Integration of TINE Control system inside Windows CE .NET

Andres Pazos (andres.pazos@embl-hamburg.de)

EMBL-Hamburg/DESY

April 2008

Version: 1.0

Abstract

This report provides a general description of the Windows CE operating system, as well as the existing tools for developing software inside this platform. In order to integrate TINE Control System a cross-compilation between desktop Windows and Windows CE has been done. A beta version is already available to operate the TINE Control System inside the Windows CE operating system

Contents

Windows CE Introduction

Windows CE is a 32-bit architecture operating system (OS) released by Microsoft Windows for embedded devices. It is a real-time, lightweight, multithreaded and composable OS, with optional graphical user interface. It is used in very different applications, from cellular phones to industrial controllers. It is not backward compatible with MS-DOS, and it covers only a subnet of the Win32 API. So the first think we learn from using Windows CE is that is not Windows. This means that specific libraries and software development kits (SDK) are necessary.

The newest release is the Windows CE 6.0 (2008). This has been an important improvement in terms of available resources of the operating system, always trying to maintain it as a small-footprint. Now is possible to have up to 32000 processes (before 32) and the virtual memory has been increased up to 2 GB (before 32 MB). Windows CE is multiplatform, supporting x86, MIPS, ARM, ARM, SH3, SH4, architectures

It is important to understand the concept of a device inside the Windows CE world. Because of Windows CE is made of components, it is possible to select the components to be running in an embedded hardware. For example, it is possible to choose if it needs network support, USB support, etc, according to the hardware dependencies. So a device (like Microsoft Mobile) releases a specific distribution of the OS Windows CE with a specific selection of components.

Other of the interesting points of Windows CE is the real-time capabilities. It has a priority based preemptive thread scheduler. Supports 256 levels of priority (0 is the highest and 255 the lowest) and synchronization objects like semaphores, mutex and critical sections. It implements an interrupt model based on an Interrupt Service Routine (ISR) and an Interrupt Service Thread (ITR). The ISR is loaded into the kernel and is assigned to a particular IRQ. It is called immediately to handle the hardware interrupt and notifies the kernel the IST to run. The IST is created by a process or by a driver. It is schedule like a normal thread in the system, and is responsible of performing the massiveness of the interrupt handling work. In order to implement real-time applications is essential to understand the hardware and the operative system itself. The user interface calls has to be separated from the real-time threads. MSQueues and shared buffers are available to communicate them. Windows CE provides the tools and libraries to develop real time applications.

Programming Windows CE Applications

The standard way of proceeding in order to develop a Windows CE application is to implement the software in a desktop PC. The first step is to choose the development environment. There are different tools available.

Microsoft embedded Visual Studio 4.0

This development environment is freely available from Microsoft download web page [1]. It is similar to the Microsoft Visual Studio 6.0. It is possible to choose between Visual C++ and Visual Basic programming languages. It provides, through the Platform Manager, the possibility of configuring the connection to the Windows CE device. It includes extra remote tools to control your windows CE device, like the kernel tracker, the process manager, the stack viewer, etc.

Microsoft Visual Studio .Net (2005 or later)

By creating a “Smart Device” inside Microsoft Visual Studio .NET is possible to compile applications to run in Windows CE. Visual C++, Visual Basic and Visual C# are the supported programming languages. Inside the *Tools* menu, there are options accessible to configure the connection to the Windows CE device.

In both cases a SDK of the specific Windows CE device has to be installed before starting a project. This is normally provided by the device manufacture and not by Microsoft. There are also standard SDK that support the operative system in general and are provided by Microsoft [2].

Before testing the software into the real device, is very convenient to use the Windows CE emulator [3]. Like with the SDK, there are specific device emulator and general ones. When you want to develop a hardware application or a network application the emulator can be useful only in first term.

It is also possible to debug the application running it directly in the device. The prerequisite is to establish a connection between the desktop PC and the Windows CE device.

There are other programming languages supported under Windows CE like Java. It is necessary to download a virtual machine compatible with your windows CE. There are also multiplatform virtual machines.

Other Windows CE Software

In this section we mention other Windows CE software available for the users and developers.

Active Sync

This is the most common tool to establish a connection between the Windows CE OS and the OS running in the desktop PC [4]. It supports USB connection, for what you need a special USB network cable [5], and serial connection. Other way of connection is using a network cable, but for that you need to set up first the connection with one of the other methods.

There is a way to connect your device through a network connection (not using Active Sync), following the next steps:

- Inside the Windows CE run the Conmandclient2.exe and the cMaccept.exe command
- Inside the desktop PC, run from Visual Studio: *Tools->options->device tools->devices* the *configure* option. Choose the *Configure transport TCP/IP* dialog and write the IP of you CE device.

Platform Manager

Platform Manager is the software that you must install in your desktop PC to create a new Windows CE device. You can use the wizard to select the supported components of you future CE device. As a result you will create a Windows CE image, a SDK and an emulator of the created device.

Remote Connection

There is a Remote Display Control for Windows CE (CeRHost.exe) that allows you to connect remotely to you CE device.

All the desktop software available is only for Windows family OS. There is not much support for Linux machines.

TINE inside Windows CE

TINE (Three-fold Integrated Networking Environment) is a distributed control system created by DESY [6]. It is multi-platform, multi-protocol, multi-architecture and supports many programming languages. It is possible to interconnect many devices through the network using TINE, but also offers general services of archiving, history, alarming and logging. For more information refer to [7].

TINE currently supports many of the existing operating systems, but it was lacking a version for the Windows CE platform. With this purpose a cross-compilation of the Windows Desktop version has been done. A first beta version has been released for the architecture x86 (February 2008). The service pipe and remote pipe options are not available for the Windows CE version. Two new files have been added to wrap the not supported windows functions under CE (wincelib.c and wincelib.h). A Windows CE version of the classical sineServer has been also compiled and tested under Windows CE 6.0. The CDI (common device interface) library of TINE with the TwinCAT plug has been also cross-compiled and tested with real hardware under CE. Most part of the changes in the porting were related with string functions and with the time library. This is because Windows CE uses Unicode coding, so in many cases it is necessary to convert ANSI functions or use the TCHAR library [8].

Use Case

Our experience is focus in running windows CE as the operative system of an embedded industrial controller. We are using an embedded PC provided by Beckhoff [9] of 1 Ghz CPU and 1 GB of memory. The operative system size is less than 20 Mbytes. This embedded PC is the controller of a line of devices connected using a Real-Time industrial Ethernet bus, called EhterCAT [10]. This includes stepper motor controllers, digital input/output, analog input/output and encoder counters. The objective of running the TINE control system inside Windows CE, is to be able to run a device server embedded in the PC and export the functionality of the connected devices through the control system. With this architecture we have a compact solution that runs close to the hardware. The server connection to the hardware is independent of the network. Real time operations are possible running the server inside the Windows CE OS. The graphical interface is optional and remote connection to the device is available.

It is also potential to run TINE clients inside embedded devices (like PDAs and cell phones). This is a new option for mobile devices to be able to access the control system.

The following figure (figure 1) shows an overview of the system architecture.

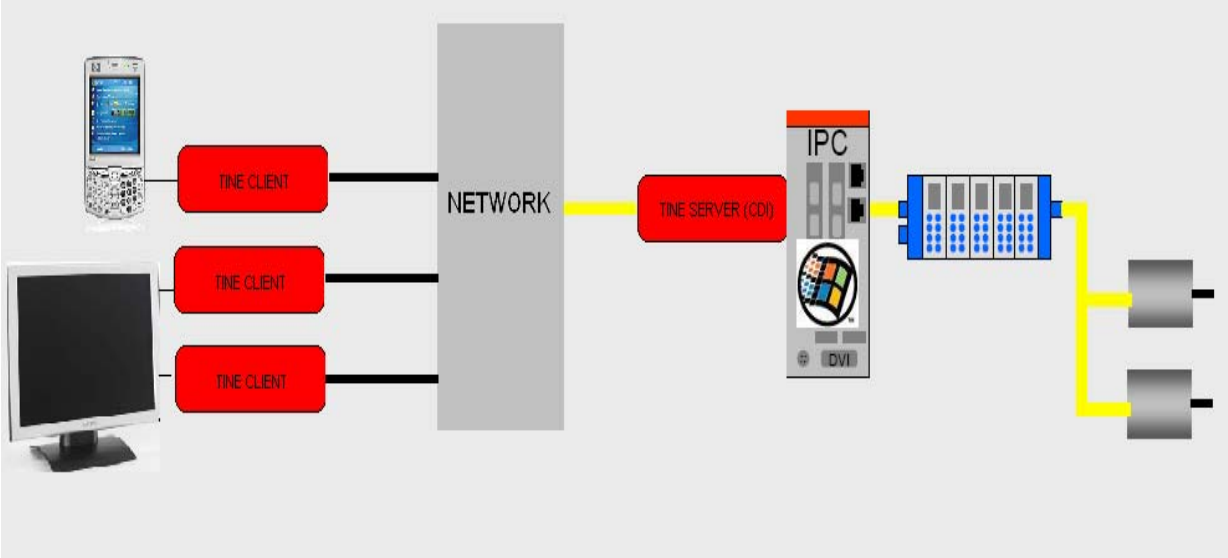

Figure 1. embedded Control System Architecture

Links and references

- [1] eMbedded Visual C++ 4.0 download web page: http://www.microsoft.com/downloads/details.aspx?FamilyId=1DACDB3D-50D1-41B2-A107-FA75AE960856&displaylang=en
- [2] Windows CE 5.0 Standard Software Development Kit: http://www.microsoft.com/downloads/details.aspx?FamilyID=fa1a3d66-3f61-4ddc-9510-ae450e2318c3&DisplayLang=en
- [3] Windows CE 5.0 emulator: http://www.microsoft.com/downloads/details.aspx?FamilyID=A120E012-CA31-4BE9-A3BF-B9BF4F64CE72&displaylang=en
- [4] Active Sync 4.2: http://www.microsoft.com/downloads/details.aspx?FamilyID=7269173a-28bf-4cac-a682-58d3233efb4c&DisplayLang=en
- [5] USB-USB Cable: http://www.hardwaresecrets.com/article/248
- [6] DESY: http://www.desy.de
- [7] TINE Control System: http://tine.desy.de
- [8] Writing portable code: http://www.metagraphics.com/index.htm?page=pubs/mgct_language-portable-code.htm
- [9] Beckhoff: http://www.beckhoff.de
- [10] EtherCAT Technology Group: http://www.ethercat.org
- *Programming Microsoft Windows CE .NET*. Douglas Boling, Microsoft Press 20033
- *Windows CE technical resources*: http://msdn.microsoft.com/en-us/embedded/aa714543.aspx
- Technical Articles: http://www.metagraphics.com/index.htm?page=pubs/mgct_language-portable-code.htm
- Windows CE developers FAQ: http://www.megawap.ru/~wap/dev/ce_dev_faq.htm
- Real Time benchmark (need registration): http://www.dedicated-systems.com