EVALUATING GUIPS FOR A MOBILE DIGITAL COMPANION (MDC)

Mónica Ayde Vallejo Velásquez
Juan Diego Echeverri Escobar
Luís Germán García Morales
José Edinson Aedo Cobo

Facultad de Ingeniería, Universidad de Antioquia
Grupo de Microelectrónica y Control

{monica, juan, lgermangm}@microe.udea.edu.co
joseaedo@udea.edu.co

ABSTRACT

In recent years, the implementation of graphic interfaces for mobile devices has become more important. This paper evaluates three of the most popular Graphical User Interface Platforms (GUIP) for mobile devices: Nano-X, MiniGUI, and Qt/embedded. The aim of this study is to select and adapt the most appropriate GUIP for a Mobile Digital Companion (MDC) being developed at the University of Antioquia. The parameters used in this evaluation were: memory usage, performance (number of operations per second), energy consumption, embedded web browser functionality, development tools, programming resources and technical support.

1. INTRODUCTION

The graphical interface system is an important component of mobile device software, and this system consumes an appreciable amount of energy when interactive applications are executed [1].

In this work, three of the most popular GUI platforms were implemented and evaluated in order to choose one for the Antioquia University MDC (UDEA MDC). Using Linux operating system, the following platforms were tested: MicroWindows (Nano-X) [2], Minigui [3] and Qt/Embedded [4].

In order to test the functionality, first the GUIPs were compiled on a normal PC and then cross-compiled to the UDEA MDC [5].

2. GUI PLATFORMS FOR THE DEVELOPMENT OF GRAPHICS INTERFACES

Microwindows (Nano-X) – The windows system Microwindows, now called Nano-X, is a platform made by Century Software [6] available as open-source software under MPL/GPL license.

MiniGui – This is a platform developed by Beijing Feynman Software Technology Co.Ltd [7], under GPL license. It works on different operative systems like Linux/uClinux, eCos, and uC/OS-II.

Qt/Embedded – A product of TrollTech [4] which is available with a commercial license and as Open-Source under the GPL license. This platform consists of a tools set for the construction of the Graphical User Interface.

3. IMPLEMENTATION

First, the GUIPs were compiled and emulated through a virtual framebuffer on a PC with an i.386 architecture and Linux Mandriva 2005 OS. Next, the GUIPs were tested on the UDEA Mobile Digital Companion (MDC). A full description of compilation and installation procedures can be found in [5]. The MDC is based on the MC9328MXL processor from the Freescale Dragonball family (based on an ARM920T core [8]) and a Spartan III FPGA from Xilinx, connected at the External Interface Module of the i.MXL processor. The figure 1 shows the MDC development kit used in this work.

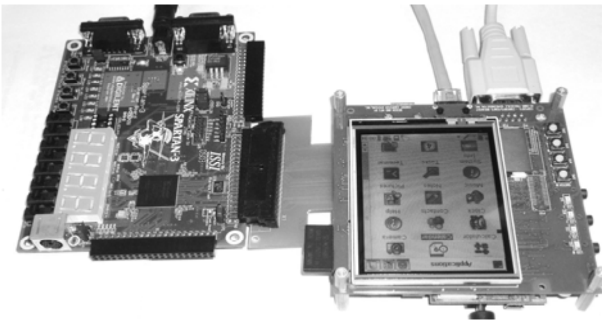

Figure 1. MDC development kit.

4. EXPERIMENTAL TEST AND RESULTS

Four Benchmarks were designed to evaluate each GUI on the develop platform. These can be downloaded from [5]. The first algorithms were codified in C++ for Qt/Embedded and in C for MiniGui and Microwindows. The fourth algorithm was written in C++ for Microwindows, using the FLNX API. The results are shown next.

Memory Space required–Table 1 shows the memory space required by each GUIP in its basic configuration.

Operations per second - The performance of the applications is measured in terms of the number of operations per second. Table 2 and Table 3 show the results.

Platform	Memory Space required	Components
Microwindows (Nano-X)	616Kbytes	Window server (nano-x) and the libraries.
MiniGui	2.8Mbytes	Window server (*mginit*) and the libraries
Qt/Embedded	8.2Mbytes	Window server and the libraries

Table 1. Memory Space required for each GUIP

Energy consumption - Defined as the product of the execution time multiplied by the power average consumed in that same period. The results obtained are shown in tables 3 and 4.

Evaluation of Web browsers and development tools - Microwindows consists of an applications development tool denominated Pixil. This provides the web browser, ViewML, but it does not work in the i.MXL platform.

MiniGui includes the Graphical User Interface, FHAS (Feynman Handheld Application Suite), and offers support for two Web browsers: Edilo, and Monqueror, but these are impossible to evaluate them.

The QT/Embedded libraries contain a highly integrated development tool named Qtopia y support the navigator Konqueror-embedded. This worked correctly in the i.MXL platform.

GUIPs	Create Window		Show Window	
	Windows /sec	Energy (Joule)	Windows /sec	Energy (Joule)
Microwindos (Nano-X)	~ 47	4,809	~56	4,019
MiniGui	~ 54	4,809	~58	4,553
Qt	~ 11	15,946	~12	15,636

Table 2. Performance and energy consumption for the create and show windows

GUIPs	Widgets		Color sequence	
	Op/sec	Energy (Joule)	Op/sec	Energy (Joule)
Microwindos (Nano-X)	~108	6,306	~ 232	11,481
MiniGui	~606	1,500	~ 212	15,604
Qt	~232	3,039	~ 666	3,21

Table 3. Performance and energy consumption for the widget and color sequence

Programming resources – Microwindows allows the use of different APIs such as nano-X, Win32 or FLNX with the object to facilitate the programming task.

MiniGui has a development environment (MFD [MiniGUI face designer]) that is very easy to use.

Qt/Embedded has tools for the creation of projects and an integrated graphical development environment for the design of applications.

Technical support - a) Microwindows has little documentation and the most recent reports about this platform were made in 2000; b) Most of the information available about MiniGui is in Chinese; similarly most support Forums are in Chinese. Also, there are not many developers that are working with this platform; c) Qt/Embedded has really good documentation and a great community of developers, who offer support through various forums.

5. CONCLUSIONS

The study and the characterization of GUIPs make in this work suggest, from the programming point of view, that Qt/Embedded has the most attractive interface and the best development suites. It offers an excellent technical support and its memory size can be reduced. On the other hand, MiniGui presents the best relationship between operations/sec and energy with benchmarks used. However, the typical applications of mobile devices are not commonly executed repeatedly. They are determined generally by the interaction with the user. Therefore, this characteristic offers opportunities to improve the power consumption of GUIPs.

6. ACKNOWLEDGEMENTS

The authors thank to the University of Antioquia and COLCIENCIAS under agreement 385-2005 for the support for this work.

7. REFERENCES

[1] L. Zhong and N.K. Jha, “Graphical User Interface Energy Characterization for Handheld Computers”, Proceedings of the 2003 International Conference on Compilers, Architectures and Synthesis for Embedded Systems, October 2003.

[2] Microwindows. http://www.microwindows.org

[3] MiniGui. http://www.minigui.org

[4] Qt/Embedded. http://www.trolltech.com

[5] Mobile Digital Companion (MDC) (http://microe.udea.edu.co/proyectos/DMA/)

[6] Century Software. http://www.censoft.com/index.html

[7] Beijing Feynman Software Technology Co., Ltd. “MiniGui User Manual - MiniGui V1.6.x (Version 1.6.8 and later)" Oct. 2005.

[8] Freescale Semiconductor, Inc. “MC9328MXL i.MX Integrated Portable System Processor Reference Manual”, 2004. http://www.freescale.com