PC-Compro – Worldwide DC Programming System

Lee, Ping Hoe and Chuah, Soo Aik

Sr. Test Systems Engineer, Sr. Staff Test Systems Engineer
Test Systems Engineering (Strategic Group)
Motorola, Penang

Email: CPL014@email.mot.com, CSC009@email.mot.com

Abstract

This paper describes the programming concept and implementation of the PC-Compro (**PC** Platform **Com**mon **Pro**gramming and Label Printing System). The main objective of this system is to provide a low cost and easy to maintain system for Worldwide DC (Radio Distribution Center) on the postponement strategy. The first radio product that work on PC-Compro is the Waris portable radio.

To address the above objective, a common system (software and hardware) platform for worldwide DC have been developed and incorporated with :

i) Microsoft Visual C++ Object Oriented Programming (OOP) language on Windows NT O/S (Operating System).
ii) System configuration files for flexible station setup.
iii) Product specific configuration files for multiple radio product supports.
iv) Profile driven test sequences to support any type of process flow.

1. Introduction

As part of the postponement strategy, WARIS mobile and portable products are incorporated a regional identifier field in the radio codeplug. This regional identifier byte will be programmed in the respective DC and not on the manufacturing line. The DC therefore requires a programming system to perform this task.

The common programming system have been discuss and brainstorming by various sites. A few systems been proposed and generally, two major platforms been presented:

i) HP-UX based PATS (Portable Automated Test Software) System.

ii) PC based Compro System

Based on analysis, PC-Compro has been selected to be the common programming platform for all DC on this postponement strategy.

2. Objectives

This project is to develop a worldwide DC common programming and label printing system in 2 months time for Waris portable for :

i) Regional identifier byte programming
ii) Nameplate verification, and
iii) Print regional label.

On top of that, the system must be able to expand to support other type of future postponement product such as Waris mobile.

3 Methodology

There are a few basic concepts and requirements that contribute to the PC-Compro.

3.1 Merging of Visual C++ OOP codes with function programming.

As mentioned, HP-UX Compro has been developed in mid of 1996 and widely use in Penang and China. By converting HP-UX platform into PC platform, it will reduce a lot of cycle time through code re-use (since the dateline that provided only 2 months).

However, MFC (Microsoft Foundation Class) in Visual C++ is more towards object oriented programming where HP-UX Compro more towards functional programming. Direct conversion is not applicable. Beside that, many low-level functions provided by HP-UX are not supported in MFC Visual C++ (it comes with own MFC Library).

PC-Compro is developed using MFC in Visual C++ and re-used codes from HP-UX Compro whichever applicable. The programming structure is actually a mix of OOP and Function Programming. The overview of the programming architecture is as shown in Figure 1.

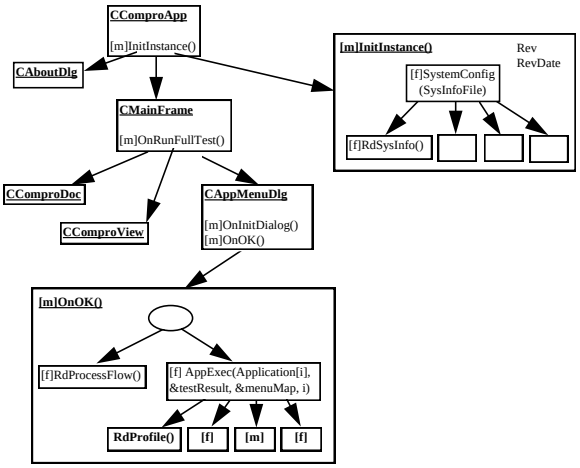

Figure 1 : Programming Flow

CComproApp, CMainFrame, CComproDoc, CComprView, and CAboutDlg are classes generated by Visual C++ AppWizard. The SystemConfig module from UNIX Compro is then insert into the member function InitInstance of CComproApp, since CComproApp is actually the starting point of the entire window program (quite similar to main() in C programming) and *CComproApp.InitInstance* is actually doing the initialization for the entire application program.

CAppMenuDlg is a Dialog Class that is developed to interface with user while running the "Full Test" (F1 or start button in the main menu is pressed). At AppMenuDlg, once the serial number barcode is scanned in, it will trigger the OnOK() member function of CAppMenuDlg. The OnOK member function will then call two major modules that are RdProceeFlow() and AppExec().

RdProcessFlow() is the module that reads in the process flow profile and AppExec() is the module that executes all the application program requested by the process flow profile.

One of the *process flow profile application* that need to be mention here is the RdProfile(). RdProfile() application is used to read in *product application profile* and extended the RdProcessFlow() to handle different product/model. The RdProfile() module is always called by the process flow profile. The technique of breaking the application loop into two parts is used to handle different process flow requirement gave by different DC.

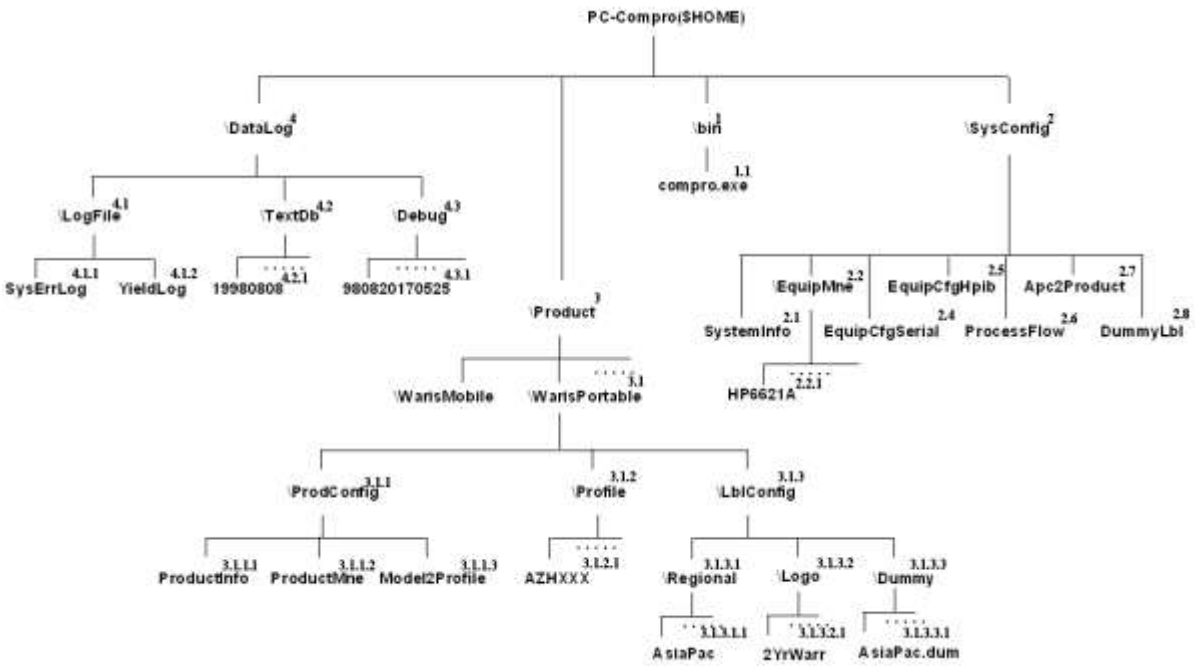

Figure 2 : Configuration File Structure

3.2 Configuration files driven software

The configuration files driven software has been widely used in most of the test software. The advantages of the configuration file driven software such as :

i) Reduce software modification time.
ii) Minimize the modification or update on source code.
iii) Updated software be delivered virtually free of defects

Figure 2 shows the configuration file structure for PC-Compro. There are 4 major directories.

3.3 Support different types of DC process.

PC-Compro is developed for all DC from any part of the world. Each DC will has their own process to handle the postponement strategy. For Waris postponement, different radio model will need different type for regional label and nameplates. So the sales model number is used to determine the test profile name. However, to decode which radio test profile to use, different DC will has their respective way to suit their environment.

To remain same single software executor for all, but able to handle any type of process, "ProcessFlow" file been introduce under system level. Same like normal test profile, the "ProcessFlow" is a list of all the test sequence to handle the different type of process on how to trigger/decode the test profile.

3.4 "Plug and Test" Concept with Standard Hardware Interface Box.

PC-Compro is designed to support multiple type of postponement product for coming future. A common hardware interface (Figure 3) is designed to facilitate the implementation of the "Plug and Test" concept. This interface box is designed with the various requirements of the multiple types of products in mind. The "Plug and Test" means only the radio fixture needs to be swapped during product changeover.

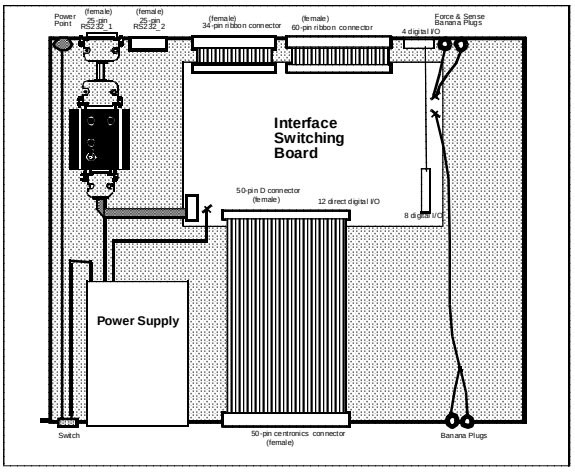

Figure 3 : Standard Hardware Interface Box

3.5 Pconplus run on same computer with PC-Compro

To support multiple products, we need the Pconplus as the protocol converter. The Pconplus allows communication between a host and a target radio. The host sends commands that are translated into a specific protocol packet for the target radio. The radio responds by sending handshaking signals that are transmitted back to the host for appropriate action. As shown in Figure 4, Pconplus is act as a middle person between the HP-UX (for PATS and HP-UX Compro) system and the radio.

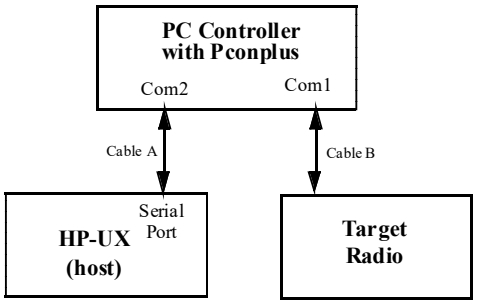

Figure 4 : Normal Pconplus Setup

For PC-Compro, both Pconplus and PC-Compro software execute on PC platform. With Windows NT operating system that can perform multi-processing at one time, we can run both software at the same machine. It will help to reduce the cost of one unit PC. As shown in Figure 4, extra serial port been used (Com3). The Com3 is act as the host where it wills transmit/receive all the data from/for PC-Compro. A "null modem" cable is connecting between the Com3 and Com2. With this technique, one unit of PC cost been saved.

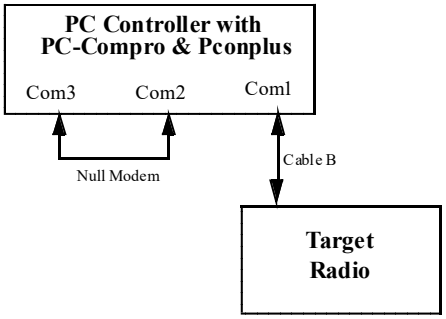

Figure 4 : PC-Compro and Pconplus run on the same PC Controller

4. Conclusion

PC-Compro is relative easy to setup and support. This well designed configuration file driven software proven that very appropriate for worldwide DC environment as there have different kind of equipment and process flow requirements.

Compare with UNIX operating system, Windows NT is easier to administrate. It is more suitable to DC type of environment where they might have less technical expertise personnel. Beside that, the cost of a PC is much lower compare with UNIX.