



AVR[®] DA Training Manual

Differential ADC Using the AVR128DA48 Curiosity Nano

Prerequisites

This section has the purpose to provide a list of all requirements to complete this training.

Hardware Prerequisites

- AVR128DA48 Curiosity Nano ([DM164151](#))
- Curiosity Nano Base for Click boards™ ([AC164162](#))
- POT Click board ([MIKROE-3402](#))
- POT 2 Click board ([MIKROE-3325](#))

Note: The POT Click boards can be replaced by any other two potentiometers. A schematic for this case will be provided.

Software Prerequisites

The software versions used for this training are presented below.

- MPLAB[®] X IDE v5.40 or above
- MPLAB XC8 Compiler v2.20 or above
- AVR-Dx_DFP (Device Family Pack) v1.1.40
- MPLAB Data Visualizer (MDV) v1.1
- MPLAB Code Configurator (MCC) v3.95
- MCC 8-bit AVR[®] MCUs Library v2.3.0

Documentation Materials

- [AVR128DA28/32/48/64 Data Sheet](#)
- [TB3245: Using 12-Bit ADC for Conversions, Accumulation, and Triggering Events](#)
- Other device related documents can be found at: [AVR128DA48 Device Overview](#)

Introduction

Considering that a lot of external stimuli are analog-type stimuli, embedded applications often rely on analog inputs, provided by analog sensors. Most Microchip microcontrollers (MCUs) are equipped with an integrated Analog-to-Digital Converter (ADC) to acquire analog data and be able to process it.

This document describes how to develop an application using the differential conversion feature of the ADC, with the AVR128DA48 Curiosity Nano evaluation kit. It provides an overview of the peripheral, and explains the steps to configure the ADC in Differential mode.

After completing this training, the user will be able to:

- Initialize the system by configuring all peripherals
- Start an ADC conversion
- Continuously read the ADC result and send it through the Universal Synchronous/Asynchronous Receiver/Transmitter (USART)
- Run the application step by step to understand the configurations
- Test the developed application using the hardware setup
- Visualize the data using the graphical interface tool

Firstly, an overview of the ADC module will be presented. Then, the proposed application will be described. After establishing an overview, all the steps needed to implement the application will be provided: Getting familiar with the software tools, initializing the system and the peripheral modules, implementing other needed functionalities, debugging the application, and visualizing the received data using the MPLAB Data Visualizer.

Table of Contents

Prerequisites.....	1
Introduction.....	2
1. Overview.....	4
1.1. Analog-to-Digital Converter Module Overview.....	4
1.2. Application Overview.....	5
2. Analog-to-Digital Converter Training.....	6
2.1. Icon Identifiers.....	6
2.2. Hardware Setup.....	6
2.3. Get Familiar with the Software Environment.....	7
2.4. Assignment: Differential ADC.....	14
3. Conclusion.....	27
4. References.....	28
5. Revision History.....	29
The Microchip Website.....	30
Product Change Notification Service.....	30
Customer Support.....	30
Microchip Devices Code Protection Feature.....	30
Legal Notice.....	31
Trademarks.....	31
Quality Management System.....	32
Worldwide Sales and Service.....	33

1. Overview

This section provides an overview of the ADC module and the content of this training.

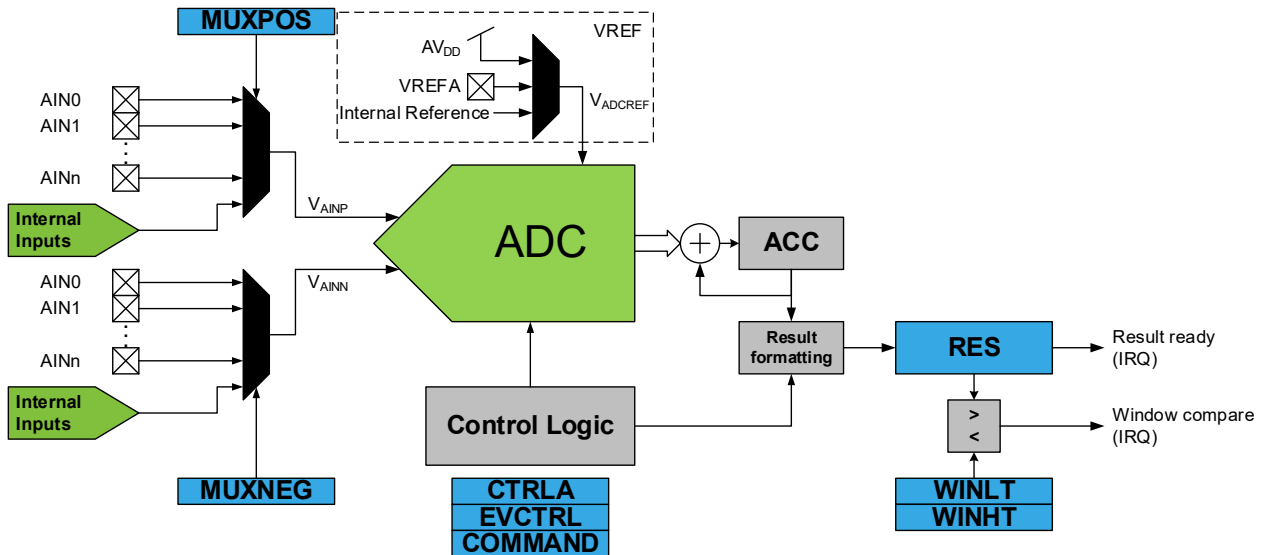
1.1 Analog-to-Digital Converter Module Overview

The device used to develop the application described in this document is AVR128DA48. It is equipped with a 12-bit resolution ADC module that provides both Single-Ended and Differential modes.

The ADC input signal is fed through a Sample-and-Hold circuit which ensures that the input voltage to the ADC is held at a constant level during sampling. The ADC voltage reference is configured in the VREF peripheral.

The block diagram of the ADC module is presented below.

Figure 1-1. ADC Block Diagram



To configure a module, the respective module registers must be used. The *Register Summary* and *Register Description* data sheet chapters provide a list of all registers of a module and describe the functionality of all the bits and bit fields of the module registers.

In this application example, the ADC will be configured in Differential mode. A differential ADC measures the voltage difference between two inputs. This can be essential in certain applications as some measurement concepts require two output signals, instead of one, to quantify the physical property of interest. Sensors that implement such concepts typically provide their output value as the voltage difference between two signals, also known as a differential signal. Other sensors might provide a differential output for added robustness even though the measurement itself generates a single-ended signal.

When connecting a differential analog sensor to an MCU, one of the signals in the differential pair is defined as the positive input, while the other is defined as the negative input. The value of the differential signal is the voltage of the positive input referenced to the negative input. The positive and negative designation of each signal determines the polarity of the differential signal, defining it as positive when the positive input is larger than the negative input, and negative if the negative input is larger than the positive input. The conversion result is given by the following equation:

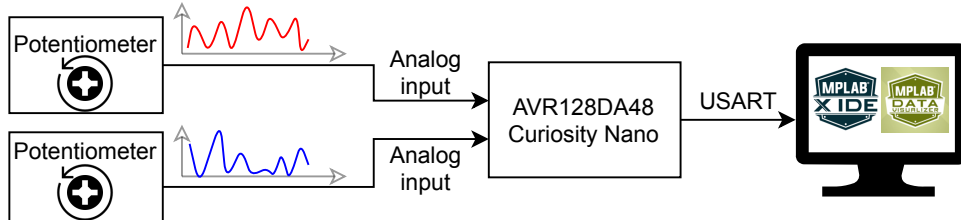
$$ADC_{result} = \frac{V_{AINP} - V_{AINN}}{V_{REF}} \times 2048 \in [-2048, 2047]$$

Where V_{AINP} and V_{AINN} are the positive and negative ADC inputs, and V_{REF} is the selected ADC voltage reference. The data format for differential conversions is two's complement with sign extension.

1.2 Application Overview

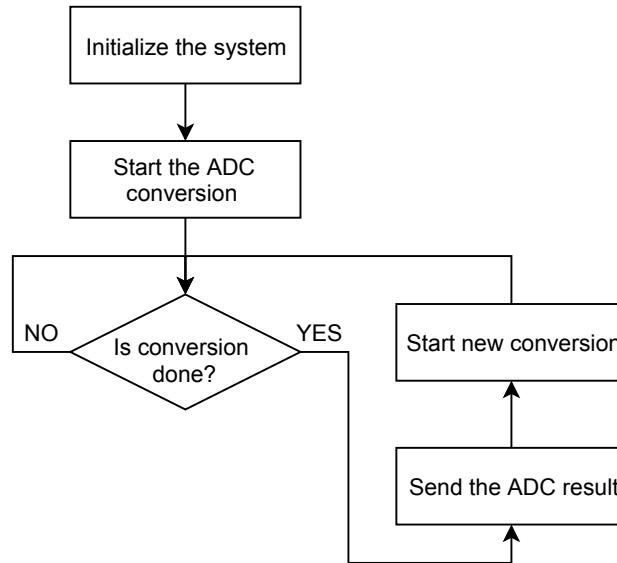
The hardware setup needed to develop and test this application is described in [Figure 1-2](#). Two potentiometers are used to provide analog input signals to the AVR128DA48 device, on the Curiosity Nano development board. The ADC will convert the voltage difference between the input signals. The result will be sent through USART and it will be plotted using the MPLAB Data Visualizer plug-in.

Figure 1-2. Hardware Setup



The software application designed for this training will implement the following software diagram. After initializing the system, the free-running ADC conversions will be started. Then, using an infinite loop, the result status will be continuously read and transmitted through USART to be displayed using the graphical interface.

Figure 1-3. Software Diagram



2. Analog-to-Digital Converter Training

2.1 Icon Identifiers

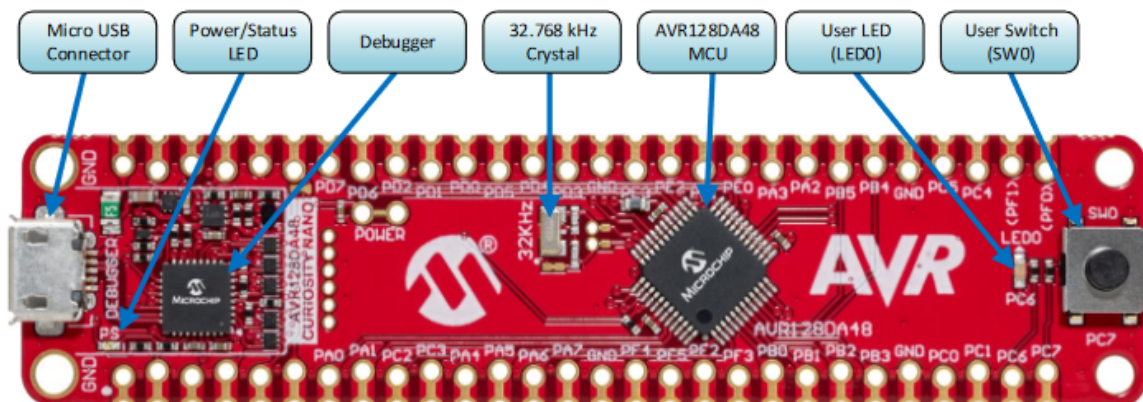
This subsection provides the icons used to guide the user through the training, and their meaning. The following icons will be used.

	Info:	This icon will be used to emphasize useful information.
	To do:	This icon will be used to show there is a task the user has to complete: To configure modules settings or to implement code.
	Result:	This icon will mark the solution to a task.

2.2 Hardware Setup

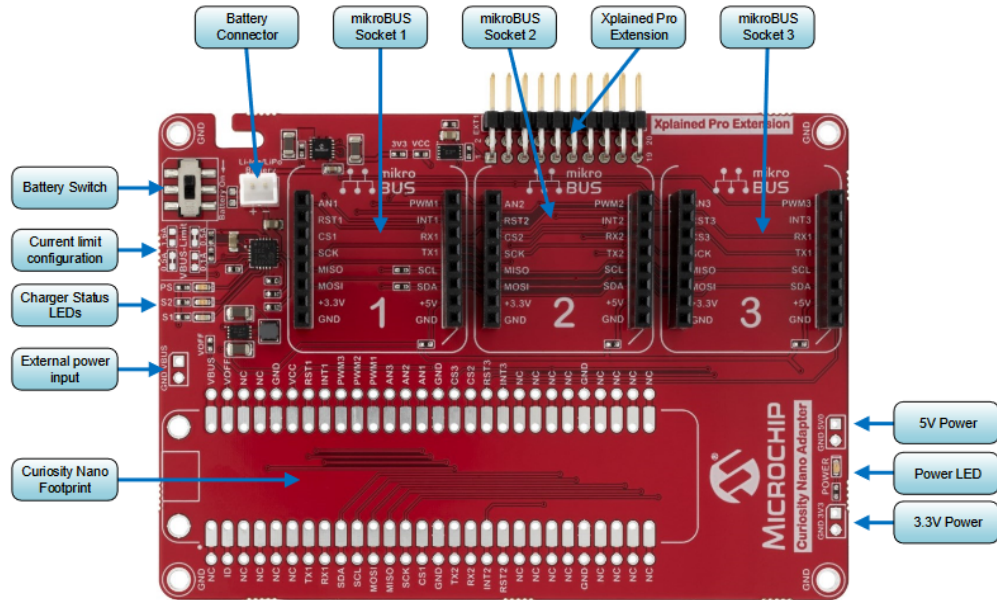
The hardware used to develop the provided training materials consists in a Curiosity Nano development board for the AVR128DA48 device, presented in [Figure 2-1](#).

Figure 2-1. Curiosity Nano Board



To easily integrate other components useful for this application, such as the POT Click board and the POT 2 Click board, the Curiosity Nano Adapter board can be used. It is presented in [Figure 2-2](#).

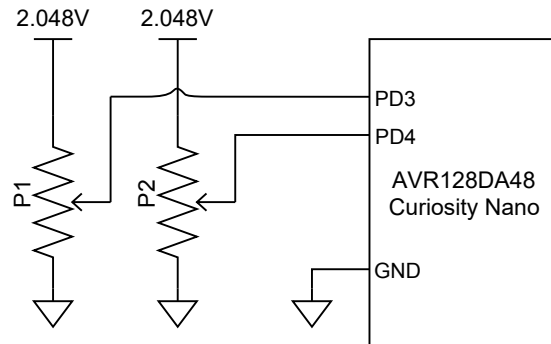
Figure 2-2. Curiosity Nano Adapter Board



Note: Two simple potentiometers are used to build the demo prototype. To easily integrate the potentiometers, the MIKROE POT and POT 2 Click boards can be used, along with the Curiosity Nano Adapter board.

The potentiometers are connected to the Curiosity Nano board, as presented in [Figure 2-3](#).

Figure 2-3. Schematic for Connecting the Potentiometers



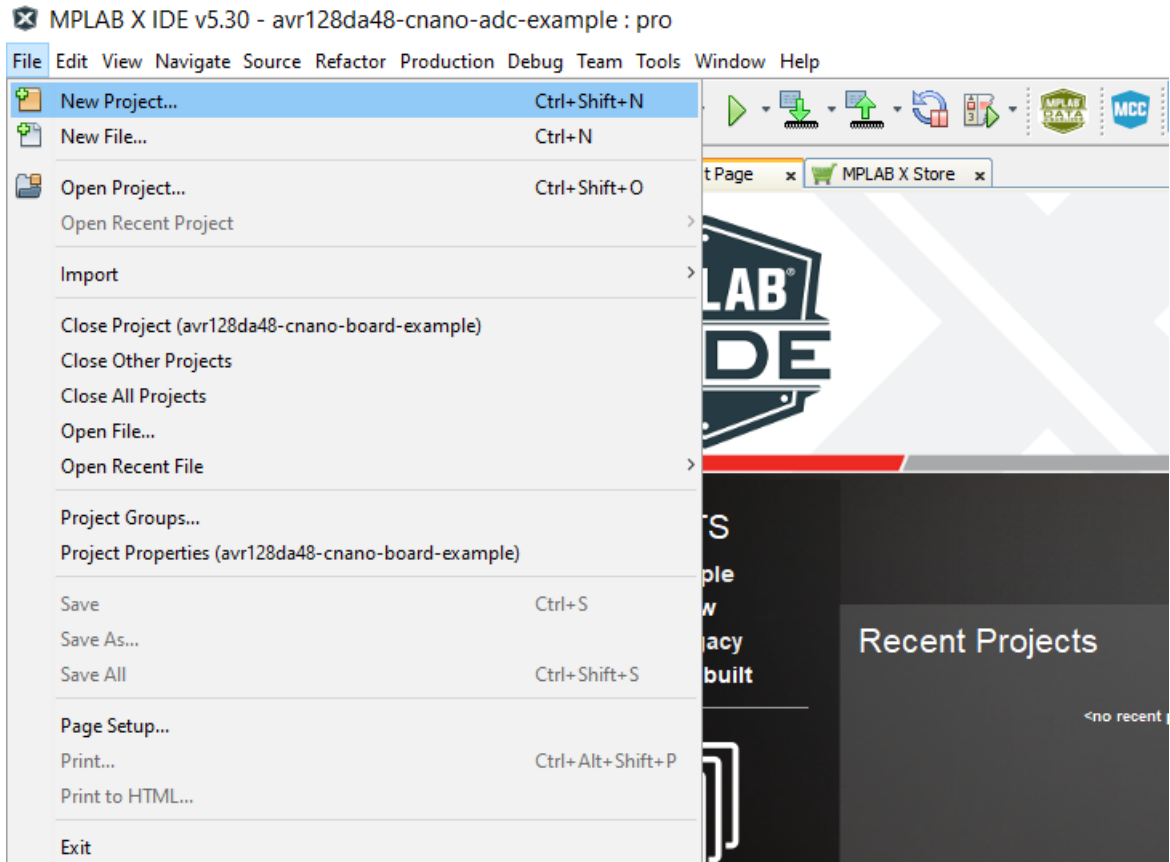
2.3 Get Familiar with the Software Environment

This section helps the user to get familiar with creating a new project, describes the necessary plug-ins, and provides the steps to install them.

2.3.1 Create a New Project Using MPLAB® X IDE

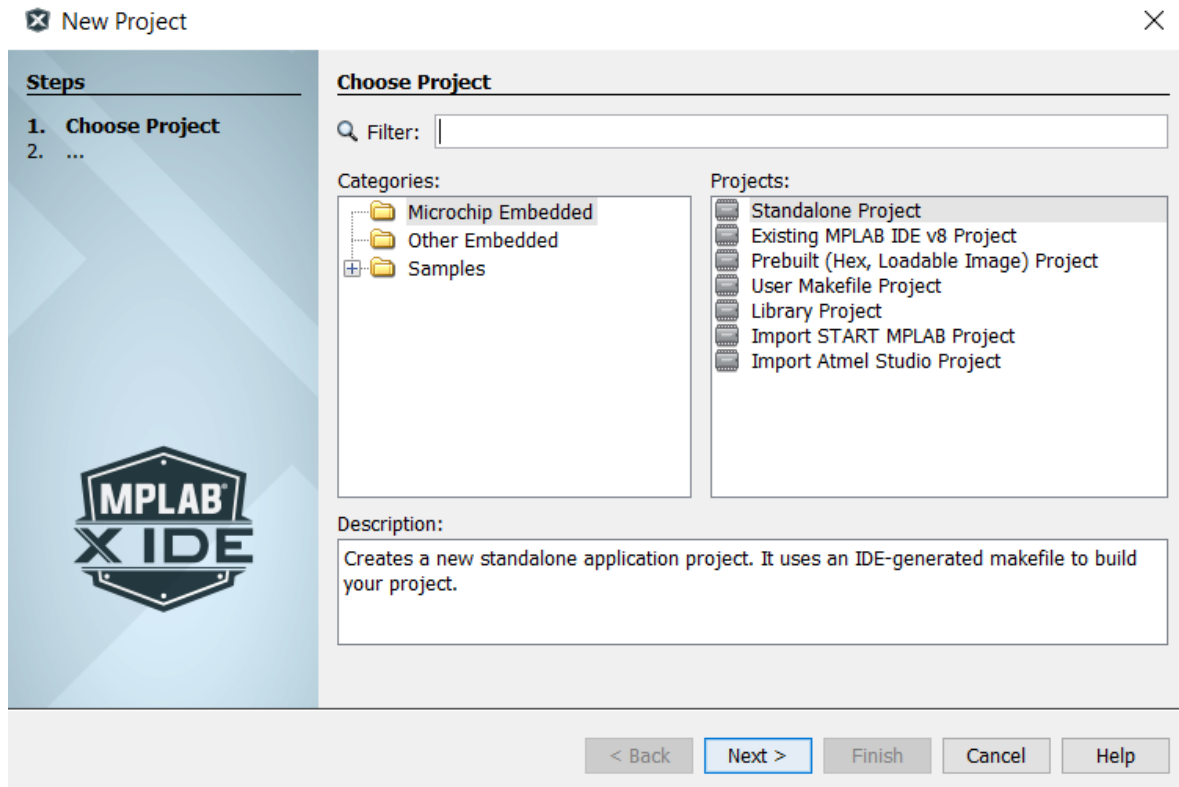
1. To create a new project using the MPLAB X IDE, go to *File* → *New Project...* and the New Project wizard will appear. This step is described in [Figure 2-4](#).

Figure 2-4. Create New Project



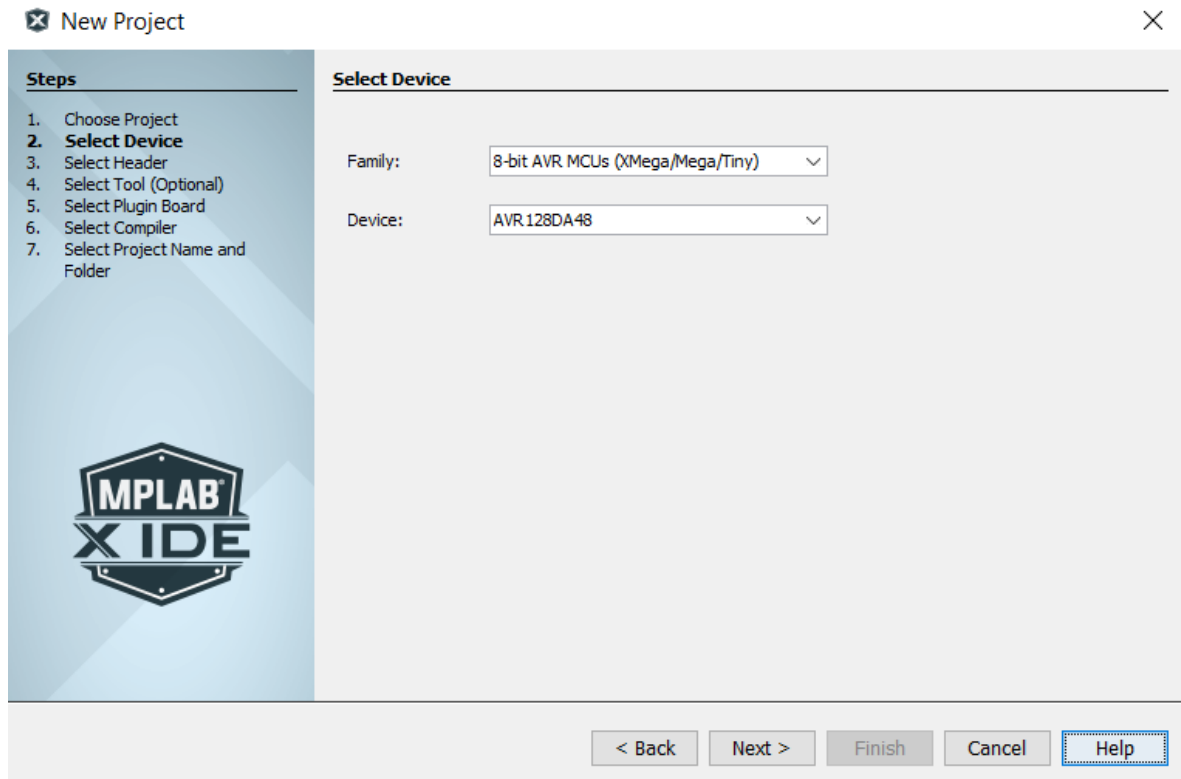
2. In the Choose Project step, select **Microchip Embedded** from Categories, and **Standalone Project** from Projects, then click **Next**. This will create a new stand-alone application project. This step is presented in [Figure 2-5](#).

Figure 2-5. Choose Project



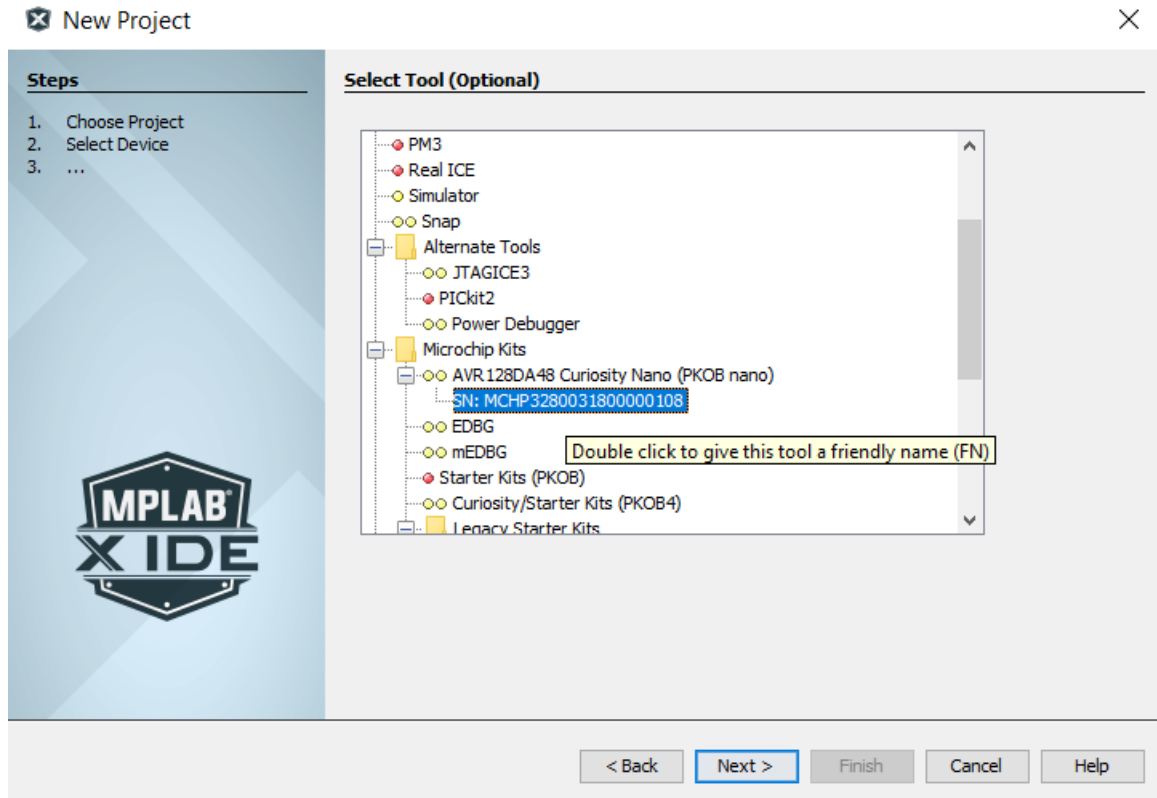
- The next step is to select the device for programming. The application presented in this document is implemented using the AVR128DA48 device from the 8-bit AVR MCUs family. After choosing this device, click **Next**.

Figure 2-6. Select Device



- Then, in the Select Tool step, from *Microchip Kits* → *AVR128DA48 Curiosity Nano (PKOB nano)* select the desired tool (SN:MCHP...). This is an optional step. If the tool is not available yet, or it is not connected yet, the user may select the **Simulator** tool from *Hardware Tools*, or leave it unselected until it is required.

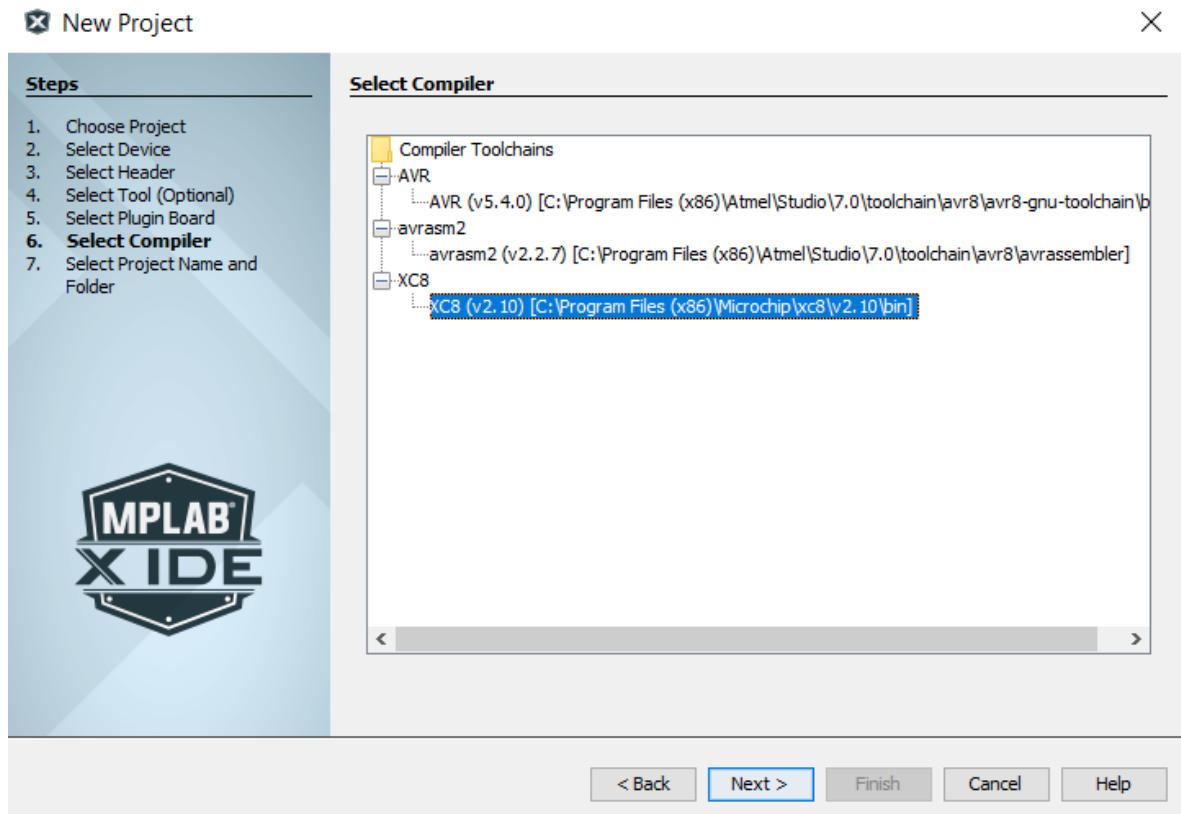
Figure 2-7. Select Tool



Info: The user can also double click on the tool name to give it a friendly name (FN).

5. The desired version for the XC8 compiler is chosen for this project, as presented in [Figure 2-8](#).

Figure 2-8. Select Compiler



- The final step is Select Project Name and Folder. For this example, the project name will be avr128da48-cnano-adc-diff. During this step, the user can also configure the project location, the project folder, and the encoding.

Figure 2-9. Select Project Name and Folder

The screenshot shows the 'New Project' dialog box in MPLAB X IDE. The 'Steps' pane on the left lists seven steps, with step 7, 'Select Project Name and Folder', highlighted. The main area is titled 'Select Project Name and Folder' and contains the following fields and options:

- Project Name:** avr128da48-cnano-adc-diff
- Project Location:** C:\Users\AVRist\MPLABXProjects (with a 'Browse...' button)
- Project Folder:** MPLABXProjects\avr128da48-cnano-adc-diff.X
- Overwrite existing project.
- Also delete sources.
- Set as main project
- Use project location as the project folder
- Encoding:** ISO-8859-1 (dropdown menu)

At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish' (highlighted), 'Cancel', and 'Help'.

After completing all the required steps, a new and empty project will appear in the Projects window. All the necessary files will be introduced in the project using the MCC plug-in.

2.3.2 MPLAB® Code Configurator (MCC)

The MCC will be used to develop this project. It is a graphical programming environment that generates easy-to-understand C-code to be inserted in the project. It provides an intuitive interface; it enables and configures a rich set of peripherals and functions specific to the desired application.

The plug-in can be installed following the steps provided on the [Install MPLAB® Code Configurator \(MCC\)](#) webpage.

After generating all the required source and header files, the user must complete the source code to obtain the desired functionality. After implementing the code, the application functionality must be tested. This will be accomplished using the MDV plug-in.

2.3.3 MPLAB® Data Visualizer (MDV)

MDV is a graphical run-time debugging tool available as an MPLAB plug-in or a stand-alone debugging tool. It graphically displays run-time variables and functions in an embedded application.

Data can be graphed as:

- A raw streaming 8-bit variable
- Multiple variables in a data streaming protocol

To install the MDV plug-in, the MCC install steps can be used, the only difference being the plug-in name. This tool will be used to display the ADC results after implementing the application.

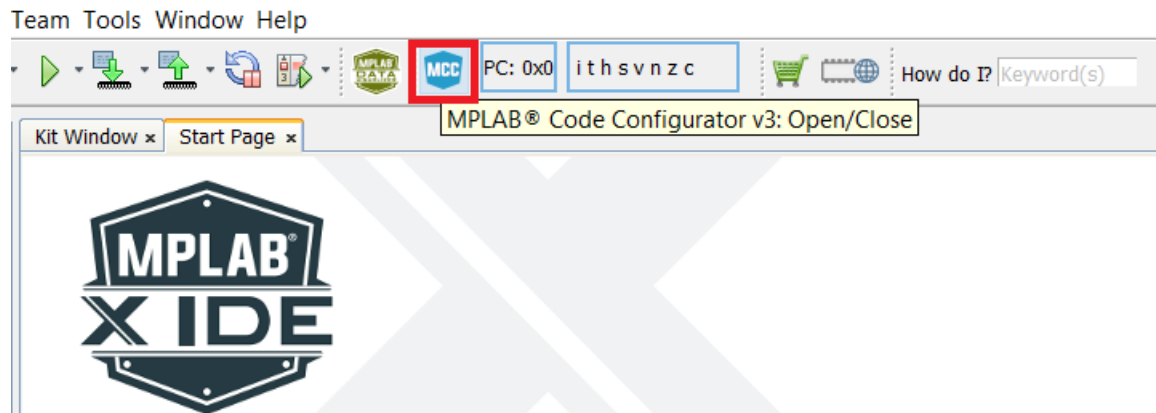
2.4 Assignment: Differential ADC

2.4.1 Initialize the Modules

This subsection will provide all the necessary steps to develop a simple application that uses the ADC in Differential mode, transmits the results through USART, and analyzes data with a graphical interface.

The MCC will be used to initialize all the desired modules. To open the plug-in, go to *Window* → *MPLAB Code Configurator* → *MPLAB Code Configurator Open/Close*, or click on the **MCC** icon, presented in [Figure 2-10](#).

Figure 2-10. Open/Close MCC



The first steps to develop this application using MCC are to provide the initialization settings and add the desired peripheral modules to the project. The user must initialize the system, by initializing all the necessary peripherals with the desired configurations. Then, using the generated source and header files, the user must implement the algorithm that will be executed by the MCU.

2.4.1.1 Configure the Device System

The first step to develop the ADC application is to configure the system clock. To find out which are the clock options, the user must consult the *CLKCTRL – Clock Controller* chapter, from the device data sheet.

For example, in this application, the chosen main clock frequency is 2 MHz.



To do: Using MCC, configure the internal high-frequency oscillator as the clock source. The frequency of the oscillator must be 4 MHz. The main clock frequency must be 2 MHz.



Result: Open the MCC and go to *Resource Management [MCC] → Project Resources → System → System Module*. From the **Easy Setup** tab, the following configuration must be done:

- Clock Control
 - Clock Source: Internal Oscillator
 - Internal Oscillator Frequency: 1-32 MHz internal oscillator
 - Oscillator Frequency Options: 4 MHz system clock (default)
 - Prescaler Enable: checked
 - Prescaler: 2X

The settings are described in [Figure 2-11](#).

Figure 2-11. MCC Clock Control Configuration

Property	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 Configure the ADC

For the application presented in this document, the ADC will be configured in Differential mode. To configure the initial settings of the module using MCC, the user must add this module to the Project Resources.



To do: Add the ADC peripheral module to the Project Resources, in MCC.



Result: Go to *Resource Management [MCC] → Device Resources → Peripherals → ADC* and add the ADC peripheral module by clicking the green + sign. The module will appear in the Project Resources window.

The next step is to configure the ADC module. The ADC will be running in Differential mode, and it will convert acquired samples continuously (Free-Running mode). The resolution will be 12 bits, the prescaler value will be 4, and the differential inputs will be PD3 (positive input) and PD4 (negative input).



To do:

- Enable the ADC module
- Enable the Differential mode conversion, Free-Running
- Set the result resolution to 12-bit
- Configure the ADC prescaler to obtain an ADC frequency of 500 kHz
- Configure the ADC input pins: ADC input pin 4 as negative input; ADC input pin 3 as positive input



Result: Go to *Resource Management [MCC] → Project Resources → Peripherals* and select **ADC0**. From the **Easy Setup** tab, the following configurations must be done:

- Software Settings:
 - Result Selection: 12-bit mode
 - Differential Mode Conversion: enabled
 - Left Adjust Result: unchecked
- Hardware Settings:
 - Enable ADC: checked

The settings are described in [Figure 2-12](#) and [Figure 2-13](#).

Figure 2-12. ADC0 Software Settings

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

Figure 2-13. ADC0 Hardware Settings

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

Additionally, the user must configure the ADC registers. To know what settings are available in which register and to find the available configurations of a register's bits and bit fields, the user must consult the device data sheet.

The Free-Running mode option is available in the Control A (CTRLA) register of the ADC module. By writing to the CTRLA register, the user can enable/disable the Running in Standby mode, select the conversion mode, set the result adjustment, select the resolution, and enable/disable the peripheral. The clock frequency for this peripheral must be configured using the Prescaler (PRESC) bit field in Control C (CTRLC) register.

To use the ADC in Differential mode, two analog inputs are needed. The positive input must be configured using the MUX Selection for Positive ADC Input (MUXPOS) register, and similarly, the negative input must be configured using the MUX Selection for Negative ADC Input (MUXNEG) register. The respective pins must also be configured as analog inputs, with digital buffers disabled.

The configuration of the registers can be done using MCC, in the **Registers** tab.

- Register: CTRLA
 - FREERUN: enabled

Figure 2-14. ADC0 CTRLA Register

Register: CTRLA	Value
CONV_MODE	enabled
ENABLE	enabled
FREERUN	enabled
LEFT_ADJ	disabled
RESSEL	12-bit mode
RUNSTBY	disabled

- Register: CTRLC
 - PRESC: CLK_PER divided by 4

Figure 2-15. ADC0 CTRLC Register

Register: CTRLC	Value
PRESC	CLK_PER divided by 4

- Register: MUXNEG
 - MUXNEG: ADC input pin 4
- Register: MUXPOS
 - MUXPOS: ADC input pin 3

Figure 2-16. ADC0 MUXNEG and MUXPOS Registers

Register: MUXNEG	Value
MUXNEG	ADC input pin 4

Register: MUXPOS	Value
MUXPOS	ADC input pin 3

2.4.1.3 Configure the VREF

The ADC voltage reference can be configured by writing to the ADC0 Reference (ADC0REF) register. It can be configured using MCC.



To do: Using MCC, add the VREF module to the Project Resources and configure the ADC voltage reference to be 2.048V.



Result: Go to *Resource Management [MCC] → Device Resources → Peripherals → VREF* and add the VREF peripheral module by clicking the green + sign. The module will appear in the Project Resources window.

Go to *Resource Management [MCC] → Project Resources → Peripherals* and select **VREF**. From the **Easy Setup** tab, the following configuration must be done:

- Hardware Settings
 - ADC Voltage Reference: Internal 2.048V reference

Figure 2-17. VREF ADC Voltage Reference



2.4.1.4 Configure the USART

After converting the received analog data using the ADC, the data can be sent further to be analyzed by the user. To transmit the data to the developing computer host, the USART peripheral module will be used. On the AVR128DA48 Curiosity Nano board, the USART1 RX (receiving) and TX (transmitting) pins are connected directly to the debugger pins, so the user will be able to send data to the computer without additional wires. Therefore, the USART peripheral module used in this application will be USART1.



To do: Add the USART1 module to the project using MCC.



Result: Go to *Resource Management [MCC] → Device Resources → Peripherals → USART* and add the USART1 peripheral module by clicking the green + sign. The module will appear in the Project Resources window.

After introducing the module to the project, some initial configurations must be done.



To do: Configure the USART1 to run in Asynchronous mode, with the baud rate of 9600, with no parity, and with 1 stop bit. The character size must be of 8 bits.



Info: There is no need for the USART RX to be enabled: The device does not need data from the computer. Only the USART TX needs to be enabled.



Result: Go to *Resource Management [MCC] → Project Resources → Pin Module* and do the configurations presented in [Figure 2-20](#).

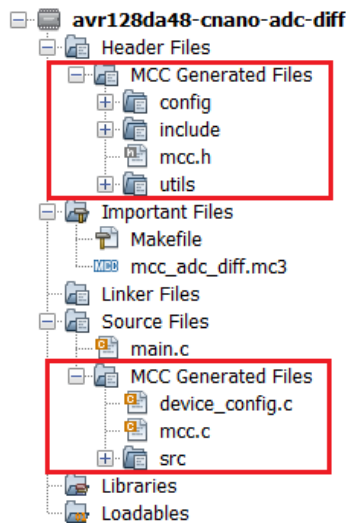
Figure 2-20. Pin Module

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 Generated Code Overview

To generate the code designed with MCC, click the **Generate** button. Then, to continue the application development, close the MCC. The MCC generated files can be seen in the **Projects** tab, under the created project.

Figure 2-21. MCC Generated Files



The generated files are included in the main source file and the system is initialized. The generated code for `main.c` is presented below.

Example 2-1. Code Listing 1 – Main Generated File

```
#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
 */
```


Example 2-4. Code Listing 4 – Reading and Transmitting the ADC Result

```
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);
    }
}
```



To do: `START_TOKEN` and `END_TOKEN` are user defined macros that must be also defined in the `main.c` file to be used in the main function, as presented below.

```
#define START_TOKEN 0x03    /* Start Frame Token */
#define END_TOKEN 0xFC     /* End Frame Token */
```



[View Code Example on GitHub](#)

Click to browse repositories

2.4.4 Application Testing: Debugging

One way to test the functionality of an application is by going step by step through it and check that all the implemented instructions are providing the expected result.



To do: Enable a breakpoint on the `SYSTEM_Initialize()`; function call in the main function. Enter the Debug mode.



Result: Create a new Line breakpoint by clicking on the editor gutter next to the file line. The respective line will appear as presented below.

Figure 2-23. Enable Breakpoint

```
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 }
```

To visualize the content of the registers, go to *Window → Debugging* and select **IO View**. This will open the IO View window. To start debugging, go to *Debug* and select **Debug Main Project**. The program execution will stop at the breakpoint line.

After clicking the **Step Over** button, the user can check, for example, if the ADC peripheral was initialized as desired, by looking into the registers in the IO View window. The registers from the IO View window are presented in [Figure 2-24](#).



To do: Check if all the configurations are done as desired.

Figure 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 (CTRLF)	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	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	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
	CTRLF	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	6 5 4 3 2 1 0
	MUXPOS	0x0608	0x03	3	6 5 4 3 2 1 0
	MUXPOS		3	3	6 5 4 3 2 1 0

By going step by step through the application, the user can see:

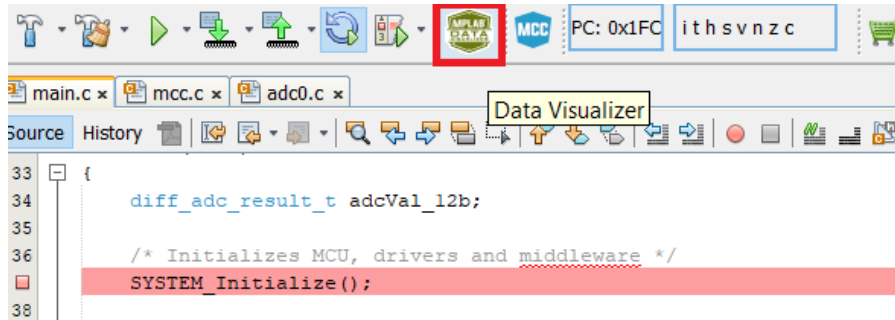
- The conversion result in the ADC Result (RES) register
- The data that needs to be transmitted through USART – in the TXDATAL register
- The value of the adcVal variable, using the Variables window

2.4.5 Application Testing: Data Visualizer

The application can be easily tested by using the MPLAB Data Visualizer. To plot the ADC results transmitted through USART, the following steps must be implemented:

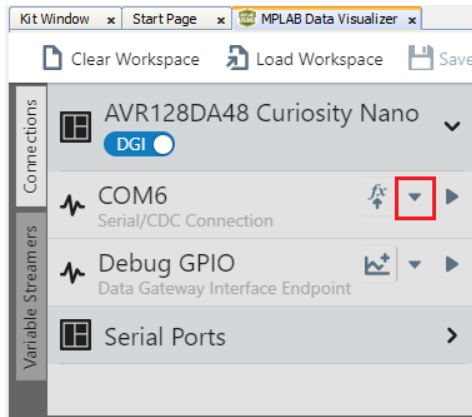
1. To open the Data Visualizer plug-in, click on the plug-in icon, as presented in [Figure 2-25](#).

Figure 2-25. Open Data Visualizer



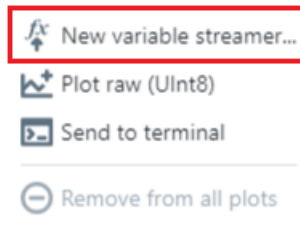
2. From the **Connections** tab, select the **Curiosity Nano communication port (COMn)** drop-down list, as presented in [Figure 2-26](#).

Figure 2-26. COMn Port – Display Drop-Down List



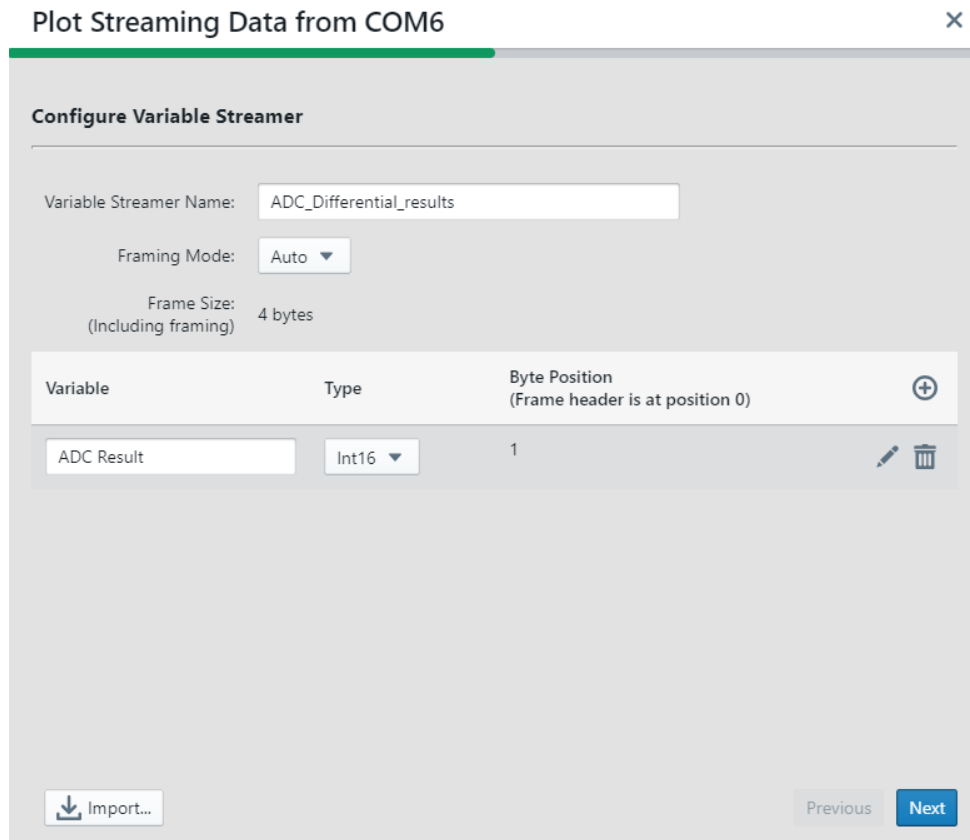
- From the drop-down list, select **New variable streamer...**, as presented in [Figure 2-27](#).

Figure 2-27. COMn Port – Select from Drop-Down List



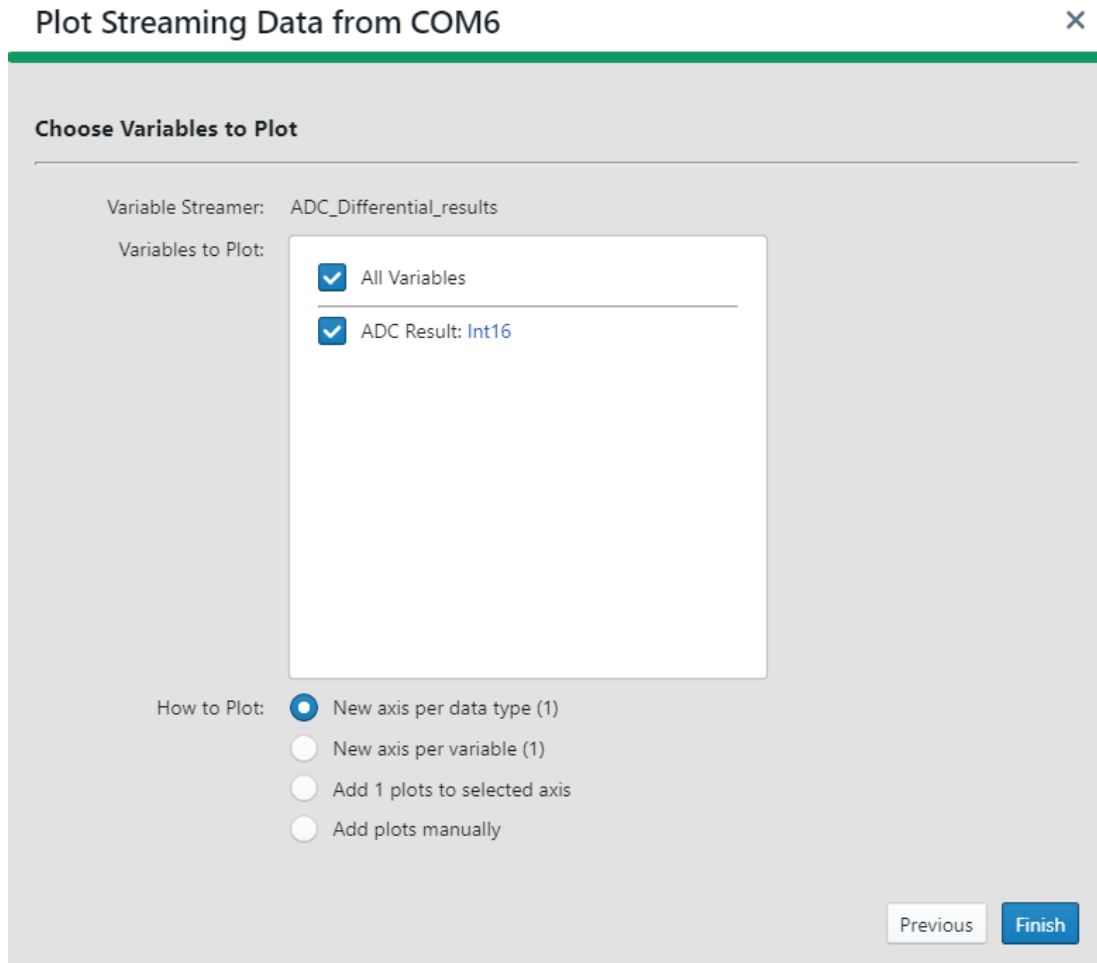
- Select a **Variable Streamer Name**, add the variables that will be received, and click **Next**.

Figure 2-28. Configure Variable Streamer



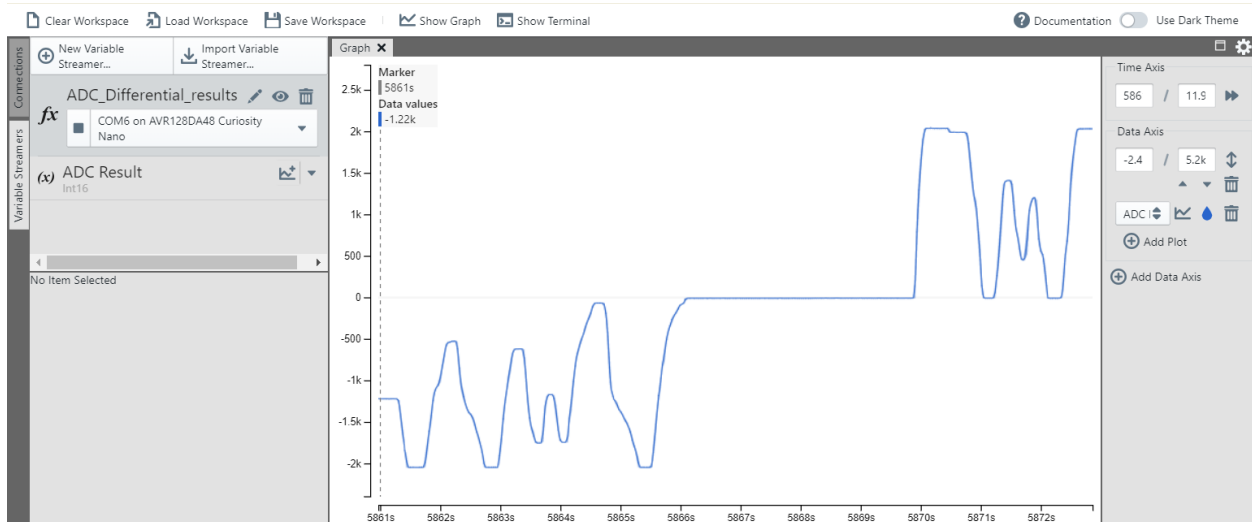
5. Choose the variables to be plotted by selecting the desired variables, and select **New axis per data type (1)** for how to plot the data. Then, click **Finish**.

Figure 2-29. Choose Variables to Plot



After randomly turning both potentiometers, the plot was obtained. The ADC results are presented in [Figure 2-30](#).

Figure 2-30. Data Visualizer ADC Results Plot



3. Conclusion

After going through the training provided by this document, the user can understand the basic features of the ADC, use the software tools needed to develop an embedded application, and independently develop a basic application using the ADC module. Furthermore, the user will understand how to configure the ADC to convert data from a differential input, how to continuously convert the data, and how to interpret the results. This training also provides the necessary steps to debug the application and visualize the results.

4. References

1. [Data Visualizer Software User's Guide](#)
2. [MPLAB® Code Configurator \(MCC\)](#)
3. [How to add a library in MPLAB Code Configurator \(MCC\)](#)
4. [Microchip webpage to download the MCC latest libraries](#)
5. [Visual Debugging with the MPLAB Data Visualizer](#)
6. [Differential and Single-Ended ADC White Paper](#)
7. [AVR128DA48 Curiosity Nano Evaluation Kit](#)
8. [Curiosity Nano Base for Click boards™](#)

5. Revision History

Document Revision	Date	Comments
A	08/2020	Initial document release

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

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

- 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 and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- 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. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

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.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Klear, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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 logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

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, KlearNet, KlearNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, 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., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

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

© 2020, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-6604-8

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
<p>Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Tel: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com</p> <p>Atlanta Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455</p> <p>Austin, TX Tel: 512-257-3370</p> <p>Boston Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088</p> <p>Chicago Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075</p> <p>Dallas Addison, TX Tel: 972-818-7423 Fax: 972-818-2924</p> <p>Detroit Novi, MI Tel: 248-848-4000</p> <p>Houston, TX Tel: 281-894-5983</p> <p>Indianapolis Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380</p> <p>Los Angeles Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800</p> <p>Raleigh, NC Tel: 919-844-7510</p> <p>New York, NY Tel: 631-435-6000</p> <p>San Jose, CA Tel: 408-735-9110 Tel: 408-436-4270</p> <p>Canada - Toronto Tel: 905-695-1980 Fax: 905-695-2078</p>	<p>Australia - Sydney Tel: 61-2-9868-6733</p> <p>China - Beijing Tel: 86-10-8569-7000</p> <p>China - Chengdu Tel: 86-28-8665-5511</p> <p>China - Chongqing Tel: 86-23-8980-9588</p> <p>China - Dongguan Tel: 86-769-8702-9880</p> <p>China - Guangzhou Tel: 86-20-8755-8029</p> <p>China - Hangzhou Tel: 86-571-8792-8115</p> <p>China - Hong Kong SAR Tel: 852-2943-5100</p> <p>China - Nanjing Tel: 86-25-8473-2460</p> <p>China - Qingdao Tel: 86-532-8502-7355</p> <p>China - Shanghai Tel: 86-21-3326-8000</p> <p>China - Shenyang Tel: 86-24-2334-2829</p> <p>China - Shenzhen Tel: 86-755-8864-2200</p> <p>China - Suzhou Tel: 86-186-6233-1526</p> <p>China - Wuhan Tel: 86-27-5980-5300</p> <p>China - Xian Tel: 86-29-8833-7252</p> <p>China - Xiamen Tel: 86-592-2388138</p> <p>China - Zhuhai Tel: 86-756-3210040</p>	<p>India - Bangalore Tel: 91-80-3090-4444</p> <p>India - New Delhi Tel: 91-11-4160-8631</p> <p>India - Pune Tel: 91-20-4121-0141</p> <p>Japan - Osaka Tel: 81-6-6152-7160</p> <p>Japan - Tokyo Tel: 81-3-6880-3770</p> <p>Korea - Daegu Tel: 82-53-744-4301</p> <p>Korea - Seoul Tel: 82-2-554-7200</p> <p>Malaysia - Kuala Lumpur Tel: 60-3-7651-7906</p> <p>Malaysia - Penang Tel: 60-4-227-8870</p> <p>Philippines - Manila Tel: 63-2-634-9065</p> <p>Singapore Tel: 65-6334-8870</p> <p>Taiwan - Hsin Chu Tel: 886-3-577-8366</p> <p>Taiwan - Kaohsiung Tel: 886-7-213-7830</p> <p>Taiwan - Taipei Tel: 886-2-2508-8600</p> <p>Thailand - Bangkok Tel: 66-2-694-1351</p> <p>Vietnam - Ho Chi Minh Tel: 84-28-5448-2100</p>	<p>Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393</p> <p>Denmark - Copenhagen Tel: 45-4485-5910 Fax: 45-4485-2829</p> <p>Finland - Espoo Tel: 358-9-4520-820</p> <p>France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79</p> <p>Germany - Garching Tel: 49-8931-9700</p> <p>Germany - Haan Tel: 49-2129-3766400</p> <p>Germany - Heilbronn Tel: 49-7131-72400</p> <p>Germany - Karlsruhe Tel: 49-721-625370</p> <p>Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44</p> <p>Germany - Rosenheim Tel: 49-8031-354-560</p> <p>Israel - Ra'anana Tel: 972-9-744-7705</p> <p>Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781</p> <p>Italy - Padova Tel: 39-049-7625286</p> <p>Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340</p> <p>Norway - Trondheim Tel: 47-72884388</p> <p>Poland - Warsaw Tel: 48-22-3325737</p> <p>Romania - Bucharest Tel: 40-21-407-87-50</p> <p>Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91</p> <p>Sweden - Gothenberg Tel: 46-31-704-60-40</p> <p>Sweden - Stockholm Tel: 46-8-5090-4654</p> <p>UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820</p>