Castillo, N. V.; Torres, D.; Gutiérrez, H.
Development and testing tool for paper handling and/or printer devices
Journal of Applied Research and Technology, vol. 1, núm. 1, abril, 2003, pp. 40-52
Centro de Ciencias Aplicadas y Desarrollo Tecnológico
Distrito Federal, México

Available in: http://www.redalyc.org/articulo.oa?id=47400104
DEVELOPMENT AND TESTING TOOL FOR PAPER HANDLING
AND/OR PRINTER DEVICES

N. V. Castillo¹, D. Torres¹ & H. Gutiérrez²

¹ CINVESTAV del IPN GDL
² Hewlett Packard de Mexico

Received: February 13th 2001 and accepted March 5th 2002

ABSTRACT

A useful development and testing tool for Hewlett Packard paper handling and/or printer devices is presented. The system provides a microprocessor based general architecture integrated by two PCB cards and a software application, used to improve new product design and testing. Paper handling devices operation is based on DC motors, stepper motors and sensors components, so a graphical user interface was developed in order to configure, drive, and test them. The implemented tool allows a time reduction greater than 50% for the development and test of prototypes.

RESUMEN

Se presenta una herramienta para la prueba de dispositivos Hewlett Packard de manejo de papel y/o impresión. El sistema tiene una arquitectura de microprocesamiento integrada por dos tarjetas PCB y un programa para mejorar el diseño y la prueba de nuevos productos. La operación de los dispositivos de manejo de papel está basado en motores DC, motores de pasos y sensores, por lo que se desarrolla una interfaz de usuario gráfico para configurarlos, manejarlos y probarlos. Esta herramienta permite una reducción de más del 50% del tiempo de desarrollo y pruebas de prototipos.

KEYWORDS: Tool development, paper handling and printer devices, hard-software codesign, testing

1. INTRODUCTION

During the last years, some companies have developed high efficient printers, e.g. Hewlett Packard, in order to satisfy large printer offices demands. By means of paper handling devices -PHDs-, new est printers are able to separate, distribute, and staple documents. In this way, most PHDs connected at printer’s paper interfaces -input and output-, are designed to move paper sheets with a high precision.

In general, the development process of electronic devices consists of several levels: conception, analysis and design, implementation, and testing. For the development of similar products, the same process is always followed. Due to delivery of electronic devices is crit ic in time, reusable control architecture is desired in order to decrease time intervals between prototypes and final products. This is possible for a device family, where products are based on similar functionality and operation procedures.

All PHDs operation is based on electromechanical components, e.g. motors and sensors. According to this, generic control architecture should be created to simplify their developing procedures. In order to provide such functionality, the platform must be able to satisfy the following requirements:

1. To generate any system paper sheet movement, providing every signal required moving and controlling DC and stepper motors.

2. To determine, at any time, paper sheets position, through electronic sensors operation.
3. To store and to execute programmable code (firmware), in a microprocessor based system.

4. To provide internal printer communication, using CAN 2.0B standard protocol.

5. To support PC communication, through RS232_C protocol.

6. To allow system-user interaction, based on a graphical software program running over a PC.

Main objectives of the present work are:

1. To design a reusable architecture for control and operation of printer paper handling devices.

2. To implement the tool for development and testing of the printer devices.

2. HEWLETT PACKARD PAPER HANDLING DEVICES –PHDs-

PHDs are attached to printers and they take the printed paper from the printer and, depending on the type of device, either puts it into a high capacity output tray, or into a collation mechanism to separate, distribute or staple documents. Most printers provide two output trays; one "face up" and one "face down" to send their printed jobs. Fig. 1 shows specific HP printers' output paper trays [1].

![Figure1. Hewlett Packard printer output paper trays](image)

In addition to normally output paper containers, HP printers have the capability to attach PHDs to receive the printed paper. They have several output additional trays, selectable by command, allowing the separation of different print jobs. Paper handling devices stack sheets and provide automatic job offset capabilities for easy handling and job separation. It also allows in-line stapling, enabling the creation of a wide range of documents ready for distribution in just one step [1].

2.1 Type of papers

For HP printers and PHDs the best paper has the following characteristics [2]:

- Cleanly and accurately cut.
- Paperweight of around 80-90 gsm (grams per square meter).
- Dry and free of dust.

Other type of papers contaminates the printer mechanism, and may skew or crease in the paper path.
Frequently is necessary to print documents over different paper types. HP printers and paper handling devices allow the use of plasticized paper, cards, transparencies, and other types of special printing materials, with different paper densities.

2.2 Paper Movement Analysis

Different paper movements, on paper handling devices, can be described by the set \( T = \{ T_1, T_2, \ldots, T_n \} \), where \( T_i \) is a trajectory determining sheet of paper path. For every \( T_i \), each sheet is moved from printer to the selected output paper container. In order to improve the transportation through devices of sheets of paper, movement profiles \( M_i \) are introduced and associated to every \( T_i \) paper trajectory. Besides, each movement profile \( M_i \) is a set \( M_i = \{ m_1, m_2, \ldots, m_k \} \), where \( m_k \) are independent elementary movements. These movements are related with DC motors, stepper motors, and sensors. These are components denoted by \( e_n \). Then, it is possible to consider a function, \( f_k \), able to generate any \( m_k \) required for a printer device; and consequently, \( m_k = f_k (e_1, e_2, \ldots, e_n) \). Paper movements require, inside of printer devices, of:

- High precision and minimum load movements.
- Realization device critical operations: such as supervision and regulation of paper movement.

Fig. 2 shows the principles of DC and stepper motors. Generally, motors are used through gears and bands system, which permits to obtain desired speed of the sheet of paper. The system allows reaching motor operation close to some optimal parameters that produce the rollers spinning and therefore paper movement. Rollers provide a proportional force originated by their own attraction force and a fixed frictional coefficient. Under optimal conditions, totally angular movement is transferred to the paper, producing a linear movement. By this way, paper speed and acceleration are directly controlled by the motor operation.

![DC and stepper motors scheme](image)

**Figure. 2 DC and stepper motors scheme**

### 2.2.1 DC motors

According to Fig. 2, the lineal speed, \( v_L \), is determined by \( v_L = P w f \) equation, where \( P = 2p R_r \), represents the attraction force between the rollers of radio \( R_r \) over every sheet of paper. Besides, the variable \( w f \) represents angular final speed, which is directly proportional to gears relationship \( E_r \), and its motor angular speed \( w_M \) \((w_f = E_r w_M)\) [3]. The \( w_M \) value is directly proportional to voltage applied to the motor, \( V_M \), and to manufacturer factor \( K_M \), i.e., \( w_M = V_M K_M \). So the final lineal speed could be described by the next equation:

\[
v_L = 2\pi R_r E_r V_M K_M \tag{1}
\]

The eq. 1 shows the relationship between the voltage applied to the motor, \( V_M \), and the system lineal speed, \( v_L \). Final paper lineal movement, \( M_L \), is a function of \( v_L \), and time. The variable time is introduced
using the step function $m(t-t_0)$ to describe the paper movement over time intervals. Finally, the paper movement is given by the relationship:

$$M_L = v_L \mu(t-t_0)$$

Consequently, the voltage applied to the motor is directly related to the final paper movement lineal speed. The polarity of the voltage regulates the motor movement direction.

### 2.2.2 Stepper motors

Stepper speed and movement analysis is presented in reference [4]. For this case, substituting in eq. (1) the voltage applied to the motor, $V_M$, by a frequency source $f$, and the manufacturer factor, $K_M$, by the equation $K_M = 2\pi/n$, where $n$ represents the steps number required to reach a full motor revolution. It is obtained that paper lineal speed is given by:

$$M_L = v_L \mu(t-t_0)$$

In addition, stepper motors have capabilities to maintain a constant force over rollers, allowing sheets of paper required fixed position, in order to avoid movement failures. Stepper motor movement direction is given by the introduction of a sequence of the electrical pulses.

### 2.3 Movement and speed profiles

Fig. 3 shows the acceleration, speed and movement profiles of paper sheet. The acceleration profiles during pre-movement stages of paper ($a_1$, $a_3$ and $a_5$) proportionate to DC and stepper motors, the required time to find their optimal operation parameters. These parameters determine the lineal speed levels required moving paper through devices and their values are very important during the product development process.

The acceleration interval duration $a_1$, $a_3$, $a_5$, can be neglected in comparison with the sheets of paper transport timings given by the periods $a_2$, $a_4$. This means that all this acceleration intervals prepare the electromechanical system to carry out the sheets of paper through the device.
3. ARCHITECTURE

PHDs, as most electronic products, are implemented by using programmable components. Microprocessors and peripheral components, such as controllers and special application modules [5], define their architecture. These application modules are well-defined subsystems to control motors and handle sensors. In this way, a generic architecture can be created and it can be used in other implementations. Therefore, engineers will use the tool in order to develop current devices in shorter time intervals than before.

The selected architecture for the tool is shown in Fig. 4. It consists of the following parts: hardware, control firmware, user application software, and communication interfaces.

![Figure 4. Paper handling and printer device development tool architecture.](image)

3.1 Hardware

It has every required physical device to move motors, read sensors and execute special tasks. A microprocessor, memories, controllers, motor drivers, communication circuits and glue logic form this block. It is divided in two PCB cards: the first one is called system hardware card, and the second one application hardware card.

System Hardware Card. It is the control system core for the development and testing tool. Besides, it is based on a programmable microprocessor, which is able to realize a group of general-purpose tasks, which are required to implement paper handling devices functionality (execute programmable code, external communication interfacing).

Application Hardware Card. Its principal function is to support every physical device necessary to move motors and read sensors. Through a connector, it is attached to the system hardware card.

3.2 Control Firmware

The firmware constitutes the software interface between hardware and application software. It is a programmable instruction set embedded into the microprocessor. This set of instructions handle and control all the input and output signals related to the microprocessor, that are required to execute the system desired operations [6].

Control firmware is divided into two basic components. The first one is a set $A = \{A_0, A_1, \ldots, A_n\}$ of dedicated algorithms $A_k$; and the second one is the access code $CA$. Algorithms $A_k$ are used to move DC and stepper motors, originating elementary movements $m_k$, which help to build every movement profile $M_i$ and, consequently the paper trajectory $T_i$. The access code $CA$ has two different functions: to
select an algorithm \( A_k \), and to communicate with the application software running into a PC over the interface RS_232_C (at speeds of up to 38400 bps). All the information used and generated by this module is interpreted and received by the third functional block: the user application software.

3.3 User Application Software

User application software runs over PC Window s's 95/98/NT operating systems. Besides, the program implements serial port communication with the control firmware, it provides a user graphical interface. It was developed using C++ Builder 3.0 tool and object oriented programming. All the generated information is coded and transferred to the control firmware; then hardware components execute the desired operations. Software must provide to the engineers with a set of required commands to configure and test devices, e.g. DC-, stepper motors, and sensor components, according to the application.

3.4 Communication Interface

All PHDs must realizes tasks such as signal detecting, component configuration, motors activation and deactivation, and activities related to device paper movement. Although, operation of most devices is determined by external variables coming up from printers or other printer controllers; a PHD need to have a communication protocol in order to know all their working conditions. According to this, HP engineers adopted CAN version 2.0B as a standard protocol to define their PHDs and printer interfaces.

3.4.1. CAN 2.0B Protocol

The Controller Area Network (CAN) is a serial communications protocol of high integrity, which efficiently supports distributed real-time control with a very high level of security. Its domain of application goes from high-speed networks to low cost multiplex wiring, -see Fig. 5. It operates at data rates of up to 1 Mb/s, providing an excellent error detection and reliability. CAN is documented in ISO 11898 (for high-speed applications) and ISO 11519-2 (for lower-speed applications) [7]. It has the following characteristics:

- CAN is a lineal multi-master bus with one logic bus line and equal nodes.
- Their bus nodes do not have a specific address.

Instead of it, the identifiers of the transmitted messages, indicating the message content and the priority of the message carry out the address information. The higher priority message is guaranteed to gain bus access as if it were the only message being transmitted. Lower priority messages are automatically retransmitted in the next bus cycle or in a subsequent bus cycle if there are still other higher priority messages waiting to be sent [8].

- The number of nodes is not limited by the protocol and can be changed dynamically without disturbing the communication between nodes.
- To support multicast and broadcast addressing.
- To provide sophisticated error-detection and error handling mechanism such as CRC check, and high immunity against electromagnetic interference. Erroneous messages are automatically retransmitted, and temporary errors are recovered.
- Non-return-to-Zero code. For synchronization purposes, bit stuffing is used.
- There is a high data transfer rate up to 1 Mb/s, at buses length of 40 meters using twisted wire pair. Messages length has a maximum of 8 data bytes.
The bus access is handled via the advanced serial communications protocol Carrier Sense Multiple Access/Collision Detecting with Non-Destructive Arbitration.

CAN is implemented in some commercial chips.

![Diagram of CAN bus: Messages transmission on the printer and paper handling devices](image)

4. HARDWARE

As we already mentioned, two PCB cards represent the hardware block: the system hardware card and the application hardware card.

System Hardware Card. Microprocessor and peripheral components selection defines paper handling hardware design. This card interfaces with other devices (see Fig. 6). Main components are described below:

- 16-bit H8S/2357 Hitachi high efficient microprocessor [9].
- 2 Mbit of Flash memory organized as 128 K x 16 bits, 512 Kbits of Static RAM organized as 32 K x 16 bits, and 512 bytes of NVRAM (serial EEPROM).
- CAN 2.0B communication protocol controller.
- Debug port through RS232 driver.

System hardware card implementation gives development tool capabilities to:

- Execution of general-purpose tasks.
- CAN 2.0B access, enabling printer and paper handling devices communication.
- PC serial port interface.
Figure 6. Development Tool: System hardware PCB card.

Application Hardware Card. It is attached to the system hardware card. It was designed to represent a kind of universal board to control steppers and DC motors. It also has the capability to read sensors and to generate an interrupt when any of them changes from active to inactive or vice versa (see Fig. 7). A list of all features included in this card is:

- 40-pin system hardware card connector.
- Necessary logic to read up to 16 sensors.
- Circuit EPM7064S PLD (programmable logic device) to implement the sensor interrupts [10].
- Drivers to handle and control up to 14 DC motors and 31 stepper motors.

Figure 7. Development Tool: Application hardware PCB card.

5. SOFTWARE

User application software provides an interactive interface to developer engineers to configure and test their devices of any electromechanical system. By the GUI interface, users are able to drive DC motors, stepper motors and sensors according to the requirements, in order to find the optimal operation parameters of many devices.
According to this functionality the software, based on object oriented programming, was developed using C++ Builder 3.0 tool [11] and it is integrated by the elements shown in Fig. 8.

Control. Principal software component implemented to control and supervise the program functionality:

a) To program environment setup, creating all the functional software objects required to reflect every development tool hardware component.

b) To implement the user graphical interface.

c) To create, update and control all system active elements (DC-, stepper motors, and sensors), according to firmware and user interaction.

d) To capture and code user data to be transmitted and executed by dedicated control firmware. At the same time, coming up firmware data decoding and interpretation. Serial port driver implementation to receive and send data through RS_232_C protocol.

Functional Objects. Components are classified according to following statements. Any hardware representative object forms first group such motor driver chips and sensors electronic components. The second group describes all the software motors and sensors logical representation, which are dynamically activated and deactivated during program execution.

5.1 Software Characteristics:
Software was developed using object-oriented techniques and the C++ Builder 3.0 tool. It runs over Window’s 95/98/NT operating system. Six own classes were implemented, 22 recipient classes used, 2050 code lines written, 157000 compiled lines, and the executable program has a volume of 475Kbytes.

Figure 8. User Application Software Architecture

Graphic Interface. An intuitive and ease of use interface, see Fig. 9, allows the user handle up to 30 DC motors, 14 stepper motors and 16 sensors [12]. Users can handle independently each motor, adjust the level current parameters, fix different speed levels to determine stepper motors operation, user-defined ramps up (acceleration) and ramps down (deceleration) in order to accomplish different paper movements, and read sensor logical level.
6. RESULTS AND ANALYSIS

A paper handling device consists of different parts: mechanical, electronic and firmware. During the normal development process, the mechanical part was implemented necessarily before the other ones. The debugging process was very difficult. Besides, the process to find good movement profiles was very time consuming, due to the load and unload of different algorithms using traditional methods. For these reasons, the development time was approximately 16 weeks.

In the present work the development process was studied to reduce its duration. Therefore, relationships between trajectories, movement profiles, elementary movements, components and functions were analyzed.

- Movements of piece of papers through printers and PHDs can be described by trajectories $T_i$. There is a movement profile $M_i$ associated with each trajectory $T_i$.

- An ordered set of independent elementary movements $m_k$ builds a movement profile $M_i$, and every $m_k$ required for a printer device is expressed as a function $m_k = f_k(e_1, e_2, ..., e_n)$, where $e_i$ denotes a component.

- Two basic components of the firmware were implemented. The first one is a set $A = \{A_0, A_1, ..., A_n\}$ of dedicated algorithms $A_k$ and, the second one is the access code $CA$. Algorithms $A_k$ are used to build every movement profile $M_i$ and, consequently the paper trajectory $T_i$.

- Some metrics such as: number of printed pages per minute and, number of jammed sheets are very important to evaluate a trajectory $T_i$ and its movement profile $M_i$.

The new implemented platform or tool contributes to the automation of the development process of PHDs and printer devices allowing to the engineer:

- To achieve a flexible composition of different movement profile $M_i$ for research purposes.

- To find good movement profiles for printer devices and PHDs, saving time.
To work on firmware in parallel with the mechanical part allowing a greater independence than before.

For these reasons testing time to put in order a PHD device was decreased from 16 weeks to only 4 weeks. Accumulated experience by engineers using this tool continues decreasing this time interval.

7. CONCLUSIONS

A development and testing tool for paper handling and/or printer devices was implemented. This platform simplifies and reduces efficiently the design and implementation of final Hewlett Packard products.

For the hardware were designed and built two PCB’s denominated system hardware board and application hardware board. An EPM7064S PLD (Altera) was included and programmed adding special system functionality to the hardware. The microprocessor firmware was written in C language. It contains the set A of dedicated algorithms and the access code CA. Algorithms Ak are used to move DC and stepper motors, originating elementary movements mk, which help to build every movement profile Mi and, consequently the paper trajectory Ti. The access code CA has two different functions: to select an algorithm Ak, and to communicate with the application software.

The application software was implemented using object-oriented techniques and it has a GUI interface. By this dedicated graphical interface, users are able to configure, operate and control any motor and sensor connected to the platform, in order to find optimal operation parameters required to move sheets of paper through PHDs.

Development tool capabilities are related to implement general purpose tasks, access to CAN 2.0B standard protocol, maintain successful communication with a PC, recognize sensors logical levels, and handle DC motors and stepper motors.

The use of this tool for the development and testing of paper handling and/or printer devices has demonstrated a high usefulness and quality to release in shorter period of time high quality products. The first tests reduced the development and testing time from 16 to 4 weeks.

Besides, it represents a new platform to research and development optimal movement profiles for printer devices and development tools for both R&D (Research and Development) laboratory and service in the field for the support engineers at Hewlett Packard.

8. ACKNOWLEDGEMENTS

The authors wish to express their gratitude to Hewlett Packard for the financial support given to this work, and in particular to Mr. H. Rodríguez.

9. REFERENCES

[1] Hewlett Packard, Hardware Drivers Documentation for Paper Handling Devices (Guadalajara), 1-58, (2000).
[2] Burrows S., 1994 A Laser Printer Book, Ed. Black Lightning Inc, (New York), 5-25.
[3] Reliance Motion Control Inc, DC Motors, Speed Controls and Servo Systems Handbook, Ed. Reliance Inc, (Minnesota), 2-114, (1996).
[4] Ericsson Components, Stepper Motor Control Handbook, Ed. Ericsson, (Sweden), 11-82, 1995.
[5] Merrill C., 1984 Microprocessor Hardware/Software Interfacing, Ed. B. Brey.
[6] Hewlett Packard, Paper Handling Firmware Development Standard, (Guadalajara), 1-32, (2000).
[7] Bosch R., 1991 CAN Specification version 2.0, (Stuttgart), 2-68, ()
[8] Philips Semiconductors, Interfacing a Stand Alone CAN controller, (Amsterdam), 1-66, (2000).
[9] Hitachi Semiconductors, H8S/2357 Series, Hardware Manual, (Tokyo), 1-1084, (1998).
[10] Altera, Max Plus II, Programmable Logic Development System, (San Jose), 12-300, (1995).
[11] Liberty J., 1998 Beginning Object-Oriented Analysis and Design: With C++, Ed. Wrox Press Inc, 35-68.
[12] Schildt H., 1997 Borland C++, Ed. McGraw Hill, 55-255.
Authors Biography

Vladimir Castillo

Received the B.E. degree in Electronics from Instituto Tecnologico de Sonora (ITSON) in 1998 and the M.S. in Telecommunication from Research Center and Advanced Studies of IPN, Mexico, in 2000. Since 2000 he is a member of R&D Department of Hewlett Packard, Guadalajara, Mexico, where he works with hardware and firmware for Printers.

Deni Torres Román

Received a Ph.D. degree in telecommunication from Technical University Dresden, Germany in 1986. He was professor at the University of Oriente, Cuba. Co-author of a book about Data Transmission. He received the Telecommunication Research Prize in 1993 from AHCIET Association and was recipient of the 1995 best Paper Award from AHCIET Review, Spain. Since 1996 he is associate professor at Research and Advanced Studies Center of IPN (Cinvestav-Mexico). He is a member of the IEEE.

Hernán Gutiérrez

Received the B.E. degree in Electronics in 1990 and the M.S. in Industrial Electronic from Instituto Tecnologico de Estudios Superiores de Occidente (ITESO), Mexico, in 1996. From 1990 to 1996 he worked at different companies as IIE and GPH Mexican. Since 1996 he is a member of R&D Department of Hewlett Packard, Guadalajara, Mexico, where he works with hardware and firmware for Printers.