

# EMBEDDED SYSTEMS PROGRAMMING\*

*Guillermo A Francia, III  
MCIS Department  
Jacksonville State University  
700 Pelham Rd North  
Jacksonville, AL 36265  
gfrancia@cs.jsu.edu*

## INTRODUCTION

The proliferation of embedded systems presents a new challenge to information technology professionals. In just about everything that we use—from automobiles to microwave ovens—embedded systems are seamlessly integrated. In a network economy, embedded systems have found yet another place in which they can further assert their pervasiveness. Although traditionally, these systems are used in isolated devices, the present trend indicates the steadily growing number of these devices that are getting connected to networks and, most importantly, the Internet.

Embedded systems present a new challenge and opportunity to the industry and the academe as well. The increasing demand of the industry for embedded system-trained professionals creates yet another academic frontier where the disciplines of computer science, technology, and engineering can collaborate.

In what follows is a description of an embedded systems programming course, its support laboratory activities, and the affordable resources that were utilized to provide a meaningful application-oriented environment for the students taking the course.

---

\*Copyright © 2001 by the Consortium for Computing in Small Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing in Small Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

## EMBEDDED SYSTEMS PRIMER

An embedded system is a combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a specific function [9]. More and more of the equipment that we use in our everyday existence have embedded systems in them. Such equipment include, but are not limited to, microwave ovens, automobiles, laser printers, network routers and switches, medical diagnostic tools, and barcode scanners.

Embedded system design and development varies greatly depending on the hardware and software components that are required. It can be as simple as writing an application for an existing device or as complicated as designing, prototyping, and implementing a complete embedded system from the ground up on both component aspects: hardware and software.

Interested readers may find an excellent introduction to embedded systems in [3].

## EMBEDDED SYSTEMS DEVELOPMENT TOOLS

### Real-Time Embedded Systems

The development tool that was used to accomplish a laboratory activity on real-time embedded systems is the ToolSuite Lite. This tool is available from Phar Lap Software Company and is bundled with the textbook by Grehan, Moote, and Cyliax [1].

To configure the embedded system, one needs two computers: the first as the development station and the other as the target station where the embedded system application would run. These computers will communicate through a direct serial or parallel connection.

The ToolSuite Lite includes the real-time Embedded ToolSuite (ETS) kernel, a real-time operating system for running real-time embedded applications. The kernel is divided into a monitor and a set of libraries that are linked with the embedded application. These libraries include, among others, the Win32 APIs, specialized ETS APIs, floating point emulation, and WinSock. The scheduler that is part of the kernel supports real-time multithreaded applications. In addition, it is designed for deterministic scheduling to make it more suitable for real-time operation.

### Palm Programming

The development kit used in Palm programming is the CodeWarrior Lite by Metrowerks. This is bundled with the book: Palm Programming: The Developer's Guide [2]--the required textbook for the

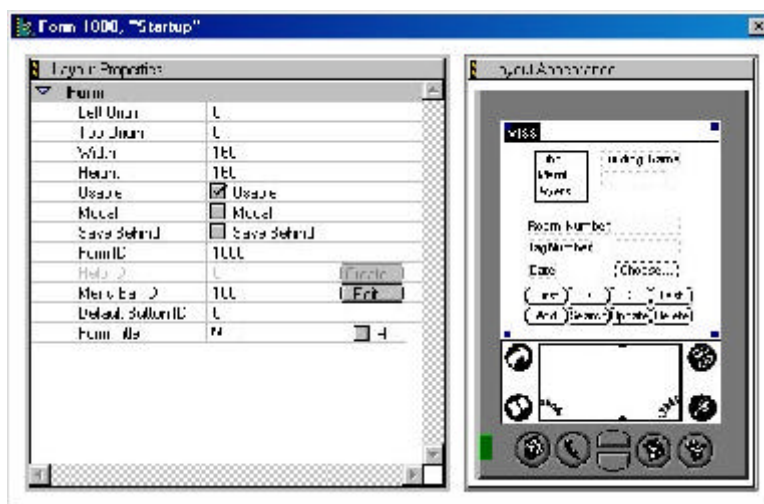


Figure 1. The Constructor IDE

course. Of course, there exist other development tools that can be used as well—it is simply a matter of developer's preferences. Some of these are the GCC toolkit from the Free Software Foundation [4], the Pendragon Forms from Pendragon [5], the Java 2 Micro Edition from Sun Microsystems [13], and the Satellite Forms from SoftMagic [6].

CodeWarrior Lite is a C/C++ development environment. It comes with an extremely useful visual tool for rapidly assembling user interfaces: the Constructor. This tool is not as elegant and easy to use as the visual tools found in a Visual Basic IDE but serves its purpose well. Figure 1 shows an instance of the Constructor IDE.

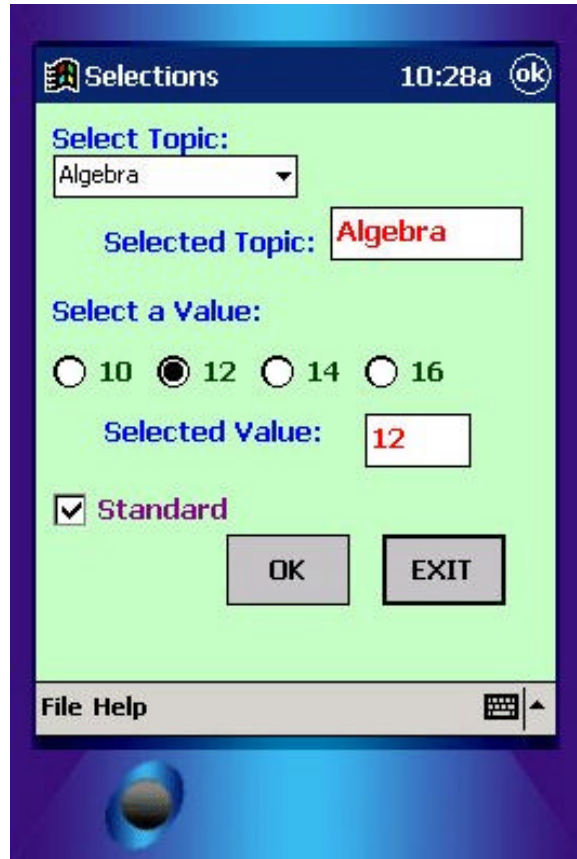


Figure 2. The Pocket PC Emulator

## Windows CE Programming

The development kit used in Windows CE programming is Microsoft's Embedded Visual Tools 3.0, which is available free of charge from the Microsoft web site [7]. The tools include Embedded Visual Basic 3.0, Embedded Visual C++ 3.0, and three (3) SDKs for each of the following CE devices: the Handheld PC Pro, the Palm-size PC, and the Pocket PC. Furthermore, the kit provides an emulation environment that can be used to develop Windows CE applications using a desktop system. The emulator is shown in Figure 2.

Embedded Visual Basic 3.0 is practically a scaled-down version of Visual Basic 6.0. Learning to use this development tool is very straightforward. Those developers who are familiar with Visual Basic will be able to put together a simple Windows CE application almost instantly.

## EMBEDDED SYSTEMS PROGRAMMING PROJECTS

### Real-Time Embedded Systems Project

The real-time embedded systems project is an adaptation of the Uninterruptible Power Supply (UPS) simulation project described in [1]. The project calls for the simulation of a UPS system that monitors an AC power source and provides emergency power to its load of one or more devices. The design of the system is divided into four (4) basic subsystems: event, windowing, keystroke, and timer. The event subsystem processes and dispatches events and interrupts. The windowing subsystem, shown in figure 3, emulates the actual display of a UPS system. The keystroke subsystem provides low-level key-input processing. The timer subsystem generates timer events to be handled by the event subsystem.

This project requires a good bit of familiarity of C++ and the ETS APIs. These APIs cover much of the routines needed in real-time systems such as event handlers, interrupt handlers, timer generators, graphical user interface modules, process control, etc. In order to not overwhelm the students with so many details, they were given a functional version of the UPS simulation project and required that they implement some enhancement on it. One such enhancement is to implement a serial I/O communication.

### Palm Programming Project

The Palm programming project calls for the design and implementation of a portable equipment inventory system for a Palm device. The system allows the creation, maintenance, administration, and querying of an equipment database. The database is made up of the following information: building name, equipment inventory tag number, room number, equipment description, and entry date. The user interface is depicted in Figure 4.



Palm programming requires a good knowledge of C/C++ and event driven functions. The students struggled at first with C++ pointers and objects during the development stage but felt more confident and at ease after doing 2 or 3 laboratory exercises. The interested reader may wish



Figure 3. The Real-Time Project Windowing System Interface

to consult [8] and [10] for some sample laboratory exercises on Palm application development.

The most challenging task on Palm programming is the part that involves the creation and the manipulation of Palm databases. A Palm database can be created either by using a desktop computer that is synchronized with a Palm device or by invoking the Palm Data Manager function, *DmCreateDatabase()*, during run-time. On the desktop, the Palm database can be created using a Java desktop application called *Palm Desktop Database Builder (PDDB)* [10]. This tool takes an input text file and generates an equivalent Palm database (pdb) file.

The target application, as specified by the project description, is intended to create and manipulate a small Palm database. This database will contain all the pertinent information required by an equipment inventory system. A code snippet of database creation that is used in the project is shown below.

```
// Create a new database in case there isn't one
if(((error=DmCreateDatabase(0,"ContactsDB-PPGU",'PPGU',
                           'ctct', false )) != dmErrAlreadyExists) &&
    (error != 0) ){
    // Handle db creation error
    FrmAlert( DBCreationErrorAlert );
    return( 0 );
}
// Open the database
contactsDB = DmOpenDatabaseByTypeCreator( 'ctct',
    'PPGU', DmModeReadWrite );
// Get the number of records in the database
NumRecords = DmNumRecords( contactsDB );
```

The interested reader is referred to Bachmann's text [8] for an extensive treatment of Palm databases.

### **Windows CE Programming Project**

The Windows CE project requirement entails the design and implementation of a personal credit card expense tracking system. The system will create, maintain, and manipulate a database that contains the following information: the place of purchase, the date of purchase, the amount incurred, and the type of card used for the transaction. The user interfaces for this project are shown in figure 5. The development environment consists of a desktop running under Windows 2000, the Embedded Visual Basic 3.0 system, and a Pocket PC emulator.

The students felt more at ease with the development process using Embedded Visual Basic because of their familiarity with the Visual Basic 6.0 IDE.

User interfaces were created much quicker and the lines of code were written more smoothly and with less error. Testing and debugging proceeded in the same manner. The Windows CE database creation is not as tedious as that in the Palm system. The database can either be pre-built using MS-Access and downloaded on the device or be formed dynamically by creating a table through an ADOCE object recordset. The following is a code snippet that will dynamically create a database:

```

strActiveConnection = "\My Documents\" & TxtDBFileName.Text
Set objRecFields = CreateObject("adoce.recordset.3.0")

objRecFields.open "CREATE DATABASE '" & strActiveConnection
                  & "'"

objRecFields.open "DROP TABLE [myTable]"

objRecFields.open "CREATE TABLE [myTable] _
                  ([CardType] text, [Date] text, [Place] text, _
                  [Amount] text, [TaxDeduct] text)", _
                  strActiveConnection, 2, 3

objRecFields.close
    
```

## CONCLUSION and FUTURE PLANS

The previously described laboratory setups and exercises illustrate the viability and affordability of a functional laboratory for teaching and learning embedded systems programming. As indicated above, the course projects can be implemented using resources that may be acquired free of charge. It is an investment of time and effort that is well spent for pedagogical reasons. After all, the reward becomes somewhat apparent after observing that students are excited and are eager to learn when they are exposed to such a learning environment.

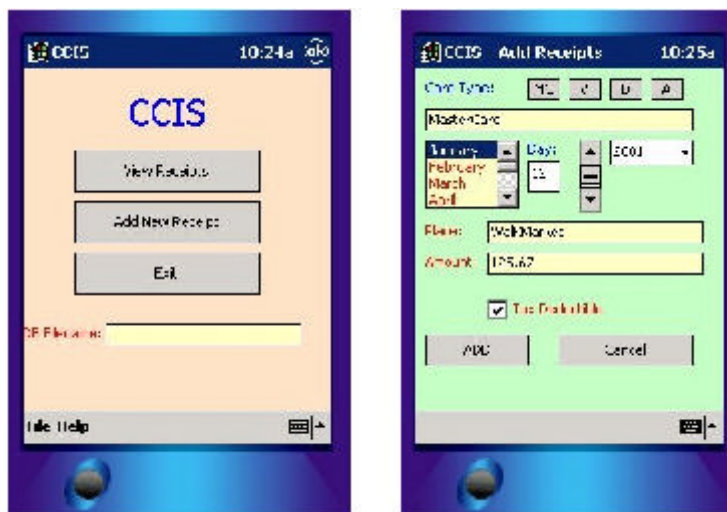


Figure 5. The Windows CE Project User Interfaces

Future enhancements of this course will include the following:

1. The acquisition of an experimental 80188 microcontroller-based board that will be used for real-time system application development.
2. The introduction of Java 2 Micro Edition [11] as a development tool.
3. The introduction of UML tools for software engineering of embedded systems [12].
4. Extend the Palm OS through shared libraries. The technique on implementing these extension packages is fully described by Winton in [14].

## REFERENCES

- [1] Grehan, R., Moote, R., and Cyliax, I. (1998). *Real-Time Programming A Guide to 32-bit Embedded Development*. Reading, MA. Addison-Wesley.
- [2] Rhodes, N. and McKeehan, J. (1999). *Palm Programming: The Developer's Guide*. Sebastopol, CA, O'Reilly & Associates.
- [3] Simon, David. (1999). *An Embedded Software Primer*. Reading, MA. Addison-Wesley.
- [4] URL: <http://www.gnu.org/fsf/>
- [5] URL: <http://www.pendragon-software.com>
- [6] URL: <http://www.softmagic.com>
- [7] URL: <http://www.microsoft.com/windows/embedded/ce/tools>
- [8] Bachmann, Glenn (1999). *Palm Programming The Authoritative Solution*. Indianapolis, IN. Sams Publishing, Co.
- [9] Barr, Michael (1999). *Programming Embedded Systems in C and C++*. Sebastopol, CA, O'Reilly & Associates.
- [10] Mann, S. and Rischpater, R. (2001). *Advanced Palm Programming*. New York, NY. John Wiley & Sons, Inc.
- [11] Giguere, Eric (2000). *Java 2 Micro Edition The Ultimate Guide to Programming Handheld and Embedded Devices*. New York, NY. John Wiley & Sons, Inc.
- [12] Douglass, Bruce (2000). *Real-Time UML: Developing Efficient Objects for Embedded Systems*. Reading, MA. Addison-Wesley.
- [13] URL: <http://www.sun.com>
- [14] Winton, Greg (1999). "Extending the PalmOS," *Dr. Dobb's Journal*, November 1999. pp.50-55.