Using uFCoder library on UWP
Version 1.1

Digital Logic Ltd.

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

Table of contents

Get uFCoder library	**3**
Add capabilities for serial communication	**3**
Include uFCoder library in a Visual Studio project - C#	**3**
uFCoder UWP library usage example - C#	**4**
Include uFCoder library in a Visual Studio project - C++	**6**
uFCoder UWP library usage example - C++	**6**
Revision history	**9**

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

Get uFCoder library

Follow the instructions below to get the latest uFCoder library.

1. Navigate to https://www.d-logic.net/code/nfc-rfid-reader-sdk/ufr-lib
2. Download or clone urf-lib repository.
3. Folder 'ufr-lib/windows/uwp' contains 'uFCoder' and 'uwp-serial' dll/lib files for the x86/x64/ARM platforms that are required to work with uFR Series readers on UWP.

Add capabilities for serial communication

Follow the instructions below to add the capability for serial communication in your UWP project:

1. Open file *"Package.appxmanifest"*
2. Add the following under the <*Capability*> tag:

```
<DeviceCapability Name="serialcommunication">
    <Device Id="any">
      <Function Type="name:serialPort" />
    </Device>
</DeviceCapability>
```

Include uFCoder library in a Visual Studio project - C#

Follow the instructions below to include the library in your UWP project:

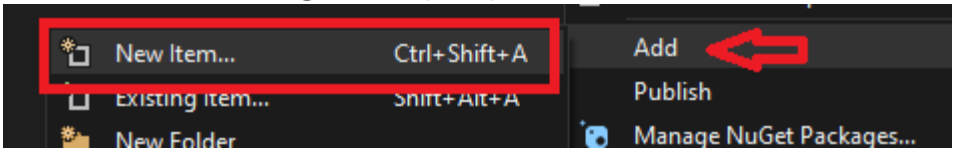
1. Create a new UWP project, or open an existing one.
2. Download *"ufr-lib"* as a submodule in your project, or if you have already downloaded it beforehand simply place it in the root directory of your project.
3. Right-click your project, select *"Add"* and then *"Existing item..."*
4. Navigate to the following path: */ufr-lib/windows/uwp*. From this folder, you need to add "uFCoder-*" and "uwp-serial-*" .dll & .lib files, depending on your platform architecture, and add them.
    a. F*or example, for the x86 platform, you will need the following files: "uFCoder-x86.dll", "uFCoder-x86.lib", "uwp-serial-x86.dll", "uwp-serial-x86_64.lib"*

5. Afterwards, your project will require a "uFCoder.cs" import file. This file should contain imports (*[DllImport]* attribute) of the methods from our API. More details in the following section: <link to section>

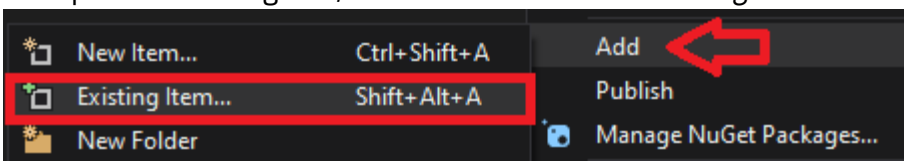
Digital Logic Ltd.

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

a. To create a new one, right-click your project name, select *"Add"* and then *"New item..."*



b. To import an existing one, select *"Add"* and then *"Existing item..."*



i. On our git, there are various C# examples containing *"uFCoder.cs" or "uFCoderMulti.cs"* import files. Explore C# projects on our git. These projects can be used as a reference.

6. Depending on your implementation of the import file, or adding an existing one, all that is left is to reference it in your source code via *"using"* keyword. E.g for our import files, simply add the *"using uFR;"* line in your code. *(namespace/class name(s) may vary depending on the import file)*

uFCoder UWP library usage example - C#

```
// various "using ..." imports for your project
using uFR;

// The Blank Page item template is documented at
https://go.microsoft.com/fwlink/?LinkId=402352&clcid=0x409

namespace DemoApp_UWP
{
  /// <summary>
  /// An empty page that can be used on its own or navigated to within a Frame.
  /// </summary>
  public sealed partial class MainPage : Page
  {
    public MainPage()
    {
      this.InitializeComponent();
    }

    private void Button_Click(object sender, RoutedEventArgs e)
    {
      textBox.Text += uFCoder.dll2str() + Environment.NewLine;
```

```
        DL_STATUS status = uFCoder.ReaderOpen();
        // alternatively you can use ReaderOpenEx() method instead of ReaderOpen
        // however it will require of the user to provide the necessary parameters
        // for example:
        // UInt32 reader_type = 1;
        // string port_name = "WINIOT:0";
        // UInt32 port_interface = 0x01;
        //  string arg = "";
        // refer to our API document & "uFCoder.h" found in "ufr-lib/include"
        // for the values of these parameters
        // DL_STATUS status = uFCoder.ReaderOpenEx(reader_type, port_name, port_interface, arg)

         if (status == DL_STATUS.UFR_OK)
        {
           byte sak = 0, uid_size = 0;
           byte[] uid = new byte[11];
           status = uFCoder.GetCardIdEx(ref sak, uid, ref uid_size);
           if (status == DL_STATUS.UFR_OK)
           {
              String uid_str = BitConverter.ToString(uid, 0, uid_size).Replace("-", ":");
              txtBox.Text += "UID: " + uid_str + Environment.NewLine;
           } else
           {
              txtBox.Text += "GetCardIdEx() has failed, status: " + uFCoder.status2str(status) +
Environment.NewLine;
           }
        else
        {
           txtBox.Text += "ReaderOpen() has failed, status: " + uFCoder.status2str(status) +
Environment.NewLine;
        }
   }
}
```


Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

Include uFCoder library in a Visual Studio project - C++

Follow the instructions below to include the library in your UWP project:

1. Create a new UWP project, or open an existing one.
2. Download *"ufr-lib"* as a submodule in your project, or if you have already downloaded it beforehand simply place it in the root directory of your project.
3. Right-click your Project name in the "Solution Explorer" and open *"Properties" (ALT + Enter)*
4. Open the following: *Configuration Properties -> Linker -> Input -> Additional dependencies*, and add the following line *ufr-lib/windows/uwp/uFCoder-x86.lib* (change "-x86" suffix according to your platform/build configuration)
5. Now, one another property should be changed. Open the *Configuration Properties -> Build Events -> Post-Build Event -> Command line* and add the following lines:
   xcopy "$(SolutionDir)$(ProjectName)\ufr-lib\windows\uwp\uFCoder-x86.dll" "$(TargetDir)AppX" /s /d /y
   xcopy "$(SolutionDir)$(ProjectName)\ufr-lib\windows\uwp\uwp-serial-x86.dll" "$(TargetDir)AppX" /s /d /y
   a. These commands should be able to copy dlls right next to the created .exe (if build was successful, you should always check the output directory, and/or do a *clean-> build* of the solution. Of course, change the scripts "-x86" suffixes so they correspond to your platform/build configuration.

uFCoder UWP library usage example - C++

```
#include "ufr-lib/include/uFCoder.h"

// various "using ..." imports for your project

// The Blank Page item template is documented at
https://go.microsoft.com/fwlink/?LinkId=402352&clcid=0x409

MainPage::MainPage()
{
        InitializeComponent();
}

constexpr char hexmap[] = { '0', '1', '2', '3', '4', '5', '6', '7',
                                                    '8', '9', 'a', 'b', 'c', 'd', 'e', 'f' };

// Helper functions for converting data types.
```


Digital Logic Ltd.

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

```
std::string toHexStr(unsigned char* data, int len)
{
	std::string s(len * 2, ' ');
	for (int i = 0; i < len; ++i) {
		s[2 * i] = hexmap[(data[i] & 0xF0) >> 4];
		s[2 * i + 1] = hexmap[data[i] & 0x0F];
	}
	return s;
}

Platform::String^ StdStrToPlatformStr(const std::string& input)
{
	std::wstring w_str = std::wstring(input.begin(), input.end());
	const wchar_t* w_chars = w_str.c_str();
	return (ref new Platform::String(w_chars));
}


void DemoCppApp_UWP::MainPage::Button_Click(Platform::Object^ sender,
Windows::UI::Xaml::RoutedEventArgs^ e)
{
	UFR_STATUS status = ReaderOpen();
	//uint32_t reader_type = 1;
	//std::string port_name = "WINIOT:0";
	//uint32_t port_interface = 1;
	// std::string arg = "";
	// UFR_STATUS status = ReaderOpenEx(reader_type, port_name.c_str(), port_interface,
(void*)arg.c_str());
	if (status == UFR_OK)
	{
		ReaderUISignal(1, 1);
		uint8_t sak = 0, uid_size = 0;
		uint8_t uid[11];
		status = GetCardIdEx(&sak, uid, &uid_size);
		if (status == UFR_OK)
		{
			std::string uid_str = toHexStr(uid, uid_size);
			String^ print_uid = StdStrToPlatformStr(uid_str);
			txtBox->Text += "UID: " + print_uid + "\n";
		}
		else
		{
			std::string ufr_error = UFR_Status2String(status);
			String^ print_error = StdStrToPlatformStr(ufr_error);
```


Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com

```
                txtBox->Text += "GetCardIdEx() has failed, status:" + print_error + "\n";
            }
        }
        else
        {
            std::string ufr_error = UFR_Status2String(status);
            String^ print_error = StdStrToPlatformStr(ufr_error);
            txtBox->Text += "ReaderOpen() has failed, status:" + print_error + "\n";
        }
}
```

Revision history

Date	Version	Comment
2022-02-02	1.0	Base document

Digital Logic Ltd.

Address: Nemanjina 57a, 12000 Pozarevac, Serbia · Tel: +38112541022· VAT: 111385444 Reg. 21473642
e-mail: office@d-logic.com · www.d-logic.com