

Working with Phidgets in WinCE

Environment and Libraries

First, we need to set up the proper environment and get the necessary files off the Phidgets web site. Visit the drivers and programming section at www.phidgets.com and get the latest:

- Phidget Framework
- Phidget21 WinCE Libraries
- ARMV4I or x86 Framework, depending on the target system
- C# - .NET Compact Framework Code Samples

You will need the C# .NET CF examples or the Phidget21CELib to use Phidgets on the target system, and the Phidget Framework to program with Phidgets on the development machine. The ARMV4I or x86 Framework is for installing the Phidget Drivers on the target machine. We also recommend that you download from the programming section the following reference materials:

- .NET or C API Manual
- Programming Manual
- The Product Manual for your device

The .NET and C API manual contains calls and events for every type of Phidget and can be used as a reference. You can find a high level discussion about programming with Phidgets in general in the Programming Manual. The Product manual for your device also contains an API section that describes limitations, defaults, and implementation details specific to your Phidget. You may want to have these manuals open while working through these instructions.

Setting up a Phidgets Project

This tutorial was written with Windows CE 6.0 and the .NET compact framework in mind. Other development environments such as Windows CE 5.0 should work as well and would be set up in a similar manner. Please see the getting started guides available on the Phidgets website for development under a particular language. This tutorial only touches on Windows CE development specifics.

Begin by installing the ARMV4I or x86 Framework on the target device. Normally this can be accomplished by simply executing the .CAB file on the target system. Otherwise, the Framework may need to be included in a custom OS image (briefly discussed under the section titled "Compiling Your Own OS Build"). Afterwards in Visual Studio, create a smart device project for the language of your choice and then add a reference to the phidget21CE.NET.dll from the examples (or the phidget21CE .lib if developing under C/C++).

Coding for Windows CE

Due to the way Phidgets uses threading and the way the .NET compact framework is designed, you need to be aware that making changes to certain controls like the GUI needs to be handled in a special way. Phidget Event handlers run on separate threads, so the following

code uses Invokes from them to manipulate the GUI in Visual Basic:

```
Protected Delegate Sub PrintGUI_delegate(ByVal arg As String)
    Private Sub PrintGUI(ByVal arg As String)
        TextBox1.Text = arg + TextBox1.Text
    End Sub

    Private Sub PhidgetInterfaceKit_Attach(ByVal sender As Object, ByVal e As Phidgets.
Events, AttachEventArgs) Handles m_ifkit.Attach
        Dim s As String = "Attached: " + e.Device.SerialNumber.ToString()
        Me.Invoke(New PrintGUI_delegate(AddressOf PrintGUI), s)
    End Sub
```

Deploying the Program

For simple programs in .NET, if the compiled Windows CE program runs on the development machine, then it should behave the same way on the target machine with the exception of the need for Invokes. It is also recommended that the program be debugged over a remote connection to the device. Make sure to include the phidget21CE.NET.dll with your distribution.

If you receive a “Can’t find PInvoke DLL” error on deployment, check to make sure that the Phidget Framework .dlls are installed to the device (normally in the Windows folder). Running the CAB installer is the easiest way to ensure the files are installed and registered correctly.

You can use the Phidget Webservice to connect to remote Phidgets if USB ports are not available. Note that Windows CE does not support mDNS and by extension some open() calls. In .NET, use open(serial,“IPAddress”, port, “pass”).

Deploying the Program on an Emulator

Your success for running Phidgets on an emulator may vary based on the type of the device being emulated, and whether it supports USB devices or the installation of the Phidget Framework. Here, if you receive a “Can’t find PInvoke DLL” error during execution, this means the emulator image does not have the Phidget Framework installed. One approach installing it on an emulator is to set up a shared folder (File | Configure... | General | Shared folder) with the ARMV4I or x86 Framework inside, then browse to its location and run the installer. Also, if you’re running the Phidget Webservice, you may want to have the network connection shared between the PC and device through ActiveSync.

Compiling Your Own OS Build

Often you’ll also find that after power cycling the device, the state of the operating system is reset and the Phidget drivers are lost. As a matter of convenience, it’s possible to include the Phidget drivers into the OS build so that they do not need to be reloaded every time. Building an OS image is often a complicated task and is not covered in detail here. However, if you are going to try this process it will involve using the SDK for the device, including the ARMV4I or x86 .dlls, and inserting entries into the registry for USB detection of Phidgets. The easiest way of getting the .dlls and registry entries is to copy them off the device after installing the CAB file. The CE kernel source is also available off the Phidgets web site.