LabSmith Application Note

LabSmith uProcess™ SDK Visual Studio Code Setup

By Nirajara Dungwatanawanich
LabSmith, Inc., Livermore, CA 94550

LabSmith's uProcess™ SDK allows for communication with LabSmith uProcess microfluidic automation devices, including valves, syringe pumps, pressure and temperature sensors, and more. LabSmith now provides SDKs for both Linux and Windows operating systems.

The Windows SDK is available on the uProcess Support webpage (login required). Email support@labsmith.com to request the Linux SDK.

Installing Visual Studio Code

Windows

Follow the instructions here: Visual Studio Code for Windows.

Linux

Open the Linux Terminal and type:

```
sudo snap install --classic code
```

If snap is not installed, do the following commands first:

```
sudo apt update
sudo apt install snapd
```

More information on installing Visual Studio Code in Linux can be found at: Visual Studio Code for Linux.

Visual Studio Code Extensions

Once the Visual Studio Code is installed, open the program and navigate to Extensions, located in the menu on the left-hand side as seen in Figure 1.

Install the following extensions:

1. C/C++
2. CMake
3. CMake Tools.

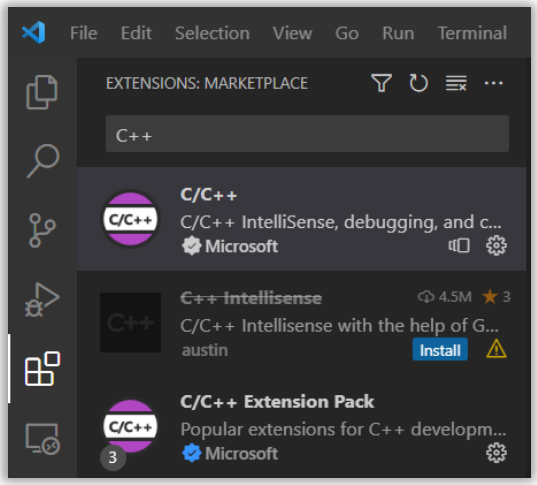


Figure 1. Installing C/C++ extension in VSCode

Building the SDK

To start working with the SDK, open the project in Visual Studio Code (Figure 2).

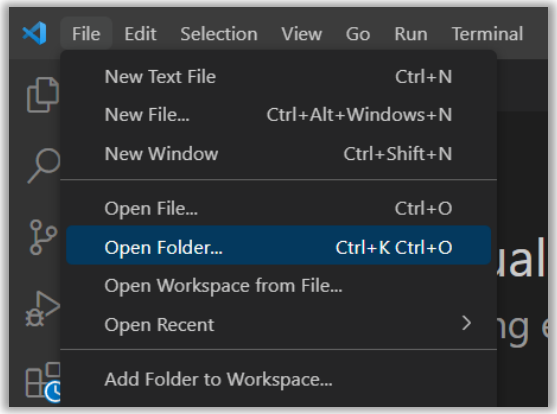


Figure 2. Opening project in VSCode

Press Ctrl+Shift+P and run "CMake: Build" to generate the makefiles for the project and build the executable (Figure 3). The executable can be found in /x64/Debug/.

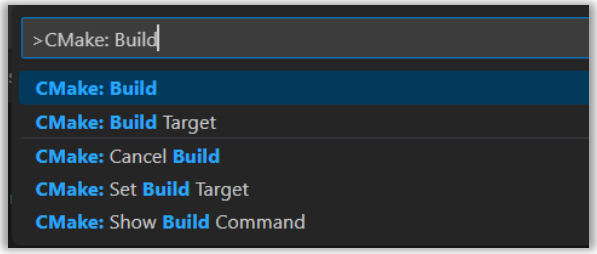


Figure 3. Building the project

Adding Files

When adding source files to the project, add the file path to their respective CMake files in the uProcessDLL or uProcessTest folder (NOT the one in the main project folder). The section for this addition can be found in the CMakeLists.txt file under the "# Source groups" line (Figure 4).

```
########################################
# Source groups
########################################
set(Header_Files
    "pch.h"
)
source_group("Header Files" FILES
${Header_Files})

set(Source_Files
    "pch.cpp"
    "uProcessTest.cpp"
)
source_group("Source Files" FILES
${Source_Files})

set(ALL_FILES
    ${Header_Files}
    ${Source_Files}
)
```

Figure 4. Adding source files in CMakeLists.txt

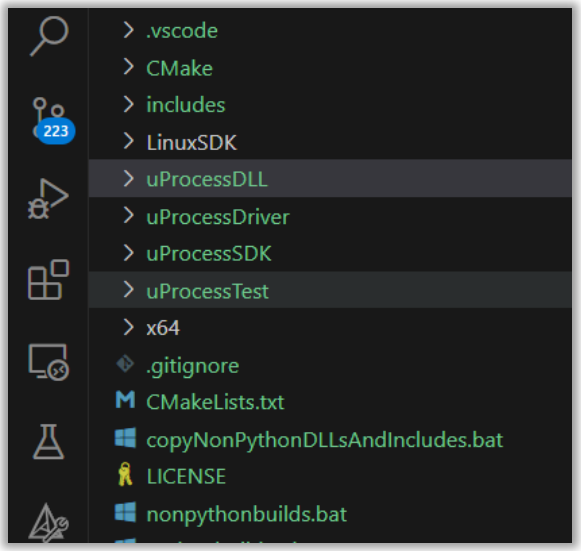


Figure 5. The uProcessDLL and uProcessTest folder

Communicating with Device

Windows

1. Connect the uProcess setup to the computer via USB cable.

2. Determine the COM port. Open Device Manager and go to Ports to find the COM port number for the device (Figure 6).

Figure 6. Device Manager

3. In the uProcessTest folder, open uProcessTest.cpp, change the variable nCom port number to the appropriate COM port.

4. Rebuild the project (see Building the SDK).

5. Run the executable. Navigate in the file manager to the location of the executable and double click uProcessTest.exe.

Linux

1. Connect the uProcess setup to the computer via USB cable.

2. Determine the COM port. Open the Linux Terminal and type:

```
ls /dev/ttyUSB*
```

substituting the COM port for the *. In Figure 7, the COM port is 0.

```
root@DESKTOP-SSC52RD:~# ls /dev/ttyUSB*
/dev/ttyUSB0
```

Figure 7. Determining COM number in Linux terminal

3. In the uProcessTest folder, open uProcessTest.cpp. Change the variable nCom port number to the appropriate COM port.

4. Rebuild the project (see Building the SDK).

5. Run the executable. Navigate to x64/Debug in terminal and run:

```
./uProcessTest
```

IMPORTANT: If the program fails to connect to the device, it may be because the Linux user is not part of the dialout group. Add the user to the dialout group through terminal:

```
sudo adduser [user] dialout
```

Determine your username with the command:

```
whoami
```

Debugging in Visual Studio Code

To start debugging:

1. Open the debugging sidebar by clicking the Run and Debug icon in the menu on the left-hand side.

2. Click the green triangle at the top of the debugging sidebar (Figure 8).

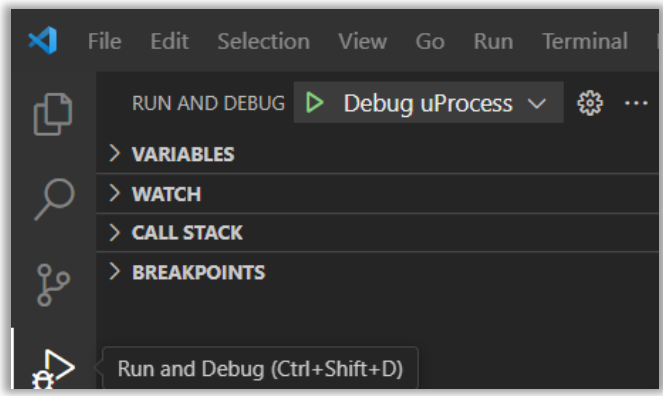


Figure 8. Debugging in VSCode

3. Add breakpoints by clicking to the left of the line number (Figure 9).

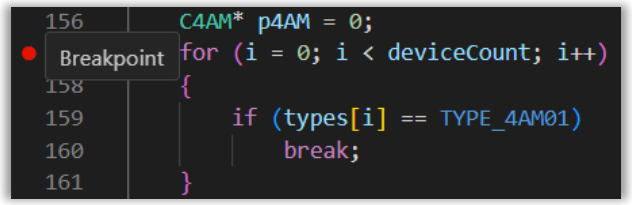


Figure 9. Adding breakpoints

4. Choose standard debugger options (stepping over, into, and out) from the menu that appears at the top center (Figure 10).

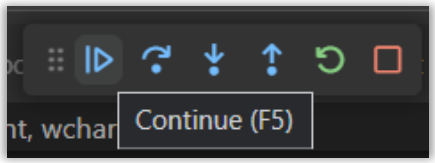
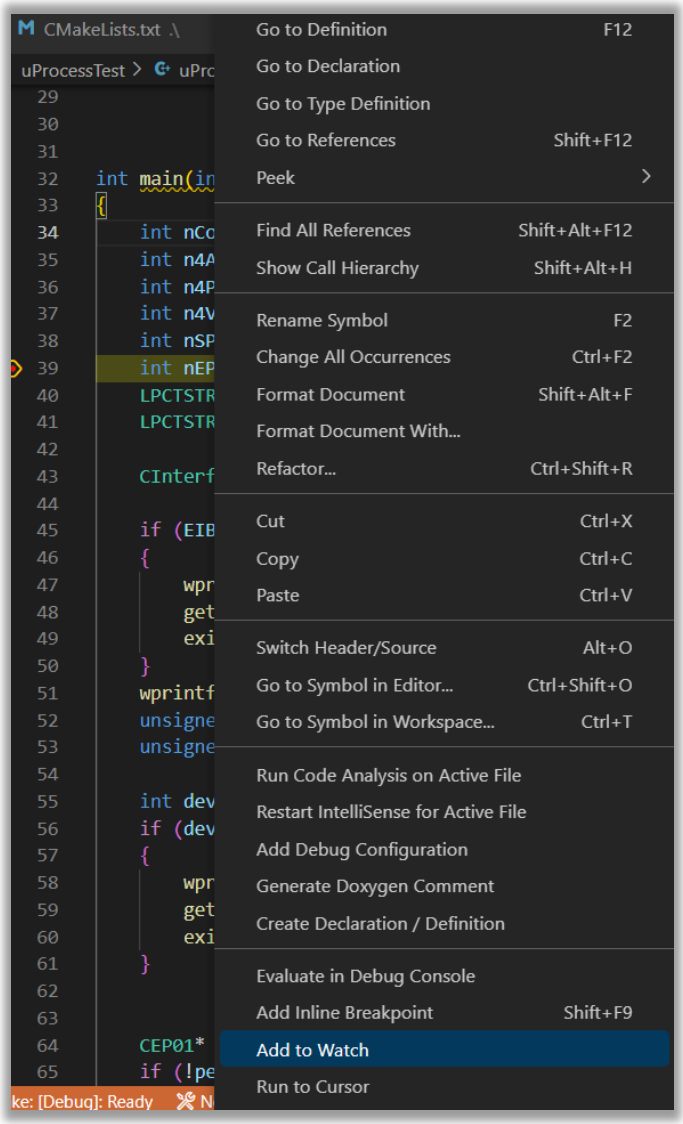


Figure 10. Navigating the debugger

5. Add variables to watch by right-clicking on variables and selecting “Add to Watch” in the panel (Figure 11).



`Figure 11. Adding variables to watch

More information on debugging in Visual Studio Code can be found in the Visual Studio Code Documentation.

uProcess™ Linux SDK with Windows Subsystem for Linux (WSL)

The SDK for Linux OS can be run on Windows using WSL. Instructions can be found here: WSL Setup Instructions.

Information on connecting to USB devices can be found here: Connecting USB Devices to WSL