Security	**TrustZone®**	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	LS00/60 only	Yes
	TrustRAM	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Yes	Yes
	Data Flash	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	LS00/60 only	Yes
	Secure Boot	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	LS00/60 only	via HSM-Lite (SG only)
	Tamper	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Yes	
	Crypto	N/A	N/A	AVR® XMEGA A, B	N/A	N/A	N/A	N/A	N/A	LS00/60 only	
	One True Random Number Generator (TRNG)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	Yes	
	Device Identity Composition Engine (DICE)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	LS00/60 only	Yes
	Physical Unclonable Function (PUF)	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	SG only

Notes:

- —Available in all families
- —Available in specific families
- —Not available

4. Core and Architecture

This section provides a comparison of the AVR and PIC32CM MCUs' cores and architectures.

4.1. AVR®

The AVR MCU core features a Harvard architecture with separate instruction and data buses, maximizing performance through simultaneous access to program and data memory. It includes a rich set of 32 general-purpose working registers directly connected to the Arithmetic Logic Unit (ALU), enabling single-cycle instruction execution for most operations. Its architecture enables high-performance and low-power consumption, making AVR MCUs popular in embedded systems and consumer electronics.

4.2. PIC32CM

The PIC32CM MCUs implement either the Arm Cortex-M0+ or M23 processors. The Arm Cortex-M0+ and Cortex-M23 are both 32-bit MCU cores designed for low-power, cost-sensitive embedded applications, but they differ in architecture, features and security capabilities.

The Cortex-M0+ is based on the Armv6-M architecture and optimized for ultra-low power and simplicity, featuring a two-stage pipeline, basic interrupt handling, and a compact instruction set. In contrast, the Cortex-M23 is based on the newer Armv8-M architecture and introduces several enhancements, such as the optional Arm TrustZone technology, which enables the separation of secure and non-secure code for improved security in IoT and safety-critical applications. It also offers improved debug and trace capabilities, enhanced interrupt handling, and better support for modern embedded software development. While both cores are suitable for similar low-power applications, the Cortex-M23 is preferred when security and advanced features are required.

For more information on the Arm Cortex-M processors, refer to the product pages on the Arm website:

- Cortex-M0+—32-bit, Low-Power Processor at an 8-bit Cost
- Cortex-M23—Low-Power Microcontroller with TrustZone Security

Table 4-1 below provides a comparison of the cores and architectures of AVR and PIC32CM MCUs.

Table 4-1. AVR® and PIC32CM Core and Architecture

	AVR®	PIC32CM M0+	PIC32CM M23
Core	8-bit RISC	32-bit Arm® Cortex®-M0+	32-bit Arm® Cortex®-M23
Architecture	Harvard (separate program/data memory)	Von Neumann, 32-bit data path	Similar to M0+ with enhanced security (TrustZone®)
Instruction Set	8-bit, optimized for simple control tasks	Arm®v6-M, efficient for low-power/cost apps	Arm®v8-M, supports TrustZone® for secure execution
Power/Performance	Low power Up to 32 MHz	Ultra-low power Up to 48 MHz	Ultra-low power Up to 72 MHz
Use Cases	General-purpose Industrial IoT	General-purpose Industrial IoT	Secure IoT Hardware-based security Safety-critical

5. Related Peripherals and Features

AVR and PIC32CM MCUs share peripherals that support a wide range of embedded applications. While the architecture and implementation may differ, the peripherals listed in the table below are typically found in both families.

Table 5-1. AVR® and PIC32CM Common Peripherals

Peripheral Type	Description
Clock System	Internal/external oscillators, clock prescalers
Nonvolatile Memory (NVM)	Nonvolatile memory for data storage
Power Management	Sleep modes, low-power operation
Reset Controller	Manages device reset operations
Interrupt Controller	Handles external and internal interrupt sources
GPIO/PORT	General-purpose digital input/output pins
Direct Memory Access (DMA)	High-speed data transfers
Event Systems	Peripheral event generation and response
Timers/Counters	8/16/32-bit timers for timing, PWM, input capture, etc.
Real-Time Counter (RTC)	Timekeeping, periodic interrupts, low-power wake-up
Analog-to-Digital Converter (ADC)	Analog-to-digital conversion for sensor interfacing
Digital-to-Analog Converter (DAC)	Digital-to-analog conversion for internal voltage references
Analog Comparator (AC)	Compare analog voltages and threshold detection.
Multi-Voltage I/O (MVIO)	Certain I/O port pins powered by a different voltage domain than the rest of the MCU
Voltage Reference (V_{REF})	Programmable voltage references to analog peripherals
Configurable Custom Logic (CCL)	Programmable logic peripheral
Universal Synchronous Asynchronous Receiver Transmitter (USART)	Serial communication (asynchronous/synchronous)
Serial Peripheral Interface (SPI)	Synchronous serial host/client peripheral interface
Inter-Integrated Circuit (I^2C)/Two-Wire Interface (TWI)	Two-wire host/client serial communication
Brown-Out Detector/Reset (BOD/BOR)	Monitors supply voltage, triggers a reset on low voltage
Watchdog Timer (WDT)	System monitoring and recovery from software faults
Temperature Sensor	On-chip temperature monitoring
Cyclic Redundancy Check (CRC)	Verification for data integrity

Some AVR and PIC32CM families also share application-specific peripherals.

Table 5-2. AVR® and PIC32CM Application-Specific Peripherals

Peripheral Type	Description/Typical Use
Waveform Extensions	Advanced waveform control for motor control, power, etc.
Op amps	Operational amplifiers for analog signal conditioning
Peripheral Touch Controller (PTC)	Capacitive touch sensing
USB	USB 2.0 device support
CAN	CAN 2.0 and CAN FD support

The following sections discuss the similarities and differences of related peripherals in AVR and PIC32CM MCUs.

5.1. System Management

5.1.1. Clock Selection

AVR and PIC32CM MCUs offer flexible and robust clock systems that support multiple internal and external clock sources, programmable prescalers, and safe run-time switching. Both MCUs allow peripherals to request and receive clocks as needed, support low-frequency and high-frequency oscillators, and provide mechanisms for clock accuracy tuning and failure detection. The main clock can be prescaled, and peripherals can operate either synchronously or asynchronously with respect to the main clock. These features enable efficient power management, high performance, and reliable operation across a wide range of applications.

The main difference between AVR and PIC32CM lies in how clock management is integrated within the MCU. AVR uses a Clock Controller (CLKCTRL), while PIC32CM MCUs (depending on the device family) implement a clock system that uses a combination of the Generic Clock Controller (GCLK), Main Clock Controller (MCLK), System Controller (SYSCTRL), Oscillator Controllers (OSCCTRL and OSC32KCTRL), and Power Manager (PM). Table 5-3 provides a comparison of features between AVR and PIC32CM.

Table 5-3. AVR® and PIC32CM Clock Selection Features

Feature	AVR® CLKCTRL (Clock Control)	PIC32CM Clock System (GCLK, MCLK, SYSCTRL, OSCCTRLs, PM)
Main Clock Source	Selectable internal/external Up to 24 MHz OSCHF, 48 MHz PLL Up to 32 MHz (XMEGA® only)	Selectable via GCLK Up to 48 MHz OSC48M, 96 MHz FDPLL
Internal Oscillators	OSCHF (up to 24 MHz) OSC32K/OSCULP32K (32.768 kHz) RC OSC (up to 32MHz) (XMEGA® only)	OSCHF (up to 24 MHz) OSC8M (8 MHz) OSC48M (48 MHz) OSC32K (32.768 kHz)
External Oscillators	0.4–16 MHz crystal (XMEGA® only) 32.768 kHz crystal External clock	0.4–32 MHz XOSC 32.768 kHz XOSC32K External clock
PLL/DFLL/DPLL	48 MHz PLL (2x/3x multiplier) 128 MHz PLL (XMEGA® only)	DFLL48M (48 MHz) FDPLL96M (48–96 MHz)
Clock Distribution	Automatic peripheral clock request system	Generic Clock Controller (GCLK)
Prescaler Range	1x to 64x	1x to 128x (GCLK, MCLK)
Clock Domains	Main clock Some async peripherals	CPU AHB, multiple APBx(1) Per-peripheral domains
Safe Run-Time Switching	Supported	Supported (GCLK, MCLK)
Clock Gating/Masking	Automatic by peripheral request	Module-level clock gating via masks
Clock Failure Detection	Limited	Safe clock switch and event output
Calibration/Auto-Tuning	Auto-tuning for internal oscillators	Factory calibration Fine tuning
Power Management Integration	Peripheral ON/OFF in Standby mode	Power domain gating Wake-up on clock events
Brown-out Detector	Limited	BOD33 integrated Can trigger wake-up

Note:

1. The Advanced High-performance Bus (**AHB**) and Advanced Peripheral Bus (**APB**) are two types of internal bus architectures used in Arm-based microcontrollers. These buses are used to connect the CPU, memory and peripherals.

5.1.2. Nonvolatile Memory Controller

AVR and PIC32CM MCUs feature a Nonvolatile Memory Controller (NVMCTRL) that manages access to on-chip Flash memory and other nonvolatile memory regions. Both MCUs support in-system programming, self-programming, bootloader support, and configurable memory protection for secure code and data storage. Both controllers are designed for reliable program and data storage, with mechanisms for memory protection and flexible access. Table 5-4 shows a comparison of features between AVR and PIC32CM.

Table 5-4. AVR® and PIC32CM NVM Control Features

Feature	AVR® NVMCTRL	PIC32CM NVMCTRL
Bus Interface	8/16-bit CPU bus	32-bit Advanced High-Performance Bus (AHB) for memory read/write 32-bit Advanced Peripheral Bus (APB) for memory control
Flash Memory	Unified Flash (program/data)	Main Flash array
EEPROM	Separate EEPROM block	EEPROM emulation (RWWEE)
Memory Protection	Boot, application, data sections	Boot, 16 individually protected regions
Signature Row	Factory data Serial number Calibration bytes	Memory-mapped to AHB
User Row	User data Writable from software/UPDI	Memory-mapped to AHB
Wait State Configuration	Fixed/Programmable	Programmable
Cache	N/A	Direct-mapped cache
Power Management	Standard	Flash block power-down in sleep Wake on access
Security Bit/Device Lock	FUSEs, lock bits	Security bit, Device protection
Access in Sleep Modes	N/A	Wake on access or exit from sleep mode

5.1.3. Power Management

AVR and PIC32CM MCUs provide advanced low-power management features to minimize energy consumption during periods of inactivity. Both MCUs offer multiple sleep modes (such as Idle and Standby), allow the CPU to halt execution while retaining SRAM and register contents, and support wake-up from sleep via interrupts or reset events. The application determines which sleep mode to enter and when, and peripherals can be selectively enabled or disabled in certain sleep modes. These features make both MCUs suitable for battery-powered and energy-sensitive applications. Table 5-5 shows a comparison of features between AVR and PIC32CM.

Table 5-5. AVR® and PIC32CM Power Management Features

Feature	AVR® Sleep Controller (SLPCTRL)	PIC32CM Power Manager (PM)
Integration	Stand-alone SLPCTRL peripheral	Integrated into the Power Manager (PM) peripheral

Table 5-5. AVR® and PIC32CM Power Management Features (continued)

Feature	AVR® Sleep Controller (SLPCTRL)	PIC32CM Power Manager (PM)
Sleep Modes	Idle Standby Power-Down	Idle Standby Hibernate Backup Off
SleepWalking	Supported (in Standby)	Supported (in Standby, Hibernate, on GCLK clocks)
Power Domain Gating	Peripheral ON/OFF in Standby	Static (ON/OFF) or Dynamic in Standby/ Hibernate
I/O State Retention	Supported	Supported
SRAM/Registers Retention	Supported	Supported
Wake-up Sources	Interrupts Reset	Interrupts Reset

5.1.4. Reset Control

AVR and PIC32CM MCUs feature a Reset Controller that manages device resets, returns the MCU to a known initial state, and allows software to identify the source of the reset. Both controllers support multiple reset sources, including Power-on Reset (POR), Brown-out Detection (BOD), external reset pin, Watchdog Timer (WDT) reset, and software-initiated reset. They also provide status registers for software to determine the cause of the last reset, supporting robust system recovery and diagnostics. Table 5-6 shows a comparison of features between AVR and PIC32CM.

Table 5-6. AVR® and PIC32CM Reset Control Features

Feature	AVR® Reset Controller (RSTCTRL)	PIC32CM Reset Controller (RSTC/PM)
Integration	Stand-alone RSTCTRL peripheral	Differs per device family: Stand-alone RSTCTRL peripheral or integrated into the Power Manager (PM) peripheral
Power Reset Sources	Power-on Reset (POR) Brown-out Detector (BOD)	POR BOD
User Reset Sources	External Reset Watchdog Timer Software Reset UPDI Reset	External Reset Watchdog Timer Software Reset System Reset Request CPU Lockup Reset
Reset Status Register	Supported	Supported

5.1.5. Interrupts

AVR and PIC32CM MCUs provide interrupt management systems that enable responsive and efficient handling of asynchronous events from peripherals and external sources. PIC32CM uses the Arm Nested Vectored Interrupt Controller (NVIC), which allows more complex interrupt nesting, priority management, and relocation. Table 5-7 shows a comparison of features between AVR and PIC32CM.

Table 5-7. AVR® and PIC32CM Interrupt Features

Feature	AVR® CPU Interrupt Controller (CPUINT)	PIC32CM Nested Vectored Interrupt Controller (NVIC)
Interrupt Prioritization	Two levels: Normal (with optional round robin) and High	Four programmable priority levels (M0+) Eight levels (M23 with Secure/Non-secure split)
Non-Maskable Interrupt	Supported	Supported
Vector Table	Supported	Supported
Interrupt Sources	Peripherals External pins	Peripherals External pins
Interrupt Flag Handling	Set/cleared in the peripheral INTFLAGS register	Set/cleared in the peripheral and NVIC registers
Global Enable/Disable	Supported	Supported
Response Time (minimum)	Four clock cycles	15 clock cycles
Priority Configuration	Fixed (except for one high-priority interrupt)	Fully programmable per interrupt
Scheduling Scheme	Optional round robin for normal priority	Hardware-based preemption, and tail-chaining
Security Context	N/A	Secure/Non-secure split (M23)
Register Access	Byte/word accessible	Word-only (little-endian)

5.1.6. Device Memory Access (DMA) Controller

AVR and PIC32CM DMA controllers enable high-speed data transfers between memory and peripheral registers without CPU intervention. This improves system efficiency, reduces CPU load, and enables real-time data handling for tasks such as serial communication, ADC sampling, and memory-to-memory transfers. Table 5-8 shows a comparison of features between AVR and PIC32CM.

Table 5-8. AVR® and PIC32CM DMA Features

Feature	AVR®	PIC32CM DMAC
Number of Channels	Up to four	Up to 16
Transfer Types	Peripheral-to-memory Memory-to-memory	Peripheral-to-memory Memory-to-memory
Trigger Sources	Peripheral events Software triggers	Peripheral events Software triggers External triggers
Addressing Modes	Fixed, increment or decrement	Fixed, increment or decrement
Burst/Block Transfer	Supported	Supported
Priority Levels	Fixed or simple priority	Programmable priorities
Descriptor Support	Basic (repeat/auto modes)	Advanced (linked list descriptors for complex transfers)
Data Width	8-bit	8/16/32-bit
Interrupt Support	Supported	Supported
CRC Check	Through separate CRC	Built-in

5.1.7. Event Systems

The AVR and PIC32CM Event System (EVSYS) peripherals enable direct and autonomous communication between peripherals without CPU intervention. This system allows peripherals to generate and respond to events, providing low-latency, predictable. and synchronized actions across multiple modules. By offloading inter-peripheral signaling from the CPU, the Event System reduces software complexity, improves real-time performance, and enables advanced features like Core

Independent Peripherals (CIPs) and SleepWalking (peripheral operation in sleep modes). Table 5-9 shows a comparison of features between AVR and PIC32CM.

Table 5-9. AVR® and PIC32CM EVSYS Features

Feature	AVR® Event System (EVSYS)	PIC32CM Event System (EVSYS)
Number of Event Channels	Up to ten parallel channels (device-dependent)	Up to 12 configurable channels (device-dependent)
Event Users/Generators	Peripherals Software	Peripherals Software
Channel Routing	Per channel: One generator, multiple users	Per channel: One generator, multiple users
Event Path Types	Asynchronous Synchronous	Asynchronous Synchronous Resynchronized (via GCLK)
Edge Detection	Configured through individual peripherals	Configurable edge detector
Sleep Mode	Supported	Supported
CPU Offloading	Yes	Yes

5.2. User Interface

5.2.1. I/O Pins

AVR and PIC32CM MCUs provide highly configurable I/O pin controllers that allow individual pin configurations, including interrupt/event generation on pin changes and peripheral multiplexing for flexible hardware designs. Table 5-10 shows a comparison of features between AVR and PIC32CM.

Table 5-10. AVR® and PIC32CM I/O Features

Feature	AVR® PORT	PIC32CM PORT
Port Group Size	Up to eight pins per PORT	Up to 32 pins per PORT
Pin Multiplexing	PORTMUX for default/alternate peripheral functions	Software-controlled, highly flexible
Read-Modify-Write (RMW)	Hardware Read-Modify-Write (RMW) via toggle/set/clear registers	Atomic RMW via 8/16/32-bit writes
Input Sensing/Interrupts	Asynchronous pin change Interrupt-on-change	Configurable input sampling Interrupt-on-Change (IOC)
Voltage Domains	Multi-Voltage I/O (MVIO) support for certain device families: A subset of pins can use a separate VDDIO2 supply	Multi-Voltage I/O (MVIO) support for certain device families: A subset of pins can use separate VDDIO2 supplies
Supply Status Monitoring	Through ADC/AC MVIO: VDDIO2 status bit interrupt/event	Through ADC/AC MVIO: VDDIO2 status bit interrupt/event
Peripheral Override	Supported, independent of supply configuration	Supported, via the pin configuration register
I/O Bus Access	Bit-accessible I/O memory space (virtual ports)	AHB/APB bridge Arm® single-cycle IOBUS
Output Driver Options	Push-pull Open-drain control Slew rate control Tri-state Configurable driver strength Programmable inversion	Push-pull Open-drain control Slew rate control Tri-state Configurable driver strength High-sink pins

Table 5-10. AVR® and PIC32CM I/O Features (continued)

Feature	AVR® PORT	PIC32CM PORT
Input Buffer Control	Enabled by disabling digital inputs on analog-capable pins	Supported Can be disabled for low power

5.2.2. MVIO

Multi-Voltage I/O (MVIO) is a feature available on specific AVR and PIC32CM MCUs that allows a subset of I/O pins to operate in a different voltage domain from the rest of the device. This hardware-level integration eliminates the need for external level shifters when the MCU must communicate with peripherals operating at different logic levels, such as a 1.8V sensor connected to a 5V system. Table 5-11 shows a comparison of MVIO features between AVR and PIC32CM.

Table 5-11. AVR® and PIC32CM MVIO Features

Feature	AVR® MVIO	PIC32CM MVIO
Integration	Stand-alone MVIO peripheral	Integrated into the Supply Controller (SUPC)
Primary Voltage Domain	VDD (main CPU and most I/O)	VDD (main CPU and VDDIO domain)
MVIO Voltage Domain	VDDIO2 (specific ports)	VDDIO2 (specific pins)
Operating Modes	Dual Supply mode: Independent VDD and VDDIO2	Dual Supply mode (independent) Single Power Supply mode (supply monitors turned off to save power)
Monitoring and Status	Status bit indicates if the voltage is in range	Status (OK) bit and Low-Power POR bit
Interrupt Support	Interrupt on VDDIO2 status change	Interrupt on VDDIO2OK or Low-Power POR events
Power Sequencing	Tri-states MVIO pins if VDDIO2 is below the acceptable range	Tri-states pins on power loss; reloads PORT configurations if VDDIO2 returns
Unique Features	Schmitt trigger levels are automatically scaled according to VDDIO2	Event System Integration: Can generate events based on the MVIO status

5.3. Timing, Measurement, and Waveforms

5.3.1. Timer/Counter (TC)

AVR and PIC32CM timer/counter peripherals are utilized for timing, waveform generation, input capture, and event handling. Both MCUs support multiple timer types with varying features for precise timing and waveform control. Table 5-12 shows a comparison of features between AVR and PIC32CM.

Table 5-12. AVR® and PIC32CM Timer Features

Feature	AVR® TCx	PIC32CM TCx/TCC
Resolution	12, 16 or up to 24 bits	8, 16 or 32 bits (selectable)
Prescaler	Programmable	Programmable
Capture/Compare Channels	Up to eight	Up to four
PWM Modes	Single/Dual-Slope Ramp	Single/Dual-Slope
Special Modes	Split mode (16-bit/2) Cascaded (8-bit x2) Ramp modes Numerical Controller Oscillator (NCO)	Circular buffer, multiple output events, error interrupt
Buffering	Double-buffered period/compare	Double-buffered period/compare/capture

Table 5-12. AVR® and PIC32CM Timer Features (continued)

Feature	AVR® TCx	PIC32CM TCx/TCC
Fault Protection	Fault handling, fast stop, overload protection	Recoverable/non-recoverable fault sources
Event System	Integrated into all types of clocking, direction, or synchronization	Integrated for input/output events, synchronization
Direct Memory Access (DMA)	Supported (AVR® XMEGA®)	Supported
Interrupts	Overflow Compare match Event-based	Overflow Compare match Input capture Error Fault
Waveform Extensions	Timer/Counter Type E (TCE), TC0/1 (AVR® XMEGA)	Timer Counter for Control Applications (TCC)

5.3.2. Waveform Extensions (WEX)

AVR and PIC32CM Waveform Extensions enhance timer/counter peripherals for precise and safe waveform control in advanced motor, power, LED control, and similar applications. In older AVR devices (AT90PWM), the waveform extension features are provided by the Power Stage Controller (PSC) module. Table 5-13 shows a comparison of features between AVR and PIC32CM.

Table 5-13. AVR® and PIC32CM WEX Features

Feature	AVR® WEX	PIC32CM TCC Waveform Extensions
Complementary Outputs	Supported	Supported
Output Distribution	Input Matrix (INMX) with multiple routing modes	Output Matrix (OTMX) with multiple routing modes
Dead-Time Insertion	Four DTI units Separate high-/low-side register 8-bit resolution Double-buffered	Four DTI units Common register 8-bit resolution Double-buffered
Swap Functionality	Four swap units for port pair or high/low side driver swap	Waveform swap option with double buffer support
Pattern Generation	Double-buffered Can distribute one channel to all pins	Double-buffered Dithering support
Dithering	N/A	Supported
Port Override/Disable	Channel distribution to all pins Output disable on selectable port pins	N/A
Fault Protection	Event-controlled, instant and predictable triggering, fault blanking, multiple event inputs	Fault protection for safe and deterministic shutdown

5.3.3. Real-Time Counters (RTC)

AVR and PIC32CM RTC peripherals are designed for accurate timekeeping, periodic interrupts, and low-power operation. The RTCs can run continuously in low-power modes, wake the device from sleep, and generate interrupts or events on compare match or overflow. Both RTCs also offer mechanisms for clock correction to improve long-term accuracy. Table 5-14 shows a comparison of features between AVR and PIC32CM.

Table 5-14. AVR® and PIC32CM RTC Features

Feature	AVR® RTC	PIC32CM RTC
Counter Resolution	16-bit	32-bit (or two 16-bit)
Prescaler	15-bit programmable	10-bit programmable

Table 5-14. AVR® and PIC32CM RTC Features (continued)

Feature	AVR® RTC	PIC32CM RTC
Compare Registers	One	1 (32-bit) or 2 (16-bit mode)
Period Register	One	One
Max Timeout (32.768 kHz)	>18 hours (1s resolution) 2s (max resolution)	>136 years (1s resolution) 36 hours (max resolution)
Clock/Calendar Mode	N/A	Supported (time/date, leap year correction)
Clock Correction	Crystal error correction	Digital frequency (prescaler) correction/ tuning
Periodic Interrupt Timer (PIT)	Separate PIT function, independent of RTC	Integrated periodic wake-up/event functionality
Interrupts/Events	Overflow, compare match, periodic, PIT	Overflow, alarm/compare, prescaler, periodic
Clear on Match	On overflow	Optional on alarm/compare match

5.4. Analog and Touch

5.4.1. Analog-to-Digital Converter (ADC)

AVR and PIC32CM MCUs provide Successive Approximation Register (SAR) ADC peripherals. ADCs convert analog signals to digital values, supporting a range of resolutions and input channels. All ADCs offer features suitable for applications requiring sensor interfacing, analog signal monitoring, and precise measurement.

Some PIC32CM device families also provide Sigma-Delta ADCs (SDADC), which are slower but offer higher resolution and better noise performance for precision applications. Table 5-15 shows a comparison of features between AVR and PIC32CM.

Table 5-15. AVR® and PIC32CM ADC Features

Feature	AVR® ADC	PIC32CM ADC
Variants	Standard ADC ADC with Programmable Gain Amplifier (PGA)	Standard ADC Sigma-Delta ADC
Resolution	10-bit, 12-bit	8, 10, or 12-bit SDADC: 16-bit
Max Sampling Rate	Up to 130 ksps (300 w/ PGA) or up to 2 Msps (XMEGA® only)	Up to 4.5 Msps SDADC: Up to 1.5 Msps
Input Channels	Up to 28, varies per device	Up to 32, varies per device
Differential Inputs	Supported	Supported
Programmable Gain	Supported in some device families 1x to 16x	1/2x to 16x
Reference Options	Internal/External	Internal/ External
Internal Inputs	Temperature sensor V_{REF} DAC	Varies per device family: Temperature sensor Band gap V_{REF} DAC Scaled supplies
Accumulation	Up to 128 (1024 with PGA)	Up to 1024
Window Comparator	Supported	Supported
Event Triggering	Supported	Supported

Table 5-15. AVR® and PIC32CM ADC Features (continued)

Feature	AVR® ADC	PIC32CM ADC
Conversion Modes	Single/Free-Running Burst Series (with PGA)	Single/Free-Running
Interrupts	Supported	Supported
Low-Power Operation	Supported	Supported

5.4.2. Digital-to-Analog Converter (DAC)

AVR and PIC32CM DAC peripherals generate analog voltages from digital values. The DAC features high drive capabilities, the ability to route DAC output to other internal analog peripherals (like comparators), and support for real-time analog signal generation. These DACs are suitable for applications such as waveform generation, audio output, and analog control.

Some PIC32CM device families have 12-bit DAC controllers with higher speed, dual-channel, and differential output options. Table 5-16 shows a comparison of features between AVR and PIC32CM.

Table 5-16. AVR® and PIC32CM DAC Features

Feature	AVR® DAC	PIC32CM DAC (10-bit/12-bit)
Resolution	10-bit 12-bit (XMEGA® only)	10-bit 12-bit (16-bit with dithering)
Max Conversion Rate	Up to 350 ksps or up to 1 Msps (XMEGA® only)	10-bit: Up to 350 ksps 12-bit: Up to 1 Msps
Number of Channels	One Two (XMEGA® only)	10-bit: One 12-bit: Two (independent) or one (differential mode)
Output Range	GND to selected V_{REF}	GND to selected V_{REF}
Trigger Sources	Enable Bit	Enable Bit Event-driven
Internal Routing	Output can be used by other analog blocks	Output can be used by other analog blocks

5.4.3. Analog Comparator (AC)

Analog Comparators (AC) compare the voltage levels of two inputs and provide a digital output based on the comparison. Both AVR and PIC32CM ACs offer features suitable for zero-cross detection, threshold monitoring, and windowed analog signal analysis. Table 5-17 shows a comparison of features between AVR and PIC32CM.

Table 5-17. AVR® and PIC32CM AC Features

Feature	AVR® AC	PIC32CM AC
Number of Comparators	One or more	2–4
Output on Pin	Supported	Supported
Input Selection	Four positive Three negative Internal DACREF	Four pins (positive/negative) Ground Band gap DAC AV_{DD} scaler (64-level)
Internal Reference	Internal reference voltage generator (DACREF)	Band gap, DAC, AV_{DD} scaler
Hysteresis	Selectable	Selectable
Output Inversion	Supported	Supported

Table 5-17. AVR® and PIC32CM AC Features (continued)

Feature	AVR® AC	PIC32CM AC
Interrupt Generation	Rising, falling, both edges Window function	Rising, falling, toggle, end of comparison Window function
Window Mode	Yes (window function, window interrupts/events)	Yes (window function, window interrupts/events)
Event Generation	Comparator output, window function	Comparator output, window function (in/out)
Response Time	Selectable	Selectable
Digital Filter	N/A	Supported

5.4.4. Voltage Reference (V_{REF})

AVR and PIC32CM MCUs provide programmable Voltage References (V_{REF}s) that supply stable reference voltages to analog peripherals such as ADCs, DACs and ACs. Both AVR and PIC32CM V_{REF}s are designed to enhance measurement precision, support low-power operation, and allow easy configuration of the reference source for different application needs. Table 5-18 shows a comparison of features between AVR and PIC32CM.

Table 5-18. AVR® and PIC32CM V_{REF} Features

Feature	AVR® V_{REF}	PIC32CM V_{REF}
Integration	Stand-alone V_{REF} peripheral	Integrated into the Supply Controller (SUPC) peripheral
Reference Voltage Options	Multiple internal voltage External V_{REF} pin	Multiple internal voltages External V_{REF} pin
Target Peripherals	ADC, DAC, Analog Comparator	ADC, DAC, Analog Comparator
Programmable Levels	Varies per device family: 0.55–4.096V	Varies per device family: 1–4.096V
Low-Power Support	Yes	Yes
Configuration	Software registers	Software registers
Additional Features	Simple routing to analog blocks	May include advanced supply management features

5.4.5. Operational Amplifiers (OPAMP)

Some AVR and PIC32CM MCUs have integrated Operational Amplifier (OPAMP) peripherals designed for flexible, low-power analog signal conditioning. The OPAMP peripherals provide up to three individually configurable opamps with rail-to-rail inputs, internal resistor ladders for feedback and gain control, and a wide range of selectable configurations. Both AVR and PIC32CM OPAMPs enable complex analog signal processing with minimal or no external components. Table 5-19 shows a comparison of features between AVR and PIC32CM.

Table 5-19. AVR® and PIC32CM OPAMP Features

Feature	AVR® OPAMP	PIC32CM OPAMP
Number of Op Amps	Up to three	Up to three
Input Selection	I/O pins, DAC, ground $V_{DD}/2$ Other OPAMPs Internal resistor ladder	I/O pins, DAC, ground Other OPAMPs Internal resistor ladder
Output Selection	I/O pins, ADC, AC Other OPAMPs	I/O pins, ADC, AC Other OPAMPs
Internal Resistor Ladder	With multiplexers for top/bottom/wiper	Configurable for internal feedback

Table 5-19. AVR® and PIC32CM OPAMP Features (continued)

Feature	AVR® OPAMP	PIC32CM OPAMP
Configurations	Stand-alone Unity gain buffer Inverting/non-inverting PGA Cascaded PGAs Instrumentation amplifier	Stand-alone Unity gain buffer Inverting/non-inverting PGA Cascaded PGAs Instrumentation amplifier Comparator with hysteresis
Calibration	Offset/gain calibration using ADC	Offset/gain measurement with ADC
Power Modes	Low-power, event-triggered operation	Four selectable modes (speed vs. power), on-demand start-up
Event System Integration	Includes dump mode for signal integration	On-demand start-up for ADC/AC
Special Features	Internal timer for a READY event Dump mode	Voltage doubler Programmable hysteresis

5.4.6. Peripheral Touch Controller (PTC)

The AVR and PIC32CM Peripheral Touch Controller (PTC) is used for capacitive touch sensing, supporting both self-capacitance and mutual-capacitance modes. It enables robust, low-power detection of touch buttons, sliders, wheels and 2D surfaces, with high sensitivity and environmental resilience. The AVR and PIC32CM PTCs are designed for easy integration, requiring only one pin per electrode and no external components, and are supported by development tools such as the QTouch® Configurator. Table 5-20 shows a comparison of features between AVR and PIC32CM.

Table 5-20. AVR® and PIC32CM Touch Features

Feature	AVR® PTC	PIC32CM PTC: Standard/ Enhanced
Self-Capacitance Buttons	Up to 46	Standard: Up to 16 Enhanced: Up to 36, mix-and-match
Mutual-Capacitance Buttons	Up to 529	Up to 256, mix-and-match with self-capacitance
Driven Shield+	Any X/Y line	Enhanced: Any X/Y line
Boost Mode	Supported	Supported
Window Monitor	Supported	Supported
Analog/Digital Accumulation	Supported	Supported
Polarity Control	Supported	Supported
Power Consumption	Low power Wake-up on touch	Low power Wake-up on touch
Channel Change Delay	Selectable	Selectable
Noise Immunity	Hardware filtering Desynchronization Driven Shield	Hardware filtering Desynchronization Enhanced: Driven Shield
Auto-Calibration	Supported	Supported
QTouch® Library	Supported	Supported

5.5. Logic

5.5.1. Configurable Custom Logic (CCL)

AVR and PIC32CM CCL peripherals act as programmable glue logic within the device. The CCL enables real-time, core-independent logic operations with custom combinatorial and sequential logic functions. Developers can add connections between internal peripherals, device pins, and

events, reducing or eliminating the need for external logic components. Table 5-21 shows a comparison of features between AVR and PIC32CM.

Table 5-21. AVR® and PIC32CM CCL Features

Feature	AVR® CCL	PIC32CM CCL
Number of Lookup Tables (LUTs)	Up to six	Up to four
LUT Inputs	Three per LUT	Three per LUT
Combinatorial Logic	Any logic expression up to three inputs Truth table	Any logic expression up to three inputs Truth table
Sequential Logic	Gated D flip-flop JK flip-flop D latch RS latch	Gated D flip-flop JK flip-flop D latch RS latch
Input Sources	I/Os, events Internal peripherals Subsequent LUTs	I/Os, events Internal peripherals Subsequent LUTs
Output Routing	I/O pins, Event System	I/O pins, Event System
Other Features	Synchronizer, filter, edge detector Interrupt on LUT output (rising, falling, both edges)	Synchronizer, filter, edge detector
Interrupt Generation	Supported	Supported

5.6. Connectivity/Communication

5.6.1. Universal Synchronous and Asynchronous Serial Receiver and Transmitter (USART)

AVR and PIC32CM MCUs provide highly flexible and robust USART peripherals. Both USART variants support buffered communication, interrupt-driven operation, and operation in low-power modes, making them suitable for a variety of embedded communication needs. Table 5-22 shows a comparison of features between AVR and PIC32CM.

Table 5-22. AVR® and PIC32CM UART Features

Feature	AVR® USART	PIC32CM USART (SERCOM)
Integration	Stand-alone USART peripheral	Mode in SERCOM (Serial Communications) peripheral
Operation Modes	Full-duplex Half-duplex synchronous/asynchronous	Full-duplex Half-duplex Synchronous/asynchronous
Protocol/ Standard Support	IrDA® LIN client SPI host RS-485	Varies per device family: IrDA® LIN host/client ISO 7816 (smart card) RS-485
Buffer/FIFO	Two-level transmit/receive buffer	16-byte transmit/receive FIFO
Baud Rate Generation	Fractional, from any peripheral clock	Baud rate generator, internal/external clock
Data Bits	5, 6, 7, 8, 9	5, 6, 7, 8, 9
Stop Bits	One or two	One or two
Parity	Odd, even, none	Odd, even, none
Data Order	N/A	LSb or MSb first selectable
Flow Control	N/A	Request-to-Send ($\overline{RTS}$) and Clear-to-Send ($\overline{CTS}$) hardware flow control

Table 5-22. AVR® and PIC32CM UART Features (continued)

Feature	AVR® USART	PIC32CM USART (SERCOM)
DMA Support	Supported (XMEGA® only)	Supported
Error Detection	Parity Buffer overflow Frame error Noise filtering	Parity Buffer overflow Frame error Noise filtering Collision detection
Sleep Mode Operation	Supported	Supported
Interrupts	Separate for TX complete, TX data empty, RX complete	Multiple, including FIFO events
Pin Mapping	Fixed	Flexible

5.6.2. Serial Peripheral Interface (SPI)

AVR and PIC32CM MCUs provide SPI peripherals that feature high-speed, full-duplex, synchronous data transfer between microcontrollers and other devices. Both SPIs support Host and Client modes, allowing communication with a wide range of external devices. Table 5-23 shows a comparison of features between AVR and PIC32CM.

Table 5-23. AVR® and PIC32CM SPI Features

Feature	AVR® SPI	PIC32CM SPI (SERCOM)
Integration	Stand-alone SPI peripheral	Mode in SERCOM (Serial Communications) peripheral
Data Buffering	TX/RX buffers, Shift registers	One-level TX buffer, Two-level RX buffer, internal FIFO
Host/Client Support	Supported	Supported
Bit Rate/Clock Speed	Up to 20 MHz	Up to 24 MHz
SPI Modes Supported	All four SPI modes	All four SPI modes
Data Order	LSb or MSb first	LSb or MSb first
DMA Support	Client mode (XMEGA® only)	Supported
Framed SPI/FSYNC	N/A	Hardware-controlled FSYNC
0-bit Extension	N/A	Supported
Wake-up from Idle	Supported	Supported
Write Collision Protection	Supported	Supported
Interrupts	Supported	Supported
Pin Mapping	Fixed	Flexible

5.6.3. Inter-Integrated Circuit (I^2C)

AVR Two-Wire Interface (TWI) and PIC32CM Serial Communication (SERCOM) provide I^2C-compatible peripherals for flexible inter-device communication on a shared bus. Both peripherals support Host and Client modes, standard and fast I^2C speeds (up to 1 MHz), multi-host arbitration, and bus error detection. Table 5-24 shows a comparison of features between AVR and PIC32CM.

Table 5-24. AVR® and PIC32CM I^2C Features

Feature/Variant	AVR® TWI	PIC32CM I2C (SERCOM)
Integration	Stand-alone TWI peripheral	Mode in SERCOM (Serial Communications) peripheral
Host/Client Modes	Supported	Supported

Table 5-24. AVR® and PIC32CM I²C Features (continued)

Feature/Variant	AVR® TWI	PIC32CM I2C (SERCOM)
Addressing	7-bit 10-bit (XMEGA® only) General call Address masking Dual address match	7-bit, 10-bit General call Address masking Dual address match
Data Buffering	Shift register	16-byte internal FIFO
DMA Support	N/A	Supported
Bus Speeds Supported	100 kHz, 400 kHz, 1 MHz	100 kHz, 400 kHz, 1 MHz
SMBus/PMBus™ Support	SMBus 2.0 compatible	SMBus and PMBus compatible
Multi-Host Arbitration	Supported	Supported
Wake-up from Sleep	Supported (on address match)	Supported (on address match)
Input Filtering	Noise suppression	Filtered inputs, slew-rate limited outputs
4-Wire Operation	No	Yes (for advanced protocols)
Data Extension	No	32-bit data extension
FIFO	No	16-byte FIFO

5.6.4. Universal Serial Bus (USB)

Some AVR and PIC32CM MCUs offer integrated USB peripherals that comply with the USB 2.0 Full-Speed (12 Mbps) device standards, supporting all four endpoint transfer types, endpoint buffering, and low-power operation with suspend/resume features, with minimal CPU intervention and interrupt load. Table 5-25 shows a comparison of features between AVR and PIC32CM.

Table 5-25. AVR® and PIC32CM USB Features

Feature	AVR® USB	PIC32CM USB
USB Specification	USB 2.0 Full-Speed (12 Mbps)	USB 2.1 (Full-Speed 12 Mbps, Low-Speed 1.5 Mbps)
Host/Device Modes	Device only	Host and Device mode
Host Mode Features	N/A	Eight physical pipes Unlimited virtual pipes Feedback endpoint SOF clock output
Endpoint Addresses	16 (32 endpoints: 16 IN, 16 OUT)	8 (16 endpoints: 8 IN, 8 OUT)
Endpoint Transfer Types	Control Interrupt Bulk Isochronous	Control Interrupt Bulk Isochronous
Max Payload per Endpoint	Up to 1023 bytes	No endpoint size limitation (typically up to 1023 bytes)
Endpoint Buffering	Internal SRAM, fully configurable	Internal SRAM, fully configurable
Multi-Packet Transfer	Supported	Supported
Double Buffering	N/A	Dual bank (ping-pong) for all endpoints
Power Management	Suspend/Resume Wake-up from sleep	Suspend/Resume Wake-up from sleep Link Power Management (LPM-L1) support
On-Chip Transceiver	Supported	Supported
DMA Support	Built-in for endpoint data and configuration	Built-in for endpoint data and configuration

Table 5-25. AVR® and PIC32CM USB Features (continued)

Feature	AVR® USB	PIC32CM USB
Crystal-less Operation	N/A	Supported
Debug Support	On-chip debug during USB transactions	N/A

5.6.5. Controller Area Network (CAN)

Some AVR and PIC32CM MCUs offer integrated CAN controllers, enabling connection to CAN networks for automotive and industrial applications. These CAN peripherals support standard CAN 2.0 A/B protocols, provide message filtering, and include error handling and interrupt capabilities. They are designed to offload CAN protocol management from the CPU, improving real-time communication performance. Both variants require an external CAN transceiver for physical layer connectivity. Table 5-26 shows a comparison of features between AVR and PIC32CM.

Table 5-26. AVR® and PIC32CM CAN Features

Feature	AVR® CAN	PIC32CM CAN
Protocol Support	CAN 2.0A/B	CAN 2.0A/B CAN-FD (ISO 11898-1:2015)
Max Data Rate	1 Mbps	1 Mbps (CAN 2.0) 10 Mbps (CAN-FD)
Message Objects/Buffers	Up to 15 MOb (Message Objects)	Up to 64 RX and 32 TX dedicated buffers Two RX FIFOs
Data Buffer Size	Eight bytes per Message Object (static allocation)	Up to 64 bytes per CAN-FD frame
Acceptance Filtering	15 identifier tags/masks (11/29 bits)	Up to 128 configurable filter elements J1939 filter
Time Stamping	Time Trigger Communication (TTC) Timer	CiA® 603 hardware timestamping External Timestamping Unit (TSU)
Loopback/Test Mode	Listening mode	Programmable loopback test mode
Error Handling	Standard CAN error handling	CAN error logging Advanced error signaling
AUTOSAR/J1939 Optimization	N/A	Supported
Number of CAN Channels	One	Up to two
Power-Down/Debug Support	N/A	Power-down Debug on CAN support
Host Memory Access	Internal static allocation	Uses system RAM via AHB (flexible allocation)

5.7. Safety

Many of the newer AVR and PIC32CM device families feature additional autonomous hardware fault detection mechanisms to detect, correct and report random hardware faults and transient faults. Table 5-27 shows a summary of available safety features for each MCU architecture.

Table 5-27. AVR® and PIC32CM Safety Peripherals and Features

Feature/Peripheral	AVR®	PIC32CM
Brown-Out Detector (BOD)	Supported	Supported
Watchdog Timer (WDT/WWDT)	Supported	Supported
CRC Engine	Supported	Supported
SRAM Memory Built-in Self-Test Module	N/A	Supported
ECC on Flash/RAM/EEPROM	Supported	Supported

Table 5-27. AVR® and PIC32CM Safety Peripherals and Features (continued)

Feature/Peripheral	AVR®	PIC32CM
Clock Failure Detection	Supported	Supported
Non-Maskable Interrupts	Supported	Supported
Safety Libraries/Manuals	Available	Available
Temperature Sensor	Supported	Supported

These sections cover common safety peripherals in AVR and PIC32CM devices.

5.7.1. Brown-out Detection/Reset (BOD/BOR)

AVR and PIC32CM MCUs provide Brown-out Detection (BOD) and Brown-out Reset (BOR) to monitor supply voltages and ensure reliable operation by resetting or interrupting the device if the voltage drops below a programmable threshold. BOD and BOR features help protect the device from erratic behavior due to insufficient supply voltage. Table 5-28 shows a comparison of features between AVR and PIC32CM.

Table 5-28. AVR® and PIC32CM BOD/BOR Features

Feature/Aspect	AVR® BOD	PIC32CM BOD/BOR (SUPC/SYSCTRL)
Integration	Stand-alone BOD	Varies per device family: Integrated into either the Supply Controller (SUPC) or the System Controller (SYSCTRL)
Supply Domains Monitored	Main supply	Varies per device family: Main (programmable) Core (internal, non-configurable) I/O, Analog, V_{REG}
Early Warning/Interrupt	Voltage Level Monitor (VLM) provides early warning interrupt	Interrupts on threshold crossing Wake from sleep
Threshold Configuration	Programmable (via fuses) VLM as % above BOD	Programmable (from NVM/User Row/CFG) per domain
Operating Modes	Continuous Sampled Disabled	Continuous Sampled (low power)
Reset Action	System reset on BOD event	System reset on BOR event I/O reset on VDDIO drop
Hysteresis	N/A	Programmable hysteresis (from calibration)
Sleep Mode Operation	Selectable modes for Active/Sleep	Monitoring in all modes, including Standby/Backup
Calibration	N/A	Threshold/hysteresis values from Flash calibration
Status Monitoring	VLM status, BOD status	Status via SUPC/SYSCTRL registers, INTFLAG, etc.
Output Pins	N/A	SUPC (for supply status indication)

5.7.2. Watchdog Timer (WDT)

AVR and PIC32CM MCUs feature a Watchdog Timer (WDT) to monitor program execution and recover from software faults such as runaway or deadlocked code. The WDT operates asynchronously from a dedicated, CPU-independent clock source, ensuring it functions even if the main clock fails. Both the AVR and PIC32CM WDTs support normal and window modes, in which the WDT must be cleared within a specific time window to avoid a system reset.

Some AVR devices also include a Synchronous WDT (SWDT) in addition to the standard WDT for safety-critical applications where software functionality and sequencing failures must be detected. Table 5-29 shows a comparison of features between AVR and PIC32CM.

Table 5-29. AVR® and PIC32CM WDT Features

Feature/Aspect	AVR® WDT/SWDT	PIC32CM WDT
Operation	Varies per device family: Asynchronous (WDT) Synchronous (SWDT)	Asynchronous
Clock Source	WDT: OSC32K, 1.024 kHz SWDT: CLK_PER	Dedicated internal oscillator
Window Mode	Supported	Supported
Time-out Periods	WDT: 8 ms to 8s SWDT: 24-bit value	Normal: 8 to 16,384 cycles Windowed: 16 to 32,768 cycles
Early Warning Interrupt	SWDT: Supported	Supported
Always-On Capability	Supported	Supported
Configuration (Change) Protection	Supported	Supported
Operation in Sleep Modes	Supported	Supported
Test/Functional Safety	SWDT: Supported	N/A
Reset Action	System reset on time-out or window violation	System reset on time-out or window violation
Interrupt on Timeout	SWDT: Supported	Supported (early warning)

5.7.3. Temperature Sensor

AVR and PIC32CM MCUs provide an integrated temperature sensor for on-chip temperature monitoring, typically accessible via the ADC. Calibration data is stored in nonvolatile memory to improve measurement accuracy and enable software compensation for precise readings. These sensors are used for thermal management, diagnostics and temperature compensation in embedded systems. Table 5-30 shows a comparison of features between AVR and PIC32CM.

Table 5-30. AVR® and PIC32CM Temp Sensor Features

Feature/Aspect	AVR® Temperature Sensor	PIC32CM Temperature Sensor
Integration	Integrated into ADC	Varies per device family: Integrated into ADC Stand-alone Temperature Sensor (TSENS) peripheral
Sensor Access	ADC input channel	ADC input channel Dedicated TSENS peripheral
Calibration Data Location	Signature/User Row	NVM
Typical Accuracy	±5°C (uncalibrated) Improved with calibration	ADC: ±2°C to ±5°C (uncalibrated) Improved with calibration TSENS: ±1°C to ±2°C (with calibration)
Temperature Range	-40°C to +125°C (typical)	-40°C to +125°C (typical)
Software Compensation	Required for high accuracy	ADC: Required for high accuracy TSENS: Minimal

5.7.4. Cyclic Redundancy Check (CRC)

AVR and PIC32CM MCUs provide hardware support for Cyclic Redundancy Check (CRC) to ensure data and code integrity, which is crucial for safety and reliability in embedded systems. Both MCUs can perform CRC checks over memory regions (such as Flash or SRAM), support standard CRC polynomials, and can trigger system responses (such as interrupts or resets) if a CRC mismatch is

detected. These features help prevent the execution of corrupted code and enable robust error detection in data transfers or memory checks. Table 5-31 shows a comparison of features between AVR and PIC32CM.

Table 5-31. AVR® and PIC32CM CRC Features

Feature/Aspect	AVR® CRC (CRCSCAN)	PIC32CM CRC (DSU, DMAC)
Integration	Stand-alone CRCSCAN peripheral	Integrated into Device Service Unit (DSU) and Direct Memory Access Controller (DMAC)
User Configuration	Selectable region and polynomial	Selectable region and polynomial (DMAC)
Supported Polynomials	CRC-16-CCITT CRC-32 (IEEE® 802.3)	DSU: CRC-32 DMAC: CRC-16-CCITT & CRC-32
Memory Regions	Flash (entire, boot, or application section)	Any memory accessible via bus matrix (Flash, SRAM)
Trigger/Control	Can run at reset, initialize, or by software	Command issued via DSU or DMAC
Status/Result	Supported	Supported
Interrupts	NMI on CRC failure	No direct NMI, but can trigger interrupts/events
DMA Integration	N/A	DMAC can perform CRC on transferred data
Software Fallback	Supported	Supported

5.8. Security

AVR and PIC32CM MCUs offer different levels of hardware security. Table 5-32 shows a summary of available security features for each MCU architecture.

Table 5-32. AVR® and PIC32CM Security Features

Features/Peripherals	AVR®	PIC32CM M0+	PIC32CM M23
Security Type	Basic	Essential	Advanced
Lock Bits	Supported	Supported	Supported
Hardware Secure Boot	Software-based only	Software-based only	Supported
CRC	Supported	Supported	Supported
Write Access Control	For system-critical registers, only via FUSE and LOCK bits and Configuration Change Protection (CCP) Program and Debug Interface Disable (PDID)	All peripherals Integrated in Peripheral Access Controller (PAC) Program and Debug Interface Disable (PDID)	All peripherals Integrated in Peripheral Access Controller (PAC) Program and Debug Interface Disable (PDID)
Debug Support	Unified Program and Debug Interface (UPDI)	Integrated in Device Service Unit (DSU)	Integrated in Device Service Unit (DSU)
TrustZone®	N/A	N/A	Supported
Hardware Crypto	N/A	N/A	Cryptographic Accelerators True Random Number Generator (TRNG) Device Identity Composition Engine (DICE) support Physical Unclonable Function (PUF)
Tamper Detection	N/A	N/A	Supported

Refer to the PIC32CM Additional Features and Peripherals section for more details.

6. PIC32CM Additional Features and Peripherals

These sections cover peripherals and features provided by most PIC32CM devices, but they are not present in AVR.

6.1. Timing, Measurement and Waveforms

6.1.1. Position Decoder (PDEC)

Some PIC32CM device families include a Position Decoder (PDEC) peripheral for precise position and speed measurement in motor control and motion applications. The PDEC integrates a quadrature/ Hall decoder and a counter with two compare channels, supporting multiple modes of operation.

Key Features:

- Internal prescaler—Allows input signal frequency scaling for optimal performance and resolution
- Selectable modes—Quadrature Decoder (QDEC), Hall Sensor Decoder (Hall), Counter
- Event and interrupt generation

6.1.2. Frequency Measurement (FREQM)

Some PIC32CM device families include a Frequency Meter (FREQM) peripheral to measure the frequency of an external or internal clock signal with high accuracy and minimal CPU intervention, aiding in system calibration and ensuring timing accuracy. It is used for clock calibration, timekeeping, communication, instrumentation, and power metering.

Key Features

- Direct frequency measurement
- Selectable reference clock
- High resolution—Provides precise frequency readings, suitable for calibration, diagnostics, and clock monitoring
- Interrupt and event generation
- Low-power operation

6.2. Math Accelerator

6.2.1. Divide and Square Root Accelerator (DIVAS)

The Divide and Square Root Accelerator (DIVAS) peripheral is a math accelerator that performs fast division and square root operations, which are typically slow when implemented in software on microcontrollers without hardware divided instructions. This peripheral is used for applications requiring real-time control, signal processing, and mathematical computations.

Key Features

- Hardware division—32-bit signed and unsigned integer division operations
- Square root calculation
- Low latency/No CPU stalling—Provides results in just a few clock cycles, freeing the CPU for other tasks
- Simple interface—Operands are written to and results are read from memory-mapped registers

6.3. Safety and Security

6.3.1. Peripheral Access Controller (PAC)

Enter a short description of your topic here (optional).

The Peripheral Access Controller (PAC) is a hardware security and safety feature that controls access to critical peripherals and reports access errors. It helps prevent unintended or unauthorized operations that could compromise an application's system integrity or safety.

Key Features:

- Access control/protection—Read, write, and execute permissions for each peripheral
- Violation reports—Interrupt generation for unauthorized access
- Locking mechanism—Locked peripherals cannot be changed until the next device reset
- Integration with security domains—Enables separation of secure and non-secure access to peripherals

6.3.2. Device Service Unit (DSU)

The Device Service Unit (DSU) is a dedicated hardware peripheral that provides advanced device-level debug, test, programming support, and memory integrity operations. It is used in development, manufacturing, and maintenance scenarios that require direct access to the microcontroller's memory and debugging features, such as diagnostic tests for functional safety standards.

Key Features:

- Programming and debug support—Allows external tools to access and manipulate device memory without CPU intervention
- Arm® CoreSight™-compliant device identification
- Access to device signature and identification registers
- Memory region operations—Enable firmware updates and diagnostics
- On-board Memory Built-In Self-Test (MBIST)
- Hardware CRC calculation—Adds integrity checks, essential for bootloader validation and safety applications
- Breakpoint and watchpoint support for advanced debugging
- Error and status reporting

6.3.3. Integrity Check Monitor (ICM)

Some PIC32CM device families include an Integrity Check Monitor (ICM) to ensure memory integrity using cryptographic hash functions. The ICM acts as a DMA controller that autonomously performs hash calculations over multiple memory regions, using transfer descriptors stored in memory (the ICM Descriptor Area). This is primarily used in applications that require high reliability, safety and security.

Key Features:

- DMA AHB host interface
- Multi-region monitoring—Up to four non-contiguous memory regions simultaneously
- Linked list support—Allows flexible block gathering and management of memory regions via linked list descriptors
- Two modes of operation:
  - Hash mode—Calculates hashes for a list of memory regions and stores the digests in memory (ICM Hash Area)

  - Active Monitoring mode—Continuously hashes memory regions and compares the result to a stored digest; if a mismatch is detected, an interrupt is raised
- Hash algorithms
  - Supports SHA1, SHA224 and SHA256
  - Compliant with FIPS Publication 180-2
- Configurable processing period
- Programmable bus burden

6.3.4. Hardware Security

PIC32CM MCUs offer a wide range of hardware security peripherals and features, such as:

- Peripheral Access Controller (PAC)
  - Controls write access to peripheral registers to protect critical system resources
- Memory Protection Unit (MPU)
  - Defines memory regions with access permissions
- Device Service Unit (DSU)
  - Provides device identification and in-system programming support
  - Supports firmware updates and device authentication
- Lock bits and security fuses
  - Disable debug/programming interfaces or lock memory regions, preventing unauthorized access to code and data

Other PIC32CM device families with Arm Cortex-M23 offer additional security features, either as stand-alone peripheral/features or integrated into a Hardware Security Module Lite (HSM Lite) peripheral:

- Arm® TrustZone® support
  - Enables trusted execution environments, IP protection, and secure partitioning of resources
- TrustRAM (TRAM)
  - Controls volatile secret data
- Cryptographic Accelerators (AES, SHA, GCM, etc.)
  - Enable secure data encryption and authentication
- True Random Number Generator (TRNG)
  - Generates high-quality random numbers for cryptographically secure keys
- Secure boot
  - Verifies firmware integrity and authenticity before execution
- Tamper detection (Anti-Tamper)
  - Detects physical tampering attempts and triggers protective actions for sensitive data
- Device Identity Composition Engine (DICE)
  - Derives cryptographically strong device identities and keys
- Physical Unclonable Function (PUF)
  - Generates unique and unclonable fingerprints

Refer to each PIC32CM device family data sheet for more information on the supported security features.

7. Getting Started With PIC32CM

Getting started with PIC32CM MCUs includes obtaining product information, setting up a development environment, and exploring example projects.

7.1. PIC32CM Products and Collaterals

Microchip's official website has product pages for each MCU family. The PIC32CM families are included on the Arm® Cortex®-based MCUs product page. This page includes links to specific family product pages and reference guides for all 32-bit MCU products.

- Product Pages: Arm® Cortex®-Based Microcontrollers
- Collaterals and solutions:
  - Cortex M0+ Family of Devices
  - Cortex M23 Family of Devices

Each family also provides starter evaluation kits, such as the Curiosity Nano boards. These boards come with basic hardware interfaces and onboard programmers and debuggers. For more information on Curiosity Nano development, refer to the main website.

- Curiosity Nano Development Platform

7.2. Development Tools

The MPLAB Development Ecosystem supports customers through all stages of embedded design, from discovery to production. It incorporates compilers, simulation tools, libraries, code examples, and other resources. The MPLAB toolchain is available in Visual Studio Code (VS Code) through the MPLAB Tools for VS Code, enabling code development in its expansive ecosystem. The following tools integrated within MPLAB are required for PIC32CM development:

- MPLAB XC32 Compiler for C/C++ development on 32-bit MCUs
- MPLAB Code Configurator (MCC) and MPLAB Harmony for generating firmware, including peripheral drivers, middleware, and example applications

AVR uses a different variation of MPLAB tools for code development. For a more detailed guide on tool migration, refer to the Migration Guide from AVR and PIC16/18F to PIC32CM Development Tools Ecosystem.

7.3. Example Projects

The MPLAB® Discover and Microchip GitHub offer a vast selection of example code and applications. Browse these websites to find examples for PIC32CM MCUs.

- MPLAB Discover References: PIC32CM Code Examples
- MPLAB Harmony GitHub: PIC32CM Reference Applications

8. Conclusion

Microchip's PIC32CM device families present an effective alternative to traditional 8-bit MCUs for applications that require the higher performance and efficiency of the Arm® Cortex®-M0+ and M23 cores. However, architectural differences in the core that translate into development may pose some challenges during migration. This document supports a smooth and informed migration by providing a walk-through of the similarities and differences between the device families. Microchip offers a wide variety of references to help users get started with PIC32CM development.

9. Revision History

Revision	Date	Description
C	04/2026	Corrections on tables 2-1 and 3-1
B	04/2026	Added the PIC32CM PL10 device family
A	11/2025	Initial document release

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: 979-8-3371-3201-3

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.