IRIS SDK FOR WINDOWS

QuickStart Guide

Version 1.0.1

CONTENTS

INTRODUCTION

The IrisSDK for windows contains examples and source code for communication with an Orca Series Motor via MODBUS RTU protocol.

This SDK will demonstrate how to build a virtual windows device, which can both communicate with an Iris Dynamics motor, and be monitored via IrisControls4 windows application. It includes libraries and tutorials to allow development of custom firmware for motor position and force control as well as creating custom GUI interfaces.

Warning

Be aware that the shaft or motor will move during operation. This software will cause the motor to create forces and motion. Ensure the shaft and motor are mounted in a safe location and are not in danger of hitting people or property.

Connecting the Motor to the Windows Machine

For a basic setup, connect the motor's power and ground to an external 24-48 V power supply and attach an ethernet splitter to the ethernet cable of the Orca. Connect an RS485 USB-to-ethernet cable to input 1 of the splitter, and an RS422 USB-to-ethernet splitter to input 2 of the splitter. Plug both cables into USB ports of the pc.

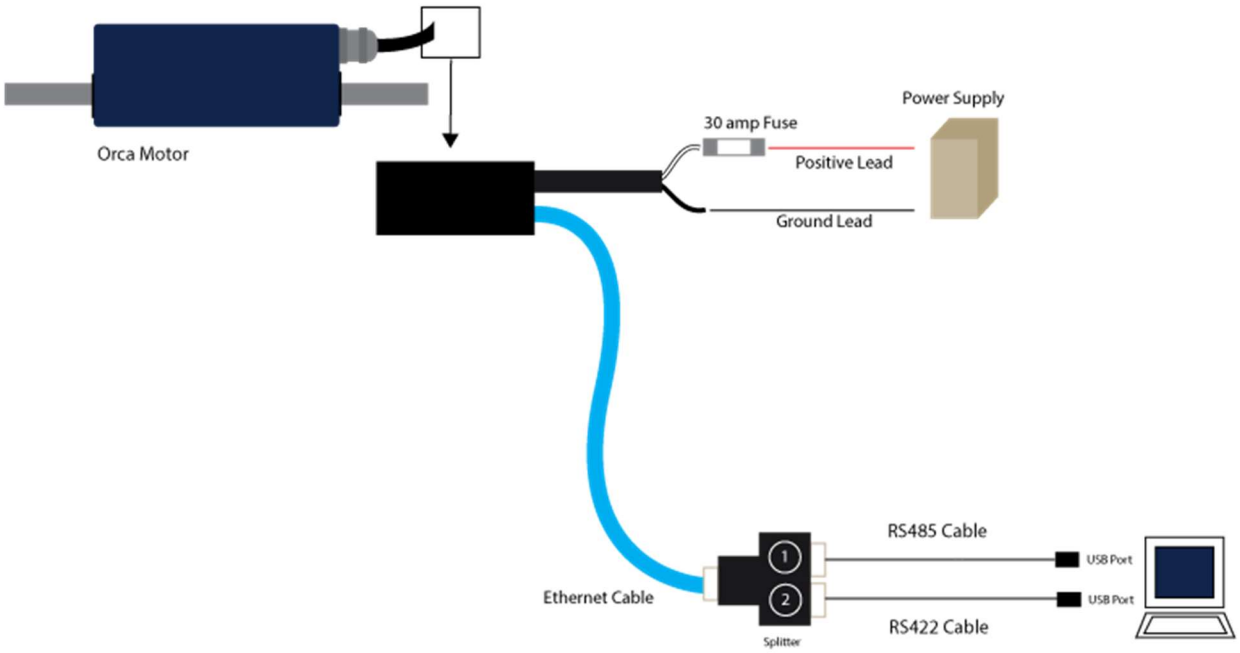


Figure 1: Connecting an Orca motor to a Windows PC

USB Connectors

In the setup above the RS485 USB-to-ethernet cable allows the motor to communicate with IrisControls4.This is useful for configuring your motor and debugging your applications (see Orca Motor Reference Manual for full details). The RS422 USB-to-ethernet cable allows the motor to communicate via MODBUS RTU. Both cables are included in the Orca Starter Kit.

Back-power

The motors in some circumstances will act like generators, this can result in current being fed backwards to power lugs. Typically, the level of back-power is low and power supplies or batteries can accommodate this. In situations where shaft or stator speeds get very high, some power supplies may go into protection.

DOWNLOADING THE SDK

Iris Dynamics' IrisSDK for Windows repository can be found on GitHub at the following link:

https://github.com/IrisDynamics/IrisSDK_for_Windows

The repository can either be downloaded as a zip by pressing the green <> Code button and clicking download ZIP. Alternatively the repository can be cloned through GIT Bash commands.

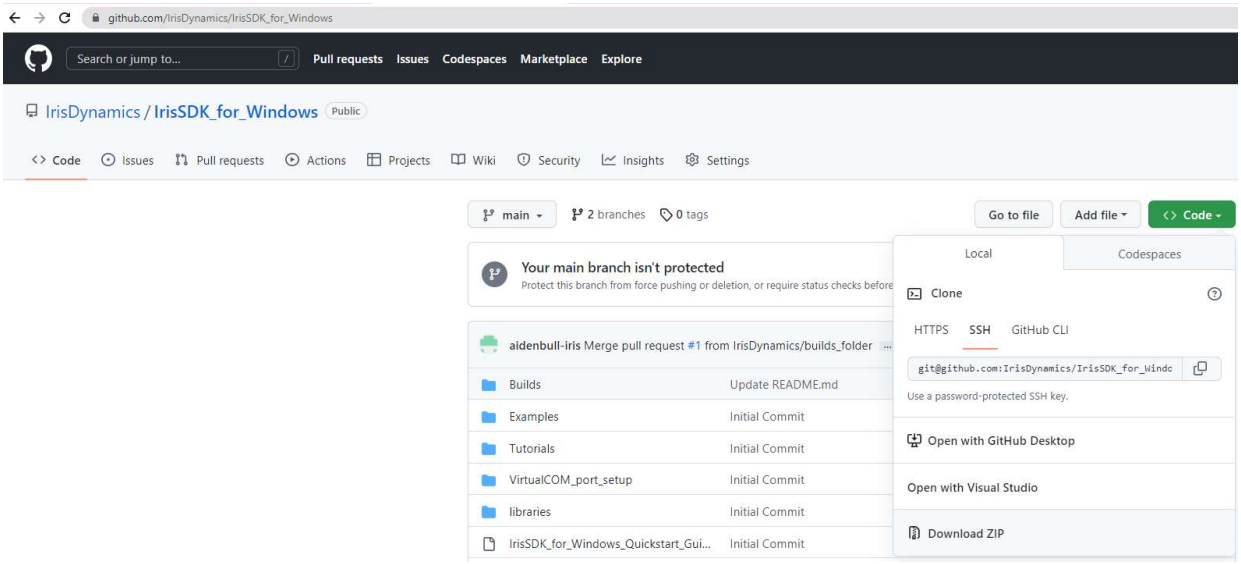


Figure 2: Accessing Git repository

RUNNING EXAMPLE PROJECTS

Motor Control GUI Example

This example requires no additional set up to run.

This using a modbus stream to establish a "connection

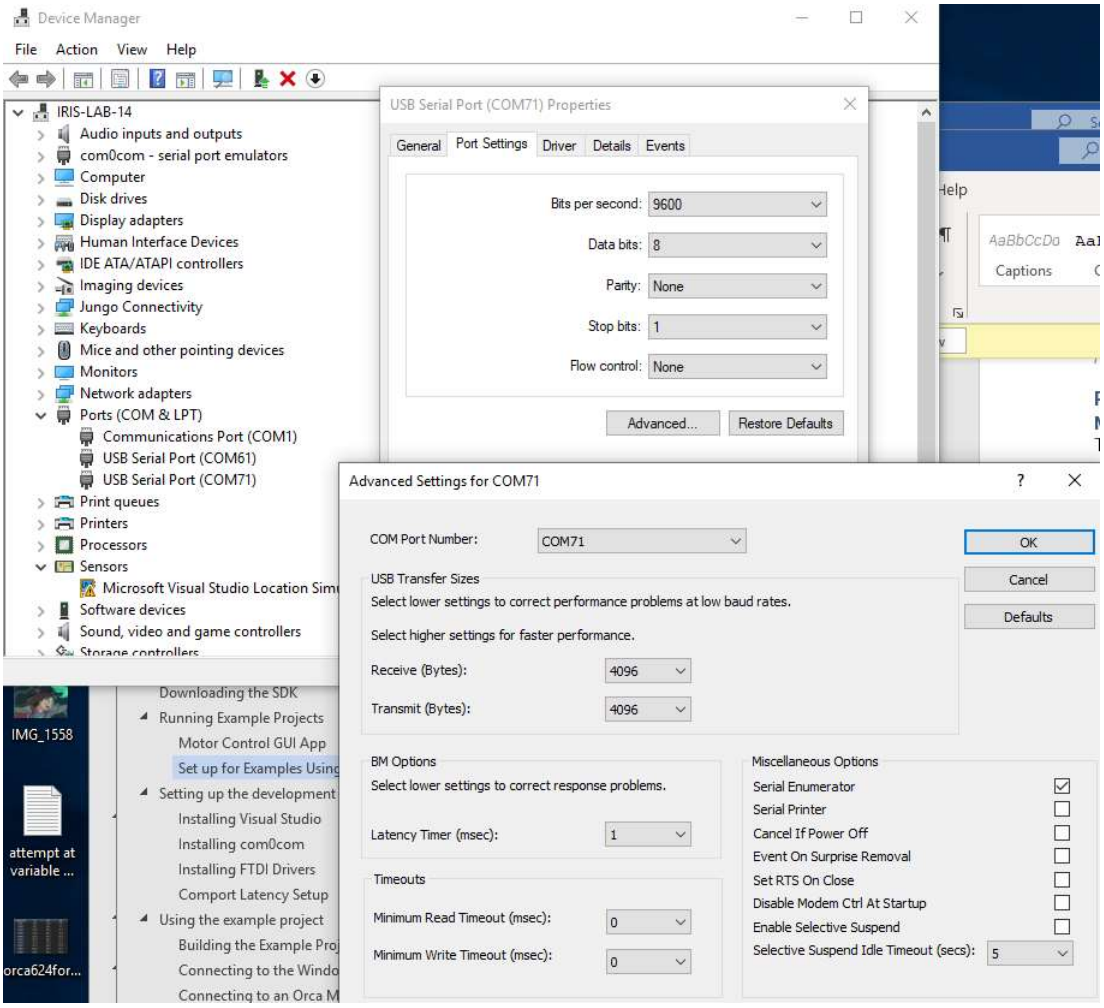
Set up for Examples Using Higher Speed Communication

The default latency on RS422 USB-to-ethernet cables slows down communication between the virtual windows device and the Orca motor. To establish a connection, this latency must be adjusted.

NOTE: These instructions require that you have administrator access on your computer.

- With the USB end of your RS422 connected to the computer, open the Windows Device Manager in Administrator Mode.

- Right click on the comport that your motor is using. Select Properties.
- In the Properties window select Port Settings at the top, and then Advanced.
- In the Advanced window set the latency timer of the port to 1 msec. Press OK twice to save these settings.



Set up for Examples Using IrisControls4 Windows Application

Setting up a virtual COM port is required to connect to the IrisControls4 windows application which is used to display custom GUIs using the ic4_library.

Com0com is Windows driver which will install a pair of virtual comports on your machine. The Windows Virtual Device and IrisControls4 will communicate via this pair of comports. If you would prefer to not use com0com see the Motor Control GUI App example project for an example of communicating with a motor not using a GUI made with a QT framework.

Note: These instructions require that you have administrator access on your computer.

- Locate the VirtualCOM_port_setup directory within the repo.
- Launch setup.exe from this directory.

- When prompted to choose components to install, leave the default options selected.
- If you are prompted to download .NET framework, say Yes.
- When the installer completes, restart your computer, and navigate to the com0com install directory. Open setupg.exe from the install directory.
- When the setup application opens, click the 'Add Pair' button on the bottom left corner.
- Click into the highlighted textbox and rename the ports to 'COM51' and 'COM52' . Select 'enable buffer overrun' for each port and click 'Apply'. You can also choose another pair if you wish.

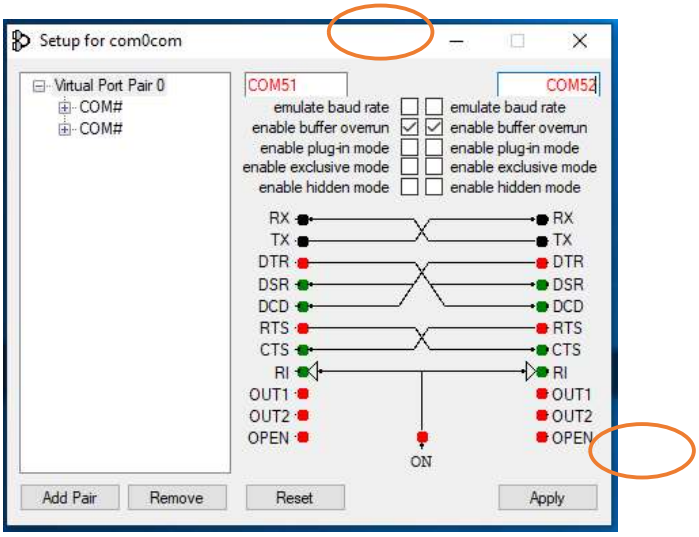


Figure 4: Renaming port pairs in com0com

- Once you have applied your changes you should see a com0com listed in your Device Manager.

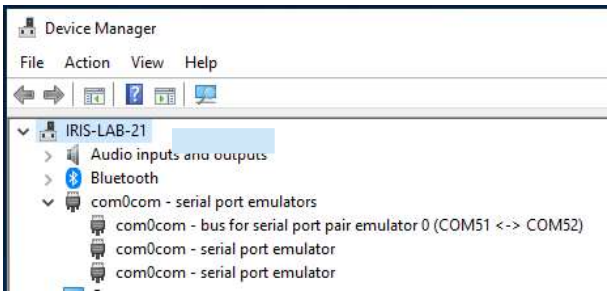


Figure 5: com0com virtual ports listed in Device Manager

Connecting to the Windows Virtual Device via IrisControls4

To connect with IrisControls4, first clone this repo: https://github.com/IrisDynamics/IrisControls4.

Navigate to where you have the repo on your computer. Open the .exe file.

Run an example project that uses IrisControls4. A console application will build and ask you which comport you would like to connect on. Type in the first number of one of your comport pairs set up though com0com and press enter. You should see a message saying that opening

the port was successful. If instead you receive an error, make sure you have com0com properly installed, and not have any other instances that connect on the port.

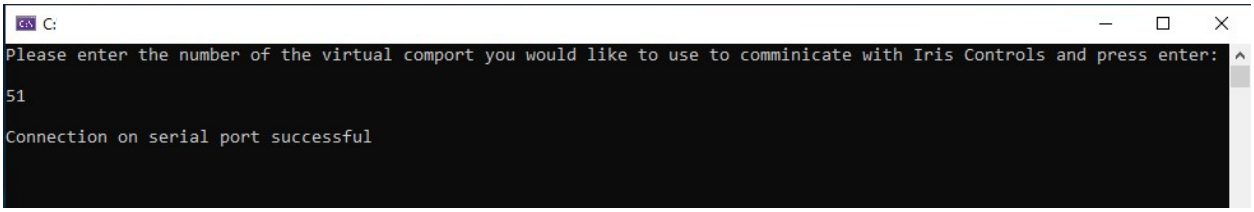


Figure 8: A successful connection to IrisControls4

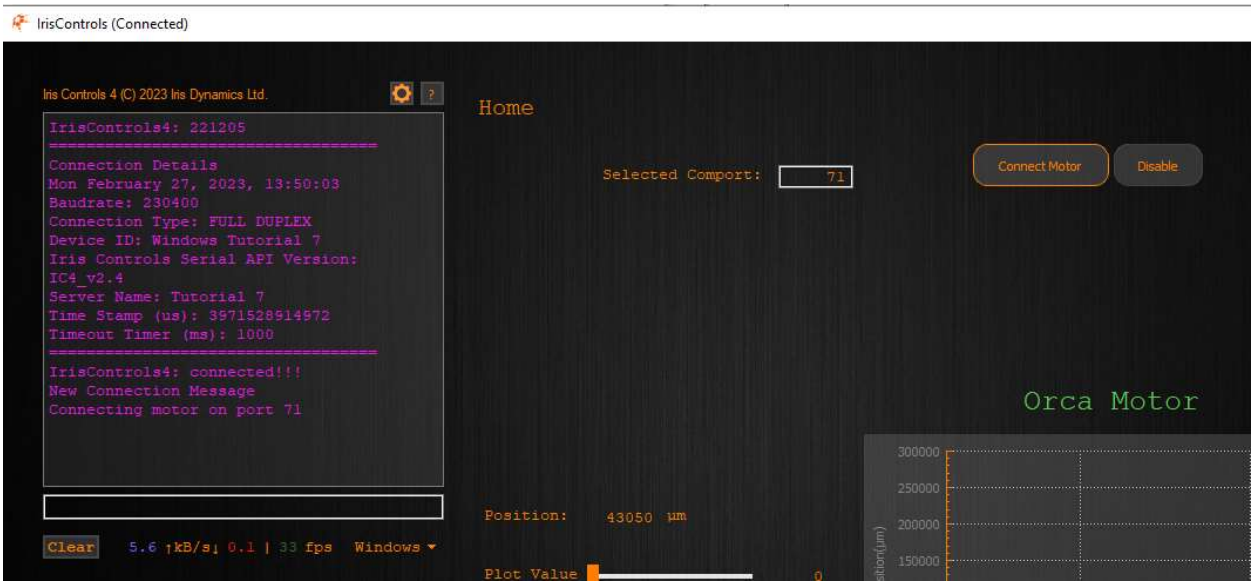
Open IrisControls4 and in the drop down on the bottom right select the second of your comport pairs from the list. This will connect to the virtual comport that you indicated through the console app.

Figure 9: The IrisControls4 console with comport dropdown highlighted.

Connecting to an Orca Motor via the Windows Virtual Device

When the project builds and a connection is made to IrisControls4, a GUI will be built in the IrisControls4 window. When a comport is entered and the Connect Motor button is pressed, a connection will be established between the Windows Virtual Device and the motor on the indicated comport.

Note: for the Comport Selector to recognize the entry, you must press enter or tab after typing in the text box. This is the case with all text fields in IrisControls4.



When a motor is connected, the graph is populated with information from the motor. The plot title will turn green when a motor is connected and has no errors. If there is no motor connected to a port, the title will be grey title, and motors which have errors will show a red plot title.

The motor's **Position** and **Power** are plotted on the graph. Under the graph there are additional values.

Voltage supplied to the motor, in Volts

Temperature of the motor, in Celsius

Power being used by the motor, in Watts

Error flags indicating the error present (see Troubleshooting section)

Frequency of communications between the Windows Virtual Device and the Orca motor

Force slider shows the amount of force being sensed by the motor, either exerted from the motor or externally applied.

This section of the GUI is the Motor Plot panel found in the irisSDK_libraries library.

SETTING UP THE DEVELOPMENT ENVIRONMENT

If you would like to start developing a custom application using the libraries, the source code for all example projects is available, as well as a step by step tutorial guide.

Installing Visual Studio

To compile and build the virtual windows device, the Visual Studio environment must be set up. Download and install the Community version of Visual Studio 2022: https://visualstudio.microsoft.com/downloads/ . Open the installer exe and follow the prompts.

When the installer asks which workloads you would like to install select 'Desktop Development with C++'.

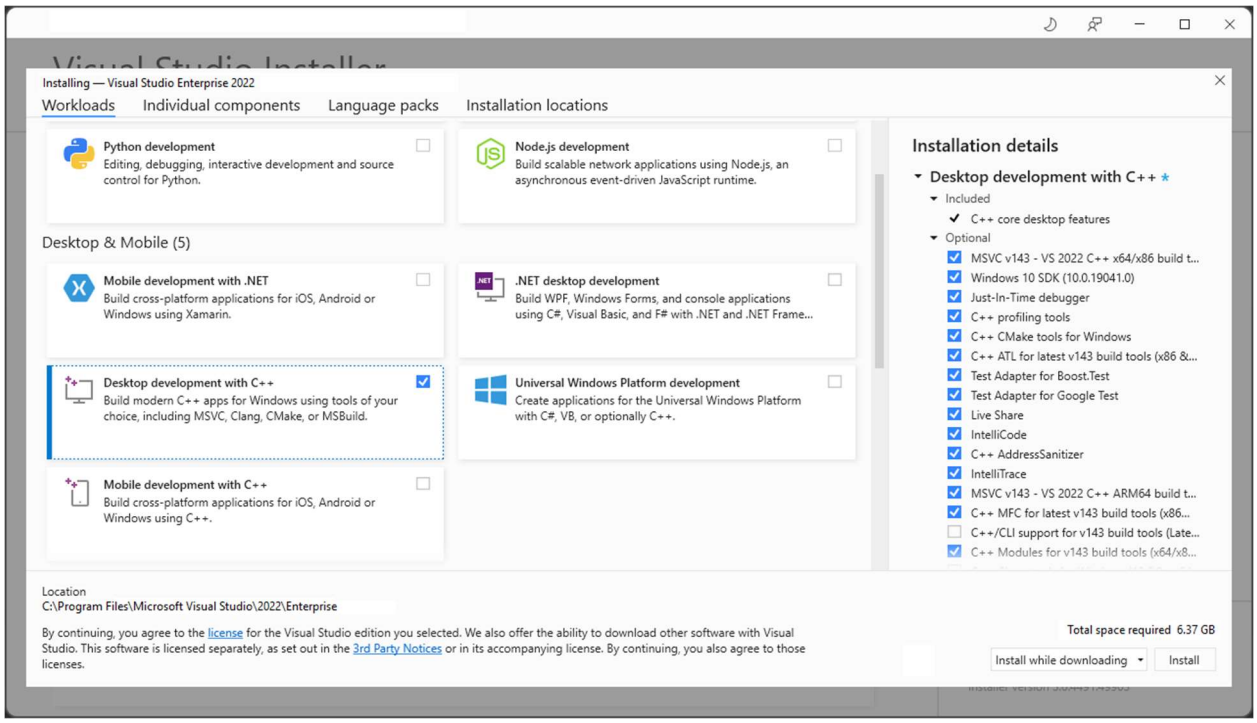


Figure 3: Visual Studio Installer

CUSTOM SOFTWARE

Building the Example Project

Source code for example projects are available in the examples folder of the IrisSDK_for_Windows repo. Double click the .sln file to open the solution in Visual Studio.

Launch the IrisSDK_example.sln file in the repo's root directory. You may be prompted to update the Platform Toolset (v142 is VS2019 and v143 is VS2022). Follow these steps to do so:

- In the Solution Explorer, right click on WindowsSDK_example and go to Properties.
- Under Configuration Properties -> General ensure that the Platform Toolset is v143.

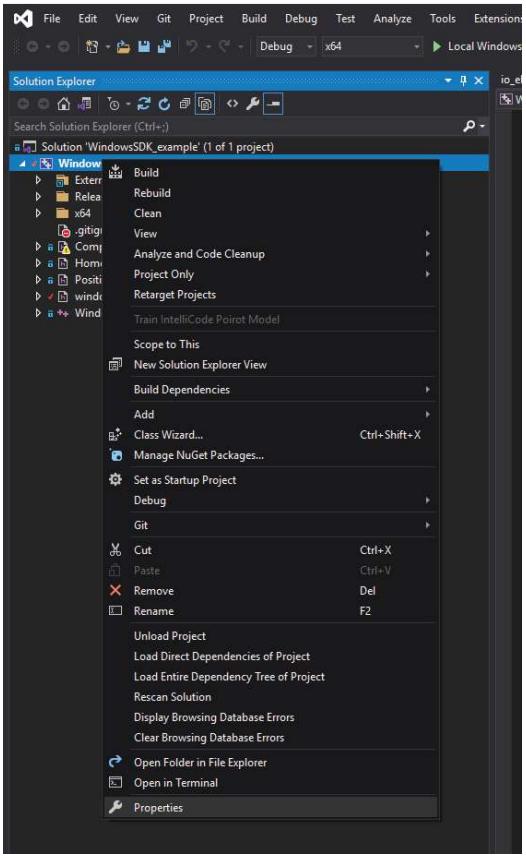


Figure 7: Opening project properties

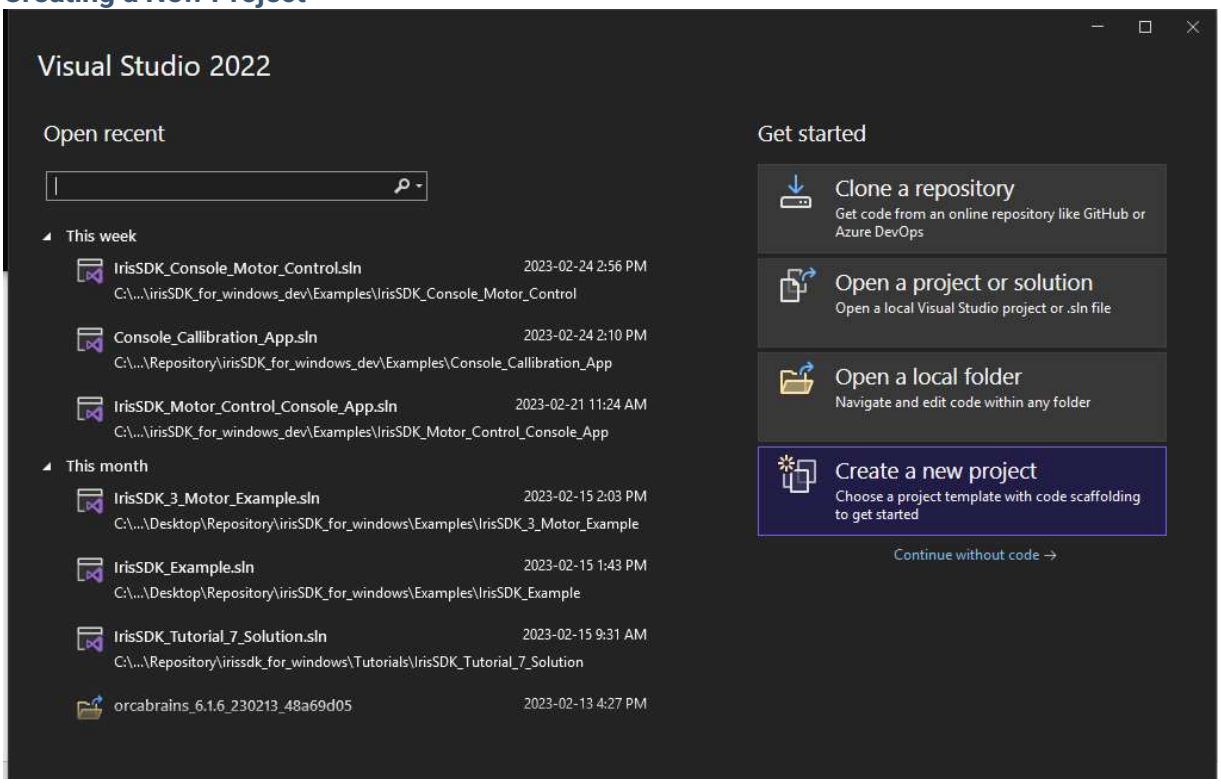
Note: To use the SDK projects you will have to add the libraries folder to your include path.

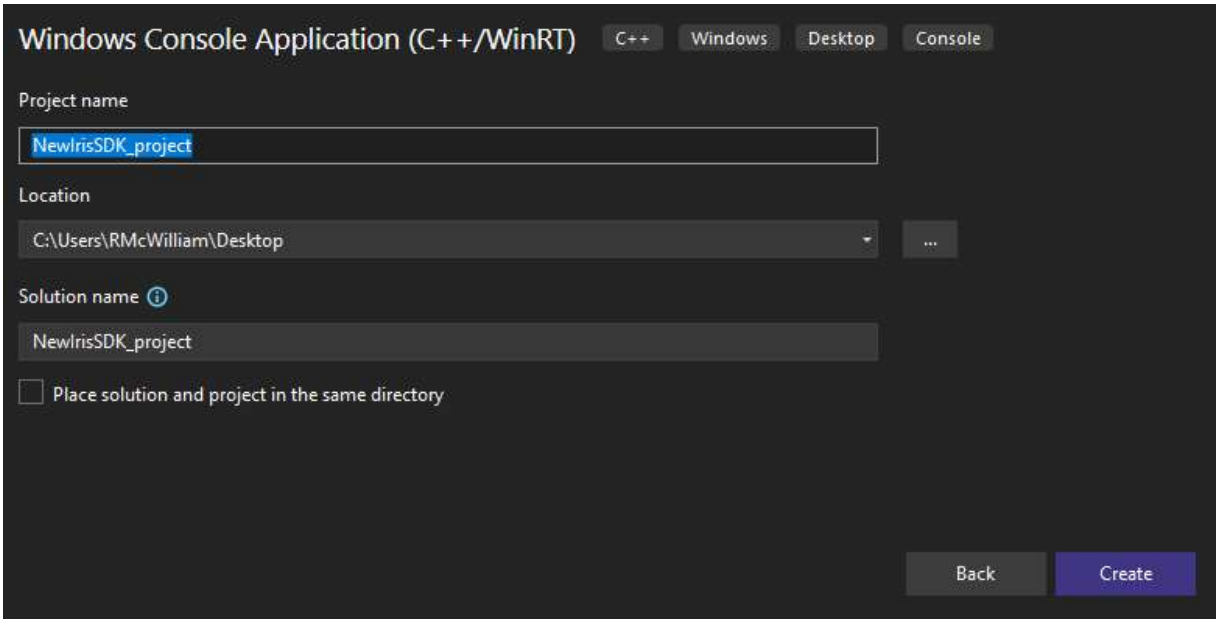
To adjust your include paths:

- In the Solution Explorer, right click on the project name and go to Properties.
- Under C/C++ -> General ->Additional Include Directories type in or use the wizard to select the folder you would like to include.

You should now be able to build the project. This project creates a virtual windows device which can be used to interact with an Orca Series Motor.

Creating a New Project





Also, make sure to the include the library_linker.h file in the main file of your project.

Included Libraries

The included libraries provide many tools for your custom development.

ic4_library : IrisControls4 API, GUI elements and serial communications with IrisControls4.exe, more info here : https://wiki.irisdynamics.com/index.php?title=IrisControlsAPI4_Overview

modbus_client : Modbus communications, contains the Actuator object (Orca API), more information can be found here: https://www.irisdynamics.com/downloads/ in the Orca API User Guide under the user manuals section.

orca600_api : Memory map associated with the Orca motor, more information can be found here: https://www.irisdynamics.com/downloads/ in the Orca Series Motor Reference Manual under the user manuals section.

irisSDK_libraries : Premade combinations of GUI elements for common feedback/functionality. More information on using these can be found in the IrisSDK for Windows Tutorial Guide. This folder also contains the library_linker.h file, which should be included in the main cpp file of all projects.

TROUBLESHOOTING

Not connecting with IrisControls4

If the Windows Virtual Device is not connecting to IrisControls4 ensure the following:

- The correct version is IrisControls4 is in use(not compatible with IrisControls2). The correct version will be available in the IrisSDK_for_Windows repository in the IrisControls4 folder.
- The virtual comport pair has been properly set up in com0com.
- The correct COM port is selected for the drop-down menu. This comport should be the corresponding comport to the one chosen in the console application.

Motor not connecting

- Motor must be powered (24-48V). The motor will also connect, but not output forces if using a splitter and having the RS485 cable connected to a PC with no additional power source.
- Motor's data cable is connected to the COMPORT that the virtual windows device is trying to connect to.

Motor not outputting forces

- Motor is receiving power (24-48V), the motor's logic can be powered through the RS485 cable with the splitter, however this will not allow the motor to output force.
- Errors will prevent the motor from outputting forces see the Orca API User Manual for full details about these errors:
    - Configuration Errors - 1
    - Force Clipping - 32
    - Temperature Exceeded - 64
    - Force Exceeded - 128
    - Power Exceeded - 256
    - Shaft Image Failed - 512
    - Voltage Invalid - 1024
    - Comms Timeout - 2048

  If multiple error codes are present, they will be added together, error 320 would be temperature exceeded and power exceeded. If a motor's plot has the title changed to red, then the motor is connected but errors are present.

Non persisting errors can be cleared by entering sleep mode or calling clear_errors() function.

GUI elements not populating

- If IrisControls4 appears to be connecting (messages in the console window, Status: Connected) but only some or no GUI elements are being populated, there may be a problem with the window sizing and/or element placement. The IrisControls4 API allows you to set the grid size and adjust element placement in the firmware code. Changing your computer/tablet display setting to change the resolution and/or scale might fix the issues (Example firmware projects have all been designed to fit 1920 x 1080 Resolution, 100% Scale). The placement of GUI elements and the grid size can also be changed. The grid numbering can be displayed by typing "guide_on" in the console and hidden by typing "guide_off".

Project not compiling

- Error message in the console "could not find file …"
  Ensure that the libraries folder of the Iris SDK for Windows repository has been added to your include path in the project properties.
- A list of errors will appear in the Visual Studio output console. A description of the error will be provided along with a link to documentation on this kind of error.

NEXT STEPS

Now that you have set up your development environment and used an example project, you can begin the tutorials included in this repo. The Tutorial Guide is located in the Tutorials folder, and walks you through how to proceed. Alternatively, finished tutorial solutions are also included in the Tutorials folder. You can also review these files and start building your own projects right away.

If you are experiencing issues setting up your motor or have any questions related to this SDK, please contact support@irisdynamics.com.

ADDITIONAL RESOURCES

Available for download at https://www.irisdynamics.com/downloads/

1. Orca QuickStart Guide
2. Orca Series Datasheet
3. Orca Motor Reference Manual
4. Orca API User Guide

REVISION HISTORY

Version	Date	Author	Reason
1.0.0	November 2022	KC	Initial Draft