Helsinki Metropolia University of Applied Sciences
Degree Programme in Media Engineering

Antti Heino

PC Based Automation Prototype

Bachelor's Thesis April 29, 2010
Instructor: Marko Huhtanen, Managing Director
Supervisor: Aarne Klemetti, Senior Lecturer

Helsinki Metropolia University of Applied Sciences **Abstract**

Author Title Number of Pages Date	Antti Heino PC based automation prototype 35 15 September 2009 (write the name of the month in full)
Degree Programme	Media Technology
Degree	Bachelor of Engineering
Instructor Supervisor	Marko Huhtanen, Managing Director Aarne Klemetti, Senior Lecturer

The aim of this thesis was to find out by building a prototype, whether it is feasible to automate old factory machinery with a PC-based system. Once ready, the system needed to be able to independently be able to carry out the whole work cycle of each machine it is used to control for. This prototype must handle automation of many different pieces of equipment, without major changes to hardware or software. The most important demand for graphical user interface was it needed to be clear, simple and yet be able to control use of different kinds of equipment. Because of the operator's previous work experience, user interface had to be very similar to common Microsoft Windows programs.

Due to the very demanding factory environment for PC equipment, system was built into a steel enclosure. To ensure sufficient air flow a fan was installed to the enclosure and a filter was placed on the air intake hole. An uninterruptible power source was placed in the enclosure to provide emergency power and to protect fragile parts from voltage spikes and surges. To control the entire system, an industrial PC was installed, which controls input and output card connected to it. The purpose of these cards was to operate the automated machines through relays.

Visual Basic was chosen to write the program needed to run the system. The main advantages of Visual Basic were ease of programming and short development time, which was considered to be especially important in order to successfully complete the project. The purpose of the program was to run small subprograms created by user. The programs consist of commands to control the automated machines via control cards installed to the system.

It was shown that PC-based automation is attainable and fairly easy to implement. During the short testing period it was not possible to find out how reliable or durable the system is. If this prototype was put to production use, it should be tested more elaborately. To provide database functionalities and a possibility to remote use, network connectivity should be implemented. To ensure a long program life cycle, the entire software part should be rewritten using .Net-framework.

Keywords	automation, factory automation, prototype, pc, Visual Basic, programmable logic controller

Metropolia Ammattikorkeakoulu Insinöörityön tiivistelmä

Tekijä Otsikko Sivumäärä Aika	Antti Heino PC-pohjaisen automaation prototyyppi 35 sivua 29.4.2010
Koulutusohjelma	mediatekniikka
Tutkinto	insinööri (AMK)
Ohjaaja Ohjaava opettaja	toimitusjohtaja Marko Huhtanen lehtori Aarne Klemetti

Insinöörityön tarkoituksena oli selvittää prototyypin avulla, onko vanhojen tehdaslaitteiden PC-pohjainen automatisoiminen mahdollista ja kannattavaa. Prototyypin tuli kyetä suorittamaan automatisoitavien laitteiden ohjelmat samalla tapaa, kuin niitä käytettiin ennen prototyypin käyttöönottoa. Samalla järjestelmällä piti pystyä käyttämään useita eri laitteita ilman suuria muutoksia ohjelmaan tai laitteistoon. Käyttöliittymän tuli olla yksinkertainen ja selkeä, mutta silti pystyä ohjaamaan eri laitteiden käyttöä. Tulevien käyttäjien taustan vuoksi sen tuli muistuttaa yleisimpiä Microsoft Windows -käyttöjärjestelmän ohjelmia.

Prototyyppi rakennettiin PC-laitteille mahdollisesti vahingollisen tehdaskäyttöympäristön vuoksi metallisen suojakotelon sisään. Riittävän ilmankierron takaamiseksi koteloon asennettiin tuuletin, jonka sisäänottama ilma suodatettiin suurimmista epäpuhtauksista. Häiriöttömän virran saamisen turvaamiseksi koteloon asennettiin myös katkoton teholähde. Prototyypin keskipisteenä toimii teollisuus-PC, joka ohjaa siihen liitettyjä ohjainkortteja. Niiden tarkoitus on käyttää automatisoitavaa laitetta releiden välityksellä.

Prototyypissä käytettävän ohjelman tekemiseen käytettiin Visual Basic -ohjelmointikieltä. Se valittiin helppokäyttöisyyden ja lyhyen ohjelman kehityskaaren vuoksi, mikä oli erityisen tärkeää projektin onnistumisen kannalta. Ohjelman tarkoituksena on suorittaa käyttäjän tallentamia ohjelmalistoja, jotka automaatiokortteja hyväksikäyttäen ohjaavat ulkopuolisten laitteiden toimintaa.

Työ osoitti, että vanhojen tehdaslaitteiden PC-pohjainen automaatio on mahdollista ja melko helppoa toteuttaa. Lyhytaikaisella testauksella ei pystytty tutkimaan prototyypin luotettavuutta eikä kestävyyttä. Mikäli prototyyppi haluttaisiin ottaa tuotantokäyttöön, sitä tulisi testata huolellisemmin. Lisäksi tiedon tallennusta ja etäkäyttöä varten verkkoyhteys tulisi ottaa käyttöön. Ohjelman elinkaaren pidentämiseksi se olisi syytä kirjoittaa uudestaan käyttäen hyväksi .Net Frameworkia.

Hakusanat	automaatio, tehdasautomaatio, prototyyppi, PC, Visual Basic, ohjelmoitava logiikka

Abbreviations

PLC	Programmable logic controller
PoC	Proof of concept
VB	Visual Basic
UPS	Uninterruptible power supply
GUI	Graphical user interface
LCD	Liquid crystal display
BIOS	Basic input-output system
AD	Analog to digital
DA	Digital to analog
ISA	Industry standard architecture
LED	Light emitting diode
IRQ	Interrupt request
I/O	Input / output
PCI	Peripheral component interconnect
COM	Component object model

Table of Contents

1 Introduction

Many companies own a lot of old machinery and equipment that are controlled only by the operator. The most common control methods are completely manual methods, where the controller uses switches and buttons and more advanced punch card controller. Punch card reader reads holes in a “control card” and switches different machine functions on and off, depending on the position of a hole in a card. Most of these machines work fine after decades of daily use, but cause some problems.

These control methods make it very difficult to supervise the operator and more importantly make quality control inaccurate. The operator has a full control over the equipment and therefore has a big impact on the outcome of the final product. Deliberate act or human error can cause flaws in the quality without anyone knowing the reason. It is almost impossible to find out afterwards what caused the problem.

Discussions with management of Siam Feather Products Co., whose factory was going to be the test site for the prototype, revealed details of these problems. The single biggest problem was seen to be quality variation as quality between product batches varied more than was seen acceptable. Manually controlled machinery was regarded particularly problematic. Especially during night shift, tired operators forgot to turn certain machine functions on or off. Sometimes during the end of operators shift, machine was left running until operator from next shift showed up, thus possibly leading to depreciation in quality. For example sometimes a batch of feathers needed to be cleaned again, as it was still stained after first run.

Even though punch card reader equipped machines were able to run the work cycle accurately, it still was not obvious whether the time between cycles was kept to absolute minimum during peak demand. This could mean that not all of the production capacity was put to use. In addition there was no way to keep track of errors and down time accurately as records of run cycles were kept by hand. Therefore optimization of production was difficult as bottlenecks in the production chain could not be found.

The aim of this project is to find out, if it is feasible to create a pc-based control system that replaces manual control of several different kind of factory machinery, by building a prototype. With this prototype users should be able to create program lists which control a work cycle of each machine and run them through a simple graphical user interface.

The focus is to create the system in a way that there is no need for big changes in the program even though different kind of equipment is used. This makes it possible to time and cost-effectively automate several machines using the same concept. Also, since the user interface remains the same, operators familiar with one machine learn to use another in a short period of time.

Even though PCs can based used to automate complete production lines, this thesis concentrates on automation of individual machines. Automation of a whole production line would require additional equipment such as lifts and conveyors, whereas the purpose of this prototype is to make running a single machine simpler.

2 PC Based Automation

In the past programmable logic controller has been the tool of choice when creating automation systems. These microprocessor based controllers were first introduced in 1960's by US automotive industry to replace various different control systems [1], [2, 429]. This made it possible to "program" the logic of the whole system with a single device instead of rewiring huge amount of relays. They are still widely used especially by many assembly automation equipment manufacturers as they are relatively cheap and reliable.

PLC's ladder programming model is enough for many needs, but for more complex operations and interfacing PLC might not be ideal choice. There is often a need to create more complicated user interfaces than PLCs allow and to interface into other equipment such as barcode scanners or printers. Some of these functions can be provided by PLC, but it raises the costs and makes programming complicated.

To overcome many of these PLC limitations, the answer has been to use PC for automation and industrial control. PC platform provides tools to handle more complicated tasks and offers a graphical user and development environment, which makes development, programming and actual use easier. PC is also much more flexible when it comes to hardware, as parts from different manufacturers work together. This also tends to keep prices down because of the fierce competition in the market. According to Keinänen, Kärkkäinen, Metso and Putkonen it is possible that PC-based automation cuts cost in half when compared to traditional methods such as PLCs [3, 12].

An even more important factor is the easy network connectivity of a PC. It can be used to connect to a database, to communicate with a third party device such as quality control cameras and to provide web based functionalities. More advanced graphical user interfaces also make it possible for operators to perform more complex tasks and configurations.

The biggest drawback of PC based automation is its relatively low reliability and stability. PC hardware is mostly designed to be used in office or server room, which makes industrial setting a very harsh environment for standard PCs. Dust is especially harmful for computer equipment with moving parts such as hard drives. Humidity and heat combined can shorten the life span of all computer equipment. Unlike PLC a computer might need to be rebooted, which leads to downtime and possible losses. However, there are several ways to protect PC equipment from less than ideal conditions, such as using industrial standard PCs.

3 Hardware

3.1 PC Hardware

PC will be often run for very long periods of time without interruption, which makes hardware reliability and durability the single most important requirement for the PC. As a most vital part of the system it must be able to withstand the high moisture level and temperature of factory environment. All hardware parts should be specially made for factory use, with low failure rates in order to maximize uptime. Absolute CPU speed is not an issue in this case, as there are no time critical processes running on the PC.

One restricting requirement for PC motherboard was a need for ISA slot. AD/DA card was only available with ISA bus connection, which is available in very few motherboards any more. Another drawback of using a ISA card is that I/O addresses, IRQs and clock speed have to be set up manually, which makes it much more complicated a task than setting up a PCI card. However according to Ph.D. Peng Zhang there is no specific need to prefer PCI over ISA, unless the aim is to use high-speed devices such as video cards [2, 329].

Other requirements for the PC:

- x86-based processor
- at least 2 PCI-slots available
- 100mbps network connection

3.2 Control Cards

There is a relatively big number of industrial measurement and card manufacturers in the market. There are a number of factors that were given thought when choosing a supplier for this project.

Reliability was a key concern, but it is fairly difficult to measure. Some manufacturers give a life expectancy estimate, while others do not. To make matters more complicated

some of the estimates are in operations before failure and others in mean time before failure. Comparison was made based on the manufacturer given life expectancy.

Most of the factory machinery is controlled by a set of relays, which in turn trigger different functions on and off. A low power electric current is used to control the relay, which acts as a switch and connects the high power circuit. For example pushing a button might connect a low power circuit to relay, which turns on a motor that feeds the material to the machine.

To be able to control a machine with certain amount of input relays we need a corresponding amount of outputs in the control equipment. As the amount of input relays varies from machine to machine, the amount of outputs in our control system needs to be very high or easily expandable. An example of industrial washing machine inputs is shown in Table 1.

Some machines use sensors as a means of control as well. For example an infrared detection sensor might measure the level of material in the machine and turn of a motor when that level has been reached. There is a wide variety of different sensors from load cells and temperature sensor to infrared detectors, but majority of the work in a similar manner. A sensor receives a signal and converts it to output voltage. It is this voltage which we can measure and use in our devices.

For most purposes we only need to know whether the signal is on or off, like in the infrared detector case above. For this purpose we need digital inputs. But sometimes we need to measure the voltage accurately. For example if there is a need to measure weight, the voltage from a weight cell must be measured and converted to kilograms. In order to do this we need analog inputs. Again, we need a matching amount of digital and analog inputs in the control system as there are sensors in the controlled machine.

Table 1. Example of industrial washing machine inputs

Input number	Description
1	Filling

2	Washing
4	Pre centrifuging
5	Recycled water
6	Water level 2
7	Water level 3
8	Town water
11	Outlet valve
12	Circulation pump
13	Soap 1
14	Soap 2
15	Bluing agent
16	Intermediate centrifuging
17	Centrifuging
18	Discharging

3.2.1 Industrial Control Card

PCI Industrial control card shown in Figure 1. is the heart of the system. It is used as a means of communication between the computer and the actual input and output cards (expansion cards). The card is connected to motherboard's PCI slot on the computer side and with parallel cable to the I/O cards. Card provides four I/O ports, which all contain 16 I/O lines, so the total number of possible I/O lines is 64, which is enough for most machines. All of these ports are programmable to be input or output channels. Industrial control card is Plug and Play capable, which makes it possible for the computer's BIOS to automatically detect the card and assign IRQ and I/O address. Operation temperature is from 0°C to 70°, humidity being 0-90%.

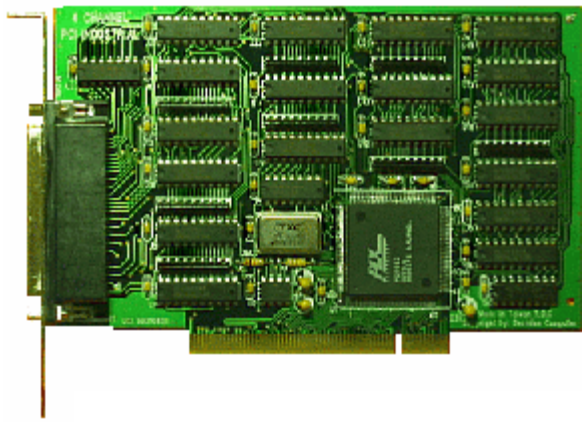

Figure 1. Industrial control card [4]

3.2.2 16 Channel Photo Isolator Input Board

The 16 channel photo isolator input board, shown in Figure 2, is used to provide 16 input channels from the controlled machine (sensor) to the Industrial control card. Photo isolation means that the signal can be floated and ground loop can be avoided. This isolated mode can be turned off. LEDs are used to indicate the digital on/off status of each input. When a led is turned on, there is a signal in the corresponding input. Board can use internal power from industrial control card or external 12V power source may be used. Features of the board are displayed in Table 2.

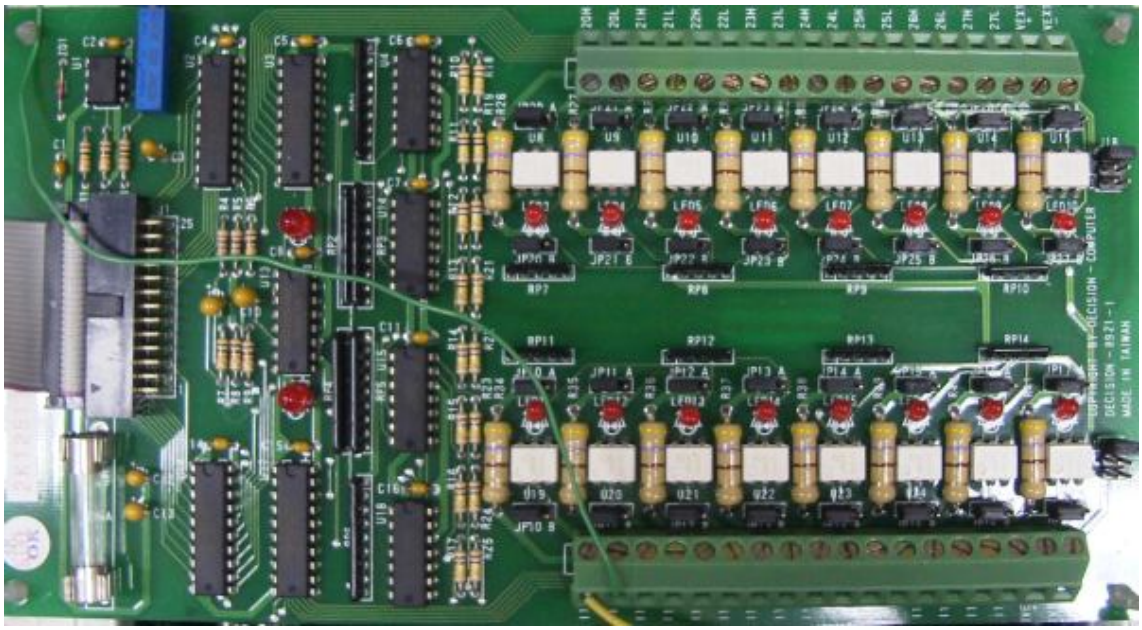

Figure 2. 16 Channel photo isolator input board

Table 2. 16 channel photo isolator board features [5]

16 channel relay output board
LED indicates when input channel is activated.
Internal and external power selectable.
Built in screw terminals for easy wiring.
Allow the input signals to be completely floated and prevent the ground loops.
Isolated or non-isolated modes selectable.
Input threshold voltage adjustable.
Breakdown voltage: 1500V DC.
Screw terminal: accept #22 to #12 awg wire.
Input current: 80mA maximum for each isolated input.
Input voltage: 30VDC maximum for each isolated input.

3.2.3 16 Channel Relay Output Board

The 16 Channel Relay Output board, shown in Figure 3, consist of 16 relays, which can be used to control external devices. However it should not be connected to high power relays beyond its operating voltage and current. As is the case with photo isolator input card, this output board has to been connected to industrial control card as it doesn't work independently. 16 LEDs are used to indicate the digital on/off status of each relay. When a relay is energized a corresponding LED will light. Board can use internal power from industrial control card or external 12V power source may be used. Features of the board are displayed in Table 3.

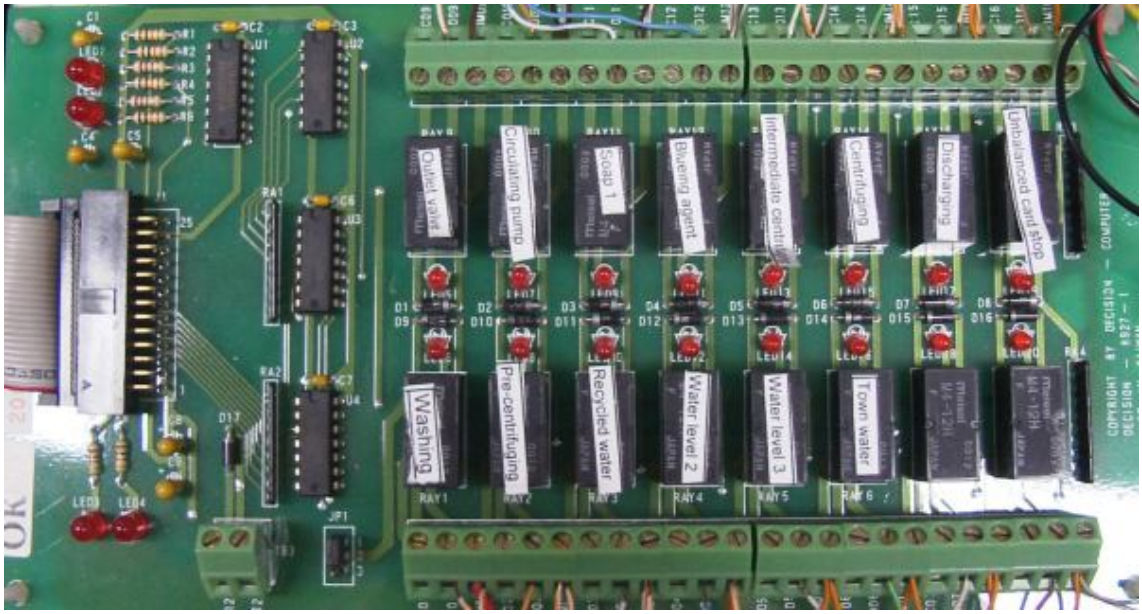

Figure 3. 16 Channel relay output board

Table 3. 16 channel relay output board features [6]

16 channel relay output board
LED indicates when relay is energized.
Internal and external power selectable.
Built in screw terminals for easy wiring.
The Normal Open (NO), Normal Close (NC), and Common contacts (COM) of each relay are brought out to the screw connector.
Max contract rating: 150V/DC 2amp, 125V/AC 2 amp.
Breakdown voltage: AC/DC 500V minimum/
Relay on time: 3 ms typical.
Relay off time: 2 ms typical.
Total switching time: 10 ms typical.
Isolation resistance: 100 M Ω minimal.
Life expectancy: 5 million operations at full load.
Screw terminal: accept #22 to #12 awg wire.
Power consumption: • +12V: 40mA for each relay, total 0.55 amps • +5V: <0.2 amps • -12V: <0.1 amps

3.2.4 12 Bit AD/DA card

12 bit AD/DA card, shown in Figure 4, provides 16 analog to digital channels and a single digital to analog channel. Analog to digital inputs are used, when we want to measure something, rather than just knowing on/off status. For example we might get 0 to 9V signal from a weight sensor. Each voltage value represents a corresponding weight within the sensors operating range. With these AD input channels, we are able to measure that voltage and to convert it to weight. Digital to analog output channel can be used to similarly control some external operation, using different voltages. For example higher voltage could mean higher rotation speed. Each channel can be used in bi- or unipolar state. Card is connected to ISA slot, so I/O port and IRQ need to be set manually. Features of the board are displayed in Table 4.

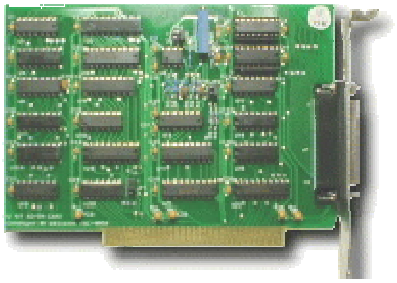

Figure 4. 12 bit AD/DA card

Table 4. 12 bit AD/DA card features [7]

Analog to digital features	**digital to analog features**
Support one 12 bits channel.	Support sixteen 12 bits channels.
Output voltage. • unipolar: 0V to 9V. • bipolar: -9V to 9V.	Input voltage. • unipolar: 0V to 9V. • bipolar: -9V to 9V.
Current setting time 500nsec.	Unipolar or bipolar selectable.
Nonlinearity 0.2%.	Successive approximation method.
	Conversion time 60usec. (each channel)

3.3 Enclosure

Operating conditions where the system is used are not ideal at all for computers. This makes it very important to choose enclosure that will protect the system from outside conditions. A Rittal steel enclosure, shown in Figure 5. was chosen for this purpose. It could house all the needed equipment, but was still compact and lightweight enough to be installed easily. Features of Rittal enclousre are displayed in Table 5.

Figure 5. Rittal AE 1045.500 Enclosure

Table 5. Enclosure specifications

Model	Rittal AE 1045.500
Width	400mm
Height	500mm
Depth	210mm
Volume	460dm
Net weight	13kg

The worst threats in a factory environment are dust, humidity and heat, which rose in test environment up to 45 degrees Celsius. The solution is to provide enough ventilation and/or cooling. In this case a special fan was installed on the top of the enclosure and fresh intake vent placed to the side. This keeps the air flow steady throughout the enclosure.

Dust other airborne particles affect this kind of system in many ways. HP: parts containing moving mechanical parts, such as disk drives can have bearing failures, due to these particles. Dust may also form a layer on components such as circuit boards and cause heating and humidity problems and lead to failure. Metallically conductive particles may also cause short circuits. To get protection from dust, the enclosure needs to be airtight and incoming air needs to be thoroughly filtered.

American Society of Heating, Refrigerating and Air-Conditioning Engineers (AHSRAE), recommends that humidity level is maintained below 60% (relative humidity RH) to guarantee long life for computer equipment [8, 8]. Even though this limit is under debate, it is known that high humidity causes galvanic actions to occur between some dissimilar metals, which will eventually lead to high resistance in connections [9]. This may cause an equipment failure. According to National Statistics Office of Thailand the average humidity percentage at the test site in Bangkok is 77% relative humidity, which is way too high for computer systems [10]. However, the only proper way to lower the humidity is to use air-condition, which wasn't appropriate in this case.

3.4 Power Supply

Uninterruptible power supply was installed inside the enclosure, as shown in Figure 6, to ensure smooth running of the equipment during line sags (brownout), when voltage levels drop, which can last for periods ranging from fractions of a second to hours. It also provides power to shutdown the system safely even if external power source is down (blackout). This makes it is possible to continue the program cycle from where it was stopped, instead of running the whole cycle again. This saves time, equipment and more importantly ensures that product quality always remains the same. UPS also

protects the system from power surges, voltage spikes and removes line noise caused by other high voltage machinery.

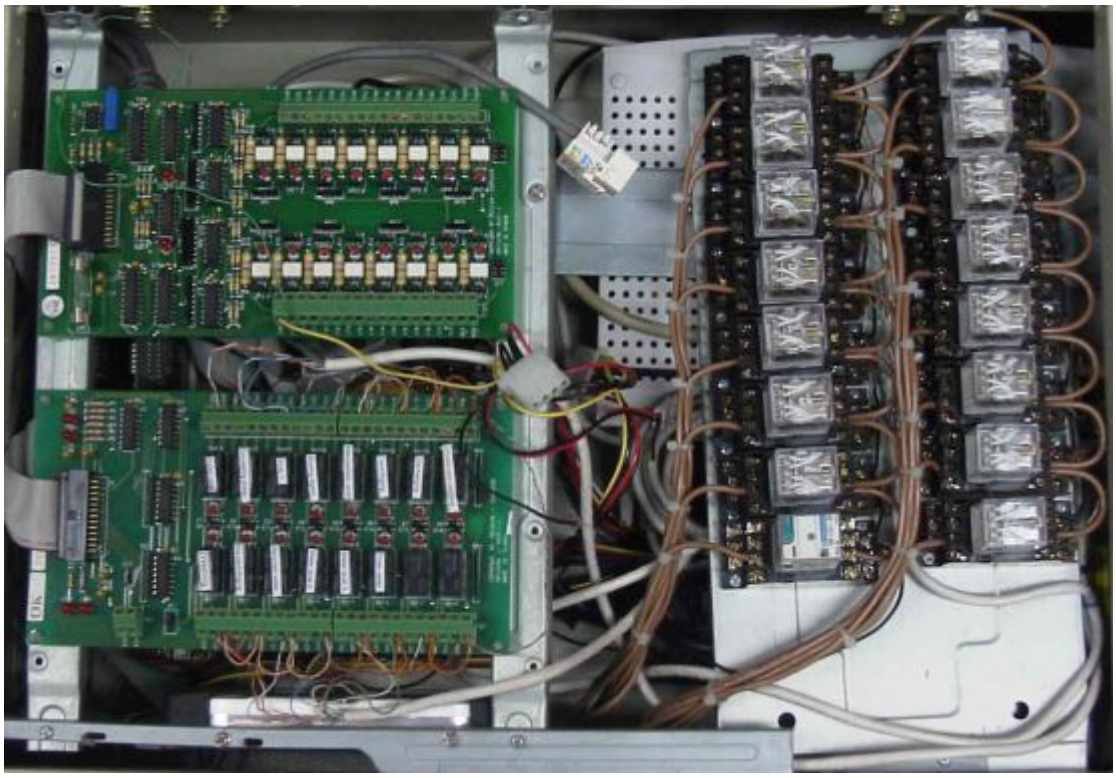

Figure 6. Inside enclosure

4 Software

4.1 Requirements

Control software in this study needed to be simple and clear, but still able to carry out all tasks needed to control the chosen hardware. The aim of the project is to be able to replace two different equipment control methods; manual control and punch card reader. Once the program has been set up it must be able to independently carry out the whole work cycle of each machine it is used to control for. There should not be any need for user input while the work cycle is running.

4.1.1 Punch Card Reader

Punch card is a plastic card, which has certain number of columns and rows. Each column represents corresponding relay in the controlled machine, while each row illustrates given amount of time.

Punch card reader turns relays on and off in line with corresponding holes in a plastic card. Card moves to next row by given time interval, a minute in this case, and checks whether each column has a hole or not. If hole a exists, a corresponding relay is set to on, otherwise relay is set to off position. These steps are repeated until reader reaches end of the card. If the cycle needs to be run again, the card needs to be entered into the reader again by hand.

System must be able to run exactly the same work cycle as is programmed to punch card. Excerpt of the cycle is shown in Table 6. Each relay must be switched on/off at given time. Once the work cycle has been started, there should be no need for operator intervention until the cycle has ended.

Table 6. Extract of a punch card control in washing machine work cycle

Time (min)	Relay channel	On/off	Description
0	0	On	Washing on
0	2	On	Recycled water on

1	2	Off	Stop recycled water
1	9	On	Circulation pump on
1	10	On	Soap on
2	10	Off	Soap off
4	9	Off	Circulation pump off
4	8	On	Open outlet valve
4	1	On	Pre centrifuging on

Work cycles that are programmed to punch cards must be reproducible with the control software. The program must handle multiple different cycles and they need to be easily modifiable with the program. One instance of the program must be able to use work cycle created on another instance and both must work in similar manner. Pre programmed work cycles need to be saved in a way that enables taking backups, in case of a system malfunction.

Operators must be able to manually control the machinery through software GUI as if they were using manual controls on the machine itself. All digital outputs can be switched on or off and analog output value can be assigned during program execution.

4.1.2 Manual Control

Manually controlled equipment is operated by buttons and switches that must be adjusted by hand. For example in dryer user turns on hot or cold air with a button and then sets the air flow rate with dial. Timing of each step in the cycle is very approximate and each cycle varies from another, which can lead to big differences in quality between batches.

The prototype must be able to run the same work cycle as operator operated cycle would be, shown in Table 7, combined with precise timing of each step. This will keep the quality constant and ensures each cycle will finish within time limits. An additional benefit is a reduced need of operator presence.

Table 7. Extract of manually controlled dryer work cycle

Time (min)	Relay channel	On/off	Description
0	0	On	Drying on
0	2	On	Cold air on
5	2	Off	Cold air off
5	3	On	Hot air on

4.2 Choosing the Programming Language

Among other languages C++ was carefully considered, as it would enable the benefits of compiled programming language, very fast to run being most important. However because of the lack of garbage collection, which leads to memory leaks and instability, problems were seen bigger than benefits.

Therefore, for this project Visual Basic was chosen as a most suitable programming language. Usually in automation applications, time and program execution speed are crucial factors. But as this project's end product is more a PoC prototype than a fully functional product, comparative slowness of Visual Basic is not a problem. Very short application development time and dependability are a lot more important in this case.

5 User Interface

5.1 Requirements

No matter how good a product is, it may not win user acceptance, if the user interface is not well designed. What makes a good user interface so important is that it is the only tool of communication between the system and a user. A simple change to a user interface can make a huge impact to productivity. In his book, Wilbert O. Galitz claims that a simple rearrangement of items in a GUI can lead to a 20 percent higher productivity [11, 5]. As the idea in this project is to be able to use exactly the same user interface to control wide variety of equipment, it is especially important to take this into consideration when designing the GUI. A good overview of system state must always be visible at a glance.

Most of the users were familiar with Microsoft programs such as Word, so the user interface design should be made very similar. As these people were also experienced operators, the GUI should also resemble a common control panel as much as possible.
The GUI must be designed to be used by normal screen, mouse, and keyboard combination, but also by touchscreen monitor.

5.2 Design Process

Despite of the touchscreen usage requirement, early on in the design process a decision was made that not all features must be usable with touchscreen. Different configuration and maintenance tasks can only be done effectively with mouse and keyboard, as they involve a lot of typing and file load/save operations. To make normal operation possible with touchscreen only, all buttons and switch must be made large enough to be comfortably pressed by average size index finger on a relatively small 12” touchscreen LCD monitor.

A very conservative color scheme was chosen; different shades of gray for background, red and green for on/off signals and switch. Areas with black text were kept white and light grey if possible. The goal was to create as natural user experience as possible, because many of the operators had never user a GUI to control machinery. Buttons are

tightly grouped by their use, to simplify the UI, as there is a need for large amount of buttons and indicators. Microsoft Visual Studio provides excellent ready made components for this kind of GUI development.

5.3 Functionality

5.3.1 Main Screen

Main screen, shown in Figure 7, is divided to five different sections, which are all used for different purposes. Inputs and digital inputs only display information of the current status of the system, while three others are used to both display and control purposes.

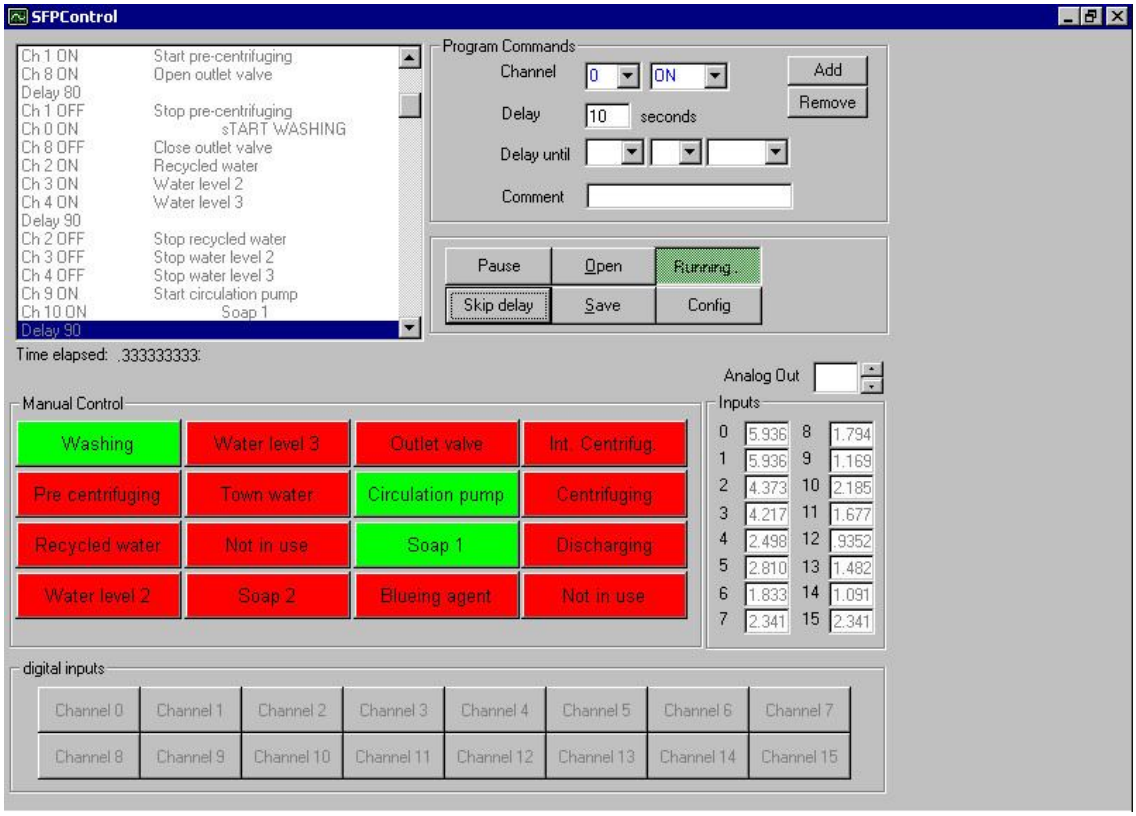

Figure 7. Main screen

List Display

This display is used to enter new “programs” and to monitor the program cycle. Program cycle is run from top to bottom of the list and the orders are executed in the same order as they are on the list display.

List is split to columns; leftmost displays the command to be executed e.g. “Channel 8 OFF” and rightmost functions as a comment field, in order to display the meaning of the command for the operator e.g. “Start program cycle”.

Program Commands

These buttons, combo boxes and text fields are used to enter new “programs” for the system. First, the user chooses which channel is turned off or on by choosing/entering the corresponding values from the combo boxes and clicks “add”. This is repeated until all the channels needed are displayed in the list. Next the time of delay is entered and added to the list. Alternatively, the user may adjust the delay to last until the defined input channel reaches desired value. Analog and digital input channels are distinguished by the operator = or >/<.

For example *“Delay until 1 > 5”* means that analog input channel 1 value must reach 5 volts, before program moves to next list item. In real life this could mean for example that weight sensor is used to measures the weight of water in our system. It is known that when weight reaches 100kg weight sensor’s output voltage is 5V. So when 100kg of water is needed we want to wait until 5V in corresponding input sensor has been reached.

“Delay until 2=ON” on the other hand means that digital input channel 2 must have value ON, before program proceeds. This could mean that water level sensor has been installed, which turns on when water level reaches 1 meter. So when 1 meter of water is needed, program should wait until signal in the corresponding input channels turns “on”.

Simple *“Delay 150”* instructs the program to wait for 150 seconds until next command is executed. This is a very commonly used command which is needed if program doesn’t wait for any input signal, but there is a need to keep output signal on for longer

period of time. *"Delay 150"* together with output signal on could be used to let motor run continuously for 150 seconds.

A command *"CH n ON/OFF"* sets corresponding output channels ON or OFF. To same extent *"Analog n"* adjust analog output value to given voltage value within output voltage limits. Digital ON/OFF can be applied for example to turn device motor on or off, whereas analog output would control rotation speed of the motor.

Comment, which describes the given command, can be added for each entry in the list. Operators can remove unnecessary entries from the list using "remove" button.

Control Group

Table 8 describes buttons that are used to run and control the program.

Table 8. Control group buttons

Pause	Pauses program until button is clicked again
Open	Opens a saved program.
Save	Saves new or changed program.
Run	Runs the program, if there are any inputs in the list (1). When program is running, same button is used to stop it.
Skip delay	Jumps to next item in the list (1)
Config	Opens the channel configuration screen. This item needs to be enabled by double-clicking the control group area and entering requested password.

Manual Control

During normal use main purpose of this group of buttons is to show, which channels are on and which off. Green light indicates channel is "on" and red "off". However same buttons can be pressed to turn channels on and off when program is running. Channel numbers and labels can be defined in a separate "configuration screen".

Digital Inputs

Digital input group shows the current value of each digital input channel, either “on” or “off”. Digital input labels are lit in purple, when corresponding input signal voltage is activated. Labels for buttons in inputs group can be modified on “configuration screen”.

Inputs

Input group displays the current value of each analog channel in voltages.

Analog Out

Analog out textbox displays the current output voltage of DA channel. Output voltage can be manually controlled with small arrow buttons next to the textbox.

5.3.2 Configuration screen

Purpose of the configuration screen, shown in Figure 8, is to be able to give different buttons a desired label and to choose which input/output channel is assigned to them. All information is stored to Windows registry. Same functionality would have been possible to obtain using INI or XML-file, but this option was seen as easier to use. Especially when considered once these settings have been saved they are seldom revised.

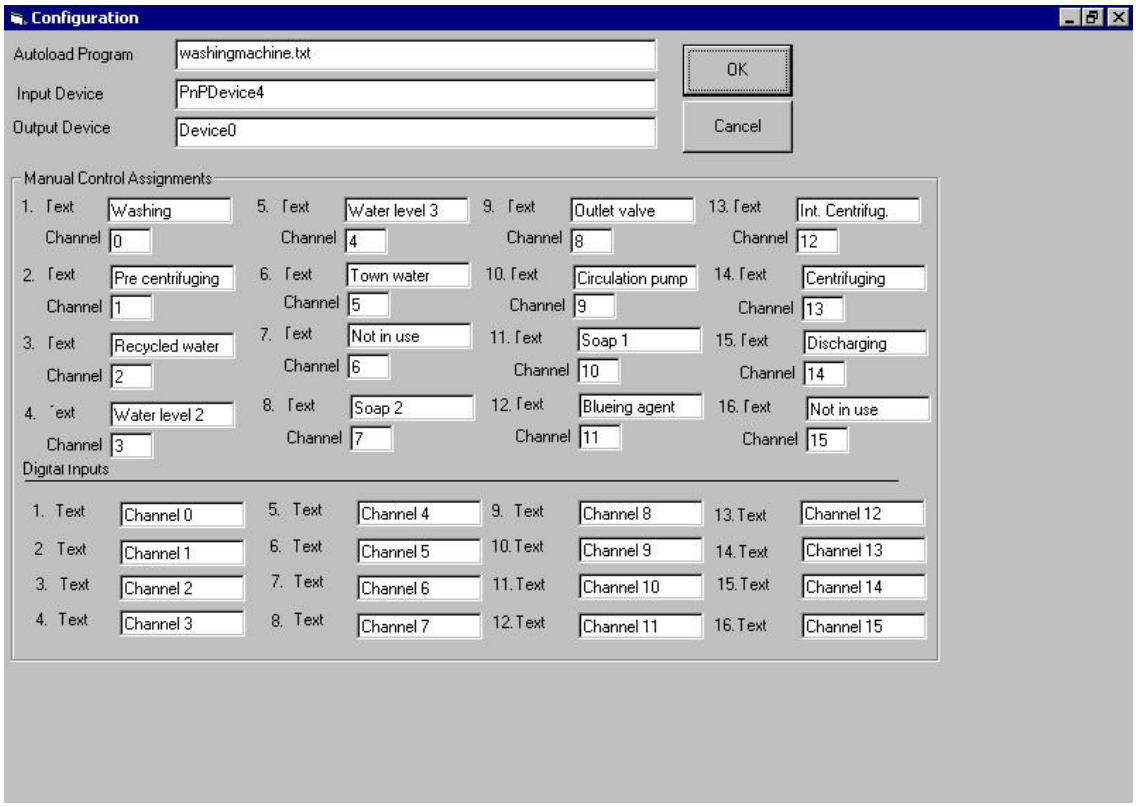

Figure 8. Configuration screen

6 Testing

Testing was conducted with L.H. Lorch industrial washing machine displayed in Figure 9. The purpose of this test was to define whether it was possible to substitute washing machine's punch card reader with pc control. This appliance was chosen because it had the most simple work cycle of the equipment available for testing.

Figure 9. L.H. Lorch industrial washing machine

The first step in testing was to set up software. Each step from washing machine's work cycle was carefully copied from punch card shown in Figure 10. to control program's list view control. Program was then saved and was set to load automatically. All control buttons labels were named after the corresponding inputs and outputs they were going to be connected to.

Electrical engineer was available to help with cabling to and from the washing machine. An extra set of relays shown in Figure 6. was needed between washing machine controls and 16 channel output cards relays as the built in relays weren't capable of handling washing machine's high operating voltage.

After setting up hardware, communication between the program and washing machine was thoroughly tested. Every input and output was individually tested for connection problems and to check that information displayed on user interface was valid and up to date. All analog channels were measured also with digital multimeter to confirm values were accurate. Before running actual work cycle all possible user actions were tested. The only problem found was two of the output connections were crossed due to inexact electrical drawing.

Figure 10. L.H. Lorch washing machine controls

Running the actual work cycle was very straightforward after everything was set up properly. According to the operator the washing program was run in exactly same order as it run with punch card reader, which is shown in Figure 10. However the actual results of the washing were only approximately checked and more precise tests need to be done. Pausing and skipping was also tested and found to be working as expected, even thought during normal production these functions are not needed.

7 Conclusions

After the initial testing period it became evident that it is feasible to build a PC based factory automation system with very limited resources. Work was completed within given time limits and the results met the predefined requirements. Chosen hardware parts work well together and software tools used were found to be suitable for the given project. Operators found the user interface to be fairly simple and easy to use. The only part that raised minor concern was how to see in which phase of the work cycle is being executed and what remains to be done.

The biggest question that remains unanswered is reliability. How will the system cope with harsh factory environment? Smatlab estimates it relay output cards life expectancy to be 5 million operations at full load [6, 2]. If each program cycle contains approximately 100 operations and the cycle and cycle is run 12 times a day, theoretically output card should last 11 years. With limited resources only realistic way to find out how reliable the system as a whole is, is to run it. Even though running the system in production requires more testing all people who participated in this project consider it successful, as predefined goals were met.

7.1 Development Areas

If this system will be put to production use, there are some features that should be implemented. The most important of these is a database connection. This would allow managers to follow the production more precisely than at the moment. At least start and end of program cycle should be recorded and knowledge of unnecessary stops would be usable information as well. Once the network has been set up, it is possible to use web servers and standard browsers to make all this information available everywhere.

As Visual Basic 6 entered Microsoft’s non-supported phase in April 2008, it would be sensible to implement the control program using .Net framework [12]. Same device drivers as used in this project should work with .Net framework seamlessly using built-in .Net COM interoperability.

An additional benefit of moving to .Net is its far better exception handling compared to VB6. In this project exception handling and errors were not a major concern, but in production use these issues require a lot of planning. For example how should the program behave when it cannot turn a relay output off? In real life application this could mean that water valve would be left open. Net's c++ derived try-catch statements allow better exception handling options than Visual Basics various different error handling procedures. As the exceptions are returned as objects, programmer can also handle them in a similar manner and find out exactly what caused the error [13]. This also makes it impossible to ignore exceptions, unlike in Visual Basic [14].

In this project a fast program development time was preferred over code quality and therefore, it lacks modularity. Code should be rewritten to allow some program functionalities to removed and added within minimum coding effort. This would allow for example to easily remove analog output control if it there is no need to control the given equipment. In modular programming a program is split to modules each of which carries out only its given function.

References

1 PLC Manual Web Site: PLC History, 2010. http://www.plcmanual.com/plc-history. WWW-document. Read 12.12.2009

2 Zhang, Peng: Industrial Control Technology, a Handbook for Engineers and Researchers. Norwich, NY, USA: William Andrew Inc, 2008.

3 Keinänen Toimi, Kärkkäinen Pentti, Metso Tommi, Putkonen Kari: Koneautomaatio 2, Logiikat ja ohjausjärjestelmät. WSOY, 2002.

4 Smatlab Industrial Control Card Operation Manual

5 Smatlab 16 Photo-isolator Input Card Operation Manual

6 Smatlab 16 Relay Output Card Operation Manual

7 Smatlab AD/DA Card Operation Manual

8 Whitepaper prepared by ASHRAE Technical Committee (TC) 9.9 Mission Critical Facilities, Technology Spaces, and Electronic Equipment: Gaseous and Particulate Contamination of Guidelines for Data Centers, 1.8.2008. http://tc99.ashraetcs.org/documents/ASHRAE_Extended_Environmental_Envelope_Final_Aug_1_2008.pdf. WWW-document. Read 4.1.2010

9 Fontecchio, Mark, Data center temperature range expanded: Who cares? 27.1.2009. WWW-document. http://searchdatacenter.techtarget.com/news/article/0,289142,sid80_gci1346115,00.html. Read 14.1.2010

10 National Statistical Office: Core Environmental Indicators. WWW-document. http://web.nso.go.th/eng/indicators/environ/environ.html. Read 22.11.2009.

11 Galitz, Wilbert O.: The Essential Guide to User Interface Design, an Introduction to Gui Design Principles and Techniques. John Wiley & Sons, 2007.

12 Microsoft: Support Statement for Visual Basic 6.0 on Windows Vista, Windows Server 2008 and Windows 7. WWW-document. http://msdn.microsoft.com/en-us/vbrun/ms788708.aspx. Read 20.2.2010.

13 Sarrel, Matthew D.: Making the Move from VB6 to VB.NET. WWW-document, August 2008. http://career-resources.dice.com/job-technology/making_the_move_from_VB6_to_VB.NET.shtml. Read 12.12.2009.

14 Richter, Jeffrey: Inside Microsoft .Net -ohjelmointi. Helsinki: IT-Press 2003.