
采用AVR128DA48 Curiosity Nano的差分ADC

准备工作

本部分旨在列出完成本培训需满足的所有要求。

硬件准备工作

- AVR128DA48 Curiosity Nano ([DM164151](#))
- 适用于 Click boards™ 的 Curiosity Nano 基板 ([AC164162](#))
- POT Click 板 ([MIKROE-3402](#))
- POT 2 Click 板 ([MIKROE-3325](#))

注：两块 POT Click 板可替换为任何其他两个电位器，相关原理图见下文。

软件准备工作

下面给出了本培训的软件准备工作。

- MPLAB[®] X IDE v5.40 或更高版本
- MPLAB XC8 编译器 v2.20 或更高版本
- AVR-Dx_DFP (器件系列包) v1.1.40
- MPLAB 数据可视化器 (MPLAB Data Visualizer, MDV) v1.1
- MPLAB 代码配置器 (MPLAB Code Configurator, MCC) v3.95
- MCC 8 位 AVR[®] MCU 库 v2.3.0

文档材料

- [AVR128DA28/32/48/64数据手册](#)
- [TB3245 - 《使用 12 位 ADC 进行转换、累加和触发事件》](#)
- 其他器件相关文档可访问以下位置获取：[AVR128DA48 器件概述](#)

简介

鉴于许多外部激励都是模拟类型的激励，因此嵌入式应用经常依赖于模拟输入（由模拟传感器提供）。大多数 Microchip 单片机（MCU）都集成了模数转换器（Analog-to-Digital Converter, ADC）来获取模拟数据，并且能够对此类数据进行处理。

本文档介绍了如何借助 AVR128DA48 Curiosity Nano 评估工具包来使用 ADC 的差分转换功能开发应用程序。其中提供了外设的概述，并逐步说明了如何将 ADC 配置为差分模式。

完成本培训后，用户将能够：

- 通过配置所有外设初始化系统
- 启动 ADC 转换
- 连续读取 ADC 结果并通过通用同步/异步收发器（Universal Synchronous/Asynchronous Receiver/Transmitter, USART）进行发送
- 分步运行应用程序以理解具体配置
- 使用硬件设置来测试开发的应用程序
- 使用图形界面工具查看数据

首先提供 ADC 模块的概述，然后介绍要开发的应用程序，最后提供实现该应用程序所需的步骤：熟悉软件工具、初始化系统和外设模块、实现其他需要的功能、调试应用程序以及使用 MPLAB 数据可视化器查看收到的数据。

目录

准备工作.....	1
简介.....	2
1. 概述.....	4
1.1. 模数转换器模块概述.....	4
1.2. 应用程序概述.....	4
2. 模数转换器培训.....	6
2.1. 图标标识符.....	6
2.2. 硬件设置.....	6
2.3. 熟悉软件环境.....	7
2.4. 任务：差分 ADC.....	14
3. 结论.....	27
4. 参考资料.....	28
5. 版本历史.....	29
Microchip 网站.....	30
产品变更通知服务.....	30
客户支持.....	30
Microchip 器件代码保护功能.....	30
法律声明.....	30
商标.....	31
质量管理体系.....	31
全球销售及服务网点.....	32

1. 概述

本章提供 ADC 模块的概述以及本培训的内容。

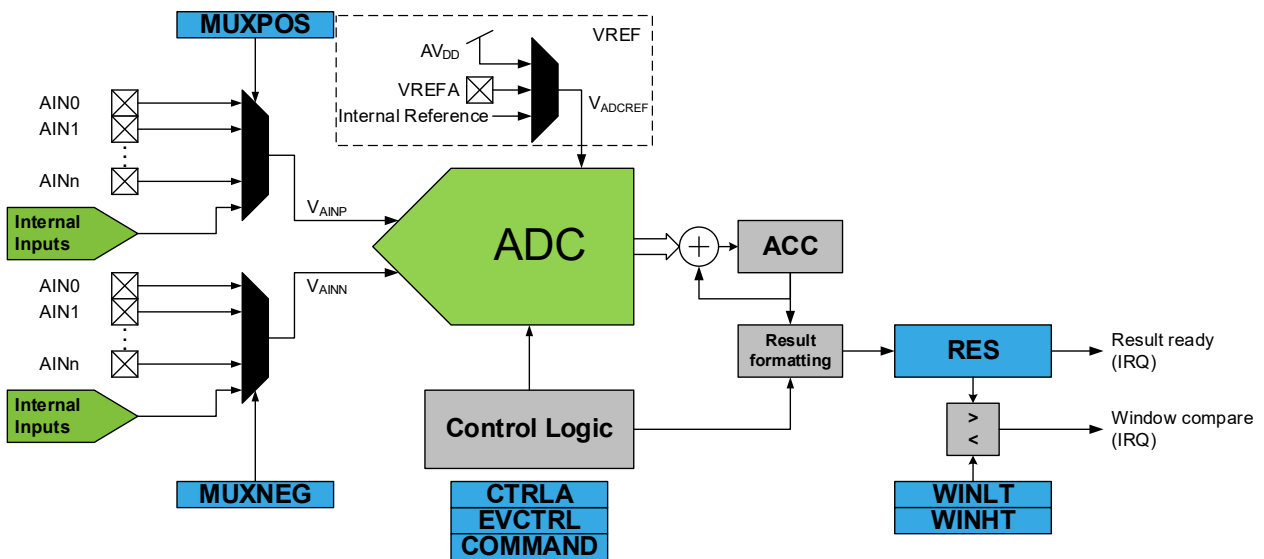
1.1 模数转换器模块概述

本文档所述的应用程序使用 AVR128DA48 器件进行开发。该器件配有 12 位分辨率的 ADC 模块，支持单端和差分两种模式。

ADC 输入信号通过一个采样保持电路馈送，可确保 ADC 的输入电压在采样期间保持在恒定水平。ADC 参考电压在 VREF 外设中配置。

下面给出了 ADC 模块的框图。

图 1-1. ADC 框图



模块必须使用相应的模块寄存器进行配置。数据手册的 *寄存器汇总* 和 *寄存器说明* 两章列出了模块的所有寄存器，并介绍了模块寄存器的所有位和位域的功能。

在本应用程序示例中，ADC 将配置为差分模式。差分 ADC 用于测量两个输入之间的电压差。这在某些应用中可能是至关重要的，因为一些测量概念需要使用两个输出信号（而非一个）来量化所关注的物理特性。实现这种概念的传感器通常以两个信号之间的电压差提供其输出值，也称为差分信号。其他传感器可能会为了增加稳健性而提供差分输出，即使测量本身生成的是单端信号也如此。

将差分模拟传感器连接到 MCU 时，差分对中的一个信号被定义为正输入，另一个信号被定义为负输入。差分信号的值是以负输入为参考的正输入电压。每个信号的正负标识决定了差分信号的极性，当正输入大于负输入时，定义为正；当负输入大于正输入时，定义为负。转换结果的计算公式如下：

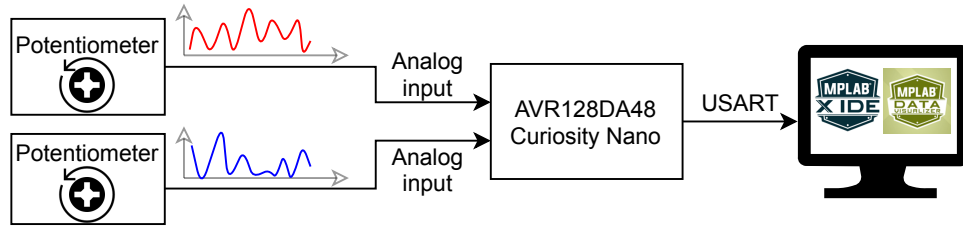
$$ADC_{result} = \frac{V_{AINP} - V_{AINN}}{V_{REF}} \times 2048 \in [-2048 \text{ 和 } 2047]$$

其中， V_{AINP} 和 V_{AINN} 是 ADC 的正负输入，而 V_{REF} 是选定的 ADC 参考电压。差分转换的数据格式是带符号扩展的二进制补码。

1.2 应用程序概述

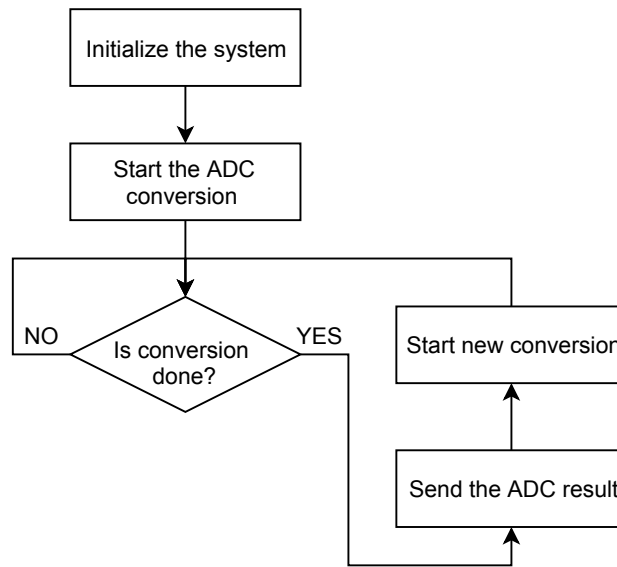
图 1-2 给出了开发和测试本应用程序所需的硬件设置。在 Curiosity Nano 开发板上，使用两个电位器为 AVR128DA48 器件提供模拟输入信号。ADC 将对这两个输入信号之间的电压差进行转换。转换结果将通过 USART 进行发送，并使用 MPLAB 数据可视化器插件进行绘制。

图 1-2. 硬件设置



本培训设计的软件应用程序将实现以下软件流程图。初始化系统后，将启动自由运行的 ADC 转换。之后，将利用无限循环连续读取结果状态并通过 USART 进行发送，最终使用图形界面进行显示。

图 1-3. 软件图



2. 模数转换器培训

2.1 图标标识符

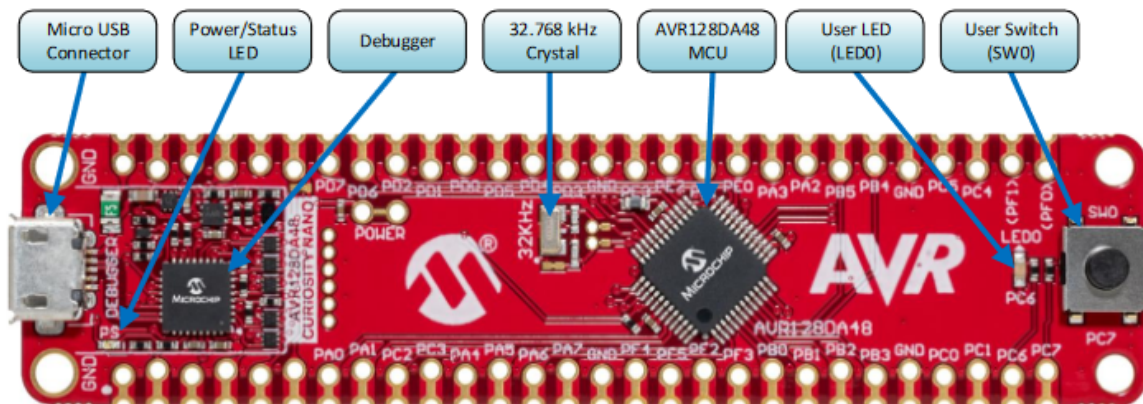
本小节提供了用于指导用户完成培训的图标及其含义，具体如下所示。

	信息:	该图标将用于强调有用的信息。
	任务:	该图标将用于指出有一项用户必须完成的任务：配置模块设置或实现代码。
	结果:	该图标将用于标记任务的结果。

2.2 硬件设置

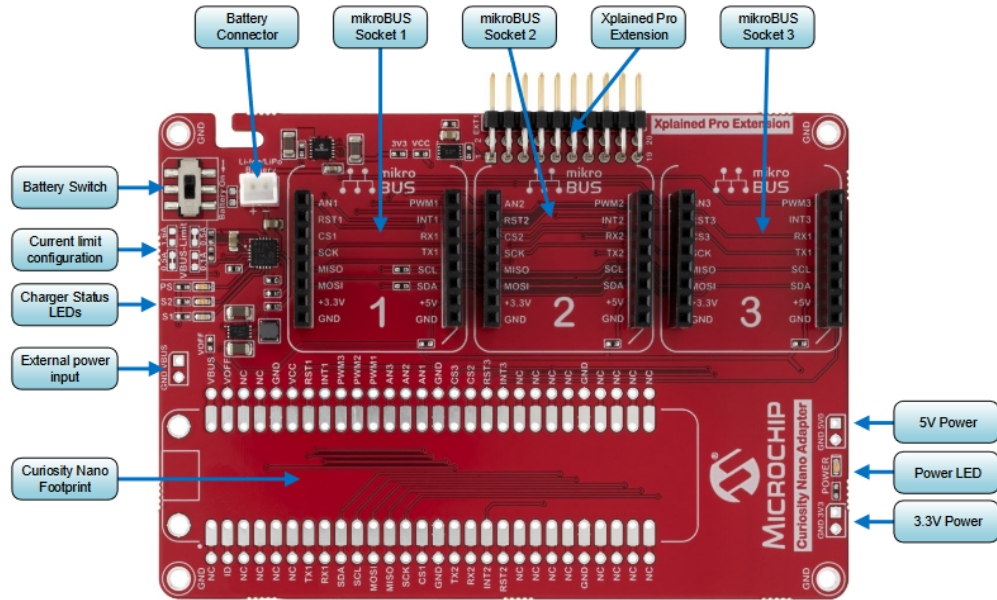
用于开发所提供的培训材料的硬件包含在用于 AVR128DA48 器件的 Curiosity Nano 开发板中，如图 2-1 所示。

图 2-1. Curiosity Nano 板



为了方便集成其他对该应用程序有用的组件（如 POT Click 板和 POT 2 Click 板），可以使用 Curiosity Nano 适配器板（如图 2-2 所示）。

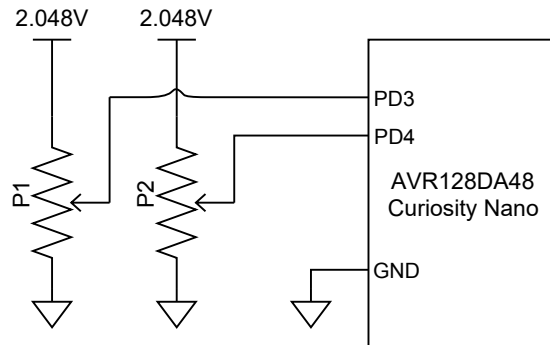
图 2-2. Curiosity Nano 适配器板



注：演示原型使用两个简单的电位器来构建。为了方便集成这两个电位器，可以使用 MIKROE POT Click 板和 POT 2 Click 板，以及 Curiosity Nano 适配器板。

电位器与 Curiosity Nano 适配器板之间的连接如图 2-3 所示。

图 2-3. 电位器连接原理图



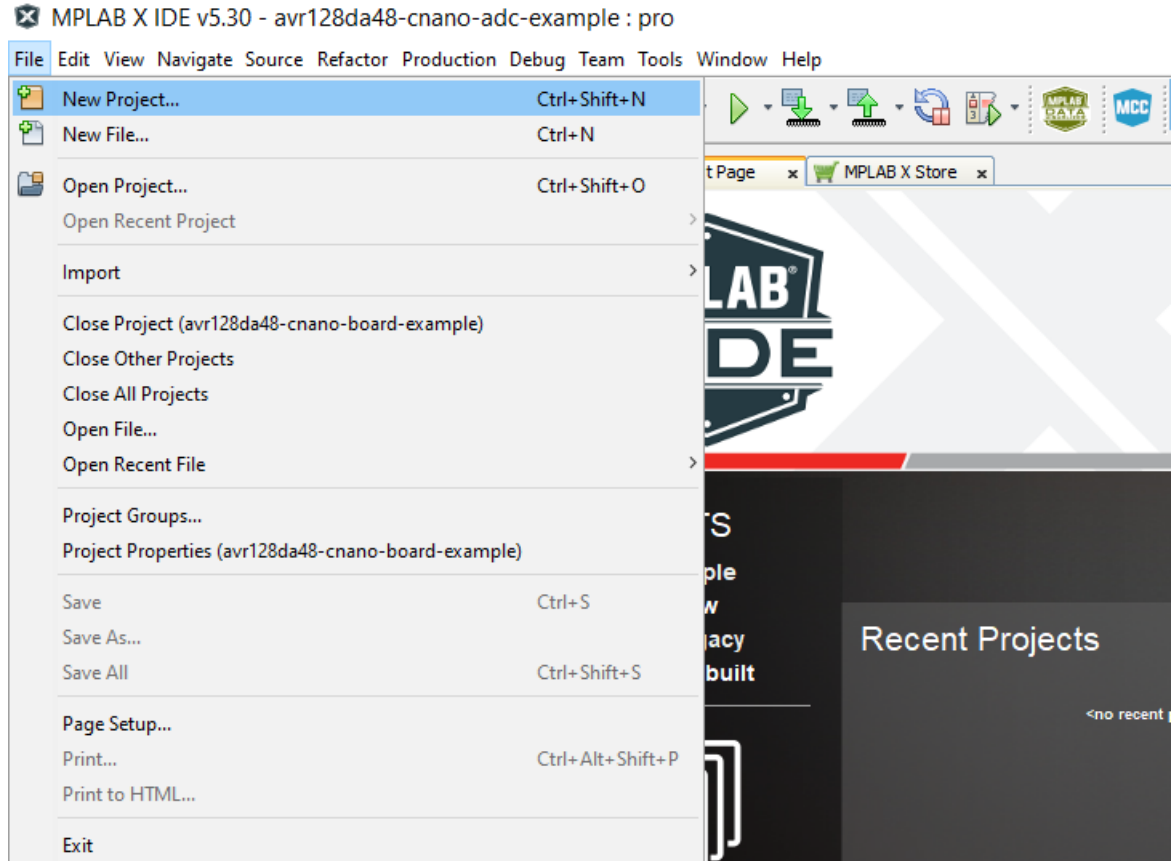
2.3 熟悉软件环境

本节帮助用户熟悉如何创建新项目，并介绍必要的插件及其安装步骤。

2.3.1 使用 MPLAB® X IDE 创建新项目

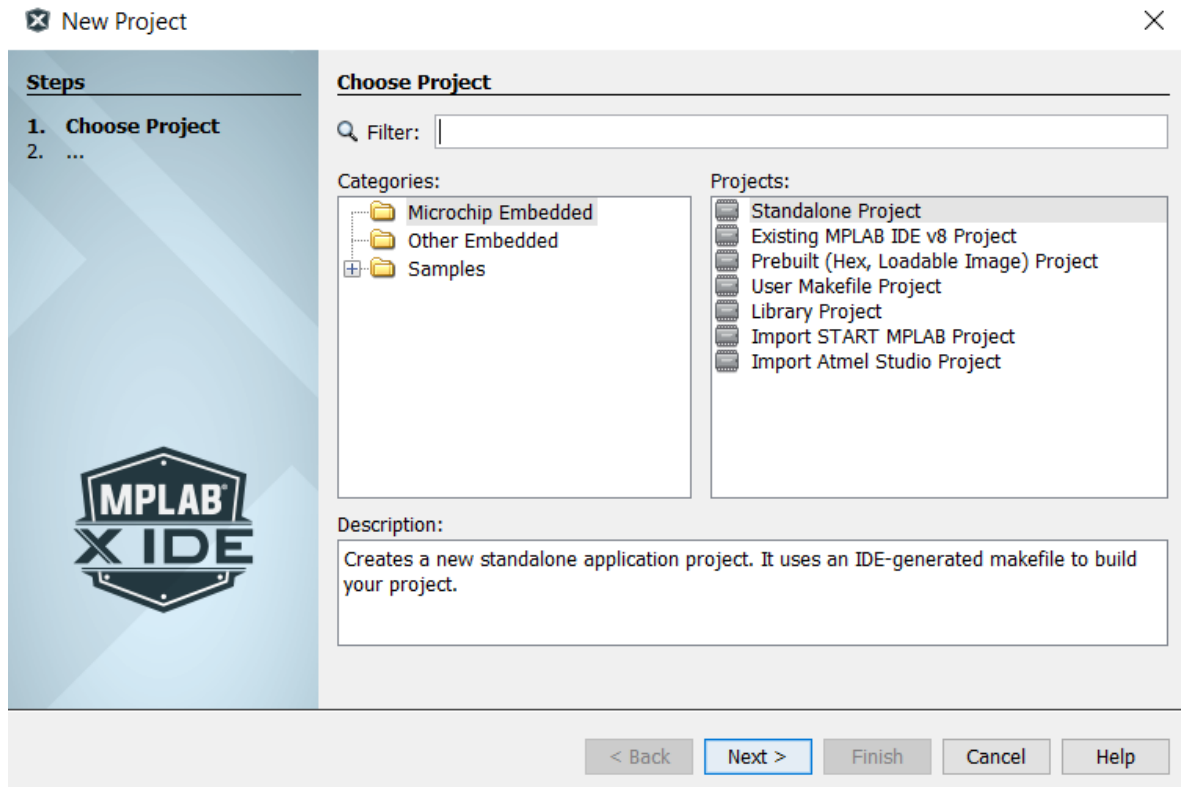
1. 要使用 MPLAB X IDE 创建新项目，请转到 **File** → **New Project...**（文件 → 新建项目...），随后将显示 New Project（新建项目）向导。该步骤如图 2-4 所示。

图 2-4. 创建新项目



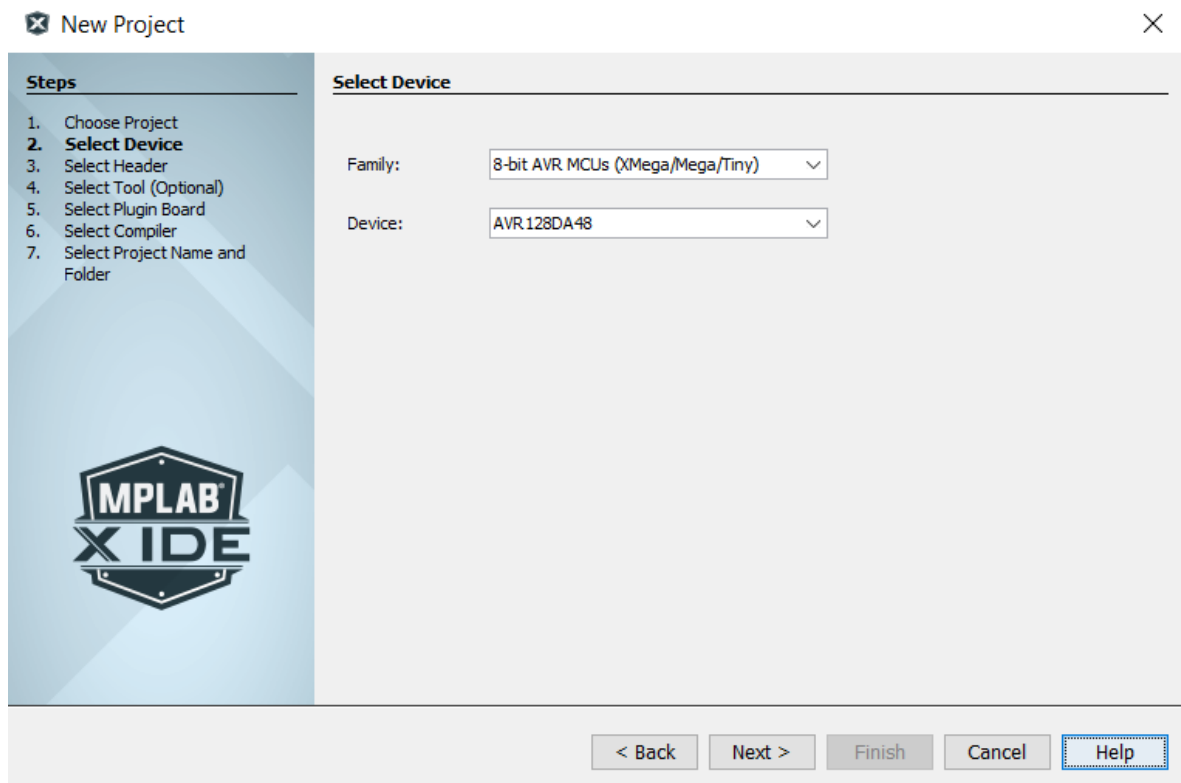
2. 在 Choose Project（选择项目）步骤中，从 Categories（类别）中选择 **Microchip Embedded**（Microchip 已安装工具）并从 Projects（项目）中选择 **Standalone Project**（独立项目），然后单击 **Next**（下一步）。随后将创建一个新的独立应用程序项目。该步骤如图 2-5 所示。

图 2-5. 选择项目



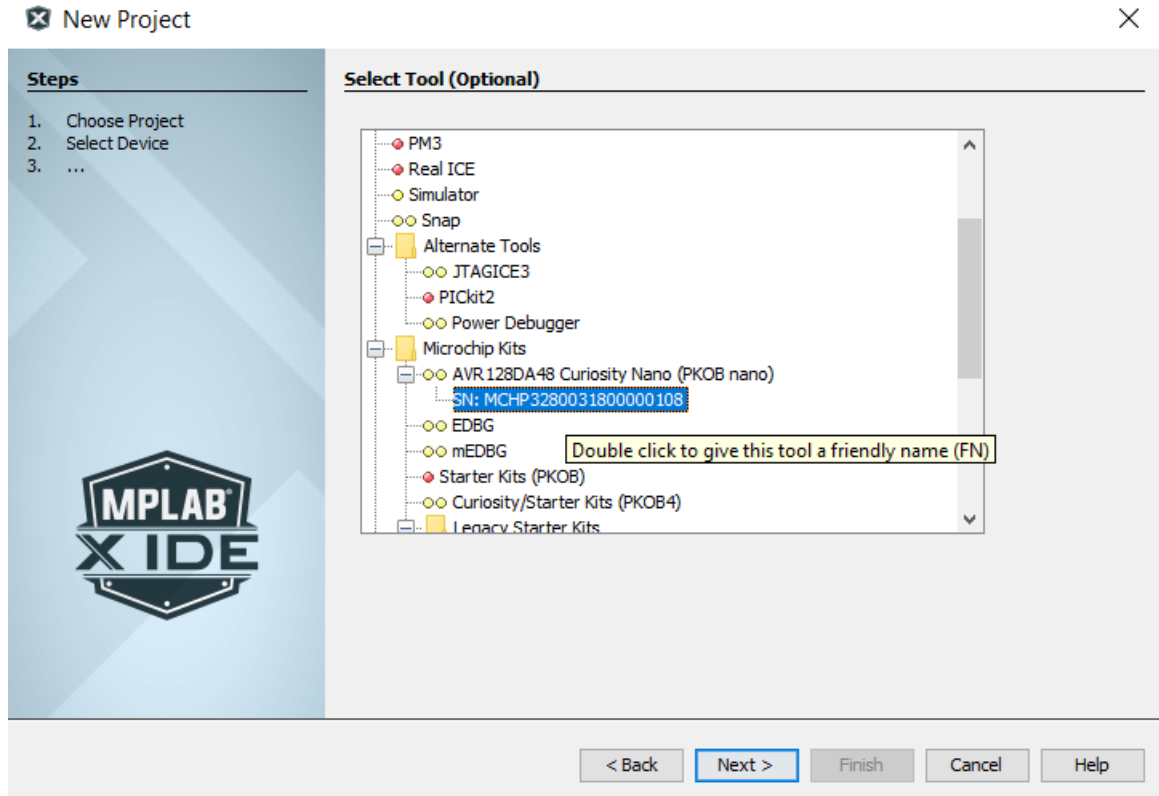
3. 下一步是选择要进行编程的器件。本文档所述的应用程序是使用 8 位 AVR MCU 系列的 AVR128DA48 器件实现。选择该器件后，单击 **Next**。

图 2-6. Select Device



4. 之后，在 **Select Tool**（选择工具）步骤中，从 **Microchip Kits** → **AVR128DA48 Curiosity Nano (PKOB nano)**（Microchip 工具包 → AVR128DA48 Curiosity Nano (PKOB nano)）中选择所需的工具（SN:MCHP...）。这是一个可选步骤。如果该工具不可用或尚未连接，用户可从 **Hardware Tools**（硬件工具）中选择 **Simulator**（软件模拟器）工具，或者等到需要时再做选择。

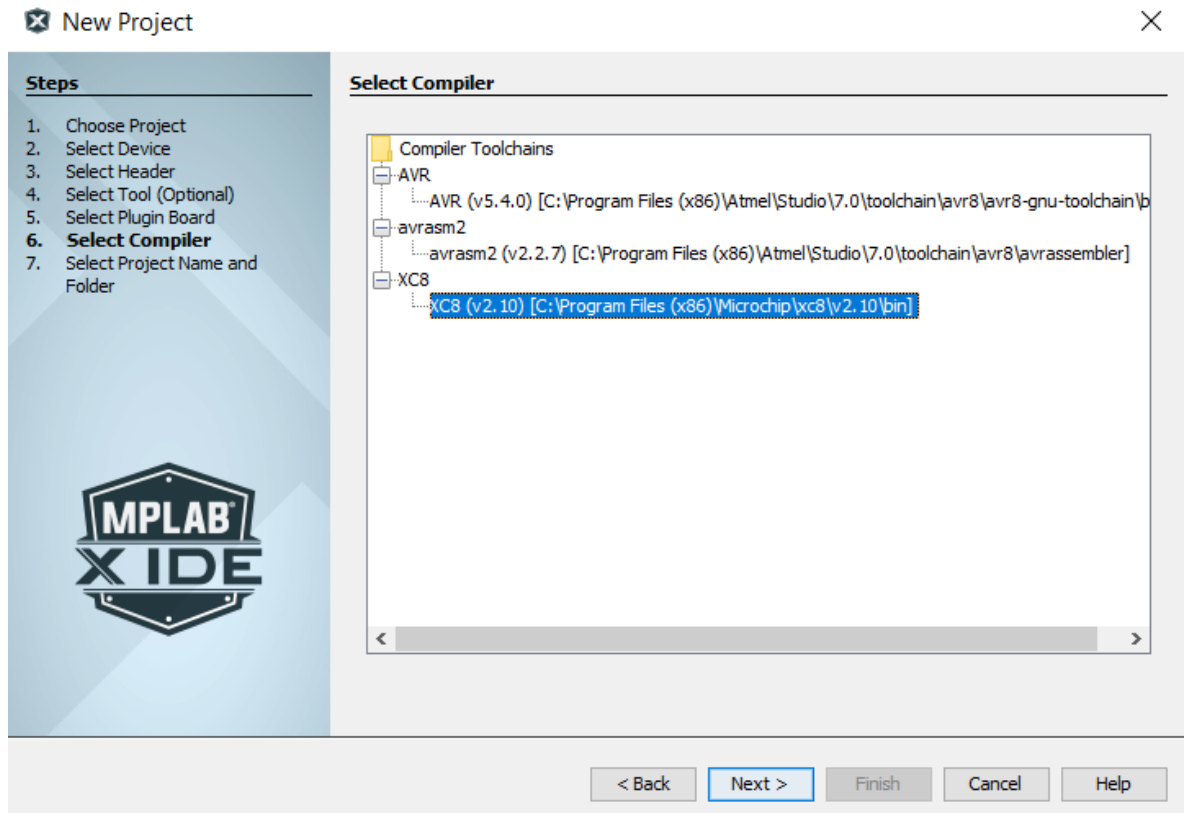
图 2-7. 选择工具



信息： 用户还可以双击工具名称，为其设置易记的名称（Friendly Name, FN）。

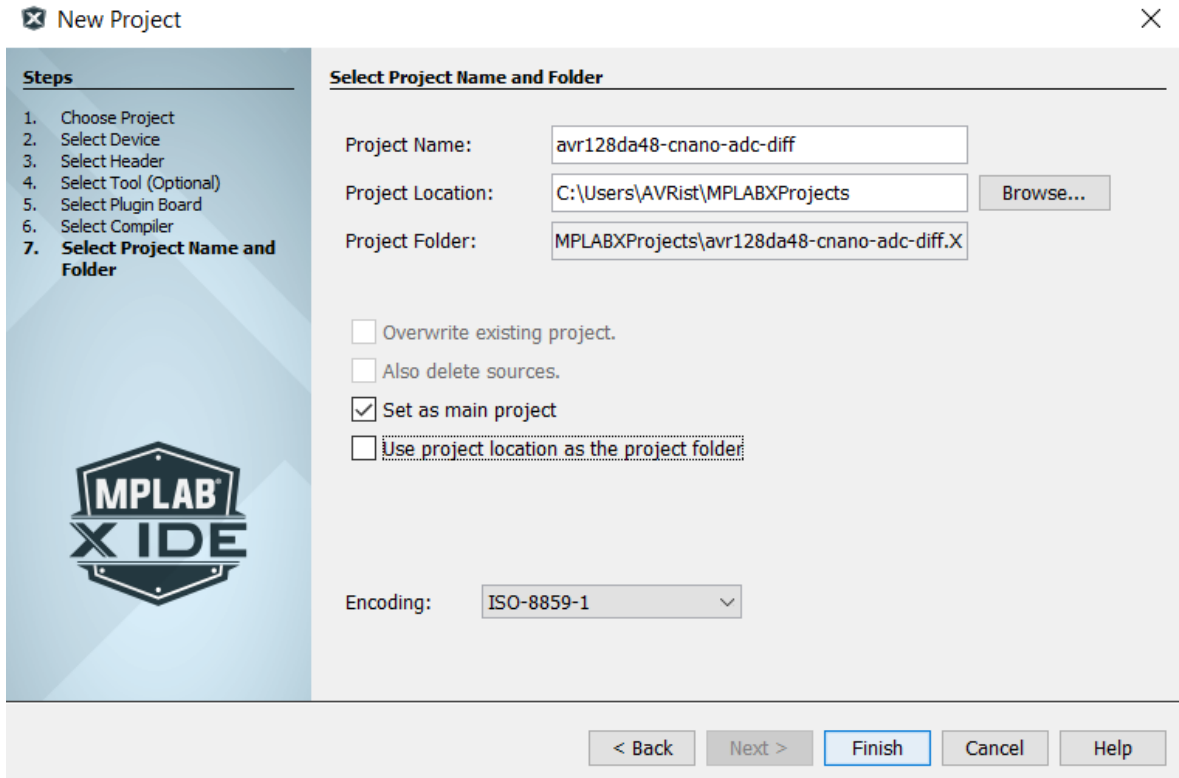
5. 选择本项目所需版本的 XC8 编译器，如图 2-8 所示。

图 2-8. 选择编译器



6. 最后一步是 Select Project Name and Folder（选择项目名称和文件夹）。在本例中，项目名称将是 avr128da48-cnano-adc-diff。在该步骤中，用户还可以配置项目位置、项目文件夹和编码。

图 2-9. 选择项目名称和文件夹



完成所有必要的步骤后，项目窗口中将出现一个新的空项目。所有必要的文件都将使用 MCC 插件引入项目中。

2.3.2 MPLAB®代码配置器（MCC）

本项目将使用 MCC 进行开发。MCC 是一款图形编程环境，可生成简单易懂的 C 语言代码，之后插入您的项目中。它采用直观界面；可针对目标应用程序来使能和配置众多外设及功能。

安装该插件时，可遵循[安装 MPLAB®代码配置器（MCC）](#)网页上提供的步骤。

在生成所有必要的源文件和头文件后，用户必须补全源代码以获得所需的功能。实现代码后，必须对应用程序的功能进行测试，具体将使用 MDV 插件来完成。

2.3.3 MPLAB®数据可视化器（MDV）

MDV 是一款图形化的运行时调试工具，既可以作为 MPLAB 插件使用，也可以作为独立的调试工具使用。该工具以图形形式显示嵌入式应用程序中的运行时变量和函数。

可以将数据以图形形式显示为以下变量：

- 原始数据流 8 位变量
- 采用数据流协议的多个变量

安装 MDV 插件时可以遵循 MCC 的安装步骤，二者惟一的差别就是插件的名称。该工具用于在实现应用程序后显示 ADC 结果。

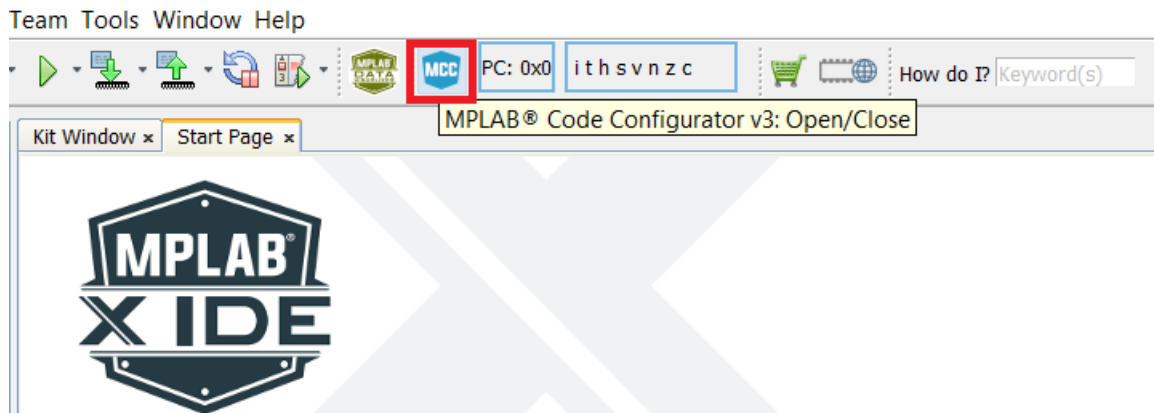
2.4 任务：差分 ADC

2.4.1 初始化模块

本小节将提供开发一个简单的应用程序所需的全部步骤，其中包括将 ADC 配置为差分模式、通过 USART 发送结果，以及使用图形界面分析数据。

MCC 将用于初始化所需的全部模块。要打开插件，请转到 *Window* → *MPLAB Code Configurator* → *MPLAB Code Configurator Open/Close*（窗口 → MPLAB 代码配置器 → MPLAB 代码配置器打开/关闭），或者单击 **MCC** 图标，如图 2-10 所示。

图 2-10. 打开/关闭 MCC



使用 MCC 开发该应用程序的第一步是提供初始化设置并将所需的外设模块添加到项目中。初始化系统时，用户必须为所有必要的外设初始化所需的配置。之后，用户必须使用生成的源文件和头文件来实现将由 MCU 执行的算法。

2.4.1.1 配置器件系统

开发 ADC 应用程序的第一步是配置系统时钟。如需了解有哪些时钟选项，用户必须查阅器件数据手册中的 *CLKCTRL*——*时钟控制器*一章。

例如，在本应用程序中，所选的主时钟频率为 2 MHz。



任务：使用 MCC 将内部高频振荡器配置为时钟源。振荡器的频率必须为 4 MHz。主时钟频率必须为 2 MHz。



结果: 打开 MCC，然后转到 *Resource Management [MCC]* → *Project Resources* → *System* → *System Module*（资源管理 [MCC] → 项目资源 → 系统 → 系统模块）。在 **Easy Setup**（简单设置）选项卡中，必须完成以下配置：

- 时钟控制
 - Clock Source（时钟源）：内部振荡器
 - Internal Oscillator Frequency（内部振荡器频率）：1-32 MHz 内部振荡器
 - Oscillator Frequency Options（振荡器频率选项）：4 MHz 系统时钟（默认设置）
 - Prescaler Enable（预分频器使能）：已选中
 - 预分频比：2X

具体设置如图 2-11 所示。

图 2-11. MCC 时钟控制配置

Parameter	Value
Main Clock(Hz):	2000000
Clock Source :	Internal Oscillator
Internal Oscillator Frequency:	1-32MHz internal oscillator
Oscillator Frequency Options:	4 MHz system clock (default)
PLL Enable:	<input type="checkbox"/>
External Clock(Hz):	1 ≤ 1000000 ≤ 20000000
Prescaler Enable:	<input checked="" type="checkbox"/>
Prescaler:	2X
Clock Out Enable:	<input type="checkbox"/>

2.4.1.2 配置 ADC

在本文档所述的应用程序中，ADC 将配置为差分模式。要使用 MCC 配置该模块的初始设置，用户必须先将该模块添加到 *Project Resources*（项目资源）。



任务: 在 MCC 中将 ADC 外设模块添加到 *Project Resources*。



结果: 转到 *Resource Management [MCC]* → *Device Resources* → *Peripherals* → *ADC*（资源管理 [MCC] → 器件资源 → 外设 → ADC），然后单击绿色的+符号添加 ADC 外设模块。该模块随后将出现在 *Project Resources* 窗口中。

下一步是配置 ADC 模块。ADC 将以差分模式运行，连续转换获取的采样（自由运行模式）。分辨率将是 12 位，预分频值将是 4，差分输入将是 PD3（正输入）和 PD4（负输入）。



任务:

- 使能 ADC 模块
- 使能差分模式转换，自由运行
- 将结果分辨率设置为 12 位
- 配置 ADC 预分频器以获得 500 kHz 的 ADC 频率
- 配置 ADC 输入引脚：ADC 输入引脚 3 作为负输入；ADC 输入引脚 4 作为正输入



结果: 转到 *Resource Management [MCC] → Project Resources → Peripherals* (资源管理 [MCC] → 项目资源 → 外设)，然后选择 **ADC0**。在 **Easy Setup** 选项卡中，必须完成以下配置：

- Software Settings (软件设置) :
 - Result Selection (结果选择) : 12 位模式
 - Differential Mode Conversion (差分模式转换) : enabled (已使能)
 - Left Adjust Result (左对齐结果) : 未选中
- Hardware Settings (硬件设置) :
 - Enable ADC (使能 ADC) : 已选中

具体设置如图 2-12 和图 2-13 所示。

图 2-12. ADC0 软件设置

Software Settings	
API Prefix:	ADC0
Result Selection :	12-bit mode
Differential Mode Conversion :	enabled
Left Adjust Result :	<input type="checkbox"/>

图 2-13. ADC0 硬件设置

Hardware Settings	
Enable ADC:	<input checked="" type="checkbox"/>
Sampling Frequency(Hz):	11363 ≤ 38461 ≤ 38461
ADC Clock(Hz):	500000
Sample Accumulation Number:	No accumulation
Sample Length (# of ADC Clock) :	0 ≤ 0 ≤ 31

此外，用户必须配置 ADC 寄存器。如需了解哪个寄存器中有哪些设置以及寄存器位和位域的可用配置，用户必须查阅器件数据手册。

自由运行模式选项位于 ADC 模块的控制 A (CTRLA) 寄存器中。通过写入 CTRLA 寄存器，用户可以使能/禁止在待机模式下运行、选择转换模式、设置结果调整、选择分辨率，以及使能/禁止外设。该外设的时钟频率必须使用控制 C (CTRLC) 寄存器中的预分频器 (PRESC) 位域进行配置。

要使用 ADC 的差分模式，需要两个模拟输入。正输入必须使用 ADC 正输入的多路开关选择 (MUXPOS) 寄存器进行配置，而负输入必须使用 ADC 负输入的多路开关选择 (MUXNEG) 寄存器进行配置。相应的引脚也必须配置为模拟输入，并禁止数字缓冲器。

可以在 MCC 的 **Registers**（寄存器）选项卡中配置寄存器。

- 寄存器: CTRLA
 - FREERUN（自由运行）: enabled

图 2-14. ADC0 CTRLA 寄存器

Register: CTRLA	Value
CONVMODE	enabled
ENABLE	enabled
FREERUN	enabled
LEFTADJ	disabled
RESSEL	12-bit mode
RUNSTBY	disabled

- 寄存器: CTRLC
 - PRESC: CLK_PER divided by 4（CLK_PER 4 分频）

图 2-15. ADC0 CTRLC 寄存器

Register: CTRLC	Value
PRESC	CLK_PER divided by 4

- 寄存器: MUXNEG
 - MUXNEG: ADC 输入引脚 4
- 寄存器: MUXPOS
 - MUXPOS: ADC 输入引脚 3

图 2-16. ADC0 MUXNEG 和 MUXPOS 寄存器

Register: MUXNEG	Value
MUXNEG	ADC input pin 4

Register: MUXPOS	Value
MUXPOS	ADC input pin 3

2.4.1.3 配置 VREF

ADC 参考电压既可以通过写入 ADC0 参考电压（ADC0REF）寄存器进行设置，也可以使用 MCC 进行设置。



任务：使用 MCC 将 VREF 模块添加到 Project Resources，并将 ADC 参考电压配置为 2.048V。



结果：转到 *Resource Management [MCC] → Device Resources → Peripherals → VREF*（资源管理 [MCC] → 器件资源 → 外设 → VREF），然后单击绿色的+符号添加 VREF 外设模块。该模块随后将出现在 Project Resources 窗口中。

转到 *Resource Management [MCC] → Project Resources → Peripherals*，然后选择 **VREF**。在 **Easy Setup**（简单设置）选项卡中，必须完成以下配置：

- Hardware Settings
 - ADC Voltage Reference（ADC 参考电压）：Internal 2.048V reference（内部 2.048V 参考电压）

图 2-17. VREF ADC 参考电压



2.4.1.4 配置 USART

使用 ADC 对收到的模拟数据进行转换后，用户可进一步发送数据以进行分析。要将数据发送到开发计算机主机，将使用 USART 外设模块。在 AVR128DA48 Curiosity Nano 板上，USART1 的 RX（接收）和 TX（发送）引脚直接连接到调试器引脚，所以用户无需额外的电线即可向计算机发送数据。因此，本应用程序中使用的 USART 外设模块将是 USART1。



任务：使用 MCC 将 USART1 模块添加到项目中。



结果：转到 *Resource Management [MCC] → Device Resources → Peripherals → USART*（资源管理 [MCC] → 器件资源 → 外设 → USART），然后单击绿色的+符号添加 USART1 外设模块。该模块随后将出现在 Project Resources 窗口中。

将该模块引入项目后，必须进行一些初始配置。



任务：对 USART1 进行以下配置：在异步模式下运行，波特率为 9600，无奇偶校验，1 个停止位。字符大小必须为 8 位。



信息：无需使能 USART RX：器件不需要从计算机接收数据。只需要使能 USART TX。



结果: 转到 *Resource Management [MCC]* → *Project Resources* → *Pin Module* (资源管理 [MCC] → 项目资源 → 引脚模块), 然后按图 2-20 所示进行配置。

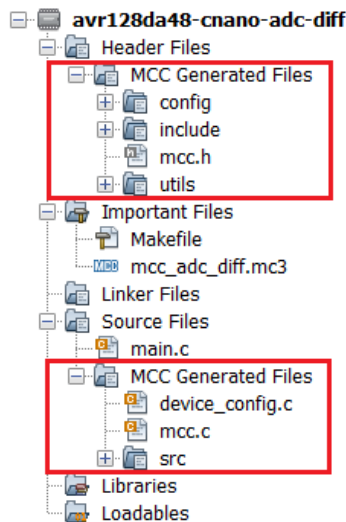
图 2-20. 引脚模块

Pin Name	Module	Function	Custom Name	OUTPUT	START HIGH	INVEN	PULLUPEN	ISC
PC0	USART1	TXD		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Interrupt disabled but input buffer enabled
PC1	USART1	RXD		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Interrupt disabled but input buffer enabled
PD4	ADC0	AINx		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Digital Input Buffer disabled
PD3	ADC0	AINx		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Digital Input Buffer disabled

2.4.2 生成代码概述

要使用 MCC 生成设计的代码, 请单击 **Generate** (生成) 按钮。之后, 若要继续进行应用程序开发, 应关闭 MCC。MCC 生成的文件可以在 **Projects** 选项卡中已创建项目的下方查看。

图 2-21. MCC 生成的文件



生成的文件被包含在主源文件中, 随后系统被初始化。下面给出了针对 `main.c` 生成的代码。

例 2-1. 代码清单 1——生成的主文件

```
#include "mcc_generated_files/mcc.h"

/*
 * Main application
 */
int main(void)
{
    /* Initializes MCU, drivers and middleware */
    SYSTEM_Initialize();

    /* Replace with your application code */
    while (1)
    {
    }
}
/**
 * End of File
 */
```

SYSTEM_Initialize 函数用于初始化系统和所有外设模块（与之前使用 MCC 配置的一样）：系统时钟、ADC、USART 和 VREF。



信息：使用 ADC0、USART0 和 VREF 实现的算法所需的全部函数已由 MCC 生成完毕。

生成的文件中提供 ADC 和 USART 函数，以帮助用户实现算法。本应用程序所关注的函数为 ADC0_GetDiffConversion 和 USART1_Write。它们在 MCC 生成的文件中的实现如以下代码清单所示。

例 2-2. 代码清单 2——用于获取差分转换结果的 ADC 函数

```
diff_adc_result_t ADC0_GetDiffConversion(adc_0_channel_t channel,
adc_0_muxneg_channel_t channel1)
{
    diff_adc_result_t res;

    ADC0_StartDiffConversion(channel, channel1);
    while (!ADC0_IsConversionDone());
    res = ADC0_GetConversionResult();
    ADC0.INTFLAGS |= ADC_RESRDY_bm;
    return res;
}
```

例 2-3. 代码清单 3——用于发送 ADC 结果的 USART 函数

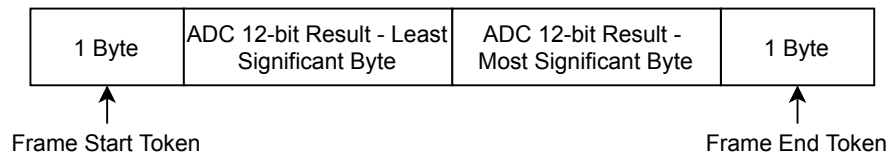
```
void USART1_Write(const uint8_t data)
{
    while (!(USART1.STATUS & USART_DREIF_bm)) ;
    USART1.TXDATA1 = data;
}
```

2.4.3 实现所需的算法

本小节将介绍如何在主文件中使用生成的文件实现算法。此外，还将提供使用数据可视化器测试应用程序所需的函数。

MPLAB 数据可视化器在编程的嵌入式器件与计算机之间提供简单的单向通信。嵌入式应用程序必须控制何时发送数据以及如何发送数据。在本例中，数据将按图 2-22 所示进行打包。

图 2-22. 数据可视化器 ADC 数据帧



发送的数据必须通过开始令牌和结束令牌打包成帧。这两个令牌彼此之间互为二进制补码。数据可视化器负责同步成帧令牌和有效负载大小。在多字节变量中，必须先发送低字节。



任务：为了读取 ADC 差分结果并将其发送给计算机，必须在主文件中实现以下代码。

例 2-4. 代码清单 4——读取 ADC 结果并发送

```
int main(void)
{
    diff_adc_result_t adcVal_12b;

    /* Initializes MCU, drivers and middleware */
    SYSTEM_Initialize();

    while (1)
    {
        adcVal_12b = ADC0_GetDiffConversion(ADC_MUXPOS_AIN3_gc,
        ADC_MUXNEG_AIN4_gc);

        USART1_Write(START_TOKEN);
        USART1_Write(adcVal_12b & 0x00FF);
        USART1_Write(adcVal_12b >> 8);
        USART1_Write(END_TOKEN);
    }
}
```



任务：START_TOKEN 和 END_TOKEN 是用户定义的宏，若要在主函数中使用，必须也在 main.c 文件中定义一次，具体如下所示。

```
#define START_TOKEN 0x03    /* 开始帧令牌 */
#define END_TOKEN 0xFC     /* 结束帧令牌 */
```



[View Code Example on GitHub](#)

Click to browse repositories

2.4.4 应用程序测试：调试

测试应用程序功能的方法之一是分步执行它，然后检查所有实现的指令是否提供了预期的结果。



任务：在主函数中的 SYSTEM_Initialize(); 函数调用上使能断点。进入调试模式。



结果：通过单击文件行旁边的编辑器装订线可创建新的行断点。相应的行如下所示。

图 2-23. 使能断点

```
32 int main(void)
33 {
34     diff_adc_result_t adcVal_12b;
35
36     /* Initializes MCU, drivers and middleware */
37     SYSTEM_Initialize();
38
39     while (1)
40     {
41         adcVal_12b = ADC0_GetDiffConversion(ADC_MUXPOS_AIN3_gc, ADC_MUXNEG_AIN4_gc);
42
43         USART1_Write(START_TOKEN);
44         USART1_Write(adcVal_12b & 0x00FF);
45         USART1_Write(adcVal_12b >> 8);
46         USART1_Write(END_TOKEN);
47     }
48 }
```

要查看寄存器的内容，请转到 [Window → Debugging](#)（窗口 → 调试），然后选择 **IO View**（IO 视图）。随后将打开 IO View 窗口。要开始调试，请转到 [Debug](#)（调试），然后选择 **Debug Main Project**（调试主项目）。程序将执行到断点处停止。

单击 **Step Over**（单步跳过）按钮后，用户可以通过查看 IO View 窗口中的寄存器来检查 ADC 外设是否按要求进行了初始化等。IO View 窗口中的寄存器如 [图 2-24](#) 所示。



任务：检查是否所有的配置都按要求完成。

图 2-24. ADC IO View

Icon	Peripheral	Option
	(ADC0)	
	Resolution selection (CTRLA)	0x0 - 12-bit mode
	Accumulation Samples (CTRLB)	0x0 - No accumulation
	Clock Pre-scaler (CTRLC)	0x1 - CLK_PER divided by 4
	Initial Delay Selection (CTRLD)	0x0 - Delay 0 CLK_ADC cycles
	Window Comparator Mode (CTRLC)	0x0 - No Window Comparison
	Analog Channel Selection Bits (MUXNEG)	0x4 - ADC input pin 4
	Analog Channel Selection Bits (MUXPOS)	0x3 - ADC input pin 3

Icon	Name	Address	Value	Decimal	Bits
	(ADC0)				
	COMMAND	0x060A	0x00	0	7 6 5 4 3 2 1 0
	CTRLA	0x0600	0x21	33	7 5 4 3 2 1 0
	RUNSTBY	0	0	0	7
	CONVMODE	1	1	1	5
	LEFTADJ	0	0	0	4
	RESSEL	0	0	0	3 2
	FREERUN	0	0	0	1
	ENABLE	1	1	1	0
	CTRLB	0x0601	0x00	0	2 1 0
	CTRLC	0x0602	0x01	1	3 2 1 0
	PRESC	1	1	1	3 2 1 0
	CTRLD	0x0603	0x00	0	7 6 5 3 2 1 0
	CTRLC	0x0604	0x00	0	2 1 0
	DBGCTRL	0x060E	0x00	0	0
	EVCTRL	0x060B	0x00	0	0
	INTCTRL	0x060C	0x00	0	1 0
	INTFLAGS	0x060D	0x00	0	1 0
	MUXNEG	0x0609	0x04	4	6 5 4 3 2 1 0
	MUXNEG	4	4	4	6 5 4 3 2 1 0
	MUXPOS	0x0608	0x03	3	6 5 4 3 2 1 0
	MUXPOS	3	3	3	6 5 4 3 2 1 0

通过单步执行应用程序，用户可以看到：

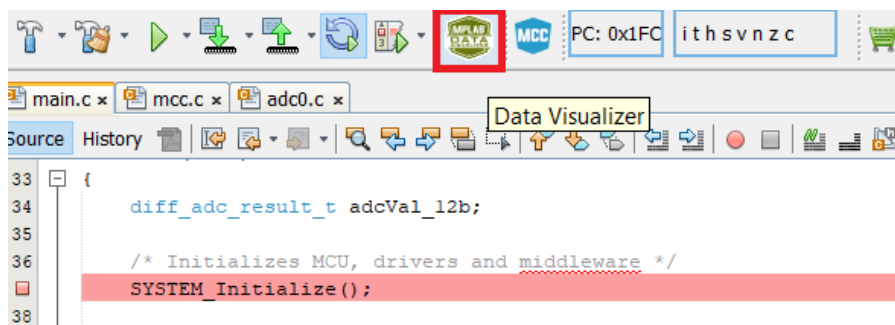
- ADC 结果（RES）寄存器中的转换结果
- 需要通过 USART 发送的数据——在 TXDATAL 寄存器中
- adcVal 变量的值，使用 Variables（变量）窗口

2.4.5 应用程序测试：数据可视化器

使用 MPLAB 数据可视化器可以轻松测试应用程序。要绘制通过 USART 发送的 ADC 结果，必须遵循以下步骤：

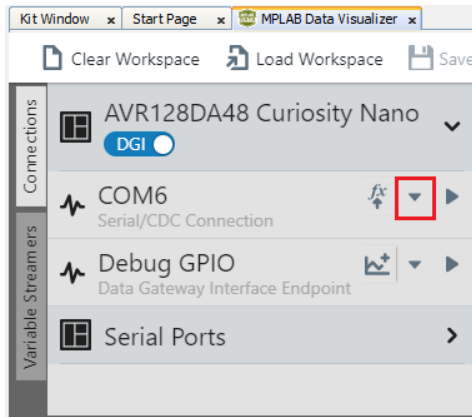
1. 单击插件图标打开数据可视化器插件，如图 2-25 所示。

图 2-25. 打开数据可视化器



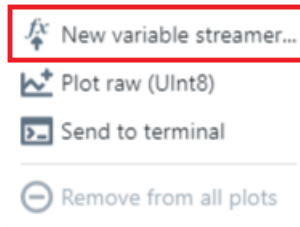
2. 从 **Connections**（连接）选项卡中选择 **Curiosity Nano communication port (COMn) drop-down list**（Curiosity Nano 通信端口（COMn）下拉列表），如图 2-26 所示。

图 2-26. COMn 端口——显示下拉列表



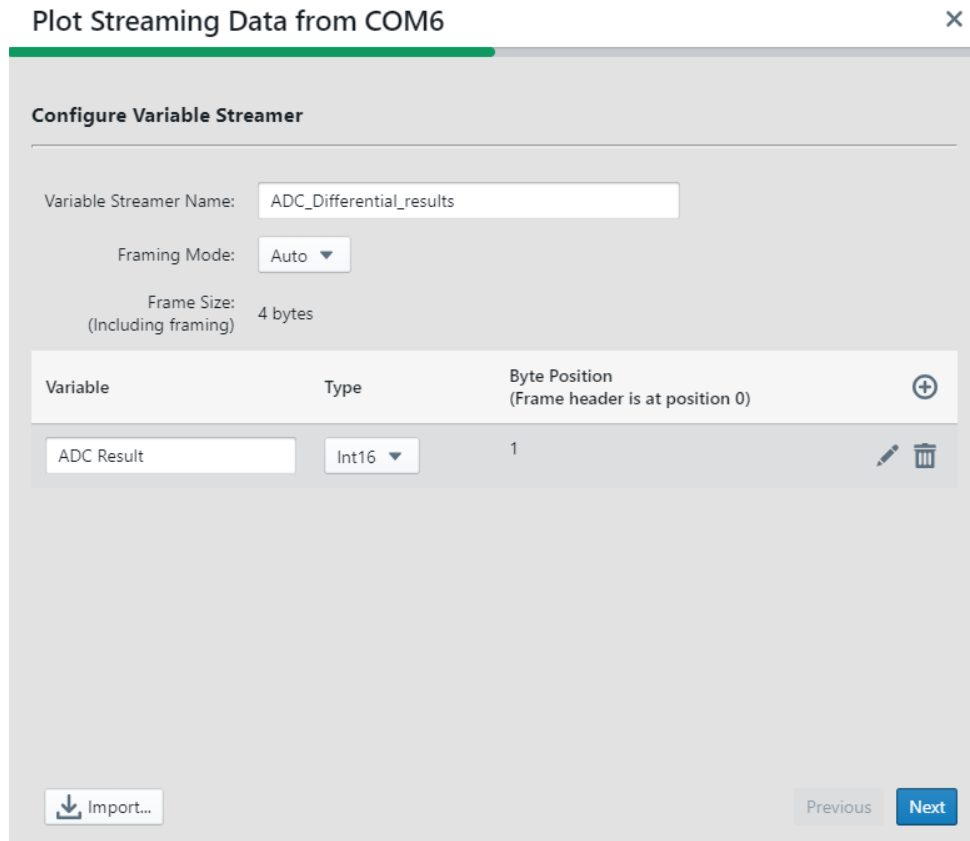
3. 从下拉列表中选择 **New variable streamer...**（新建变量流...），如图 2-27 所示。

图 2-27. COMn 端口——从下拉列表中选择



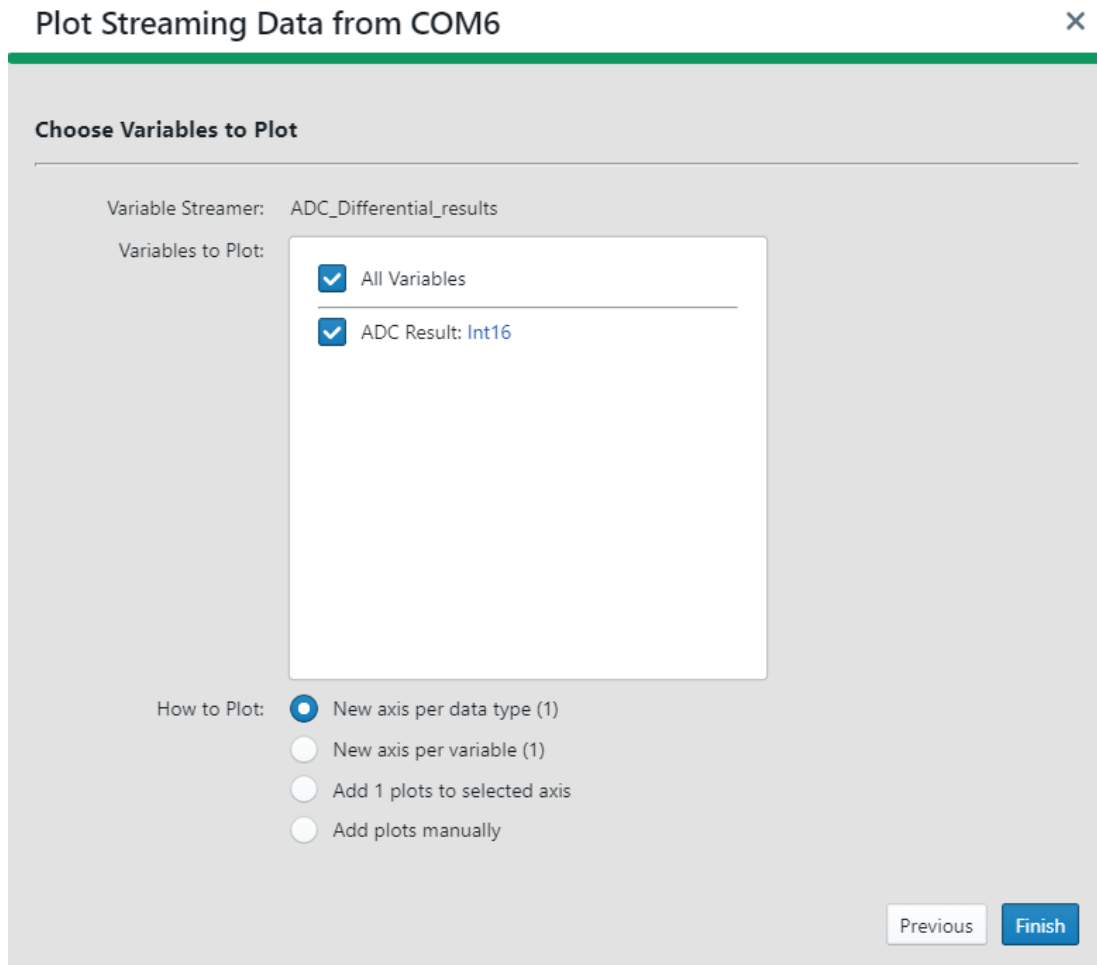
4. 选择 **Variable Streamer Name**（变量流名称），添加将接收的变量，然后单击 **Next**。

图 2-28. Configure Variable Streamer（配置变量流）



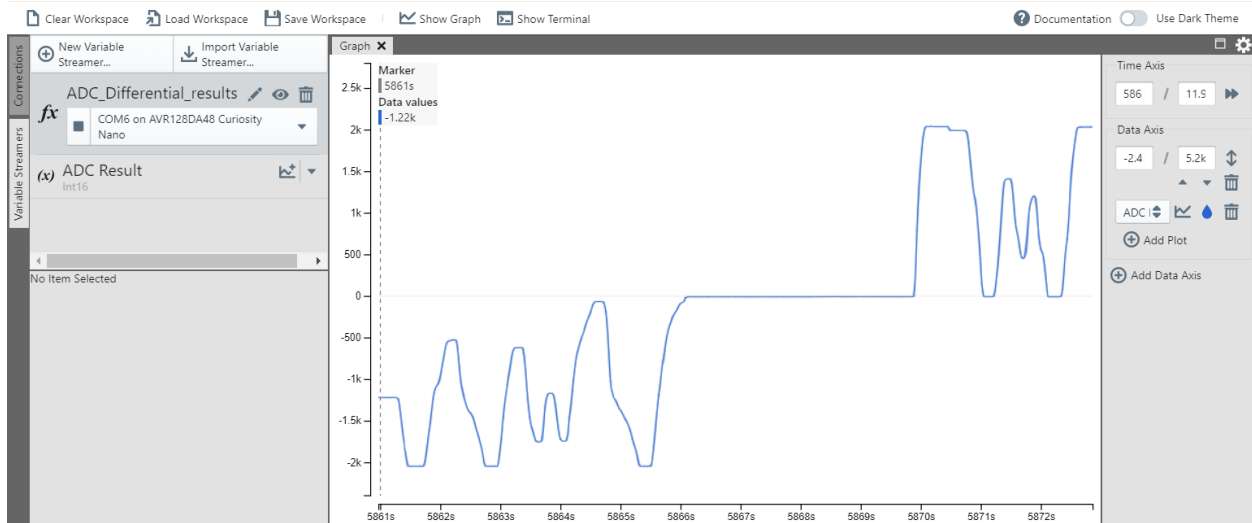
5. 根据需要选择要绘制的变量，然后选择 **New axis per data type (1)**（按照数据类型新建轴（1））作为绘制数据的方式。最后，单击 **Finish**（完成）。

图 2-29. Choose Variables to Plot（选择要绘制的变量）



随机转到两个电位器就获得了绘图。ADC 结果如图 2-30 所示。

图 2-30. 数据可视化器 ADC 结果绘图



3. 结论

完成本文档提供的培训后，用户可以了解 ADC 的基本特性，了解如何使用开发嵌入式应用所需的软件工具，以及使用 ADC 模块独立开发一个基本的应用程序。此外，用户还将了解如何配置 ADC 以转换来自差分输入的数据，如何连续转换数据，以及如何解析结果。本培训还提供了调试应用程序和查看结果所需的步骤。

4. 参考资料

1. [数据可视化器软件用户指南](#)
2. [MPLAB®代码配置器 \(MCC\)](#)
3. [如何在 MPLAB 代码配置器 \(MCC\) 中添加库](#)
4. [下载 MCC 最新库的 Microchip 网页](#)
5. [使用 MPLAB 数据可视化器进行可视化调试](#)
6. [差分和单端 ADC 白皮书](#)
7. [AVR128DA48 Curiosity Nano 评估工具包](#)
8. [适用于 Click Boards™ 的 Curiosity Nano 基板](#)

5. 版本历史

文档版本	日期	备注
A	2020 年 8 月	文档初始版本

Microchip 网站

Microchip 网站 (www.microchip.com) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。我们的网站提供以下内容：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 设计伙伴计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

产品变更通知服务

Microchip 的产品变更通知服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请访问 www.microchip.com/pcn，然后按照注册说明进行操作。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (ESE)
- 技术支持

客户应联系其代理商、代表或 ESE 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 www.microchip.com/support 获得网上技术支持。

Microchip 器件代码保护功能

请注意以下有关 Microchip 产品代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术规范。
- Microchip 确信：在正常使用且符合工作规范的情况下，Microchip 系列产品非常安全。
- Microchip 注重并积极保护其知识产权。严禁任何试图破坏 Microchip 产品代码保护功能的行为，这种行为可能会违反《数字千年版权法案》(Digital Millennium Copyright Act)。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。

法律声明

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物及其提供的信息仅适用于 Microchip 产品，包括设计、测试以及将 Microchip 产品集成到您的应用中。以其他方式使用这些信息都将被视为违反条款。本出版物中的器件应用信息仅为您提供便利，将来可能会发生更新。如需额外的支持，请联系当地的 Microchip 销售办事处，或访问 <https://www.microchip.com/en-us/support/design-help/client-supportservices>。

Microchip “按原样”提供这些信息。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对非侵权性、适销性和特定用途的适用性的暗示担保，或针对其使用情况、质量或性能的担保。

在任何情况下，对于因这些信息或使用这些信息而产生的任何间接的、特殊的、惩罚性的、偶然的或间接的损失、损害或任何类型的开销，Microchip 概不承担任何责任，即使 Microchip 已被告知可能发生损害或损害可以预见。在法律允许的最大范围内，对于因这些信息或使用这些信息而产生的所有索赔，Microchip 在任何情况下所承担的全部责任均不超出您为获得这些信息向 Microchip 直接支付的金额（如有）。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切损害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任。除非另外声明，在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、Microchip 徽标、Adaptec、AnyRate、AVR、AVR 徽标、AVR Freaks、BesTime、BitCloud、chipKIT、chipKIT 徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi 徽标、MOST、MOST 徽标、MPLAB、OptoLyzer、PacTime、PIC、picoPower、PICSTART、PIC32 徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST 徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTrackr、TimeSource、tinyAVR、UNI/O、Vectron 和 XMEGA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的注册商标。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath 和 ZL 均为 Microchip Technology Incorporated 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICKit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQL、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Incorporated 在美国的服务标记。

Adaptec 徽标、Frequency on Demand、Silicon Storage Technology 和 Symmcom 为 Microchip Technology Inc. 在其他国家或地区的注册商标。

GestIC 是 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2023, Microchip Technology Incorporated, 美国印刷, 版权所有。

ISBN: 978-1-6683-2486-8

质量管理体系

有关 Microchip 的质量管理体系的信息，请访问 www.microchip.com/quality。

全球销售及服务中心

美洲	亚太地区	亚太地区	欧洲
公司总部 2355 West Chandler Blvd. Chandler, AZ 85224-6199 电话: 480-792-7200 传真: 480-792-7277 技术支持: www.microchip.com/support 网址: www.microchip.com	澳大利亚 - 悉尼 电话: 61-2-9868-6733 中国 - 北京 电话: 86-10-8569-7000 中国 - 成都 电话: 86-28-8665-5511 中国 - 重庆 电话: 86-23-8980-9588 中国 - 东莞 电话: 86-769-8702-9880 中国 - 广州 电话: 86-20-8755-8029 中国 - 杭州 电话: 86-571-8792-8115 中国 - 香港特别行政区 电话: 852-2943-5100 中国 - 南京 电话: 86-25-8473-2460 中国 - 青岛 电话: 86-532-8502-7355 中国 - 上海 电话: 86-21-3326-8000 中国 - 沈阳 电话: 86-24-2334-2829 中国 - 深圳 电话: 86-755-8864-2200 中国 - 苏州 电话: 86-186-6233-1526 中国 - 武汉 电话: 86-27-5980-5300 中国 - 西安 电话: 86-29-8833-7252 中国 - 厦门 电话: 86-592-2388138 中国 - 珠海 电话: 86-756-3210040	印度 - 班加罗尔 电话: 91-80-3090-4444 印度 - 新德里 电话: 91-11-4160-8631 印度 - 浦那 电话: 91-20-4121-0141 日本 - 大阪 电话: 81-6-6152-7160 日本 - 东京 电话: 81-3-6880-3770 韩国 - 大邱 电话: 82-53-744-4301 韩国 - 首尔 电话: 82-2-554-7200 马来西亚 - 吉隆坡 电话: 60-3-7651-7906 马来西亚 - 槟榔屿 电话: 60-4-227-8870 菲律宾 - 马尼拉 电话: 63-2-634-9065 新加坡 电话: 65-6334-8870 台湾地区 - 新竹 电话: 886-3-577-8366 台湾地区 - 高雄 电话: 886-7-213-7830 台湾地区 - 台北 电话: 886-2-2508-8600 泰国 - 曼谷 电话: 66-2-694-1351 越南 - 胡志明市 电话: 84-28-5448-2100	奥地利 - 韦尔斯 电话: 43-7242-2244-39 传真: 43-7242-2244-393 丹麦 - 哥本哈根 电话: 45-4485-5910 传真: 45-4485-2829 芬兰 - 埃斯波 电话: 358-9-4520-820 法国 - 巴黎 电话: 33-1-69-53-63-20 传真: 33-1-69-30-90-79 德国 - 加兴 电话: 49-8931-9700 德国 - 哈恩 电话: 49-2129-3766400 德国 - 海尔布隆 电话: 49-7131-72400 德国 - 卡尔斯鲁厄 电话: 49-721-625370 德国 - 慕尼黑 电话: 49-89-627-144-0 传真: 49-89-627-144-44 德国 - 罗森海姆 电话: 49-8031-354-560 以色列 - 若那那市 电话: 972-9-744-7705 意大利 - 米兰 电话: 39-0331-742611 传真: 39-0331-466781 意大利 - 帕多瓦 电话: 39-049-7625286 荷兰 - 德卢内市 电话: 31-416-690399 传真: 31-416-690340 挪威 - 特隆赫姆 电话: 47-72884388 波兰 - 华沙 电话: 48-22-3325737 罗马尼亚 - 布加勒斯特 电话: 40-21-407-87-50 西班牙 - 马德里 电话: 34-91-708-08-90 传真: 34-91-708-08-91 瑞典 - 哥德堡 电话: 46-31-704-60-40 瑞典 - 斯德哥尔摩 电话: 46-8-5090-4654 英国 - 沃金厄姆 电话: 44-118-921-5800 传真: 44-118-921-5820
亚特兰大 德卢斯, 佐治亚州 电话: 678-957-9614 传真: 678-957-1455 奥斯汀, 德克萨斯州 电话: 512-257-3370 波士顿 韦斯特伯鲁, 马萨诸塞州 电话: 774-760-0087 传真: 774-760-0088 芝加哥 艾塔斯卡, 伊利诺伊州 电话: 630-285-0071 传真: 630-285-0075 达拉斯 阿迪森, 德克萨斯州 电话: 972-818-7423 传真: 972-818-2924 底特律 诺维, 密歇根州 电话: 248-848-4000 休斯顿, 德克萨斯州 电话: 281-894-5983 印第安纳波利斯 诺布尔斯特维尔, 印第安纳州 电话: 317-773-8323 传真: 317-773-5453 电话: 317-536-2380 洛杉矶 米慎维荷, 加利福尼亚州 电话: 949-462-9523 传真: 949-462-9608 电话: 951-273-7800 罗利, 北卡罗来纳州 电话: 919-844-7510 纽约, 纽约州 电话: 631-435-6000 圣何塞, 加利福尼亚州 电话: 408-735-9110 电话: 408-436-4270 加拿大 - 多伦多 电话: 905-695-1980 传真: 905-695-2078			