ARBOR DIO

User's Guide

Version 2.0.1

Revision History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Descriptions** |
| 1.0.0 | 2020/08/14 | Initial release |
| 2.0.0 | 2023/10/20 | Modify architecture for all in one case |
| 2.0.1 | 2025/02/26 | Add screenshot for install and DIO configuration tool. |

Contents

[1. Setting Environment 3](#_Toc191556493)

[2. SDK Index 3](#_Toc191556494)

[3. Install 3](#_Toc191556495)

[4. DIO Library 6](#_Toc191556496)

[5. Function 6](#_Toc191556497)

[5.1 DIO Version 6](#_Toc191556498)

[5.2 DIO Open 6](#_Toc191556499)

[5.3 Initial Digital I/O Mode1 6](#_Toc191556500)

[5.4 Initial Digital I/O Mode2 6](#_Toc191556501)

[5.5 Set Digital Output Data1 7](#_Toc191556502)

[5.6 Set Digital Output Data2 7](#_Toc191556503)

[5.7 Get Digital Input Status1 7](#_Toc191556504)

[5.8 Get Digital Input Status2 7](#_Toc191556505)

[5.9 DIO Pin Mode 7](#_Toc191556506)

[5.10 DIO Pin Output 7](#_Toc191556507)

[5.11 DIO Pin Input 7](#_Toc191556508)

[6. Sample Code and Demo Application 8](#_Toc191556509)

# 1. Setting Environment

|  |  |
| --- | --- |
| Operating System | Windows7 above |
| Required SW | .Net framework 4.5.1 |

# 2. SDK Index

/

├── Sample Code

├── dll

│ ├─ x64

│ └─ x86

├──doc

│ ├─ ARBOR\_DIO\_User\_Guide.pdf

│ └─ CHANGELOG.md

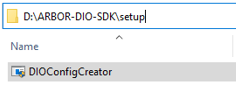
└─ setup

├─ ARBOR-DIO-Setup.exe

└─ DIOConfigCreator.exe

# 3. Install

**Step 1**: install ARBOR-DIO-Setup.exe

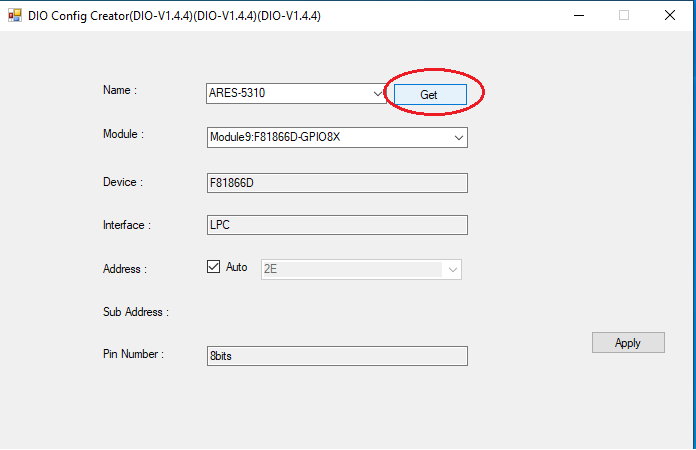


P1

**Step 2**: Run DIOConfigCreator.exe.

P2

**Step 3**: Click the "Get" button to the right of the "Product Name" field, and the app will read the system information to display the product name and refresh the "Module", "Device", "Interface", "Address", "Sub Address" and "Pin Number" fields.

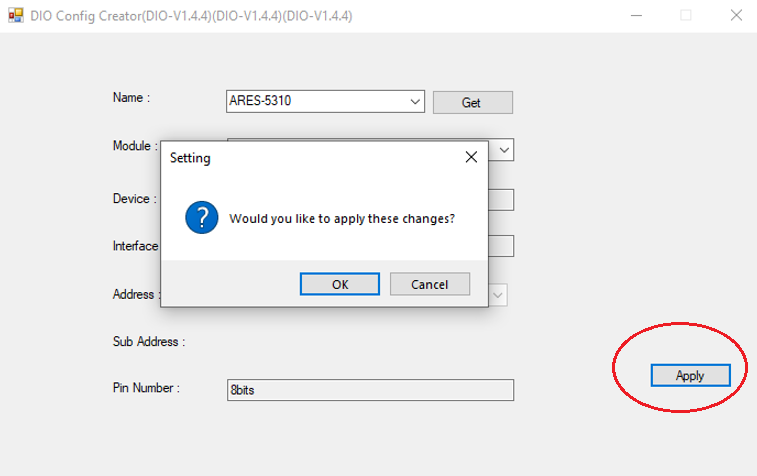


P3

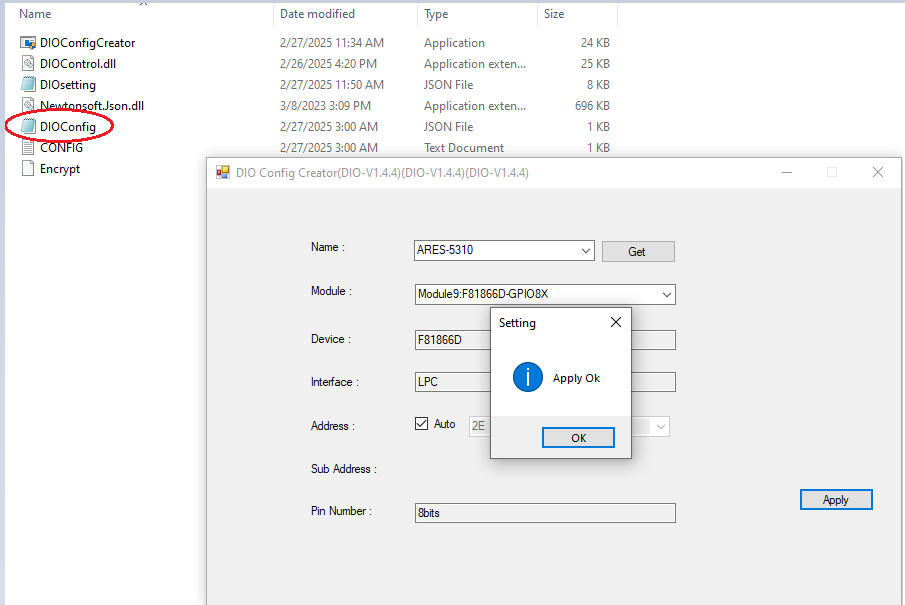
**Step 4**: Click the "Apply" button. The SDK configuration setup will completed.

**Note**: Besides using the "Get" button, you can also select a product using the combo box, where you can see the products supported by the SDK.

**Note**: The "Module" field will display the DIO Module number of the currently selected product, and this number corresponds to the classification in the sample code.



P4

****P5

# 4. DIO Library

This SDK provides 32/64-bit DLLs. Please place the DLLs and the user-developed application in the same directory.

# 5. Function

This section describes the APIs provided by the SDK. The supported APIs vary depending on the number of DIO pins in different products. For example, products with 8-pin DIO support DIO\_Data1() and DIO\_Status1(), while products with 16-pin DIO may also support DIO\_Data2() and DIO\_Status2(). For details on which products support which APIs, please refer to the sample code. Here, we only provide explanations for each API.

## 5.1 DIO Version

This function gets the version of the DLL file.

char\* DIO\_Ver()

Example:

char\* sVersion = DIO\_Ver();

## 5.2 DIO Open

This function enables the ARBOR-DIO functionality. The return value is true if the enable was successful.

bool DIO\_Open()

Example:

bool Result =DIO\_Open();

## 5.3 Initial Digital I/O Mode1

This function sets the input/output mode. The parameter "iMode" bit 0 ~ 7 mapping to pin 0 ~ 7. If the bit set to 1, then the pin is set to output mode.

void DIO\_Mode1(int iMode)

## 5.4 Initial Digital I/O Mode2

This function sets the input/output mode. The parameter "iMode" bit 0 ~ 7 mapping to pin 8 ~ 15. If the bit set to 1, then the pin is set to output mode.

void DIO\_Mode2(int iMode)

## 5.5 Set Digital Output Data1

This function sets the output level High/Low. The parameter "iValue" bit 0 ~ 7 mapping to pin 0 ~ 7. If the bit set to 1, then the pin is set to high level.

void DIO\_Data1(int iValue)

## 5.6 Set Digital Output Data2

This function sets the output level High/Low. The parameter "iValue" bit 0 ~ 7 mapping to pin 8 ~ 15. If the bit set to 1, then the pin is set to high level.

void DIO\_Data2 (int iValue)

## 5.7 Get Digital Input Status1

This function gets the input level High/Low. The return value bit 0 ~ 7 mapping to pin 0 ~ 7. If the bit is "1", then the pin is high level.

int DIO\_Status1()

## 5.8 Get Digital Input Status2

This function gets the input level High/Low. The return value bit 0 ~ 7 mapping to pin 8 ~ 15. If the bit is "1", then the pin is high level.

int DIO\_Status2()

## 5.9 DIO Pin Mode

This function sets the input/output mode of a single pin. The parameter "index" corresponds to the pin number. The parameter "bMode" sets the input/output mode of the pin, with "0" for input and "1" for output.

void Pin\_Mode(int iIndex, bool bMode)

## 5.10 DIO Pin Output

This function sets the output level of a single pin. The parameter "index" corresponds to the pin number. The parameter "bValue" sets the High/Low of the pin, with "0" for Low and "1" for High.

void Pin\_Output(int iIndex, bool bValue)

## 5.11 DIO Pin Input

This function gets the level of a single pin. The parameter "index" corresponds to the pin number. The return value is the level of the pin, with "0" for low level and "1" for high level.

int Pin\_Input(int iIndex)

# 6. Sample Code and Demo Application

This SDK provides C++/C# sample code. In the Sample Code folder, the code is organized according to different DIO modules. Users can use DIOConfigCreator.exe to check the DIO Module number.

Each DIO module supports different APIs, so users should refer to the examples in the Sample Code before developing their own applications.