Using C# API DLLs in C++

- Background
- Instructions

Background

Most AML products have a C# API available to allow easy integration of the product into your own application, without having to write an API yourself. The C# APIs provide an easy way to access all aspects of configuration and control of the product.

Using these APIs from C# is the easiest option. However, you can use the APIs with other languages such as C++ or Python.

Instructions

1. Download the latest copy of the API DLL for your product

Visit the product page for your product and look for the C# API download, for example for the SMD4, visit the link below, and look for the downloads tab at the bottom of the page. https://arunmicro.com/products/smd4-stepper-motor-drive/

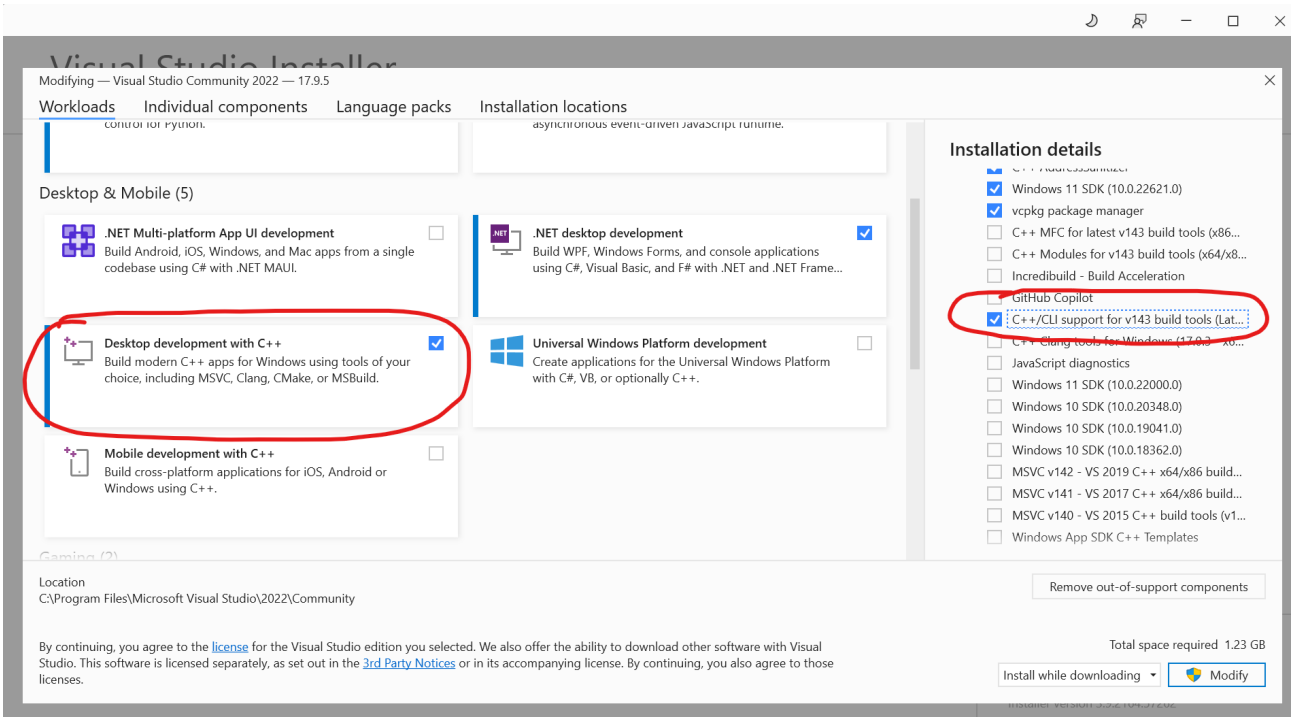
2. Install workloads for Visual Studio

To install a workload, or check what is installed, open Visual Studio and go to 'Tools > Get Tools and Features...'

Install the following workloads:

- .NET desktop development
- Desktop development with C++, with the default installation components PLUS the following:
  - C++/CLI support for v143 build tools

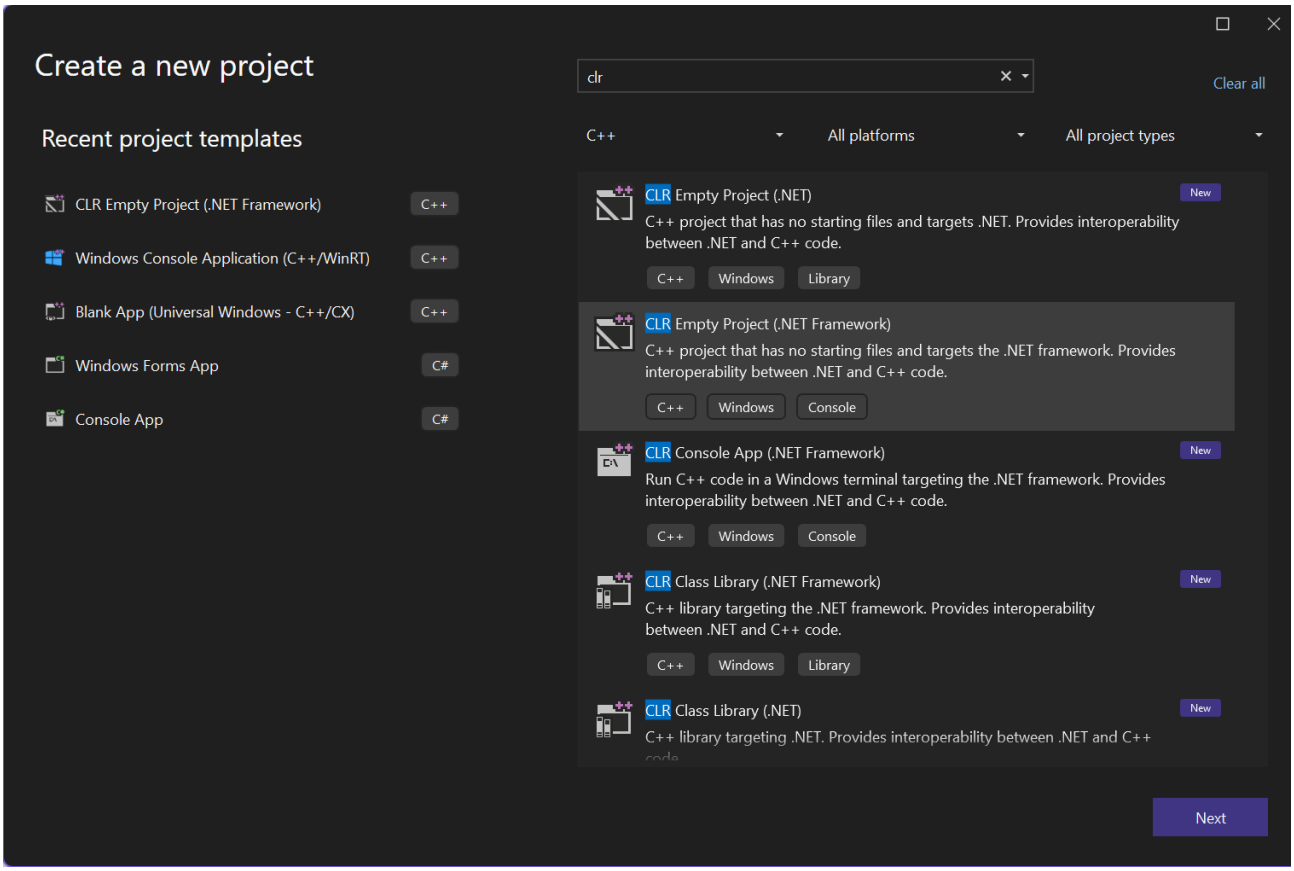
Use install or modify as required.



3. Creating the project

Start the new project wizard (File > New > Project). Choose one of the following, then proceed to create your project as normal.

- CLR Empty Project (.NET Framework)
- CLR Console App(.NET Framework)

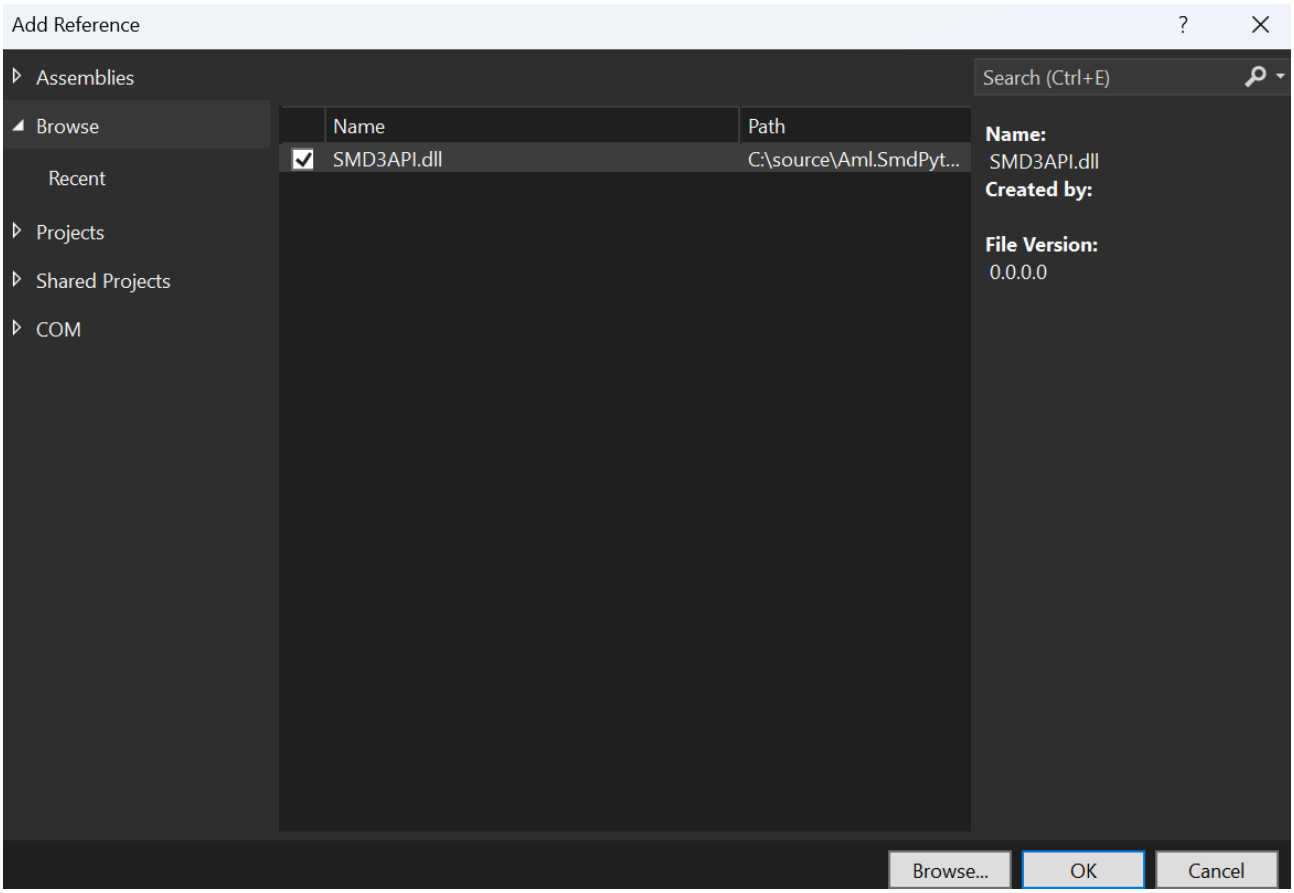


4. Referencing the API DLL file

The API DLL file needs to be referenced in your project. To do this go to:

- Solution Explorer
- Right click on the project, Add > Reference...
- Choose ‘browse’ and select the DLL file you downloaded previously (example below shows SMD3 API)

Finally, click OK and the DLL will be added as a reference to your project.



5. Using the API

Example code (main.cpp), demonstrating connecting and moving the motor.

```
#include <windows.h>
#include <iostream>
#include <msclr/marshal_cppstd.h> // Include for string conversion

// Include CLR headers
#include <vcclr.h>
#pragma comment(lib, "mscoree.lib")

// Import the mscorlib library
#import "C:\\Windows\\Microsoft.NET\\Framework\\v4.0.30319\\mscorlib.tlb"
raw_interfaces_only \
    high_property_prefixes("_get","_put","_putref") \
    rename("ReportEvent", "InteropServices_ReportEvent")

using namespace System;
using namespace msclr::interop; // Namespace for string conversion


// Main function to use the DLL in a native C++ application
int main() {
    try {
        // Create a instance of the class SMD3 from SMD3API namespace in the dll
```

```
        auto smd3 = gcnew SMD3API::SMD3();

        // Connect the SMD3 on COM3
        smd3->Connect("COM3 TEXT");

        // Start motor spinning clockwise
        smd3->MoveVelocity("+");

        // Wait for user before stopping motor
        std::cout << "Press Enter to stop motor" << std::endl;
        std::cin.get();
        smd3->Stop();
    }
    catch (Exception^ ex) {
        // Convert System::String^ to std::string
        std::string errorMessage = marshal_as<std::string>(ex->Message);
        std::cerr << "Exception: " << errorMessage << std::endl;
    }
    return 0;
}
```