Digi C# Building Your First Application

1. Start with Digi C# API

1.1. What is it?

The main function of this API is to make the application development easier to the user, this mean, hide the hardest part of develop application with the standard functions, hide handlers, configurations that are always the same and so on.

1.2. Installation

Run the Setup.exe, after the installation you can find the API in the folder **%Program_Files%\Digi\Digi C Sharp Embedded Library** with the following contain:

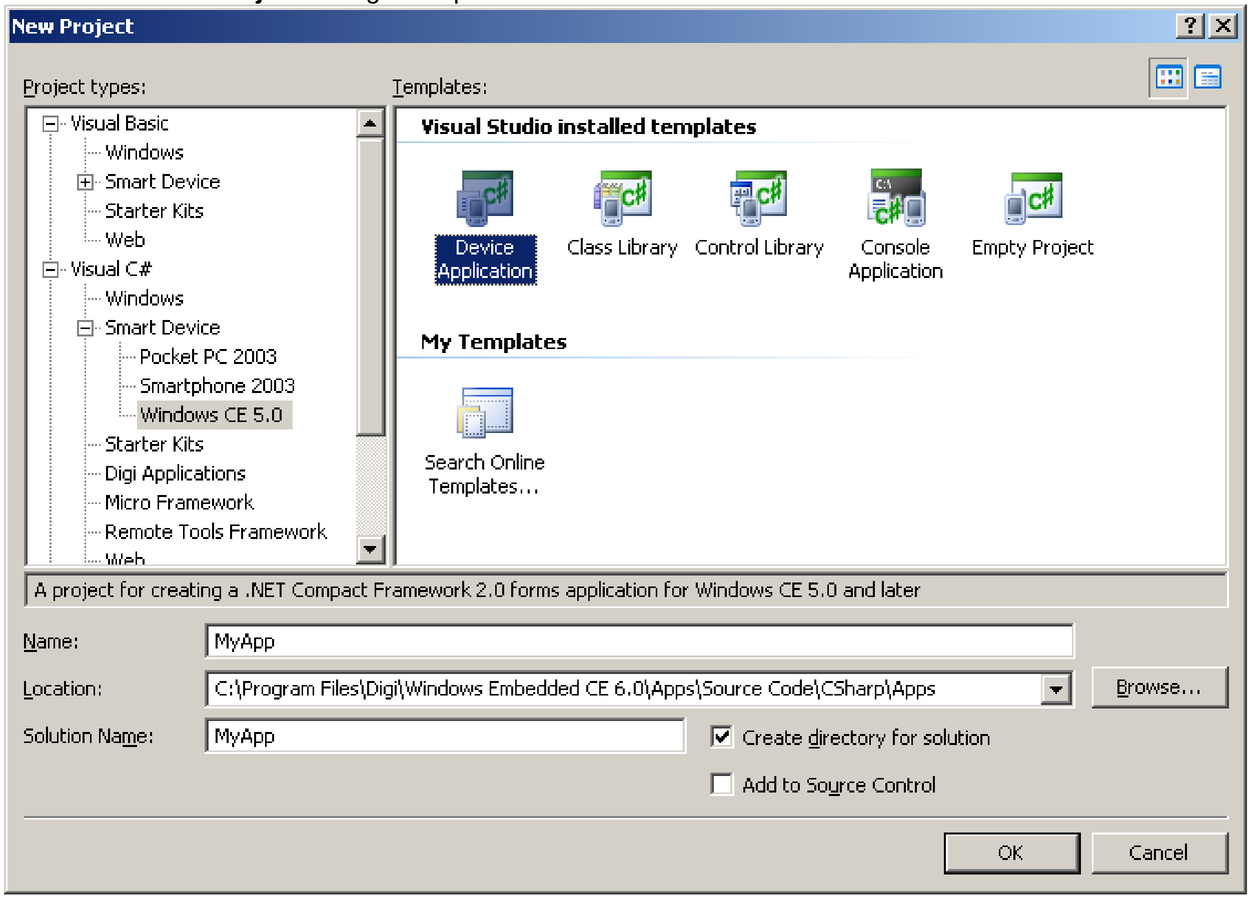
- **Apps**: Demo application source code
- **Bin**: folder that contains the compiled dlls
- **Doc**: Documentation of the API.

2. Develop an application with Visual Studio 2005 and Digi C#API

This task creates and builds a simple application using Visual C#® and the Digi C# API. In this example the user learns how to integrate the Digi C# API into his project with an application that manages the gpio driver in ConnectCore MX51 platform.

2.1. Create the project

1. Select **File > New > Project**.
   The **New Project** dialog box opens:



New Project

Project types:

- Visual Basic
  - Windows
  - Smart Device
  - Starter Kits
  - Web
- Visual C#
  - Windows
  - Smart Device
    - Pocket PC 2003
    - Smartphone 2003
    - Windows CE 5.0
  - Starter Kits
  - Digi Applications
  - Micro Framework
  - Remote Tools Framework
  - Web

Templates:

Visual Studio installed templates: Device Application, Class Library, Control Library, Console Application, Empty Project

My Templates: Search Online Templates...

A project for creating a .NET Compact Framework 2.0 forms application for Windows CE 5.0 and later

Name: MyApp

Location: C:\Program Files\Digi\Windows Embedded CE 6.0\Apps\Source Code\CSharp\Apps

Solution Name: MyApp

[✓] Create directory for solution

[ ] Add to Source Control

2. Do these steps:
   In the **Project Types** tree select **Visual C#** (if Visual C# is not your default language in Visual Studio, expand **Other Languages > Visual C#**) select **SmartDevice**, and **Windows CE 5.0** (This selection is also valid for Windows Embedded CE 6.0.)

   In the **Templates** section, select **Device Application**.

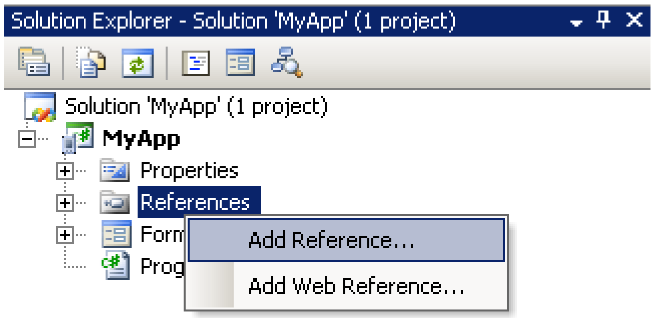
   In the bottom part of the dialog box, in the **Name** input box, enter name for project.

   In the **Location** input box, enter the location for the source files.

   When everything is entered, click **OK**.

2.2. Add the C# API dlls

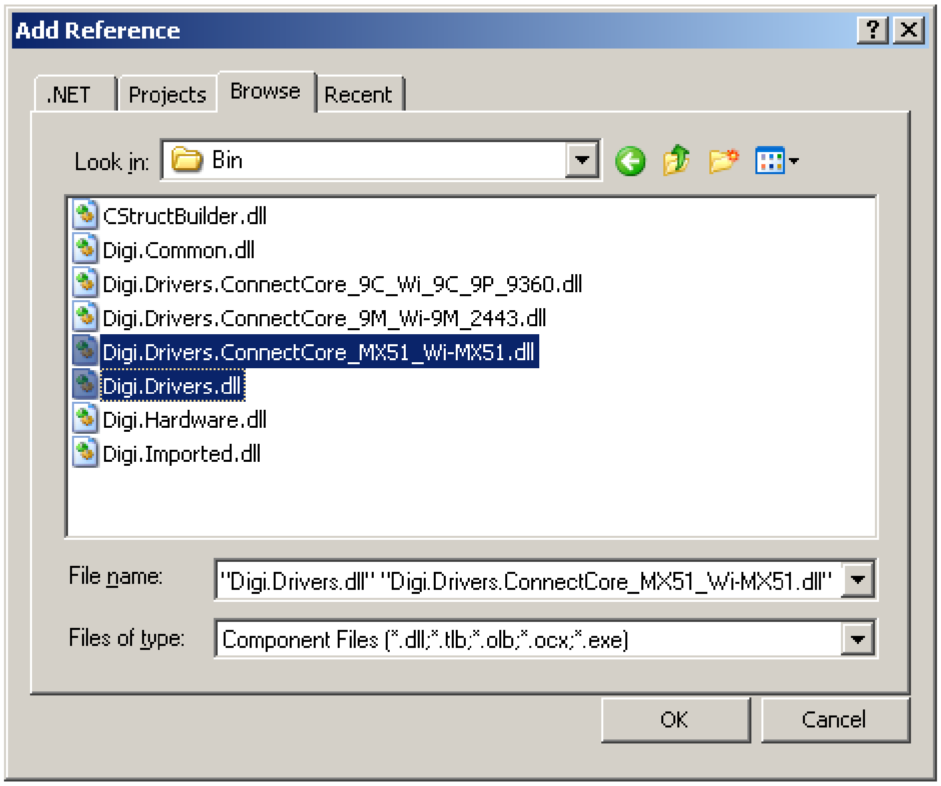
1. Select the Solution Explorer view, **View > Solution Explorer**.
2. Right-click on **References**, and select **Add Reference…**


Solution Explorer - Solution 'MyApp' (1 project)
Solution 'MyApp' (1 project)
MyApp
Properties
References
Form
Prog
Add Reference…
Add Web Reference…

The **Add Reference** dialog box opens:

3. In the Browse tab navigate through the browser to the Bin folder in **%Program_Files%\Digi\Digi C Sharp Embedded Library\Bin** and select **Digi.Drivers.dll, Digi.Drivers.<Platform>** and click **OK.**

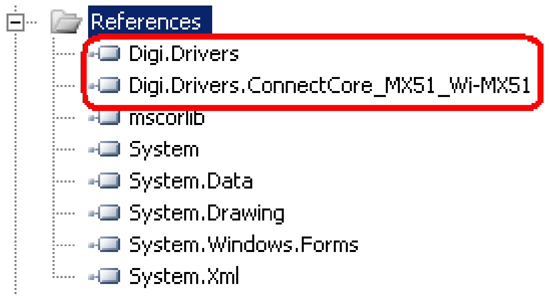
<Platform> will be substituted with the platform you are going to develop for.



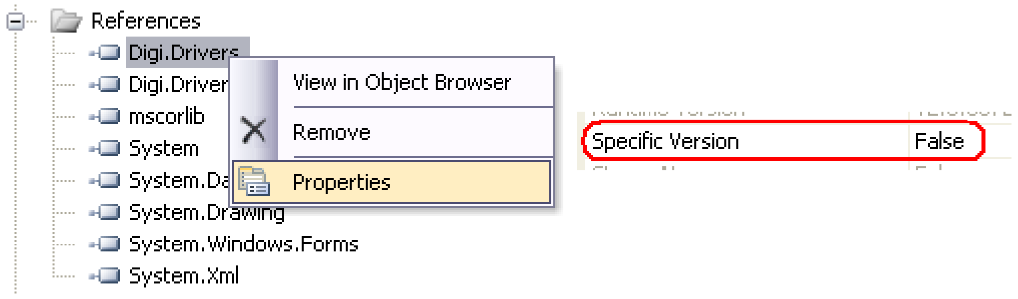
CStructBuilder.dll
Digi.Common.dll
Digi.Drivers.ConnectCore_9C_Wi_9C_9P_9360.dll
Digi.Drivers.ConnectCore_9M_Wi-9M_2443.dll
Digi.Drivers.ConnectCore_MX51_Wi-MX51.dll
Digi.Drivers.dll
Digi.Hardware.dll
Digi.Imported.dll

File name: "Digi.Drivers.dll" "Digi.Drivers.ConnectCore_MX51_Wi-MX51.dll"
Files of type: Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe)

4. See that the Digi.Drivers.dll and the Digi.Drivers.ConnectCore_MX51_Wi-MX51 are added in the Reference tree:



5. In each dll you add be sure you set the **Specific Version** property to **false**. (To see the properties of the dll right click on it select properties)



Specific Version False

If Specific Version property is set to true, you must work with the version specified in the Specific Version property, if not you will have build or run errors. For this reason is strongly recommended to set it to false

6. To be able to use dll's classes, in the top of the main file (in the example case Form1.cs) add the following code:

```
using Digi.Drivers.ConnectCore_MX51_Wi_MX51;
```

2.3. Develop the application

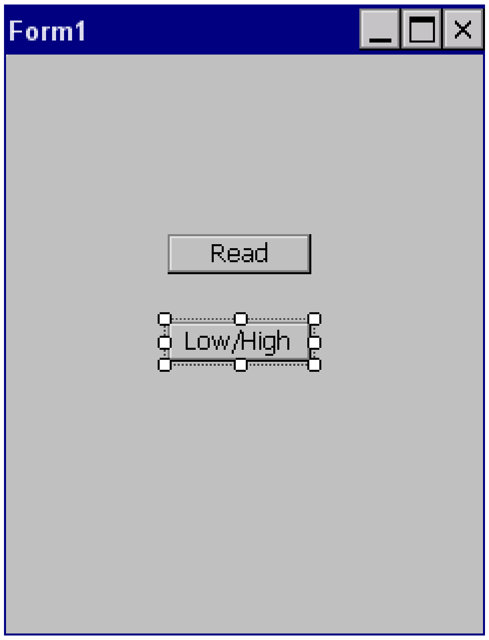
The application consists in a gpio configured as output (to change his value Low/High) and a button that shows through a message box the value of a gpio configured as input.

2.3.1. Generate the interface

Now add some information in the form to generate the interface:

1. To open the toolbox, select **View > Toolbox**.

2. Drag and drop two buttons into the empty form so the interface looks like this:



3. Right-click the first button and select **Properties**. Then, in the **Appearance > Text** field, enter **Read** as the new text. Right-click the second button and select **Properties**. Then, in the **Appearance > Text** field, enter **Low/High** as the new text.

2.3.2. Generate the source code

Next, add some code in the source code and buttons's click methods.

1. To open the first button's click method source code, double-click the first button on the form. Code similar to this is displayed:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Digi.Drivers.ConnectCore_MX51_Wi_MX51;

namespace DeviceApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {

        }
    }
}
```

2. The same with the second button, double-click the second button on the form. Code similar to this is displayed:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Digi.Drivers.ConnectCore_MX51_Wi_MX51;

namespace DeviceApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {

        }
        private void button2_Click(object sender, EventArgs e)
        {

        }
    }
}
```

3. Now define the two gpio variables:
   Configure one gpio variable as input.
   Configure the other gpio as output and assign it the value “Low”.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Digi.Drivers.ConnectCore_MX51_Wi_MX51;

namespace DeviceApplication1
{
    public partial class Form1 : Form
    {
        GPIO Input1 = new GPIO(GPIO.Port.Port3, 6, GPIO.DirectionValue.Input,
                                   GPIO.IoValue.Low);
        GPIO Output1 = new GPIO(GPIO.Port.Port3, 10, GPIO.DirectionValue.Output);

        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {

        }
        private void button2_Click(object sender, EventArgs e)
        {

        }
    }
}
```

4. Fill the methods, button1_Click correspond to button **Read**, we add the code to Read the Input1 value and show it.

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Button1 value: " + Input1.Value.ToString());
}
```

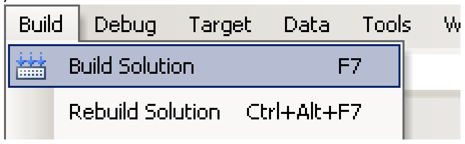
5. The button2_Click correspond to button Low/High, add the code to change the output value of the Output1:

```
private void button2_Click(object sender, EventArgs e)
{
    if (Output1.Value == GPIO.IoValue.Low)
         Output1.Value = GPIO.IoValue.High;
    else
         Output1.Value = GPIO.IoValue.Low;
}
```

6. Save the file

2.4. Build the sample application

To build the sample application, select **Build > Build Solution**:



When the build completes, messages in the output window (**View > Output**) indicate:

- That the build process was successful
- The location of the executable image

3. Transfer the application to the target

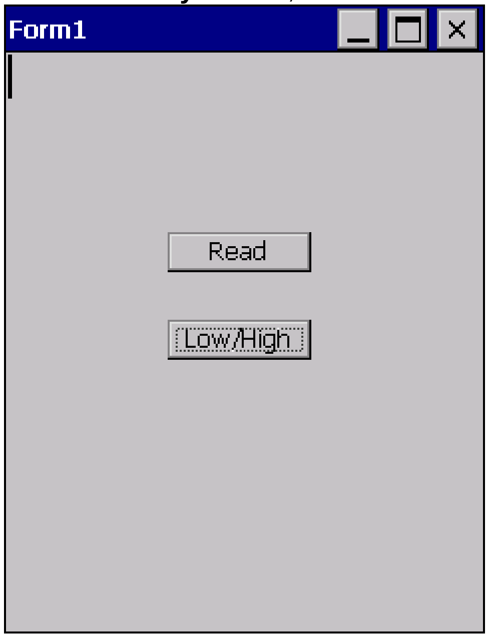
Copy the dlls and the exe file of the application to the target. There are located it in the folder bin/Release of the project.
To run the application correctly are necessary MyApp.exe, Digi.Common.dll, Digi.Drivers.ConnectCore_MX51_Wi-MX51, Digi.Drivers.dll Digi.Imported.dll and CStructBuilder.dll.

In the develop Digi.Drivers and Digi.Drivers.ConnectCore_MX51_Wi-MX51 are added to the project, but this dll uses other dlls that are necessaries to run the application correctly.

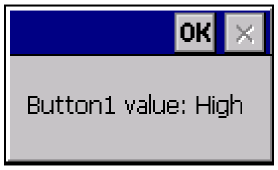
4. Run the application

Now go to the target device and run the application.

1. On the target's desktop, double-click **My Device**, and the double-click **MyApp**.



2. Click the **Read** button to read the value of the Button1.
   A message box is displayed with the value of the Button1:



Button1 value: High

3. Click the **Low/High** button to change the value of the gpio configured as Led1, you will see that the gpio changes his value

4. Close the application by clicking the **X** on the application's title bar.