A BANKNOTE NUMBERING MACHINE EMBEDDED COMPUTER CONTROL SYSTEM

Sami M. Sharif* and Sayed Harron AbdelKarim**

ABSTRACT

This paper is discussing the design and implementation of embedded computer based control system for a banknote-numbering machine, "Numerota". The design is based on the embedded system ELAN-104NC with windows CE preinstalled in it. ELAN is a single board PC with inputs outputs cards; it is a rack mounted system. The application software is designed by using the data and signal flow diagrams and coded with Microsoft Visual C++ 3.0.

1. INTRODUCTION

Sudan Currency Printing Press (SCPP) has many printing machines; some of them are old, but mechanically they are functioning well. Their relay based control systems, beside its disadvantages, are very old; many of the electrical components are no longer produced or have been superseded by newer designs. Therefore, these machines are constantly stopping due to electrical faults, which render low production**.**

SCPP decided to upgrade the control systems of those machines by contracting companies who work in this field. One company proposed an upgrade cost for one machine at a minimum cost of €130,000 EURO. The grade included the installation of Programmable Logic Controller (PLC). It has been noticed that 75% of this cost was just labor. Thus, it is decided to do the work locally (design and implementation) and the total cost for this upgrade was estimated to be about €21,647.

The design and implementation of the controlled system is based on embedded system. The embedded system may be defined as a "hardware and software which forms a component of some larger system and which is expected to function without human intervention"[10]. A typical embedded system may consists of a single-board microcomputer and software in ROM, which starts running some special purpose application program as soon as it is turned on and will not stop until it is turned off (if ever). Generally, they are used in automobiles, planes, trains, space vehicles, machine tools, cameras, consumer and office appliances, and other handheld as well as robots and toys.

In embedded systems, traditionally, the software was permanently set into a read-only memory such as a ROM or flash memory chip, in contrast to a general-purpose computer that loads its programs into RAM each time. However, nowadays, modern embedded systems, which are usually implemented with a single board, multi-purpose PC, operate under an embedded operating system (e.g. Windows CE, Embedded Windows XP, Embedded Linux, etc.), and thus functioning the same as a general-purpose computer, but used for a specific application such as the control of an industrial plant. Sometimes, single board and rack mounted general-purpose computers are called "embedded computers" if they are integrated in systems to operate in this way.

An embedded system is a real time system; it has the ability to make certain decisions in a timely manner. These decisions have a deadline for completion.

The issue of what happens if a deadline is missed is a crucial one. For example, if the real time system is part of an airplane's flight control system, it is possible for the life of the

* Associate Professor, Department of Electrical and Electronic Engineering, University of Khartoum
** Chief Engineer, Sudan Currency Printing Press (SCPP)

passengers and crew to be endangered by a single missed dead line. However if the system is evolved in a satellite communication, the damage could be limited to a single corrupt data packet.

Embedded software unlike software of general-purpose computers cannot be run on other embedded systems; it needs significant modifications. This is due to the variety in the underlying hardware. The hardware in each embedded system is tailored specifically to the application, in order to keep system costs low.

2. OPERATION OF THE BANKNOTE NUMBERING MACHINE "NUMEROTA"

The Numerota is a sheet fed letter press numbering machine, specially made for the requirements of manufacturing securities and used for printing numbers, signatures and text parts. This press is made up of four printing units, as shown in Figure 1 and 2. namely these units are (1) the feeding unit, (2) The registration unit, (3) The printing unit and (4) The delivery unit.

The type of the feeder for this press is a sheet–fed type. The trip of the sheet in the press starts from the "in feed pile" in the feeder, then it goes through registration unit to the printing unit, and finally the sheets are collected in two piles in the delivery unit Figure 3 shows a flowchart the sequence of operation of the machine . The feeder is an air vacuum device; it has two motors, limit switches and mechanical latches.

Registration is the process of controlling a sheet before it enters the printing unit, to get image position consistency in every printed sheet. In our machine, the goal of registration is to print the two serial numbers in the exact position in every printed banknote. As the sheet enters the registration unit, it is halted by the head stop, then the rotary motion of the pull guide for proper positioning pulls it, then the head stop moves out of the way so that the sheet can enter the printing unit. In this press, a two point-pull rotary system is used for registration

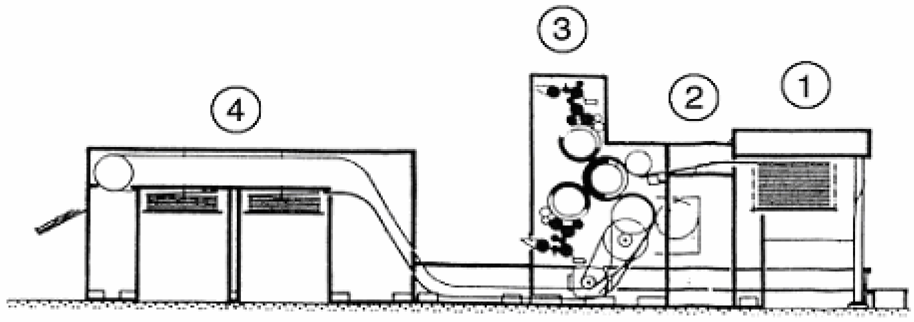

Figure 1: Numbering Machine Components

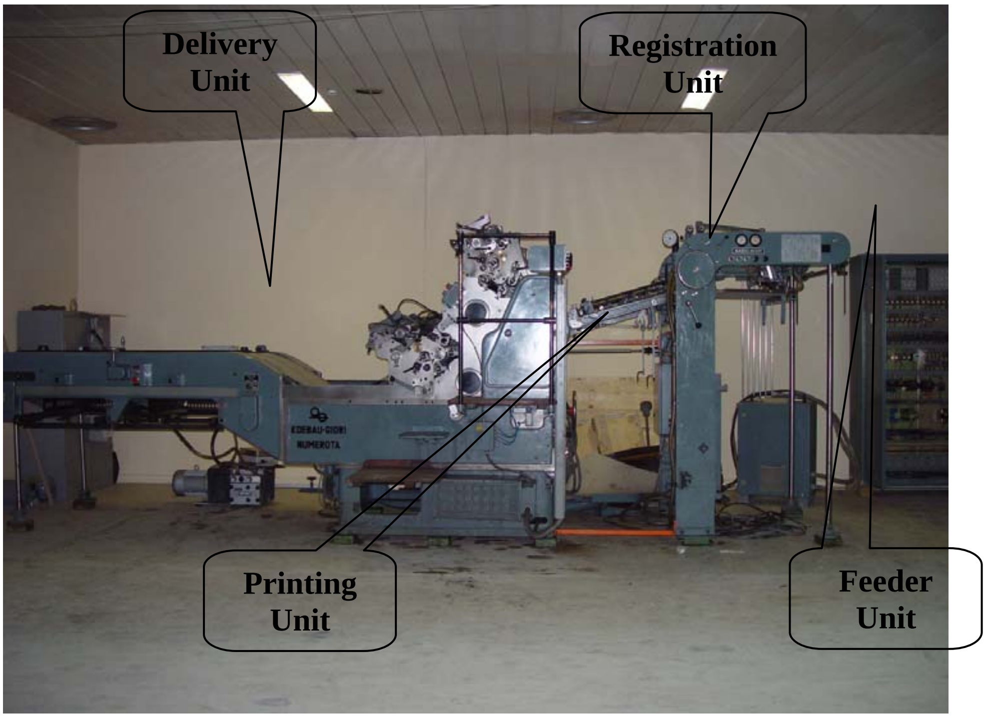

Figure 2: Numerota Machine manufactured in 1963

The goal of the printing unit is to place ink on the numbering unit, and then to transfer image to the paper and finally to deliver the paper to the delivery unit. The printing unit for this press is made up of two parts: the Cylinder system and the Inking system.

The cylinder system is made up of three cylinders, Upper Numbering Cylinder, the Impression Cylinder and the Lower Numbering Cylinder. The numbering units are fixed in the numbering cylinders . The function of the numbering cylinder is to hold the numbering units and revolve them in contact with the impression cylinder during the printing process.

The ink from rollers contacts the numbering units in the numbering cylinders.

Consequently, the numbers in the numbering units are inked. Then a sheet is fed between the numbering cylinder and the impression cylinder, and then the images of the numbers are transferred to the paper. The grippers in their way to the delivery unit receive the paper.

The function of the inking unit is to transfer a uniform thin layer of ink to the numbering units. The ink stored in a reservoir and then fed in small quantities through different types of rollers. During this process, the ink worked into the type of the delivery system used in this press is the chain gripper system; it is made up of two closed loops of chains. There are bars fixed across the two chains, the grippers are fixed on the bar. When the sheet leaves the printing unit, its front edge is grabbed by the grippers, pulled out of the printing unit and dropped on one of the two delivery tables of the press, then the grippers move back to receive another paper. At the end, a neat stack of paper is formed on the two delivery tables. The chain of the gripper system moves at the same rate of the printing unit, registration unit and the feeder unit. That means the three units move in synchronization.

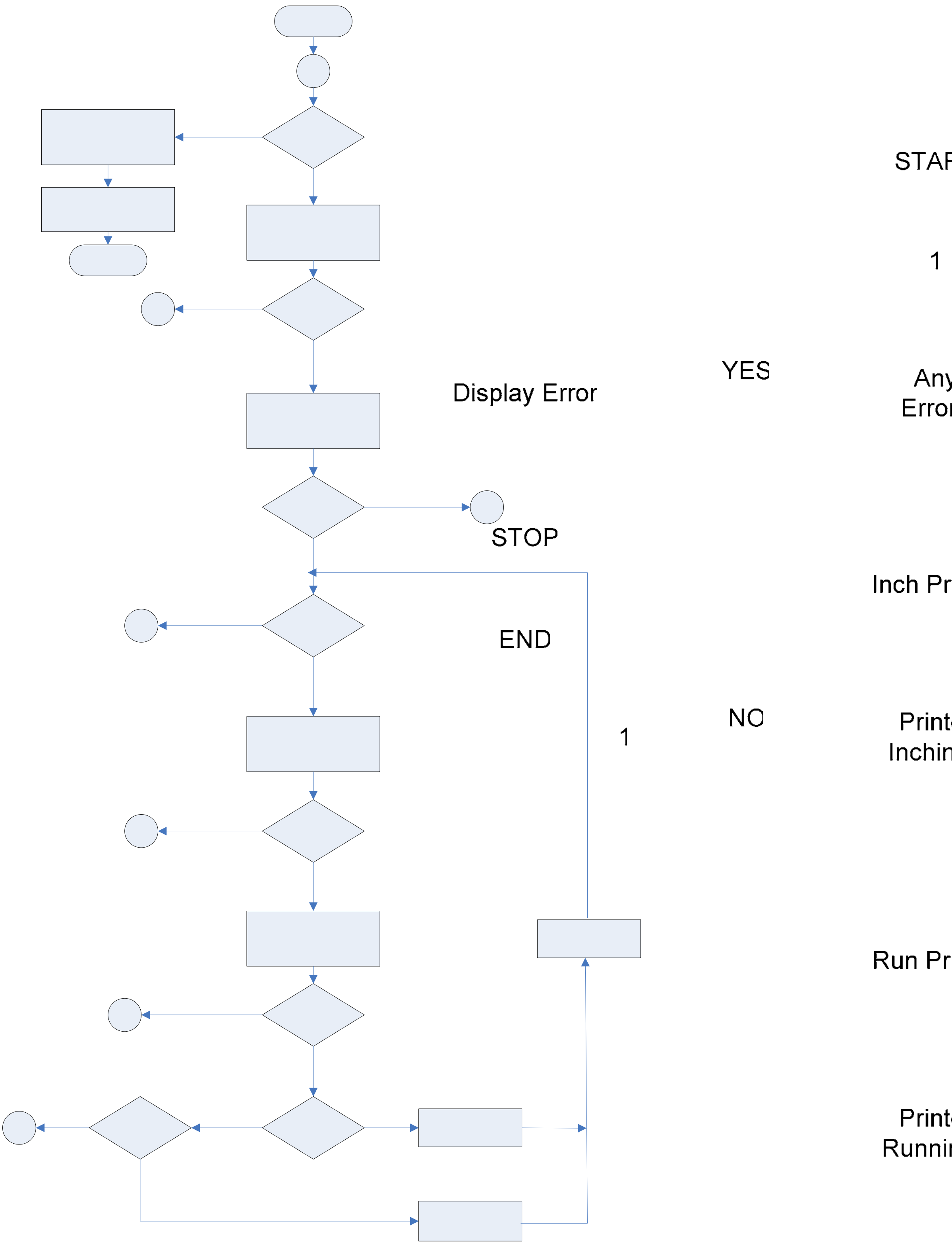

Figure 3: Flowchart for the operation of Numbering Machine

3. HARDWARE DESCRIPTION

For upgrading the control of the banknote numbering machine, an embedded system called Standard Europe PC (STEpc) using ELAN-104NC single board PC was used. This system consists of the following: Microprocessor, Memory, Inputs and outputs cards, Peripheral components and Software. It is a rack mounted system. Figure 4 shows the STEpc from inside.

Figure 4: STEpc with single board computer and I/O cards

The AMD Elan (SC400) processor is an Am486 class processor without a floating-point unit. It is a fully integrated PC/AT (Personal Computer Advanced Technology (IBM)) compatible architecture. The SC400 is a 32-bit x86 compatible device. A 100MHz processing unit is used on the ELAN-104NC.

The SC400 processor is packaged in a 256 pin Ball Grid Array. The SC400 processor is a low power device and no heat sink is required for this device on the ELAN-104NC. Without a heat sink, the processor can operate at ambient temperatures up to 70°C when run at 66MHz or 50°C when running at 100MHz.

The ELAN-104NC board supports up to 8MB flash memory. This memory is configured as a read/write silicon disk drive. The flash drive uses a 32KB memory window to access the device and two I/O address locations are used to select the appropriate flash area. The FLASH status LED will illuminate whenever the Flash drive is accessed.

The ELAN-104NC uses the Real Time Clock internal to the Elan SC400. The date and time functions are stored in the real time clock when the main power is removed and the battery backup supply is enabled.

The battery backup circuit maintains the Real Time Clock and CMOS settings when the main power input is disconnected. A lithium cell provides the battery backup supply and has a capacity of 170 MAH. This battery will provide sufficient support for at least 3 years continuous backup. If the main supply is present on the board the battery is automatically disconnected from the Real Time Clock circuitry.

The ELAN-104NC contains two independent watchdog timers, which can be used to protect against application software conditions that may cause the ELAN-104NC to 'hang'. The watchdog timers, once started, will trigger a CPU reset if they are not re-triggered within a set timeout period.

The watchdog timers can be disabled permanently by removing jumpers on the board one corresponding to each timer.

The PC/104 interface supports 8/16 bit ISA (Industry Standard Architecture) style PC/104 signals. Add-on boards can be used to enhance the functionality of the main board. A large number of companies have adopted the PC/104 standard and boards are available which support a wide range of interfaces. This bus can be used to add digital I/O, analogue I/O, serial ports, video capture devices, PC CARD interfaces, motion control devices etc.

Any board plugged into this interface will be accessed as if it were part of the main board. Therefore, it may conflict with I/O and memory devices onboard - if it has not been configured. Before using an expansion board, it should be checked that it could be configured to work alongside the peripherals already incorporated onboard. The PC/104 bus signals are fully compatible with the ISA bus electrical timing definitions.

4. SYSTEM HARDWARE DESIGN

The control cabinet included the following hardware components: An embedded computer, Interface Connectors, Interface Relays, Power Converter, Main Isolating Switch, Transformers, Rectifiers, Overload Switches, Main Contactors, Measuring Devices, User Control Panel, Fusses and Cooling Fans.

The embedded computer is a rack mounted STEbus chassis fitted with an Elan104/NC single board PC attached to serial interface card which provides a PC/104 to STEbus interface bus connector. The following I/O cards also attached to the system.

- One digital input card providing 32 digital input channels.
- One relay output card providing 16 changeover relays
- An analog I/O card providing an analog output channel for controlling the run operation of the power converter to the main drive and an input channel providing analog input channel for sensing the rate of the main drive via its tachometer.
- In addition, another analog I/O card providing an analog output channel for controlling the inch operation of the power converter to the main drive.

Four connectors have been installed for each of the input and output cards fitted in the embedded controller. These connectors link the embedded computer's input and output channels with the printing press's electrical system.

A series of various rated interface relays have been installed in order to protect the relay channels that control high current components on the printing press. These components include:

1. The main drive brake contactor
2. The power converter enable switch
3. The impression solenoids
4. The delivery solenoid
5. The feeder enable switch

The original motor is an AC motor and controlled by resistance or potentiometer. This kind of motor with its traditional control has many disadvantages. It cannot be controlled smoothly and there is much energy loss due to use of resistance in motor's speed controller, and more importantly, it cannot be controlled by an embedded system. Thus, the main drive was replaced by a Baumüller DC motor with the corresponding 520VDC power converter, which has an electronic based control system. The converter is very efficient, reliable, heavy-duty, economical in the consumption of the energy and easily controllable by a computer based controlled system. Three line reactors have been installed before the main input voltage enters the controller to protect from surge voltages. Additionally, two reactors have been fitted before the field input voltage enters the controller for the same purpose.

The main switch provides isolation and protection of the cabinet. Two transformers have been fitted. One transformer rated at 650 VA, 220/60VAC is connected to a 20 amps rectifier producing 60VDC used for supplying the solenoids. Another transformer rated at 450 VA, 220/24 VAC used for supplying the control of the entire cabinet.

Six overload switches have been used to protect the six motor in the system, as follows: Main drive motor, Main drive cooling fan motor, Feeder compressor motor, Main feeder drive, Delivery compressor motor, and Hydraulic pump motor. Six contactors have been used to control the above motors.

Three measuring devices have been installed for monitoring 1) the mains AC voltage, 2) the DC voltage of the main drive, and 3) the current of the main drive... An emergency stop button cuts power from the entire cabinet when it is pushedt. A selector switch used for checking the voltages in different phases in the mains AC voltmeter.

Three 50 amp ultra-rapid fusses are installed for the protection of the main drive and its electronic controller. Two 63 amp ultra-rapid fusses are used to protect the motor and the circuit of the field inside the electronic controller. Two 40-watt cooling fans are fixed high on the left and right sides of the cabinet for ventilating generated heat from the electronic components.

Table 1: Input Wiring Assignments

Input Channels			
Channel Name	Card/Address	Channel on Card	Pin on Connector
Stop Circuit	IN32/0x120	0	1
Main Controller Over Temperature	IN32/0x120	1	20
Main Controller Over Current	IN32/0x120	2	2
Main Controller Field Loss	IN32/0x120	3	21
Main Controller Tacho Loss	IN32/0x120	4	3
Main Drive Over Load	IN32/0x120	5	22
Main Drive Brake Sensor	IN32/0x120	6	4
Air Pump 1 Over Load	IN32/0x120	7	23
Air Pump 2 Over Load	IN32/0x120	8	5
Front Table Out Sensor	IN32/0x120	9	24
Back Table Out Sensor	IN32/0x120	10	6
Main Controller Ready	IN32/0x120	11	25
Inch Button	IN32/0x120	12	7
Accelerator Button	IN32/0x120	13	26
Decelerator Button	IN32/0x120	14	8
Lower Numbering CAM Control Switch	IN32/0x120	15	27
Upper Numbering CAM Control Switch	IN32/0x120	16	11
Optical Delivery Paper Sensor	IN32/0x120	17	29
Feeder Ready Sensor	IN32/0x120	18	12
Main Power Available Sensor	IN32/0x120	19	30
Main Controller Tacho	ANALOG/0x160	0 IN	1

Table 2: Output Wiring Assignments

Output Channels			
Channel Name	Card/Address	Channel on Card	Pin on Connector
Delivery Table Solenoid	RELAY16/0x150	0 NO	20
Main Power LED	RELAY16/0x150	1 NO	3
Lower Left Numbering Solenoid	RELAY16/0x150	2 NO	23
Printer Ready LED	RELAY16/0x150	3 NO	25
Lower Right Numbering Solenoid	RELAY16/0x150	4 NO	26
Buzzer/Alarm	RELAY16/0x150	5 NO	9
Main Controller Enable	RELAY16/0x150	6 NO	10
Upper Right Numbering Solenoid	RELAY16/0x150	7 NO	11
Printer Stop LED	RELAY16/0x150	8 NO	12
Main Drive Brake Contactor	RELAY16/0x150	9 NO	13
Upper Left Numbering Solenoid	RELAY16/0x150	11 NO	15
Main Controller Drive Voltage	ANALOG/0x160	0 OUT	12

The input and output wiring schemes are given in table 1 and 2 respectively. These components were installed and wired according to a wiring diagram.

Figure 5 is a photo for the complete wiring of the embedded control system and the other components, hosted in a cabinet for the simulation process.

Figure 5: Simulation process for the embedded system

5. SOFTWARE DESIGN ENVIRONMENT

The embedded controller selected for this electronic based system is operated under Microsoft Windows CE 3.0. The software is coded with Microsoft visual C++ 3.0.

The software development was done in four stages of development before being integrated to control the Numerato printing press. The software was developed using Object Oriented approach. The stages were as follows:

Stage One: Understanding the original operation of the Numerota printing press.

Stage Two: Connecting to the embedded controller and communicating with the embedded input and output interface cards.

Stage Three: Initial software implementation and the simulation environment.

Stage Four: The final implementation design – a Windows CE 3.0 application.

The Numerota printing press application software consists of two windows, one visible and the other invisible, a motor object that contains two threads for controlling acceleration and deceleration, and a third thread to monitor the input signals coming in via the ST-INPUT32 Arcom STE/pc interface card and generating corresponding event messages. This type of design was selected in order to catch or to trap error events as soon as they occur irrespective of other events that are simultaneously occurring. By scanning the input interface card and controlling the acceleration and deceleration in separate threads, the application is able to detect if another event, perhaps an error event, occurs while another task is in progress and respond immediately, appropriately, responsively. In case of error events, the printing press can be stopped immediately while accelerating or decelerating, or while any of the control buttons are being pressed or released.

6 DATA AND SIGNAL PATH DIAGRAM

Figure 6 shows the complete data and signal flow diagram. The flow of the data or signals originates at the buttons, limit switches, sensors, or contactors, or from Graphical User Interface on the monitor. Every input signal is processed and an event message is sent to the Arcom Control window described later. This particular window reacts to each event message that is sent to it by effecting:

1. Internal variables, like the paper counter, or the "impression on" button status for example.
2. Relay channels connected to the ST-RELAY16 interface card, used for controlling the main drive brake and main controller, or indicator lamps, like the stop and ready lamps.
3. Analog Output channels used for inching and running the printing press.

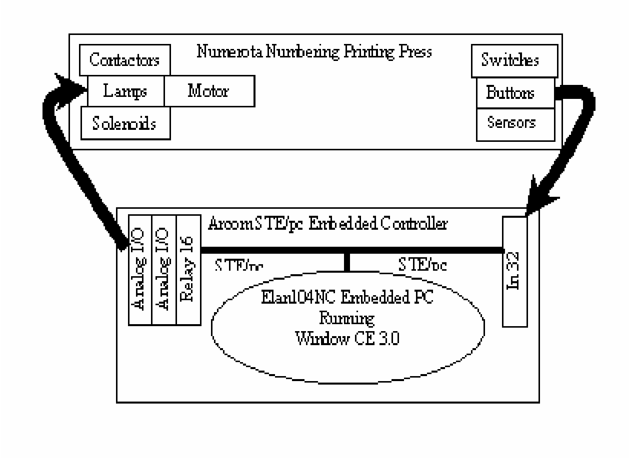

Figure 6: Complete Data Path Diagram

Figure 6 shows the software to hardware access path diagram. Generally, The Arcom STE/pc Embedded Controller contains four interface cards.

1. ST-IN32: receives all printer input signals including buttons, sensors, and limit switches.
2. ST-RELAY16: controls all output printers signals including indicator lamps, solenoids, contactors, and switches.
3. ST-ANALOGIO one to control the running operation
4. ST-ANALOGIO one to control the inching operation

In reference to Figure 7, these physical hardware interfaces are accessed via C++ object interfaces, which internally utilize the Elan104/NC board support package, which is represented by the rectangle labeled "Hardware Abstraction Layer (Halio.cpp)" in Figure 7. It is specifically designed for the Elan104/NC microprocessor, in order to read from and write to the physical interface cards on the STEbus.

The Numerota printing press application consist of four components as mentioned before:

1. The GUI Control window
2. The Arcom Control window
3. The Motor (MainDrive) object (contains two threads for controlling acceleration and deceleration.
4. The Input Manager thread

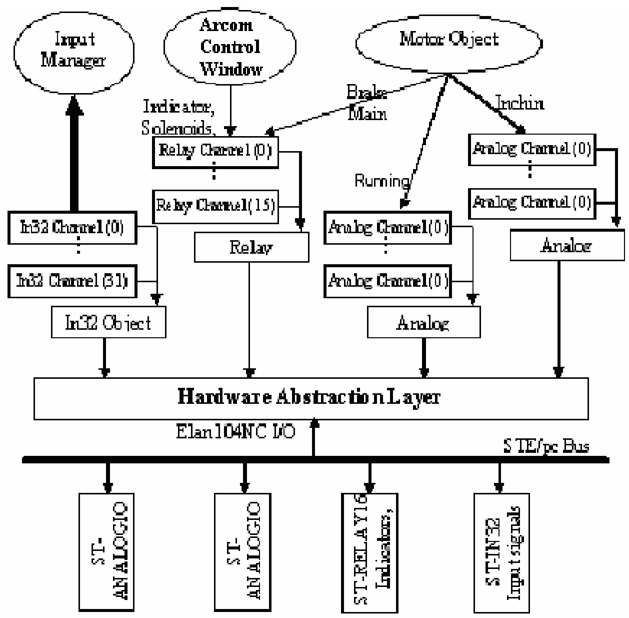

Figure 7: Software to Hardware Access Path Diagram

Figure 8 shows the Event Messages' Paths via Windows CE 3.0 The beginning of a Windows application is in a function called, WinMain(), which is found in the papp.cpp source file. It first begins by instantiating windows and objects, initializing them, and initiating the necessary threads and synchronization utilities, namely two signaled events and a mutex.

Once the initialization has completed, the Win Main() enters the controlled message loop, waiting for message sent by the Windows Operating, and from within the application itself. The message loop receives messages and dispatches them to either the GUI Control window or the Arcom Control window or any of the GUI components within the GUI Control window depending on the destination of the message.

The following are two lists naming and describing the source code and definition files used in the Numerota numbering printing press application.

Source Code Files

Analog.cpp contains software code for controlling the ST-Analogue I/O Arcom STE/pc bus interface card.

.Arcom.cpp contains software code for controlling the operations of the Numerota numbering printing press

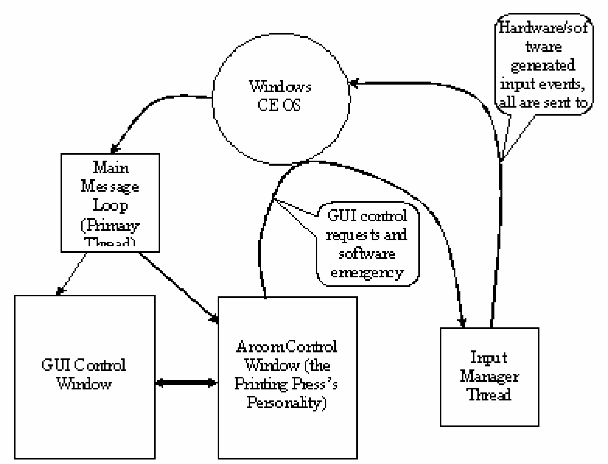

Figure 8: Event Messages' Paths via Windows CE 3.0

Halio.cpp contains software code for interfacing with microprocessor's I/O bus interface in order to communicate with the Arcom interface cards on the STE/pc bus.

In32.cpp contains software code for controlling the ST-INPUT32 Arcom STE/pc bus interface card.

Inmgr.cpp contains software code for managing the various input signals coming from the Numerota numbering printing press and firing appropriate events to the control software contained in the Arcom.cpp software code file.

Motor.cpp contains software code for managing the operational features of the main motor (i.e. stopping, applying and releasing the brake, accelerating, decelerating, and inching the main drive.

Papp.cpp contains software code initializing the main window, and child windows for the Windows CE 3.0 printer application (papp.cpp). In addition, it contains the control function for controlling the entire Graphical User Interface feature displayed on the computer terminal.

Papp.rc This file is generating by Microsoft Embedded Visual C++ 3.0. It describes the Graphical User Interface components (e.g. buttons, edit controls, progress bars, etc.). It is processed and compiled by Microsoft Embedded Visual C++.

Relay16.cpp Contains software for controlling the ST-RELAY16 Arcom STE/pc bus interface card.

Stdafx.cpp This file is generated by Microsoft Embedded Visual C++ 3.0 file.

Timers.cpp Contains the software code for handling software timers in Windows CE 3.0.

Definition (Header) Files

Analog.h contains the object definition of the ST-Analogue I/O Arcom STE/pc interface card, and the object definition of an in/out analogue channel.

Arcom.h contains the interface functions for the software contained in the arcom.cpp source code file.

Common.h contains the definition of the global configurations data structure, and the application specific event identifiers for the Numerota Software application.

Halio.h contains the definitions of the I/O bus interface functions used by the Arcom STE/pc interface card control objects.

In32.h contains the object definition of an ST-INPUT32 Arcom STE/pc interface card and the object definition of an input channel.

Inmgr.h contains the interface functions for the input manager software code contained in the inmgr.cpp source code file.

Motor.h contains the object definition of the main motor.

Relay16.h contains the object definition of an ST-RELAY16 Arcom STE/pc interface card and the object definition of a relay channel.

Resource.h Microsoft Embedded Visual C++ 3.0 file. It should not be edited.

Stdafx.h Microsoft Embedded Visual C++ 3.0 file. It should not be edited.

Timer.h contains the object definition of a Windows CE 3.0 software timer.

7. SYSTEM OPERATION

The application is initialized by the InitInstance() function, Fig 7. It registers and creates the main window for the application. In the creation of this main window, Windows CE sends a WM_CREATE message to its controlling function, WndProc() found in the file papp.cpp. The response to this message initializes the entire Numerota numbering printing press application. The following list describes the initialization in order.

1. Create a status bar window at the bottom of the main window, CreateStatusWindow().
2. Create the dialog control window used for the Graphical User Interface (GUI), CreateDialog().
3. Initialize the motor object, named MainDrive, InitMotor().
4. Initialize the invisible Arcom Control window, which is the personality of the printer itself, initArcom().
5. Initialize the Input Manager thread, InitInputManager().

8. CONCLUSION

Generally, the implemented system has the following advantages:

- The complication of wiring and sequential relays for implementing control logic is eliminated, it is provided by modern operating system (i.e. Windows CE 3.0). This makes the design of the control system have elegance, simplicity and functionality.
- Embedded systems are growing in popularity; their phenomenal growth is closely linked to the increasing availability of more powerful and less expensive processors, as well as to the decreasing price points of low-cost, high-density memory and the development in their operating systems.
- Many programming languages can program it; some of them are known and easy, like Java and C++.
- They are accessible and controllable through LANs, WANs, and the Internet, which makes it easier for the suppliers to monitor performance, update configuration information, and give technical support to customers.
- The system provides a very powerful tool for the diagnosis of faults that occur during operation of the machine, for this project a diagnostic feature was added by showing errors on the screen

Figure 8 showing the new control cabinet with the embedded system while the machine is running.

Figure 8: The new control cabinet

REFERENCES

1. Kurtis Johnson. *Process Control Instrumentation.* Gazera University. Library, 1999.
2. Groupe Schneider, Modicon *Modsoft*, SCPP Electrical Section, 1997.
3. E.A Parr Bsc, MIEE, *USE 11500 Version 2.0 Upgrading and Repairing PCs*, 12th Edition, SCPP Electrical Section, 1996.\
4. Brian W. Kernighan, Dennis M. Ritchie, *The C Programmable Language*, Proposed ANSI-Draft Edition.
5. E.A Parr Bsc, MIEE, *Programmable Controllers and Engineers' Guide*, SCPP Electrical Section, 1996.
6. Siemens, *Programmable* Data Sheet - Simatic-S5-100- U PLC, SCPP Electrical Section, 1995.
7. E.A Parr Bsc, MIEE, *Programmable User Manual 890*.
8. William Stallings, *Computer Orgnization And Architecture*, Fourth Edition, SCPP Electrical Section, 1996.
9. Michael Barr, *Programming Embedded System In C And C++*, SCPP Electrical Section, 1996.
10. hyperdictionary, online http://www.hyperdictionary.com/