

Newbie's Guide to Industrial PC's

As an experienced Hardware and Software Engineer, when a new project came up; to design a system using an Industrial PC, I had a lot of questions and not many answers. I decided on the UNO-2059E series for the embedded PC, because of its reliability and robust design.

There was a major choice to make - what operating system to use; WinCE or Windows XP Embedded. It was not clear what the advantages or disadvantages either OS would have for this project.

I tried to accumulate all I could on WinCE and WinXPE. I was able to find a few books on WinCE, but little on the newer WinXPE. The price difference was minimal and this project was on a very tight schedule. Since WinCE required some unique development tools and I already had the tools to develop Windows XP applications, I decided on WinXPE.

While waiting for the UNO delivery I had some trepidation about what to expect from the OS. I was planning on using VB.Net 2005 to develop our software but wasn't sure what differences I might encounter between developing for a desktop PC and developing for the UNO.

After receiving the UNO I plugged in a spare monitor and keyboard to its ports and powered it up. What a surprise! It was obvious that it ran a fully functional version of XP. All I had to do was develop my application on my desktop system then simply copy the .exe file to the UNO's Flash disk. No muss no fuss. Everything worked the first time. No learning curve for new software. No complicated linking or building options. It just worked right out of the box.

Based on my experience, if you're a developer that's comfortable with writing desktop applications, I would recommend using WinXPE over WinCE. I'm sure there are compelling reasons for CE as well, but the benefit of using familiar development tools made development fast and easy, and you have all the power of a real XP system.

I was able to write a real time multi-threaded application that acted as a smart datalogger, pre-processor and analysis program. I used multiple serial ports running at 115K, with multiple timers, and all the standard .Net controls. Everything worked the first time with no modification and no limitations.

Just copying the .exe file to the Startup folder (under Documents and Settings) ran the program automatically when the UNO was powered up. I did not have to convert my software to a Windows Service application which would have added another layer of complication to the project.

I found the easiest way to load the program files on the UNO was to use a USB Flash memory stick. Simply plug it in and copy the files to the appropriate folders.

One unique consideration for embedded systems is that power can be abruptly removed.

Would power shutdown corrupt the data files that were writing to disk?

After running extensive tests, cycling power on and off while continuously writing to disk; the UNO never corrupted any files.

All in all, I was very pleased with the operation of the UNO and the development environment.

I developed everything on my PC with full debugging capabilities and no new environment to learn and simply copied the executable application over to the UNO with no changes or other complications.

Frank Fortino
Contract Engineer



International Headquarters: 707 Dayton Road - PO Box 1040 - Ottawa, IL 61350 USA
815-433-5100 Fax 433-5104 www.bb-elec.com orders@bb-elec.com support@bb-elec.com

European Headquarters: Westlink Commercial Park - Oranmore, Co. Galway, Ireland
+353 91 792444 Fax +353 91 792445 www.bb-europe.com orders@bb-elec.com support@bb-europe.com