

Automated production test of FLEX KRAFT

Fredrik Carlsson

Henrik Matsum

Kraftelektronik AB
Surte 2005

Department of Energy & Environment
Elektrotekniklinjen 180p
Chalmers University of Technology
Göteborg 2005

Abstract

In this thesis an automated test system for a modular rectifier unit was developed. The implemented test procedure was made using the software LabVIEW. To present the test data, an Excel program organizes all data in various tables. The raw data is stored in a text file. As a result of automating the test procedure, a reduction of the test time was made by 60 percent. Another important improvement is the nearly eliminated possibility of errors made by the test staff when handling and reading measurement instruments. It is now possible to make statistical calculations based on the stored test data. By automating the test system, a big step is taken towards getting a fully modernized production line. This is an important aspect when trying to stay ahead of the competitors.

Preface

This master thesis project was carried out during the spring of 2005 as a final part of our engineering education. The project was performed on the behalf of Kraftelektronik AB and took place at Kraftelektronik in Surte.

About Kraftelektronik AB

Kraftelektronik AB was founded in 1935 and is one of the world's leading manufacturers of equipment for electrical power transformation in specific areas. Initially, Kraftelektronik produced battery chargers but then progressed through converters for traction applications and welding equipment. Today, the main area is high rated switched rectifiers for applications such as surface plating and electrostatic filters. Their products are designed to withstand extreme conditions and still remain reliable in every situation and environment. Reliable power supply is very important today to ensure efficient operation. As the loads gets more technically advanced they also get more sensitive and therefore it is more important now than ever before to have uninterrupted power.

Kraftelektronik AB focuses mainly on the international market and the company employs about 100 people worldwide. Internationally, Kraft is highly regarded for the level of knowledge and experience and acts as a business partner rather than just a supplier. Long term relationships benefit Kraftelektronik as well as their customers

Kraftelektronik AB's latest product, the rectifier Flex Kraft goes through a production test consisting of a series of measurements of voltage and current. Today, these tests are performed manually and results in a substantial cost in the production of Flex Kraft. Therefore Kraftelektronik has established this master thesis work with the main objectives to automate the test procedure.

We would like to thank the following persons:

First of all, we would like to thank Sven Sjöberg for giving us the opportunity to carry out our master's thesis at Kraftelektronik AB.

We would also like to thank our examiner, Torbjörn Thiringer for not putting any constraints on our work but instead inspire us by letting us work in a pace that felt comfortable.

Finally, we would like to thank all the employees at Kraftelektronik AB. They have been very helpful and understanding of our demands. Especially the employees of the development department helped us a lot with equipment needed for the measurement experiments.

Contents

1	Introduction	5
1.1	Problem background	5
1.2	Flex Kraft – “A new generation of rectifiers”	5
1.3	Goal & purpose	6
1.4	Layout of thesis	7
2	Theory	8
2.1	Communication Standards	8
2.1.1	RS-232.....	8
2.1.2	RS-485.....	8
2.1.3	GPIB (IEEE 488)	8
2.1.4	Modbus.....	9
2.1.5	VISA.....	10
2.1.6	USB	10
2.2	Instruments & Equipment	10
2.2.1	Fluke 45.....	11
2.2.2	Tektronix TDS 220.....	11
2.2.3	NI USB-6008 Multifunction DAQ.....	12
2.2.4	Agilent 82357A USB/GPIB interface	12
2.2.5	Westermo MA 42	13
2.2.6	USB/RS-232 HUB	13
2.3	Separate or external modular instruments?	14
2.3.1	PXI & VXI	14
2.4	Software	15
2.4.1	LabVIEW	15
2.4.2	LabVIEW Application Builder	16
2.4.3	Other software	16
2.4.4	LabVIEW Example	16
3	The test system	18
3.1	Development	18
3.1.1	Phase one: In the office	19
3.1.2	Phase two: In the lab	20
3.1.3	Phase three: In the factory	21
3.1.4	Load Control	23
3.2	The test Program	24
3.2.1	Program overview	24
3.2.2	Visual test.....	25
3.2.3	Current control test.....	25
3.2.4	Voltage control test	26
3.2.5	Graphical interface	26
3.2.6	Report generation	28
3.2.8	Debugging	31
3.3	Practical implementation.....	31
3.3.1	Programming	31
3.3.2	Communicating with Flex Kraft	32
3.3.3	Communicating via GPIB and RS-232	34
3.3.4	Implementing the data storage	35
3.4	New system vs. Old system.....	36
3.5	Future development.....	38
4	Summary	39
5	References	40

1 Introduction

1.1 Problem background

The development towards the use of automated tests has been increasing over the last decades. There are many reasons for this but the availability of modern technique and the increasing competition and demands in the area of production are worth mentioning. Due to the production of Kraftelektronik AB's new rectifier Flex Kraft a better system for product testing was needed. Until now, the rectifiers have gone through a thorough manual test before leaving the production. Since the production is expected to increase, Kraftelektronik AB wants automated tests to save time and money but also increase the liability of the test procedure

1.2 Flex Kraft – “A new generation of rectifiers”

Flex Kraft is a modular switch mode rectifier system designed to run trouble free in harsh environments such as the plating industry. This new product is a concept built on putting together one to ten rectifier modules to achieve the desired power for the specific application. The customer specifies their power rating demands and then buys a perfectly customized rectifier containing a controlling module, and one to ten rectifier modules. If the customer's power need is changed, modules can be added or removed with minimal efforts. Flex Kraft can deliver up to 6000A (0-20V) and the user can easily change the voltage and current parameters via the manual or remote interface. Another benefit of the unit is the low ripple. Low ripple means you can run a process without having to add an additional filter and therefore keep the investment cost lower than before. Flex Kraft is accessible from the front, which provides the possibility to save floor space as you can install the units side by side in a small area. Other existing rectifiers for plating power is not accessible from the front but from the side and therefore it is not possible to put two or more units side by side without having to move them during maintenance and repair. Figure 1 show a picture of a Flex Kraft containing nine rectifier module and figure 2 shows the control panel of Flex Kraft.



Figure 1. Flex Kraft, nine modules



Figure 2. Control Panel

1.3 Goal & purpose

The aim was, as the thesis title implies, to develop an automated production test for Kraftelektronik AB's latest product Flex Kraft. The test is to be performed from a PC, controlling the test subject as well as the measurement instruments. The test program should follow Kraftelektronik AB's testing criteria for Flex Kraft but also generate a test report after the completion of a test. Regarding the purpose of this project, it is of increasing importance for Kraftelektronik AB to implement effective and automated tests because the sales volume is expected to go up with the new introduction of Flex Kraft. Therefore it is beneficial to incorporate automated testing to keep the error statistics low. Kraftelektronik AB also wanted to acquire knowledge in the field of automated testing. In the future Kraftelektronik AB plans on using the acquired knowledge for developing more automated test systems. These future systems are not only to be used in production tests but also in service and repair work on existing installations.

1.4 Layout of thesis

The report starts with an introduction followed by a problem description in chapter one. In chapter two the different communication standards are discussed. This chapter also contains information about the used instruments, equipment and software. A detailed description of the system development and programming is given in chapter three which may be the most important and interesting part of this report. The reader can here follow the different stages of developing an automated test system and learn some of the difficulties and problems that occurred during the work. At the end of chapter three there is also a part describing ideas for future development of the test system.

2 Theory

2.1 Communication Standards

2.1.1 RS-232

The serial or RS-232 was firstly developed in 1969 and it was commonly used for connecting peripherals such as mice and keyboards to PC's. In instrument control, it is mostly used for controlling laboratory and analytical type instruments such as scales and bar code readers. RS 232 can achieve transfer rates up to 28.8 Kbytes/s and distances of up to around 30 meters. The RS-232 signals are represented by voltage levels with respect to ground. There is one separate wire for each signal, together with the ground signal (reference for voltage levels). RS-232 is useful for point-to-point communication at slow speeds. The main benefit is that it is inexpensive for instrument vendor to implement since the RS-232 UART is built in to most microprocessors.

When setting up a RS-232 communication it is very important that all communicating devices have exactly the same values on the various parameters such as baud rate, parity, flow control etc. Our personal reflection which also seems to add to the public opinion is that it is definitely more complicated to set up a RS 232 communication system than a GPIB communication system. However, the RS-232 is so widely available that it is certain to stay with us for some time to come.

2.1.2 RS-485

The RS-485 is used in industry for control systems and data transfers where the communication is exposed to electromagnetic interference from other equipment. RS-485 uses a different principle than for example RS-232: Each signal uses one twisted pair (TP) line - two wires twisted around them selves. This is called "Balanced data transmission", or "Differential voltage transmission". For RS-485 the distance can be up to 1200m, and commonly available circuits work at 2.5 MB/s transfer rate.

RS-485 is used for multipoint communications i e. several devices may be connected to a single signal cable, similar to e.g. ETHERNET networks, which use coaxial cable. Most RS-485 systems use Master/Slave architecture, where each slave unit has a unique address and responds only to packets addressed to this unit. These packets are generated by Master (In our case a PC with LabVIEW), which periodically polls all connected slave units.

2.1.3 GPIB (IEEE 488)

In 1975, the IEEE 488 Standard Digital Interface for Programmable Instrumentation was released. The IEEE 488 contained the electrical, mechanical, and functional specifications of an interfacing system. In 1990, the IEEE 488.2 added the Standard Commands for Programmable Instrumentation (SCPI) document. SCPI specifies commands that each instrument class must obey. The drawback is that not many instruments are SCPI compliant. HS488 is a high-speed extension of the IEEE 488 standard that defines a handshaking method that increases the bus performance by nearly 10 times.

The GPIB bus is still extremely popular today, more than 30 years after its creation. The main reason for its popularity is the very large installed base of GPIB instruments (about 5-10 million instruments). Other reasons are that the GPIB bus is robust and reliable, its defined handshaking mechanism and its shielded and rugged industrialized connectors. Regarding the speed, GPIB has always been faster than other buses and is still today relatively fast with 1.5 Mbytes/s transfer rate (IEEE 488). The HS488 is very competitive with its 8Mbyte/s transfer rate.

The GPIB standard is considered being easy to use and during this project we managed to set up the GPIB communication with little or no effort. It is really noticeable that LabVIEW is designed to work well with GPIB. The biggest encountered problem was to obtain the suitable GPIB commands from the instrument manuals. This was sometimes rather difficult. Because of the relatively high transfer rate and its widely spread base, GPIB will be the most common standard for several years to come. Figure 3 shows a GPIB or IEEE 488 cable.



Figure 3, GPIB cable

2.1.4 Modbus

Modbus is an open serial communication protocol that is based on the Master/Slave architecture. Since it is easy to implement on all kinds of serial interface, it has gained wide acceptance. The Modbus protocol was originally developed in order to link controllers into a network, but has however frequently been applied for connecting input/output modules. Because of the low transmission rate, which has a maximum of 38.4 kbits/s, Modbus is most suitable where the number of devices on the bus is small and where low demands are placed on response time.

The bus consists of a master station and of a number of slave stations, and the communication is controlled entirely by the master. Modbus contains two basic communication mechanisms and data can be transmitted either in ASCII code or packed into RTU format:

Question/answer (polling): The master sends an inquiry telegram to any one of the stations, and then waits for the answer telegram.

Broadcast: The master sends a global command to all the stations on the network, and these execute the command without providing any feedback.

2.1.5 VISA

Virtual Instrument Standard Architecture (VISA) is a standard defined by Agilent Technologies and National Instruments for communicating with instruments regardless of the interface. These interfaces are: GPIB, VXI, PXI, Serial, Ethernet, and/or USB.

VISA provides the programming interface between the hardware and development environments such as LabVIEW, LabWINDOWS/CVI, and Measurement Studio for Microsoft Visual Studio. NI-VISA is the National Instruments own implementation of the VISA I/O standard. NI-VISA includes software libraries, interactive utilities such as NI Spy and the VISA Interactive Control, and configuration programs through Measurement and Automation Explorer.

2.1.6 USB

Universal Serial Bus, or USB, is a computer standard designed to eliminate the hassle of opening the computer case for installing cards needed for certain devices. USB as its name would suggest is a serial bus. It uses four shielded wires of which two are power (+5v and GND). The remaining two are twisted pair differential data signals. It uses a NRZI (Non Return to Zero Invert) encoding scheme to send data with a sync field to synchronize the host and receiver clocks. USB supports plug'n'play with dynamically loadable and unloadable drivers. The user simply plugs the device into the bus. The host detects this addition, identifies the inserted device and loads the appropriate driver in a matter of seconds. The end user does not need to think about such things as IRQ and port addresses, or rebooting the computer. Once the user is finished, they can simply pull the cable out, the host will then detect this and automatically unload the driver. The USB standard is widely spread and popular for its user-friendly handling and there are also a variety of peripherals such as fans and lamps that is supplied with power from the USB port. Concerning the speed of USB there is in to days version (USB 2.0) three different USB speeds, Hi-Speed that allows for speed up to at least 480 Mbits/s, Full-Speed that has the same speed as USB 1.1 at 12 Mbits/s and Low-Speed with 1.5 Mbits/s.

2.2 Instruments & Equipment

This part of the report describes the different instruments and interfaces used. The instruments are only described in general and for more in depth information please refer to the individual instrument manuals.

Instruments used for this master thesis work include a Tektronix TDS 220 oscilloscope and Fluke 45 multimeters along with an Agilent 82357A USB/GPIB interface. Both the Tektronix oscilloscope and two of the Fluke multimeters use a GPIB pc interface and are all connected to the operating pc using the Agilent 82357A USB/GPIB interface. The other two Fluke 45 multimeters are connected to the pc via RS-232 because there were no GPIB modules installed on those. Although there is a possibility to use either RS-232 or GPIB on both the oscilloscope and the multimeters the GPIB interface standard were chosen where available because of its simpler operation and easier implementation in LabVIEW.

2.2.1 Fluke 45

The Fluke 45 is a dual display multimeter that can be controlled from a pc via either RS-232 serial communication or GPIB. In this project four Fluke 45s were used to measure both voltage current and voltage ripple. The GPIB control is an extra option as opposed to the RS-232, which comes as standard on the Fluke 45 therefore only two of the four used Flukes had GPIB control. A Fluke 45 is shown in figure 4 below.



Figure 4. Fluke 45 multimeter

2.2.2 Tektronix TDS 220

The Tektronix TDS 220 is a two channel digital oscilloscope that has become a benchmark for low-cost digital oscilloscopes, offering a combination of performance, reliability and versatility. By sampling at 16 times the bandwidth on both channels, the TDS220 provide accurate real-time acquisition up to full bandwidth. Features available include automatic measurements, peak detect, storage of two reference waveforms and auto set. Peak detect along with high sample rates help to minimize aliasing. It has a backlit liquid crystal display that has a fast update rate and good detail. The layout of the buttons and dials on the front is similar to that of an analog oscilloscope but they are grouped by function, and each channel has its own dedicated scale and position dials. Two different extension modules are available for the TDS 220, the Communication extension module allows RS-232 and GPIB control of the oscilloscope and the Measurement extension module combines the features of the Communications module plus Fast Fourier Transform (FFT) analysis and four additional measurements. Here the TDS 220 was used to check for voltage transients on both channels of the tested rectifier. Therefore only one oscilloscope was needed because both of its channels could be used. In figure 5 the reader can see a Tektronix TDS 220 oscilloscope.



Figure 5. TDS 220 Oscilloscope

2.2.3 NI USB-6008 Multifunction DAQ

The NI USB-6008 is a compact and low cost multifunction USB DAQ designed for applications where low cost and portability are valued more than high performance. Ease of use is also high on the list of features found on the NI USD-6008. There are also different types of ports on the NI USB-6008 namely eight 12-bit analog inputs, 2 analog outputs, 12 digital I/O lines and one counter. Additionally there is also a 5V 200mA output, which as the rest of the unit is powered from the USB cable. Operation from Lab VIEW is easy and the NI USB-6008 comes with readily made VIs for all different operations possible. Below, figure 6 shows the USB-6008 multifunction DAQ.



Figure 6. USB-6008 multifunction DAQ

2.2.4 Agilent 82357A USB/GPIB interface

The 82357A from Agilent is a USB to GPIB interface that uses USB1.1 and has a transfer rate of over 850KB/sec. This device had already been purchased by Kraftelektronik AB before the start of this master thesis project and it were chosen instead of a similar USB/GPIB interface made by National Instruments because it costs about 1000 SEK less. Some trouble solving was needed to get the Agilent USB/GPIB interface to function properly with LabVIEW. Based on these facts another USB/GPIB interface that was purchased by Kraftelektronik AB for another project was from National Instruments. After all problems were sorted there were in fact a software update released by Agilent that would have solved most of the problems quicker. Figure 7 shows the 82357A from Agilent.



Figure 7. Agilent USB/GPIB interface

2.2.5 Westermo MA 42

The Westermo MA 42 is a serial interface conversion unit that is made for converting signals between a RS-232 master unit and a RS-422/485 unit or network. It can be used with either two or four communication cables for the RS-422/485 transmissions but in this master thesis project it was used with two cables and RS-485 to correctly match the specifications on the Flex Kraft control unit. The MA 42 is no longer in production and has been replaced with the MA 45 but the MA 42 was used in this project because it was already available at Kraftelektronik AB and it has all functions and all the performance that was needed. The Westermo Ma 42 is shown in figure 8 below.



Figure 8. Westermo MA 42

2.2.6 USB/RS-232 HUB

This is a standard no name USB to RS-232 converter with two RS-232 ports and it is shown in figure 9. It is used to connect two of the Fluke 45 multimeters to the computer because the standard serial port on the computer is connecting the Westermo MA 42.

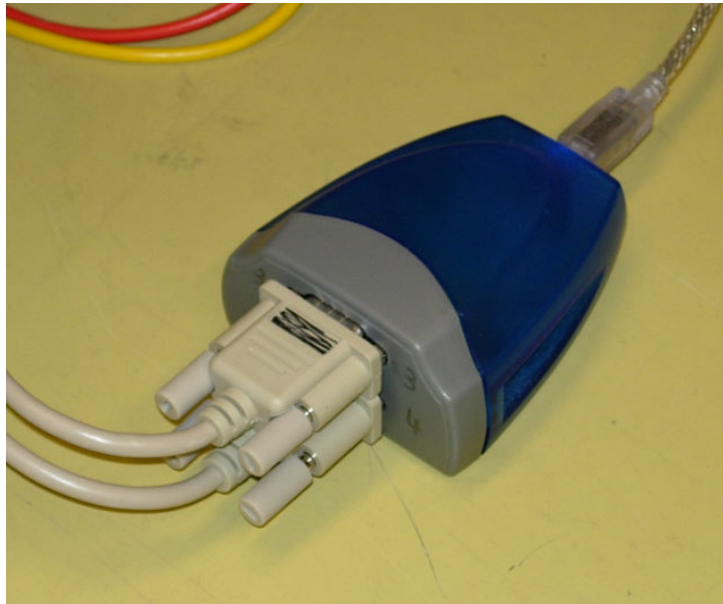


Figure 9. RS-232 Hub

2.3 Separate or external modular instruments?

The obvious advantages of modular instruments like PXI or VXI are the small and very compact physical size and construction.

Regarding drawbacks, the first one noticed is to get relevant customer support for modular instruments from different vendors. For instance if an error occurs in a system consisting of a blended mix of instruments from different vendors and you were to contact one vendors support, would they be able to give you an objective solution? It is likely that they would point out one of the other vendor's product as the source of your problem. With traditional instruments communicating e.g. via GPIB it is more obvious where the areas of responsibility lies. If all instruments but one works well in your system you naturally get in contact with that instrument vendor's support. Secondly, modular instruments like PXI and VXI uses Windows as operative system. This is not very suitable for industrial applications because of its vulnerability.

2.3.1 PXI & VXI

PXI, (PCI Extensions for Instrumentation) is a PC-based platform for measurement and automation systems that combines PCI bus features with a modular mechanical packaging. Specialized synchronization buses and key software features are added and this makes it both a high-performance and low-cost platform for measurement and automation systems. PXI is modular, which means that the user easily can customize the system to obtain the desired solution. VXI (Versa Module Europe Extension for Instrumentation) is a predecessor and similar to PXI but utilizes the VME bus instead of the PCI bus.

2.4 Software

2.4.1 LabVIEW

In 1987 National Instruments created LabVIEW1, a graphical programming language that is short for Laboratory Virtual Instrument Engineering Workbench. Since then LabVIEW has gone through a rapid development to become one of today's most used software for data acquisition and measurements. This growth is fueled by the tremendous advances in personal computers and consumer electronics. Virtual instrumentation is not just about the price benefits or its high performance with today's PC's. It is a new way of thinking and the functionality of an instrument is no longer determined by the vendor but by the user. For example, the user can choose how the measurement is made and what analysis to be done but also how the data is to be presented, stored, or transmitted over a network. One of the most important features of virtual instrumentation is the development time compared to traditional instruments. Now, when a new data analysis is developed it can be installed from a disk or from the network within minutes. The development time for traditional instruments is far too long and the price becomes too high.

By graphical language means that instead of traditional text based programming, the code is build by icons and wires. The user interface is known as the front panel and the code to control this interface is located in the block diagram. The interface is built by using a set of tools and objects. However, the main part of the programming is made in the block diagram. If organized properly, the block diagram resembles a flowchart.

In LabVIEW, the program execution is determined by data flows whereas typical programming language uses instructions to determine the execution. A LabVIEW program is called a VI, virtual instrument and as the name suggests that is exactly what it is in most cases. The main area of use is within instrument control and data acquisition. Programming LabVIEW is fairly easy and although basic programming skills are useful it is a very accessible language even for beginners. Of course the user has to be familiar with the standard concepts of programming such as the for/while-loops, strings and arrays and different data types. Especially the presentation of data in plots and graphs is very straightforward and to develop advanced interfaces is much less complicated than in other programming languages such as for example JAVA or C++.

Similar to traditional text based programming is it in LabVIEW suitable to use subroutines to simplify and organize your program to make it well designed and easy to understand. In LabVIEW we call them subVIs and they are represented by building blocks just like every other function. One particularly useful feature is that the subVIs are executable which really makes debugging simple. It is recommended to change the visual appearance of the sub VIs so that they easily could be distinguished from the rest of the blocks in the program. Otherwise the overall function of the program is hard to understand because all the blocks look exactly the same.

As is the case of most Software LabVIEW is available in several different versions with the simplest and most basic one being the "LabVIEW Base Package". This version differs from the more advanced versions when it comes to analysis of data but also the report generation possibilities is very different through the versions. In fact, in the "LabVIEW Base Package" there is no report generating functions whereas in the more advanced version "LabVIEW Developer System" there is a built in ready to use function for this purpose that definitely would have saved hours of work during this project.

2.4.2 LabVIEW Application Builder

The LabVIEW Application Builder is an add-on software to create stand-alone executable applications for running on computers without LabVIEW installed. It is very useful to be able to run LabVIEW programs on other PC's than those used for development. Since Kraftelektronik AB only bought one LabVIEW license, it was necessary to buy Application Builder.

2.4.3 Other software

There are several data acquisition software tools on the market today. We chose LabVIEW because of its wide spread and user friendliness but also because we learnt that National Instruments supposedly has good customer support. Since we are beginners in this specific area, we thought it would come in handy for us during this project. However, before we decided to go with LabVIEW we investigated a number of software such as LabWINDOWS from National Instruments and DasyLAB from Dasytec.

2.4.4 LabVIEW Example

When creating a new VI it is often suitable to start in the front panel by placing the necessary objects forming the user interface. From the menu, "controls", the programmer can access a large number of objects such as Graphs, Indicators and Controls. The last one typically represented by a volume knob or a slider. The following example shown in figure 10 explains farther how the front panel/user interface is built up. The user controls the tank level by setting the volume knob in the left corner of the interface. The level is also shown in the text indicator above the tank.

Objects on the front panel are represented by symbols (blocks) in the block diagram shown in figure 11. To connect these blocks by wiring is necessary for making the program work. Before doing so, the blocks are just defined. Not used. Wiring is a central part in LabVIEW programming and it is done by selecting the wiring tool in the tools palette and then just left click on one block with the mouse and move the cursor to the target block and left click again. Depending on which data type the wired blocks has, the wire changes color. For instance if the wire is blue, the data type is Integer. If it is orange, it is double. In this example the controls are wired to the indicators and placed within a while loop that uses the stop button as a running condition. One should have in mind that this is a very basic example but still shows some of the most used components.

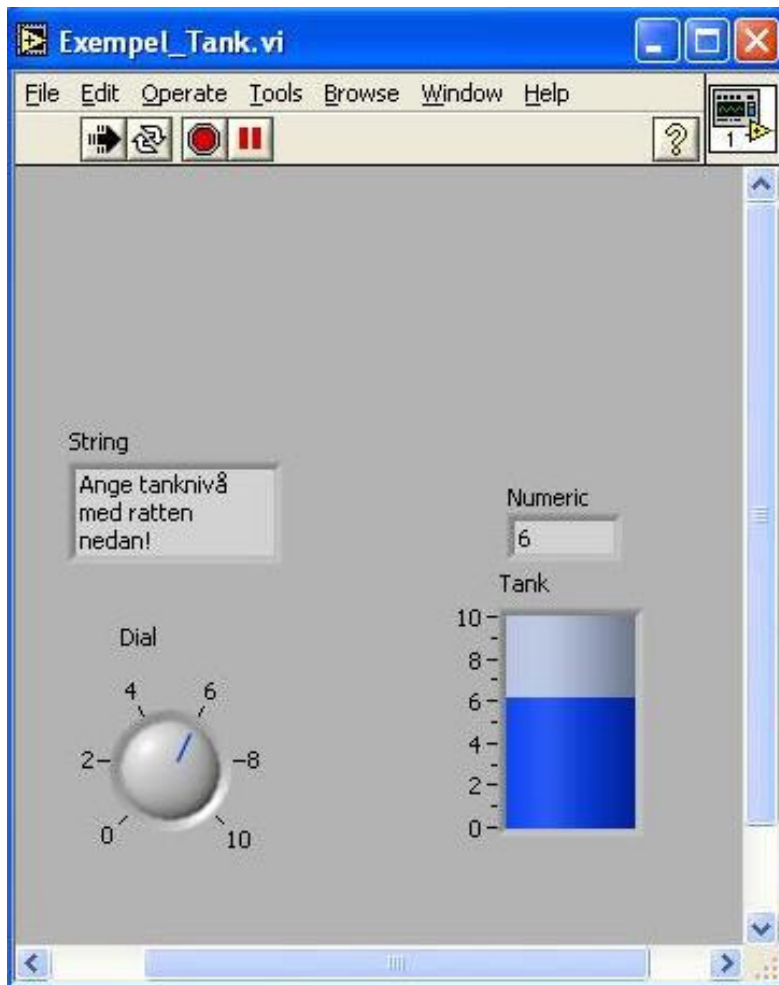


Figure 10. Lab VIEW example front panel

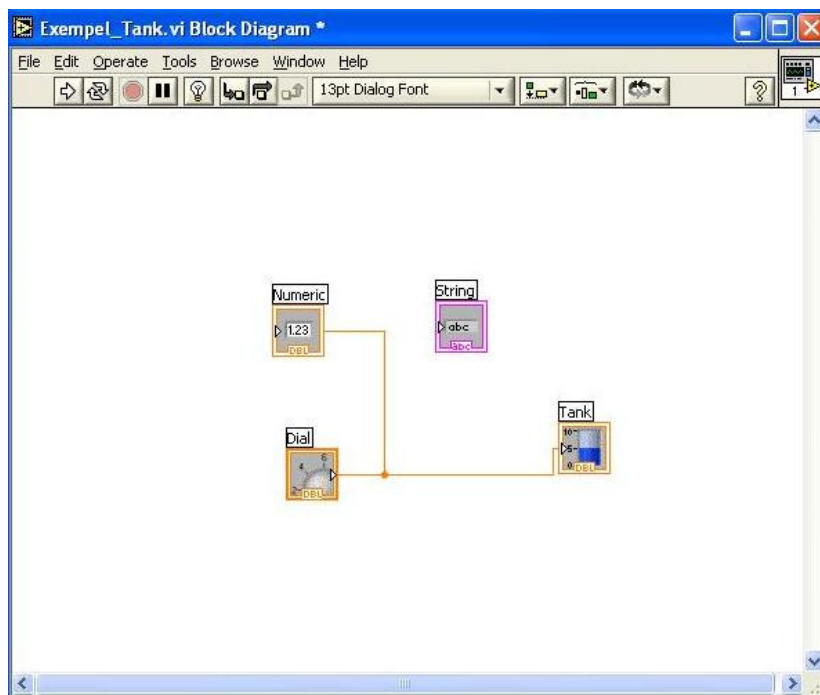


Figure 11. Lab VIEW example block diagram

3 The test system

3.1 Development

Since this project was entered with little or no knowledge and experience of automated measurements or about the software used to implement the test protocol, it had to start with a large literature study. The literature study was conducted mainly via the Internet, but also from reading books and performing interviews. There is a whole world of literature concerning this area and an endless number of different manufacturers, vendors, communication standards, protocols, ideas and examples. It took weeks just to get some sort of overview of the automated measurements area and figure out what was needed to accomplish the desired goals. When it stood clear which measurement instruments that were available within the company the problem and the relevant information could be narrowed down significantly.

The entire system has a basic structure that can be seen in the system overview in figure 12 below.

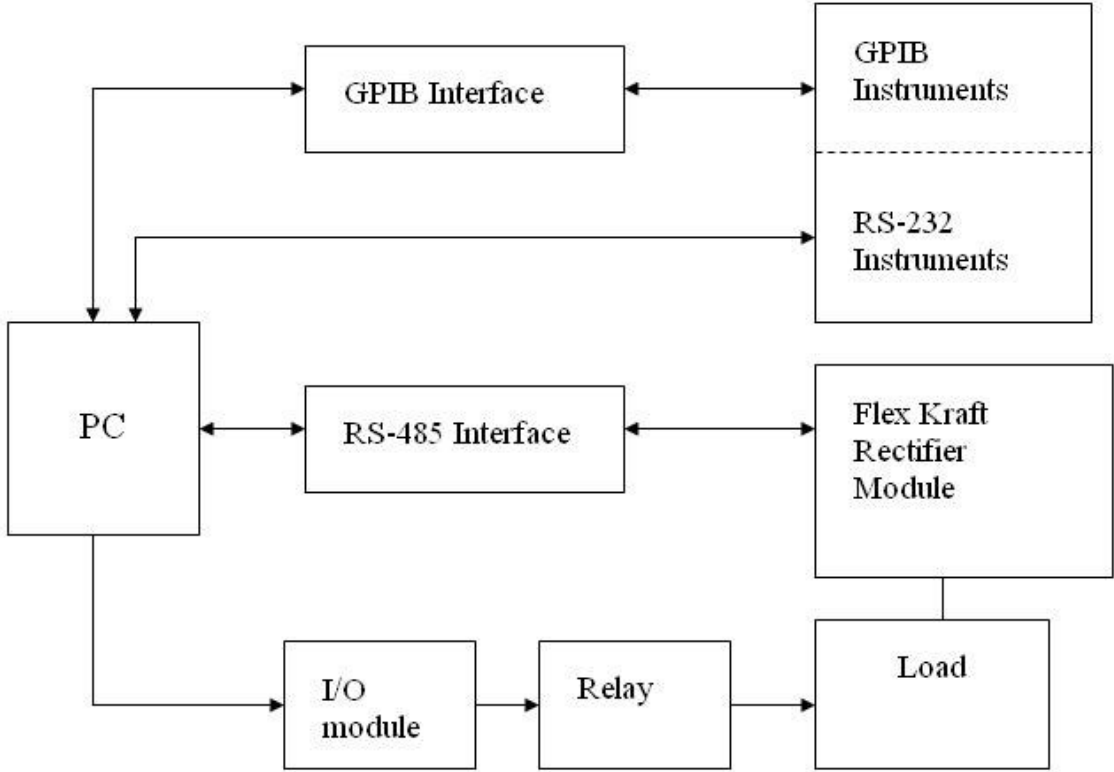


Figure 12. System overview

3.1.1 Phase one: In the office

From the start of the development phase of the project different parts of the test system was set up in the office for compatibility testing and programming verification. Early on there were only some instruments at a time but as the project progressed and the control program became more ready more and more components were interconnected.

In the beginning of the communication programming stage only the USB/GPIB interface and one Fluke 45 were needed to verify that the programmed code was working with the multimeter. The same procedure was carried out when the code for the oscilloscope were written. The biggest, or maybe only problem, when writing the code for the GPIB instruments were to understand each instruments instruction manual regarding the different commands, measurements and operations available. There were also some uncertainties as to how the GPIB commands should be combined with each other.



Figure 13. Office setup

If the GPIB programming was easy and problem free, then the Modbus programming of the Flex Kraft control unit was the exact opposite. Initially there were some major questions how to perform Modbus communication via LabVIEW and not even the people at NI Sweden knew how to do it. Eventually the LabVIEW and Modbus problems were solved after using special Modbus VIs from NI International.

When the test program was almost ready it was time to connect all components of the test system in the office so that the entire test program could be run through. This was delayed around one week because one of the four Fluke 45 multimeters was sent away for calibration.

The office setup of the instruments at this point can be seen in figure 13. Prior to this it had become clear that not all four of the Fluke 45s were equipped with GPIB, two of them only had RS-232 interface. This however didn't cause any major problems since the program is using the Instrument I/O assistant to send the VISA commands to the GPIB instruments and since VISA also works via RS-232 it was only a matter of changing the I/O assistant settings for the two Fluke 45s using RS-232.

At the end, all components of the test system except for a rectifier module were connected and running in the office.

3.1.2 Phase two: In the lab

When the time came to start testing the system in the lab with an actual rectifier module some initial problems arose. Firstly there were quite some time spent trying to get a working rectifier module. Initially two different preproduction modules were tested but they did not work properly and kept blowing up their power transistors. After blowing up several transistors the modules were replaced with a working module taken directly out of the production line.

Before the actual simulation of the test procedure could begin the adjustable load that was used had to be measured up to determine the two different load levels that are used in the current control testing and in the voltage control testing. This experimental load which is shown in figure 14 is adjusted by connecting and disconnecting resistive wires by the means of knife looking switches. There are 15 different resistive wires with varying resistive values that can be switched in or out. The load wires are when switched in connected in parallel to each other. The load has three external connection points two of which are connected to each side of the load wires via the switches and one that is connected to the middle point of each load wire.



Figure 14. Variable load used in the lab

When the two different settings for the load were determined the actual testing began. During the testing there were no major problems and the time was spent on perfecting the program

and trimming all delays. Because there were no possibility to automatically change the setting on the load this had to be done manually and therefore every test cycle took much longer to complete than the time it takes to complete one test cycle in the production line where the load is automatically changed. A picture of the total test system in the lab can be seen in figure 15 below.

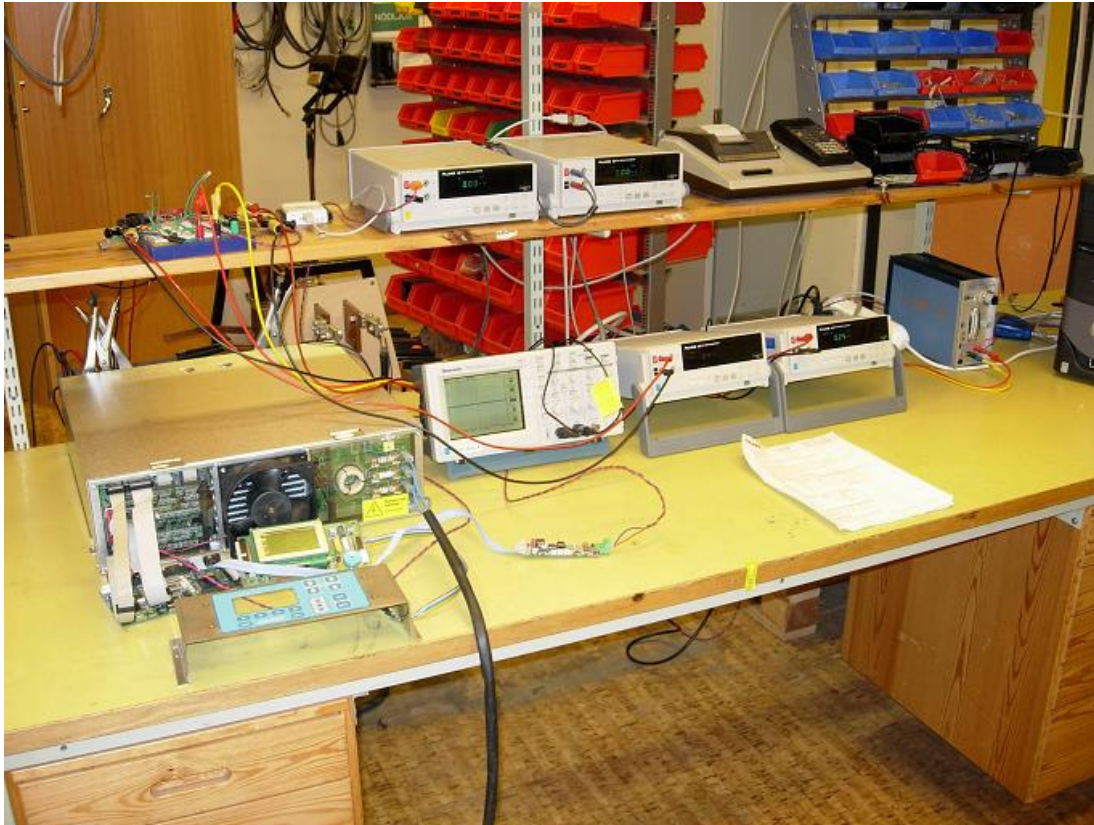


Figure 15. Test system in the lab

3.1.3 Phase three: In the factory

When all testing and verification was done and the entire test system was working completely it was time to set it up in the production line in the factory. After consulting the production manager and the workers that are to use the test system, a layout of the testing station was determined. A flat screen monitor was purchased and mounted on the side of the already existing cabinet. There were no technical difficulties regarding the system itself when mounting it at the test station but there were some difficulties mounting the oscilloscope inside the existing cabinet, this was mostly because of the limited space available inside. The end result however was to everyone's liking and is pictured in figure 16.

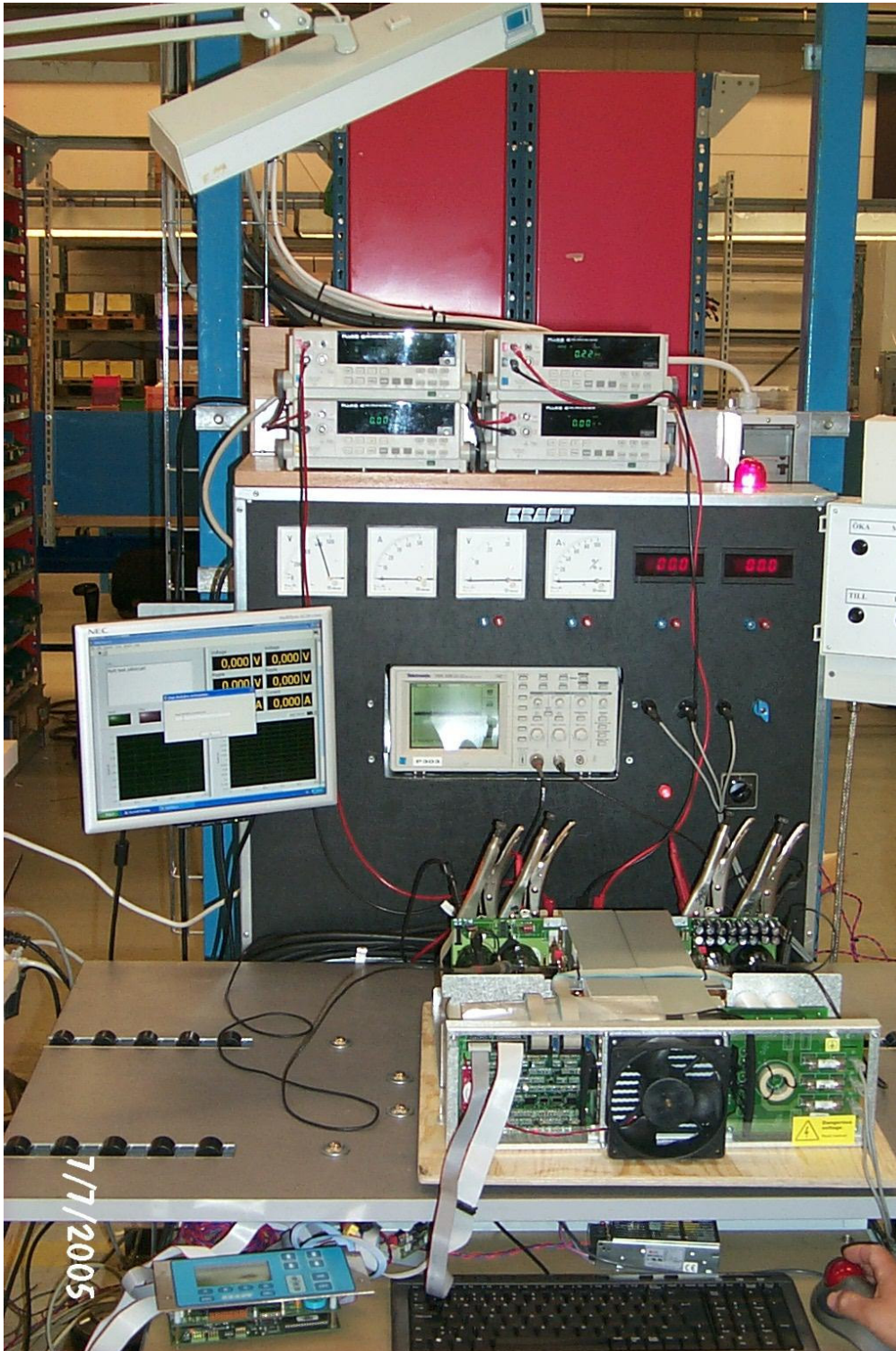


Figure 16. The test station

After some early testing of the finished test station it became clear that the custom made load that is pictured in figure 17 has a little too high resistance when set for current control testing. This causes the tested rectifier modules to go into voltage control instead of current control on the left channel when set at 12V 600A. During the previous manual testing this hadn't been noticed because the precision of the instruments used during the manual testing were lower.



Figure 17. Production load

3.1.4 Load Control

Since there is an compulsory step in the test that requires a load shift, a data signal have to be send from the computer to control the load. For this task a NI USB-6008 multifunction DAQ was purchased so that the computer could give a steady data signal that could be controlled from LabVIEW. The I/O module gives a 5V data signal of maximum 8mA DC and the contactor that controls the load is operated by 230VAC, therefore the chosen I/O module can't activate the contactor of the load directly and a specially designed "help circuit" was first considered necessary. Solving this problem took some time and a circuit was built containing an optic coupler, a driver circuit and a help relay. Some new facts appeared later on though and it became obvious that the "help circuit" could be replaced by a solid state relay as shown in figure 18. The solid state relay is installed in between the I/O module and the load contactor. The 5V data signal from the I/O module activates the solid state relay and the solid state relay activates the load contactor. The solid state relay is fed by 230VAC. The decision to use the solid state relay instead of the help circuit was based primarily on reliability and longevity. The fact that it is easier to replace one item if it fails rather than an entire circuit also contributed.

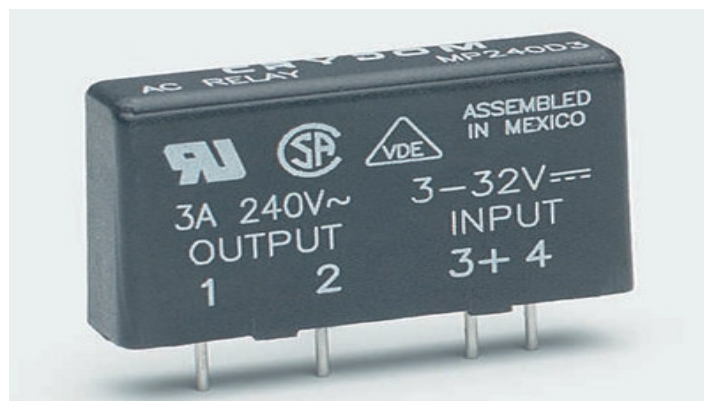


Figure 18. Solid state relay

3.2 The test Program

3.2.1 Program overview

The basic structure of the program is based on the previous instructions for the manual test. Some parts of the test were not suited for automated testing and are therefore still done manually examples of such steps are visual inspection and insulation testing. The structure of the program can be seen in figure 19 below.

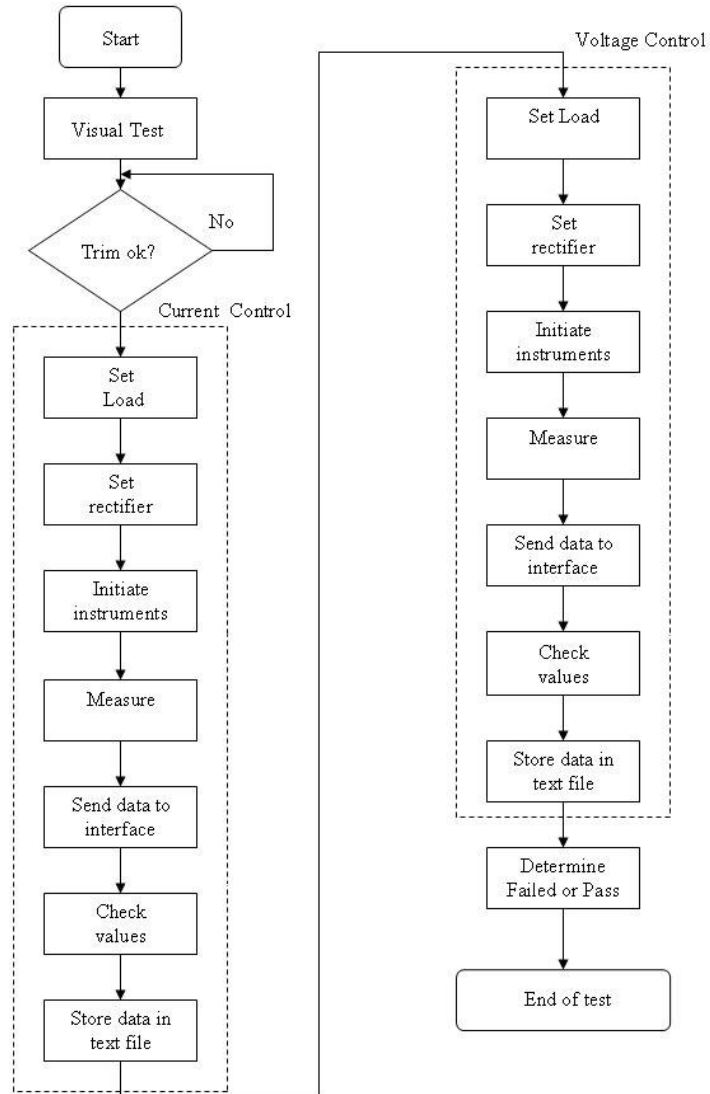


Figure 19. Program flowchart

The test program is designed so that as little manual work as possible must be done and in that way the operator doesn't have to watch over the testing all the time. However some manual input from the operator is always needed so to make the testing as efficient as possible all manual input is asked for in the early stages of the test cycle.

The test is divided into three main parts Visual test, Current control test and Voltage control test all of which are based on the corresponding steps in the old manual test.

3.2.2 Visual test

At the beginning of the automated test cycle the operator is asked to check some of the status LEDs and confirm whether they are lit or not. The operator is also asked to trim the current offset for both of the rectifier channels. After these initial steps the testing is fully automated.

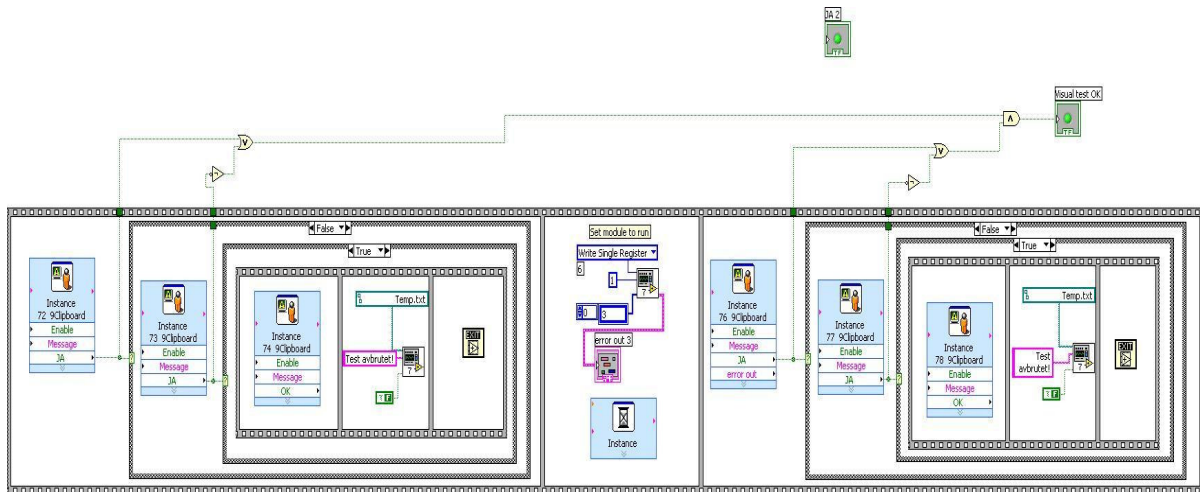


Figure 20. Visual test block diagram

3.2.3 Current control test

During this stage the module is tested at different operating levels via current control. Essentially this means that the rectifier is set to operate a load at a certain voltage- and current level. After measurements are taken and checked the rectifier is set to operate at the same voltage and a new current, this is done for five different current values.

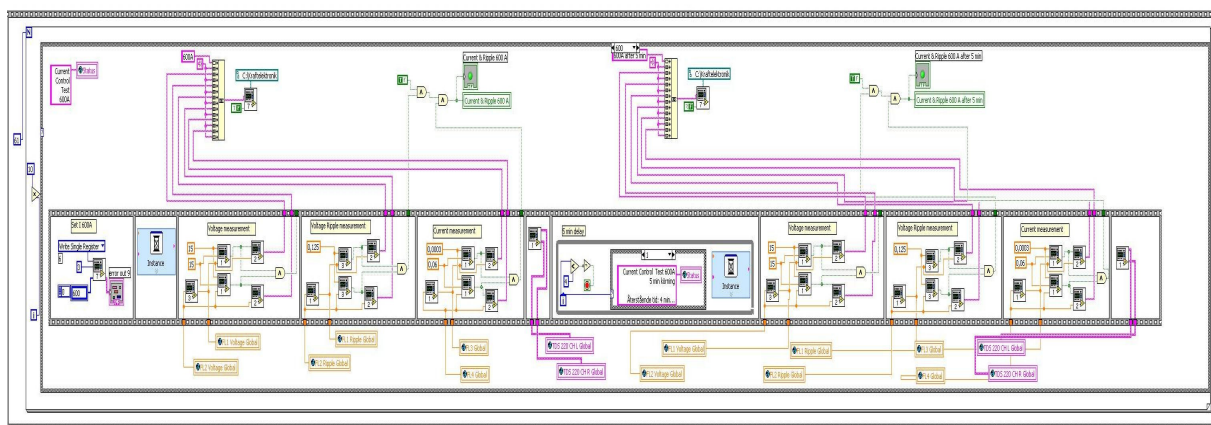


Figure 21. Current control test block diagram

3.2.4 Voltage control test

As in the current control test the module is tested at different operating levels but this time via voltage control. In other words, at four different voltage levels but only one current value.

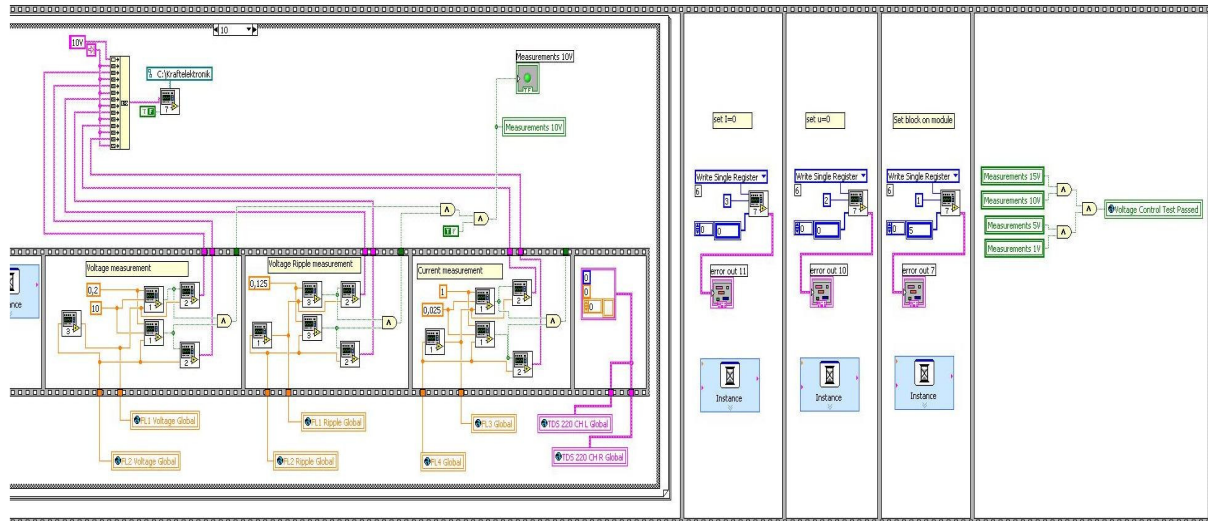


Figure 22. Voltage control test block diagram

3.2.5 Graphical interface

The graphical interface as seen by the operator on the computer screen, were designed so that it would be easy to quickly get a view over, and read the status of every measurement instrument. Design wise it has a layout that resembles that of regular instruments but with only the necessary information. The interface is composed of four panels on which different components are placed. In the upper right panel there are six similar displays that represent voltage, current and ripple. They are all designed to always show the same number of digits. Different colors and sizes were tried to get a satisfying result. For this particular application it is important that the numbers can be read from some distance and not just only from sitting right in front of the computer screen. In the upper left panel there is an information window that at all times keeps the operator informed about where in the test cycle the program currently is working. In this way the operator can feel at home with this new automated system and see the different measurement steps being performed in the same order as before. Since the load is changed during the test, an information window for the load is also present in the interface. As an indication that the test is started, a green lamp is placed on the same panel together with two indicators for failed and passed. On the two bottom panels the operator can view the left and right oscilloscope images during the whole test.

The development process of the interface included gathering opinions from the test personal. Their input was the main criteria that resulted in the interface that's being used today at Krafterlektronik AB. After all, they are the ones who shall use the program and therefore they must have a positive attitude towards this step of automated tests.

Implementation of the graphical interface required some thoughts as to how it was to be designed and also how it should be updated in the test program. The interface is fed with data from global variables which are assigned to the displays on the interface. Each display has its

own global variable and in the test program this variable is addressed continuously with the measured data. In order to keep the displays updated frequently, the measurement sub routines are placed within while loops together with delays in the main program. This is necessary to detect sudden changes of the measured values. The graphical interface front panel and block diagram can be seen in figure 23 and 24.

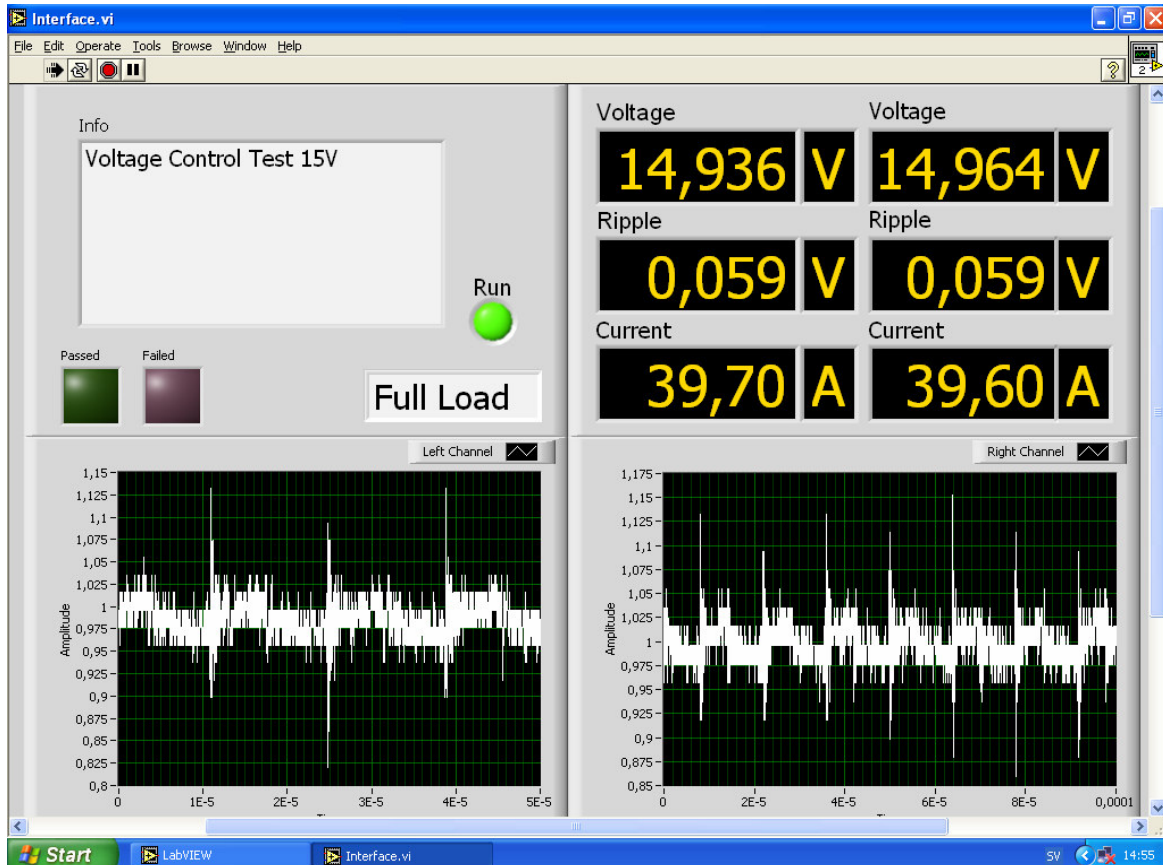


Figure 23. Graphical interface front panel

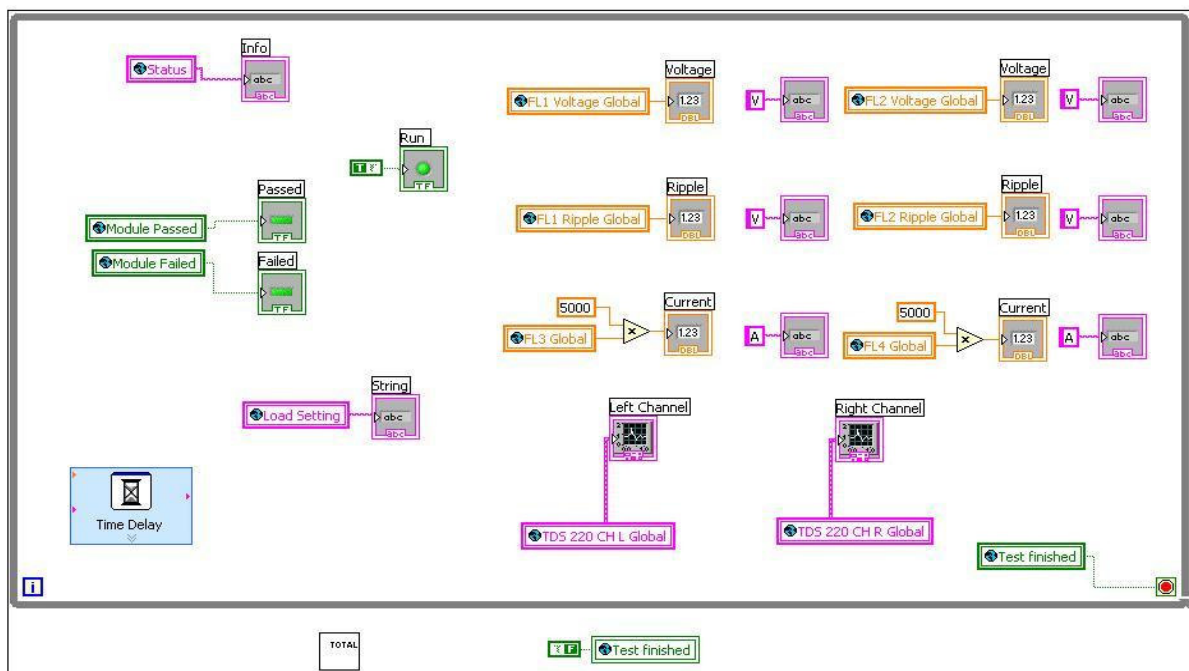


Figure 24. Interface block diagram

3.2.6 Report generation

From the beginning of the project there was a wish that every test and measurement should be saved in a data base for future reference and statistic calculations. During the test every measurement is saved in a text file that has one row for every test performed. At the beginning of every row the date and time that the particular test was performed is recorded as well as the serial number of the tested module. The format of the text file is not very easy to read or too get a quick overview of, so therefore the text file is automatically loaded into a specially made Excel file. In the Excel file all the data from the text file is shown in a format that resembles that of the previous manual test instructions. There is also the possibility to select any previous test in a scrollbar and show the measurement values from that particular test. Additionally it is easy to see whether a test was passed or failed and read comments written down by the operator at the end of the test cycle. Any measured value that is not within the predetermined boundaries and therefore not approved of is marked with three exclamation points after it so that it is easier to see what or which measurements that made a test fail. Figure 25 shows the generated test protocol in Excel.

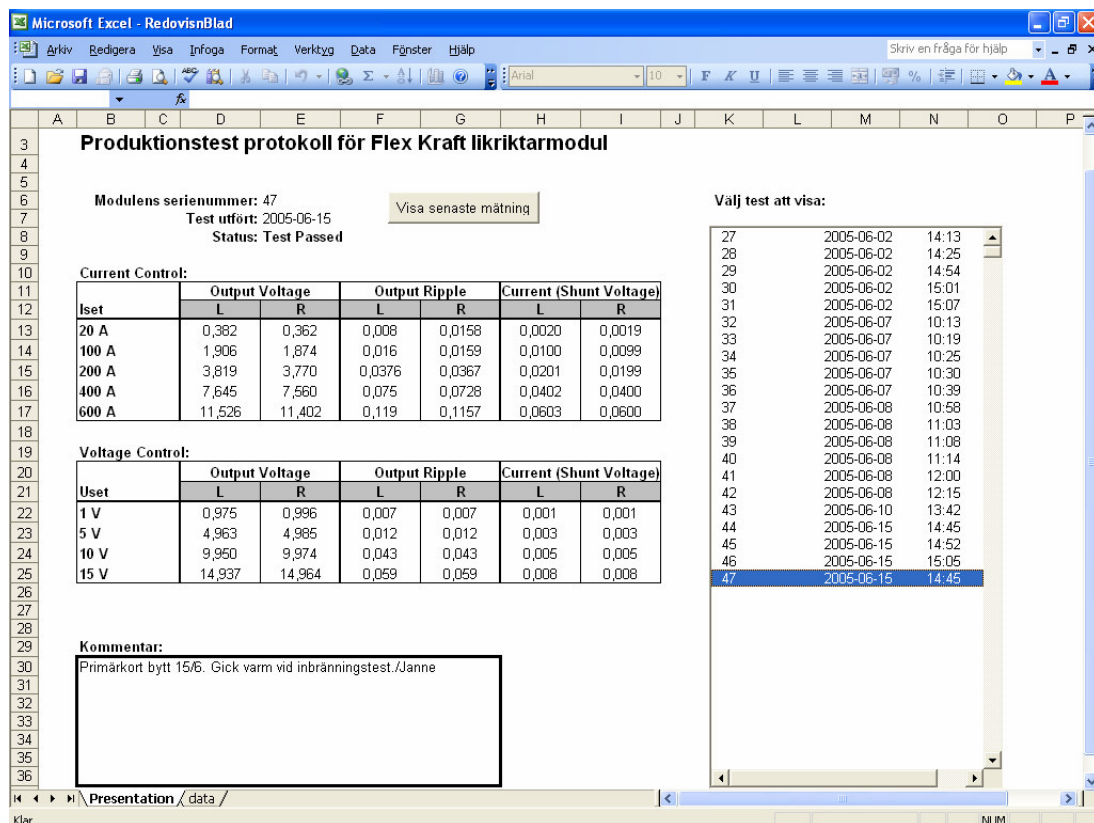


Figure 25. The test protocol

To get the data to be continuously stored away during the test, the data had to be saved from all of the performed measurements directly after the measurements took place. A “TAB” was also inserted after each value to make it easier to view. Microsoft Excel is a powerful presentation tool for reports of this kind. It has build in functions for importing data from text files and also for updating the tables and fields whenever it is desired. Layout wise Excel produces, with a fairly small amount of effort, a professional look to the presented results. From utilizing the build in help function, the authors could actually manage to create an

adequate Excel program for generating the wanted test report without having any previous skills in Excel.

3.2.7 VI and sub VI descriptions

The following part briefly describe the most commonly used VIs and sub VIs in this project. These are the main building blocks of the test program.

Check measurement Max value.vi

This VI examines whether or not the input values lies below the given maximum limit. The limit is given as an input parameter of type double. It gives a true/false signal as output.

Check measurement Max&Min Value.vi

By the use of Boolean functions it examines if the given input value lies within the given limit parameters. The input parameters are given as a reference value and a maximum allowed deviation. The result is given as a true/false output signal.

Current control test.vi

This is one of the bigger VIs in the program and it contains many of the other sub VIs listed here. It contains the “current control” part of the program and uses global variables to communicate with the Interface. The main task that’s performed here is to carry out the required measurements for when the rectifier is operated in current control mode.

Current measurement FL3&FL4.vi

Takes current measurements from Fluke 45 number three and Fluke 45 number four and output the numerical values as the type double. This VI is used many times in the test program.

Interface.vi

This VI contains the user interface in the front panel and that is what the operator sees when running the test program. It is the only VI that has to be started, all other VIs are sub VIs in the Interface VI. It updates the visual interface via global variables.

Load Control.vi

It controls the load setting via the NI USB-6008 multifunction DAQ and the solid state relay. The load control VI is only used when switching the test load from the setting used for current control testing to the setting used for voltage control testing.

Modbus controller.vi

To send commands to the Flex Kraft module via Modbus protocol the Modbus controller VI is used in almost every part of the test program. It takes “function code”, “starting address” and “data/value” as input parameters.

Numeric to string & error ind.vi

Converts the measured numerical data to string and also has a true/false input. If false, three exclamation points are added to the string after the value. These exclamation points are used for easier identification of faulty values in the generated report. The true/false input is taken from the output of Check Measurement VI.

Ripple measurement FL1&FL2.vi

Performs measurements on the Fluke 45s with GPIB address one and two. The returned numerical values are representing the AC voltage measurements from each multimeter. This VI is also used in several locations in the test program.

Totalprogram.vi

This is a Sub VI to the Interface.vi and contains the main program code.

Visual test.vi

This VI contains the “Visual Test” part of the program where the operator is asked to check some LEDs and trim the current offset. It returns a Boolean value that is used to determine if the tested module is ok or not.

Voltage control test.vi

This VI contains the “Voltage Control” part of the program and it communicates with the interface and Totalprogram VI with global variables. Just as the Current control test VI, this is one of the bigger VIs.

Voltage measurement FL1&FL2.vi

Performs measurements on the Fluke 45s with GPIB address one and two. The returned numerical values are representing the DC voltage measurements from each multimeter. This VI is also used in several locations in the test program.

Voltage measurement TDS220.vi

Perform a measurement using a Tektronix TDS220 scope on GPIB address 5. It returns the graph image from each of the two oscilloscope channels. The graph image is returned as a cluster consisting of two numerical values for graph scaling and one array of numerical values containing the graphical information. The VI also returns a numerical peak- peak value for each channel.

Write to temp.vi

Writes either a string or a numeric value to the text file “temp.txt”. It has one input for the file path to the file, one input for the string that is to be written, one input for a numerical value and one Boolean input. The Boolean input selects if a string or a numeric value should be written to the selected file. If the selected file doesn’t exist it is created. The VI can not create folders just text files. If the file exists the written data is appended at the end of the file.

3.2.8 Debugging

As always, debugging is an important aspect of developing a new system. In LabVIEW debugging is a natural extension of testing and running a simulation. Making a small modification and testing again is done in a few seconds. This leads to extremely short development/testing cycles, something generally considered important for experimental development, and helps end-users.

3.3 Practical implementation

3.3.1 Programming

To just start and write a program that performs all the required tasks included in an assignment is often hard to do and it is even harder if the programmer have little or no experience in the actual area. This test program was written in different stages but the first stage was to identify the different parts of the test. By doing so it was possible to decide and understand which sub routines that were needed and how many measurement steps that was included in a test. LabVIEW has a function that resembles a film frame in which one can place other pieces of code and in this way control the order of execution. In this particular case it was of great importance that each step was carried out at the right time. Otherwise the values in the report would be mixed up and give a misleading result. An example of LabVIEW code containing film frames is given in figure 26.

Another thing that's important is that every step in the test is allowed to be finished before the following step starts to execute. If so is not the case, a timeout error will occur. By placing a number of film frames that matched the number of measurements and at first just write a short description text inside them, it was possible to have a first overview of the total program. A list of all the needed "happenings" that could be made as subroutines was put together. This list was the main task for several weeks. All the different pieces of the total program had to work before it was possible to create a working test program.

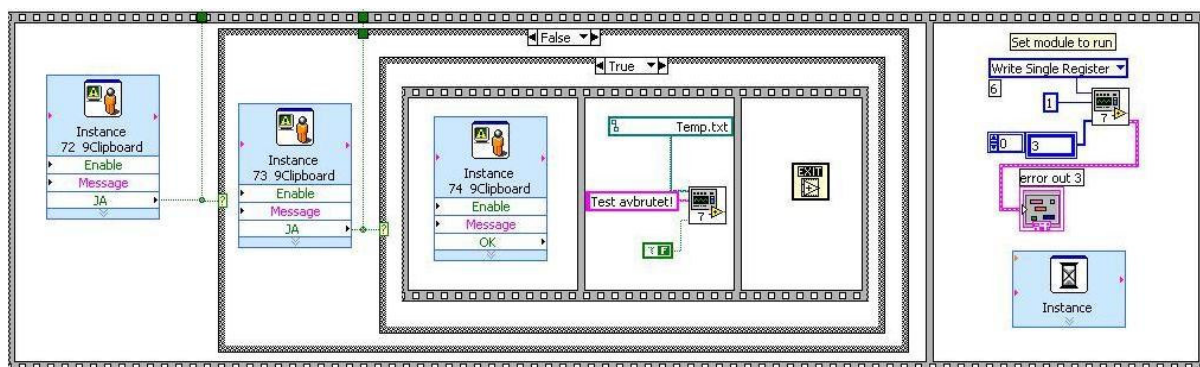


Figure 26. Film frame example

Many of the measurements in the test are repetitive and therefore could for example a sub routine like "Voltage Measurements TDS220", be used many times with out having to alter the code within it. While-loops and For-loops were used in combination with certain conditions to take measurements and send commands to the rectifier. By saying that the test is repetitive it is meant that it does the same thing over and over again with little or no changes. An example of this is when the output voltage of the rectifier is set at 1V, 2V, 5V, 10V and

the measurements are exactly the same in all of these cases. Tests that are repetitive are specially suitable for automation. The explanation is simple, for each measurement a certain amount of time is saved. Multiply the number of repetitions by the time saved and you have the total sum of saved time. Of course if the number of repetitions increases, so does the time saved. The most time saving part of the test was the repetitive ones.

Some parts of the test could be done very fast while other parts had to go slower. Regarding the most basic measurements such as voltage measurement with Fluke 45, these measurements could be performed at a relatively high speed. The amount of information that's transferred from the Fluke 45 to LabVIEW is significantly less than the amount transferred by the TDS 220 digital oscilloscope. This means that it takes longer time to carry out a measurement with the oscilloscope than with the Fluke 45 multimeter. To insure that certain instructions have enough time to execute properly delays were introduced in the program. By placing a delay component in a film frame and configure the delay time properly the program is forced to wait until the delay time has expired before it moves on to the next frame. The time setting can be tuned to make the program work as effective as possible. At first, all delay settings were too large but by testing the program and change the settings after each test, they all finally became well tuned. One parameter that has to be considered is that sometimes the instruments can have problems triggering and therefore needs extra time to measure the correct value.

To keep track of the modules that didn't pass the test and to provide some information on why they failed an operator input procedure was added to the test program. When a test is at its final stage, the operator is asked to write a short description of the specific module. A dialog box appears on the screen and the operator uses the keyboard to input his information. An example of this could be that the operator writes that the module seemed to have problems with a certain component. This information is very helpful when repairing a faulty module. The test personal or operators have special skills in the area and it is obvious that their information could speed up the repair process a lot. All given information is stored in the text file of every module and is represented in the test report. Another benefit is that if a module is repaired several times, the test personal inputs this fact and after a pre determined number of repairs the module is taken out of the production and goes to waste. Without this procedure a module could be repaired many times and still be faulty. This costs a lot of money and leads to decreased efficacy in production. Having an efficient production is one of the main variables for a successful company.

3.3.2 Communicating with Flex Kraft

The communication between the computer and the Flex Kraft modules that are to be tested is done via a Flex Kraft control unit. The control unit that is used has an extra communication circuit attached and this communication circuit supports Modbus RTU data communication via RS-485. The Flex Kraft communication unit also has an option of a different communication circuit that supports Profibus data communication instead. The Modbus communication language was chosen over Profibus because it seemed more suitable to implement in LabVIEW. A regular computer does not have any RS-485 ports so therefore the Westermo MA 42 was used to convert the RS-232 signals from the communication port on the computer to RS-485 that the communication circuit on the Flex Kraft control unit uses.

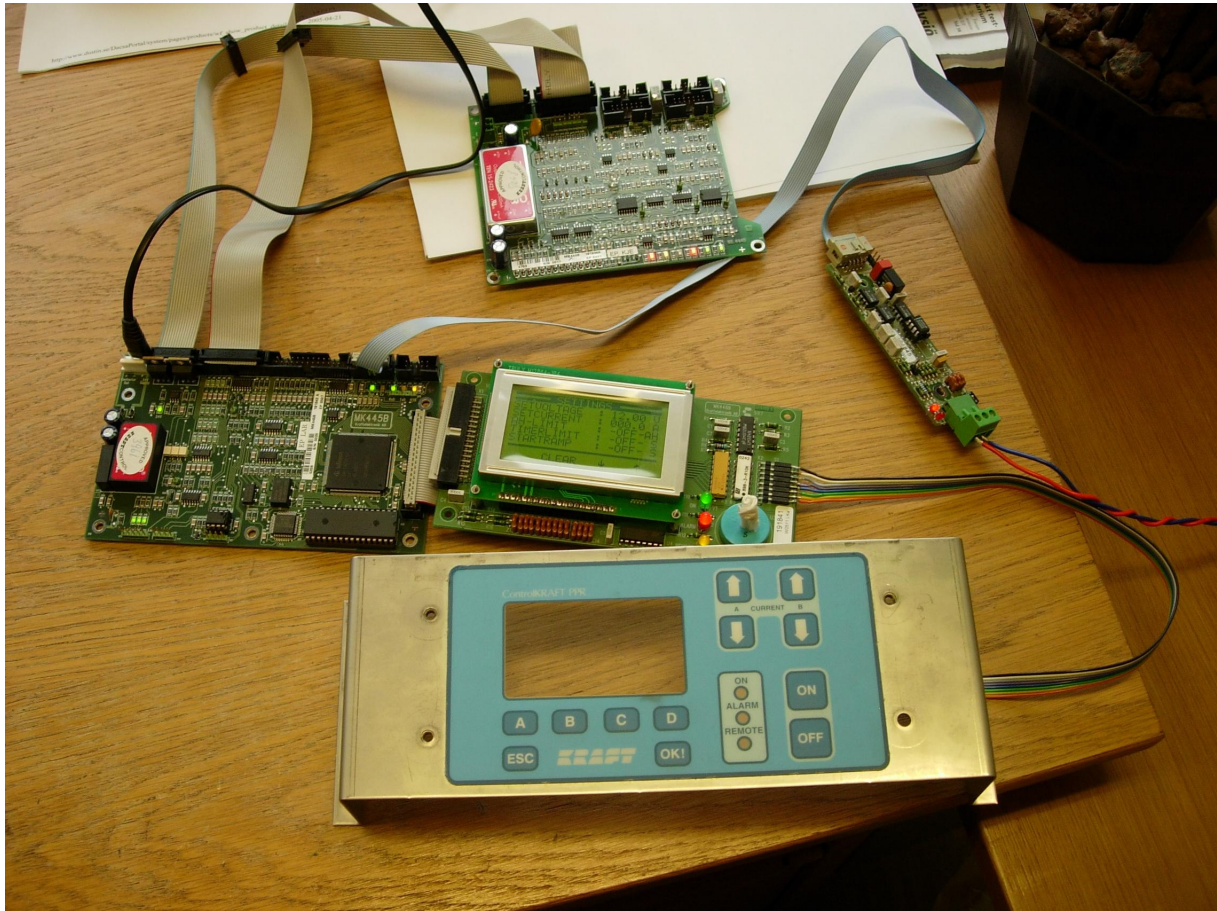


Figure 27. Flex Kraft Modbus communication components

Setting up the Westermo MA 42 and getting the RS-485 data communication to function between the computer and the Flex Kraft communication circuit was not very difficult and proved to be by far the easiest part of getting the Flex Kraft communication working. The problems started however when it was time to get the Modbus communication to function in LabVIEW. There are readily made NI Modbus VI libraries available for LabVIEW that can be downloaded from National instruments homepage but the supplied VIs didn't function properly at first. It took quite some time to get the NI Modbus VIs to work properly for this application and be compatible with the registers in the Flex Kraft control unit. To achieve this, a special VI called Modbus controller VI that can be seen in figure 28 had to be created. The Modbus controller was built with some of the NI Modbus VIs but was custom made for the application. When the communication started working it was not working entirely frictionless because of timeout errors that kept coming up in LabVIEW. The timeout errors were unexpected and took some time to figure out how to solve, but after testing it was solved by adding a time delay within one of the NI Modbus VIs.

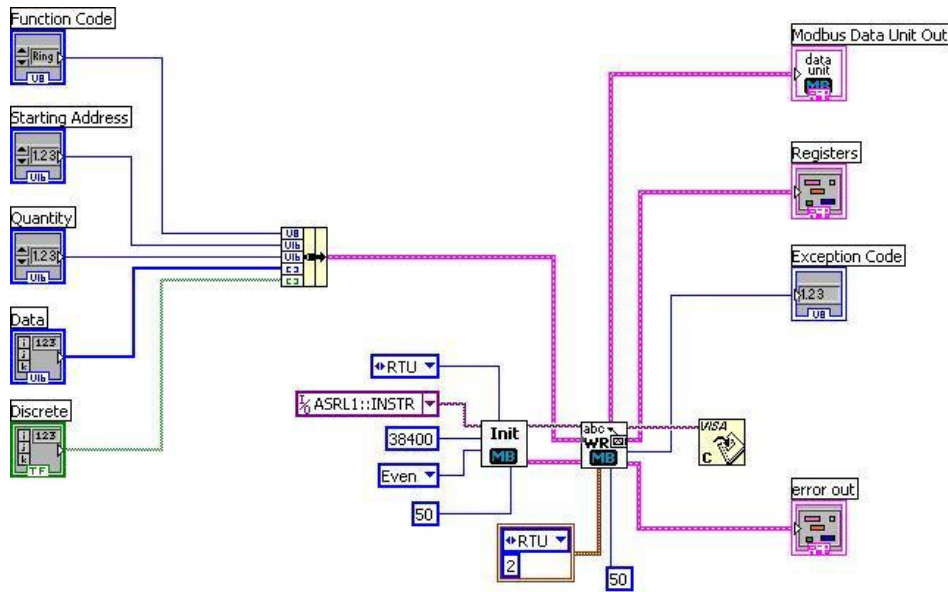


Figure 28. Modbus controller VI

After this the communication between LabVIEW and the Flex Kraft control unit worked flawlessly during all the initial test program development in the office but when it was time to start testing in the lab with an actual Flex Kraft rectifier module plugged in to the control unit the timeout errors once again surfaced. This time however it was easier to solve and the reason that caused them to appear were that the response time from the control unit was marginally longer when a rectifier module was plugged in. The solution to the problem was simply to add a small time delay in LabVIEW between each command to the Flex Kraft unit.

3.3.3 Communicating via GPIB and RS-232

The instrument to computer communication was mostly done with GPIB but as have been mentioned before, two of the Fluke 45 multimeters were missing a GPIB communication module and therefore had to use RS-232 communication to interact with the computer.

When setting up the GPIB communication there were, as earlier discussed, some problems with getting the Agilent 82357A USB / GPIB interface to function properly together with LabVIEW. These problems started because Agilent and NI are using different VISA versions. In the beginning if the NI VISA was installed as primary VISA it was not possible to see and use the Agilent 82357A from LabVIEW but if the Agilent VISA was installed as primary VISA it was possible use the 82357A from LabVIEW but it was not possible to use any other I/O or communication devices at the same time. The final solution was to use the NI VISA as primary VISA and to activate a setting in the NI MAX called tulip passport. After this was done the Agilent 82357A worked problem free.

In LabVIEW the GPIB communication was done using the input output assistant which is a express VI that was introduced in LabVIEW 7.0. The input output assistant was very easy to use and the only problem with the GPIB communication apart from the initial problems with the Agilent 82357A was to find the right commands for all instruments.

When it became clear that two Fluke 45 multimeters had to use RS-232 communication instead of GPIB this was not a major problem, even though the program code in LabVIEW

had already been made to use GPIB for all Fluke 45s. The reason for this is the fact that the input-output assistant had been used to write the code. The input-output assistant works as a simplified programming interface but the commands to the instruments are regular VISA commands and can therefore be used with RS-232 as well as GPIB.

The only thing that had to be done when changing to RS-232 was to change the addresses for the Fluke 45s in the corresponding input-output assistants. In theory this would have been true but the Fluke 45s were designed to send a reply after every instruction sent with RS-232. Therefore an extra RS-232 read step had to be inserted after every command sent to the Fluke 45s via RS-232. This read step was necessary in order to clear the buffer memory on the RS-232 bus otherwise no more RS-232 commands could be sent. It must also be said that in order to be able to connect the two Fluke 45s using RS-232 to the computer, a two port RS-232 / USB hub had to be used. The reason for this is that only one device can be connected to a RS-232 port at a time and the Westermo MA 42 was already connected to the standard RS-232 port on the computer.

3.3.4 Implementing the data storage

As creating a test report was a part of the project description, the measured data had to be not only controlled but also stored in a data base or text file. At first, the general idea was to generate the report during the test by storing the measured data in certain positions in a pre filled form. A first version showed however that this was not a professional way of handling the data since there clearly were some limits and boundaries to how the data could be manipulated. For instance, if the report is generated in a plain text file (.txt), then there is no possibility to have bold letters or larger letters for headings. The result is a report that's hard to read and also bad looking. Because of these facts, the idea of creating a report directly in a text file was abandoned.

To actually produce a professionally looking report demanded that the report generating functions that are build in to the more expensive versions of LabVIEW, had to be implemented manually in this project. One way of doing this was to store all the measured data in a text file and then export it into Microsoft Excel. Of course a program that creates the desired report layout had to be written in Excel. This procedure is further explained in part "3.2.6 Report generation". The actual saving of data in the test program is carried by a sub routine that comes in the LabVIEW Base package. There are a number of configurations that have to be made in order to get the wanted result and while this seems to be trivial, it takes some time to set them up correctly. Parameters such as pathname, file name, file type and so on have to be given as arguments to the sub routine. When the final configuration is made, the sub routine was placed in all needed locations in the program.

The result of having a well documented automated test is of course that you save a lot of time since the test personnel doesn't need to manually fill out a report form. There is also an elimination of errors made by the personnel while filling out the form. In this stage alone a time saving of about 10 percent is made. Before, it was not possible to make any calculations on the data without manually transferring it into a computer for further handling. Because time is money, this was rarely done. With this new test system it is a trivial task to perform calculations on the stored data for a person with some skills in a program similar to Microsoft Excel. It is also easy to share the stored data with other divisions within the company by sending the text file as email.

3.4 New system vs. old system

Before this project the production testing of the Flex Kraft rectifier modules were carried out manually, meaning that all steps of the testing had to be conducted by the test personnel. A typical test followed the same basic steps as the automated testing. Firstly the rectifier module had to be rolled into place at the test station and connected to the load, the power supply and the control unit. After everything was connected the operator had to turn everything on before the testing could begin. In addition to the earlier mentioned, all measuring instruments also had to be connected and turned on. When everything was ready to begin the test started with the same procedure as the automated testing. First the visual test, but with the difference that the operator had to manually alter all settings for the rectifier on a Flex Kraft control panel.

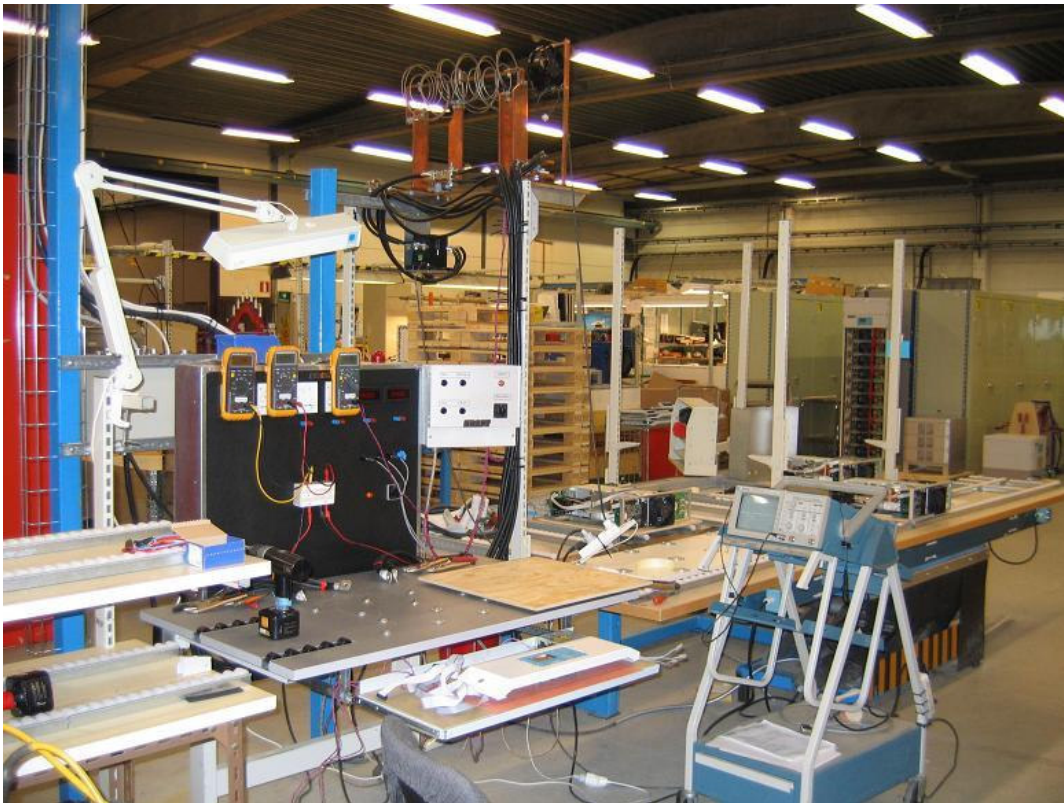


Figure 29. Manual test site

The major drawback with manually changing all the settings via the control panel is that voltage and current values can't be set instantaneously. In other words if the operator wants to change the current setting from ten to 100A then it would be necessary to scroll thru all current values in between. After the visual test part, the testing progressed thru to the current control testing. Once again the operator had to manually alter all settings for the rectifier. However in the current control part of the test the operator also had to take readings of all instruments and record the values in a test protocol. Because the current control test is done at five different current values, the operator has to set the rectifier, read the instruments, and then record the values five times for each current control test. There are two channels on the rectifier and both current and voltage values were to be read and recorded at each current test value. Because this had to be done on both channels it resulted in 20 instrument readings and value recordings just for the current control part of the testing.

When the current control test was ready the operator had to manually change the test load so that the rectifier could be tested in voltage control mode. The voltage control testing was done in the same manual manner as the current control test but at four different voltage values. As a

result, the operator had to change the rectifier parameters four times and take 16 instrument readings and record them. All these steps made the manual testing somewhat monotonous and it was easy for the operator to misread or incorrectly record a value.

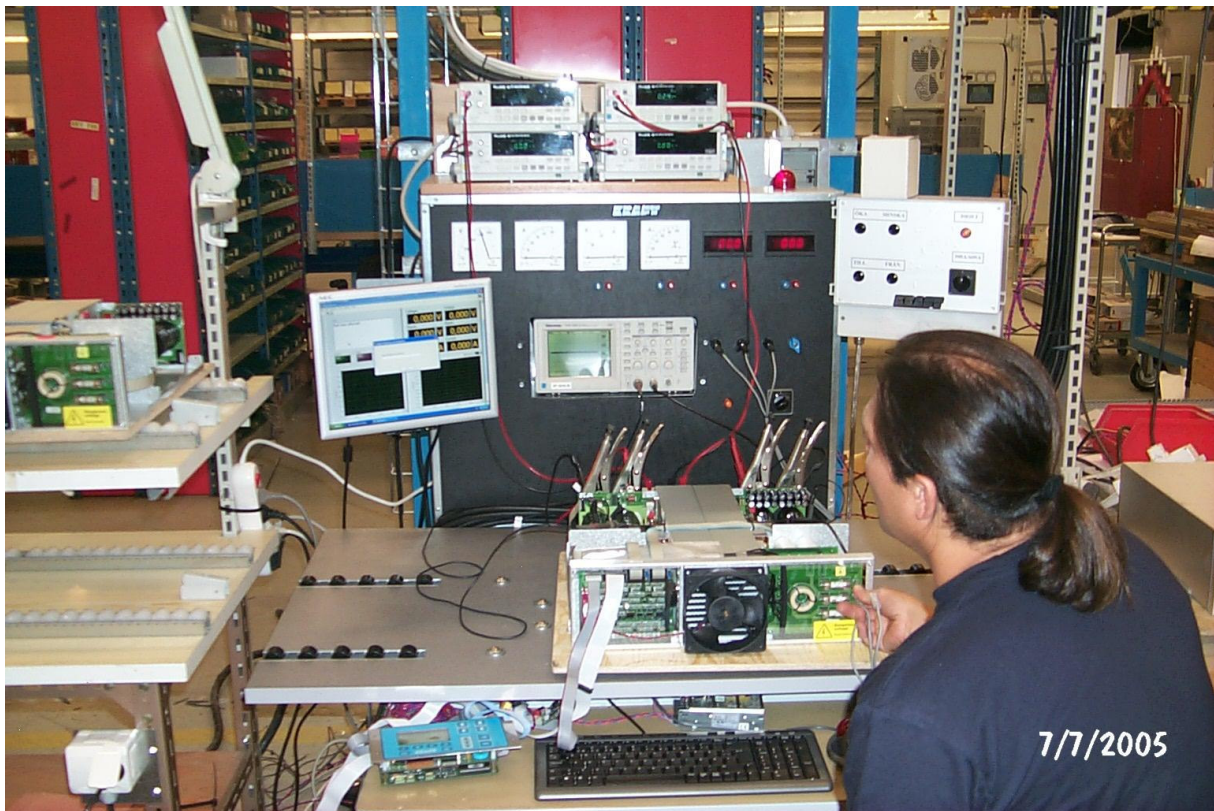


Figure 30. Automated test site

Even though the testing is now automated, some of the above mentioned steps still has to be done manually. For example the rectifier module still has to be rolled into place at the test station and connected to the load, the power supply, the instruments and the control unit. In the visual test part the operator still has to determine whether or not the different status LEDs are lit at the right time but all settings for the rectifier are given as commands by the controlling computer. From the current control part of the test it is fully automated and all settings of the rectifier and instrument readings are done automatically. This means that the operator no longer has to monitor every step of the test. One of the reasons that the test system doesn't need continuous monitoring from the start of the current control test and forwards is that the test program automatically determines whether a tested rectifier module has passed the test or not. The automatic failed/passed decision is possible because every measured current, voltage or ripple value that is acquired during the test is checked so that it lies within a predetermined interval that is specific for just that measured value.

At the end of the automated test process all measured values have to be approved of for the computer to pass the rectifier module. The test load is now also controlled by the computer and the only thing the operator has to do at the end of the test is to write any possible comments that are to be saved in the test report file with all the measured values.

Earlier the operator had to be involved in every step of the test procedure and therefore didn't have any extra time to do other tasks except during the five minute warm up stage of the test. Now it is possible for the operator to make some other tasks during the test time such as insulation testing of the next module in line for testing and putting the lid on the previous

module that were tested. Both of these tasks earlier had to be done in the time between each tested module.

3.5 Future development

A small part of the production test is still carried out manually by the operator. The main reason for this is that the project had a time limit of 20 weeks but also because there is a part that requires manual trimming of potentiometers. If there is to be a development of the present system, automating the first part of the test called “Visual test” is recommended from the authors of this master thesis. By doing so, it would be possible to reduce the test time even further. One way to implement this is to tap the internal signals and feed these signals to the DAQ unit. The program could then detect whether or not the LEDs are on or off. Another improvement towards fully automated testing is to exchange the current offset potentiometers in the module to electronically adjustable potentiometers as opposed to the mechanically adjustable ones currently used.

It is also recommended that statistical calculations are implemented in the Excel program for further analysis of the measured data. To even further automate the test procedure, an automatic paper print out of the generated report would be practical.

4 Summary

An automated test system is installed in the production line and is working according to Krafterlektronik AB's specification. From working with this project we learned that LabVIEW is a fairly easy programming language as long as you want to communicate via the most common standards and protocols. However, when we wrote the program for communicating via RS 488 and Modbus protocol, we encountered problems not easily solved.

Of course this project has been very educating and interesting but also some times challenging and frustrating. As an example we would like to mention the problems that occurred with the GPIB-USB interface from Agilent that was said to be totally compatible with LabVIEW. Now afterwards, we realise that so was not the case. We would have saved a lot of time not using the Agilent interface but a one from National Instrument instead.

We gained knowledge about communication and automated measurement systems in general while working in this project. It feels like very valuable skills for using within the coming years of working as an engineer.

5 References

Internet

Most links were regularly visited between February and June, 2005.

www.ni.com

www.fluke.com

www.kraftelektronik.se

www.agilent.com

<http://www.iit.edu/~labview/Dummies.html>

<http://forums.lavausergroup.org/index.php?s=c75e145e4f1c242ac79cc60a926c9a30&showforum=49>

<https://www.westermo.com/Resource.phx/content/se/support/faq-rs-422.htx>

www.tektronix.com

www.elfa.se

<http://www.tequipment.net/TektronixTDS220.html> 2005-05-13

Litterature

[1] Bengtsson, Lars
“LabVIEW från början”
7th edition, 2004, Studentlitteratur, Lund.
ISBN 91-44-03798-8

[2] Johnson, Gary W
“LabVIEW Graphical Programming”
2nd edition, 1997
ISBN 910-07-032915-X

Appendix

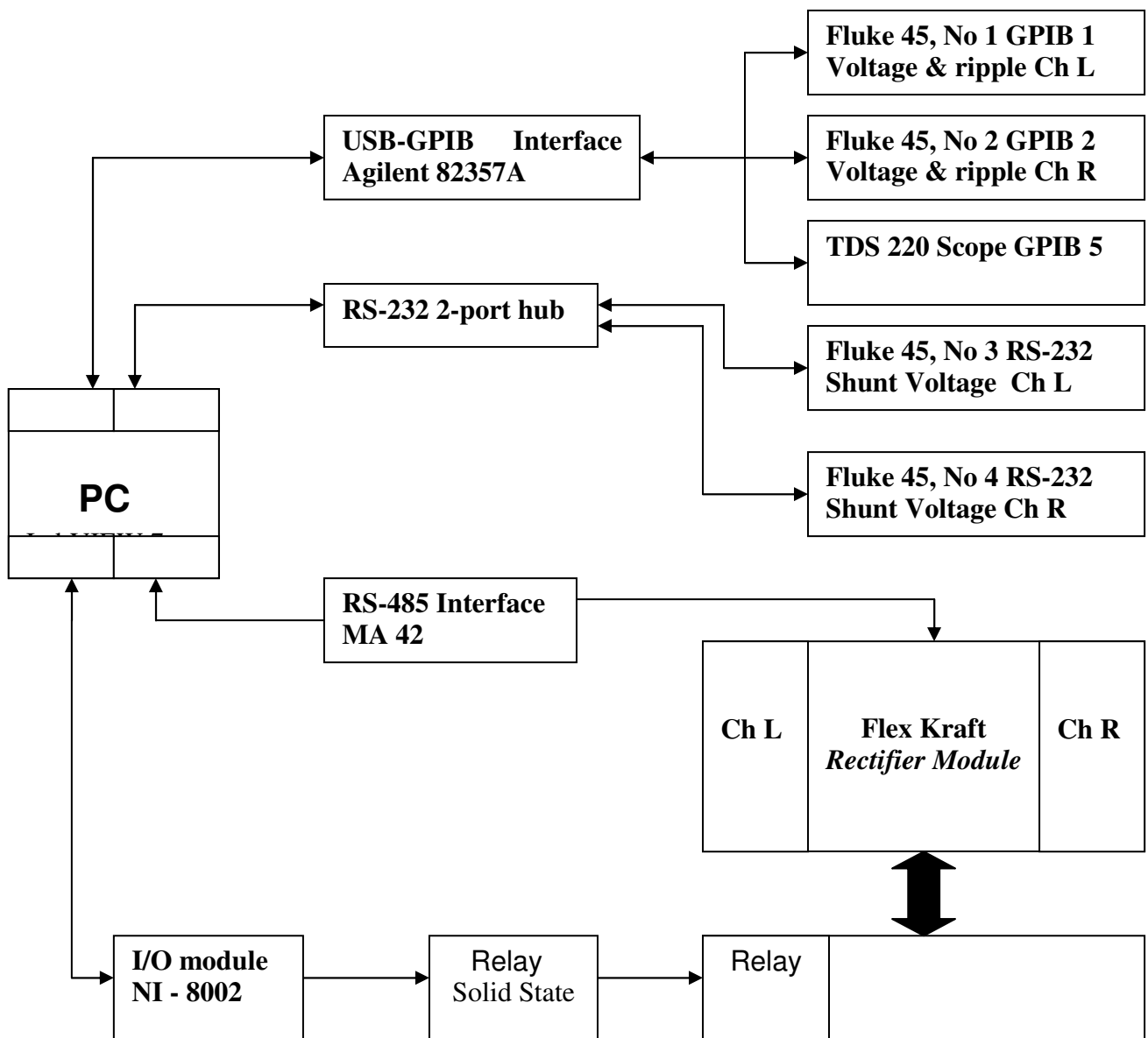
Documentation for Kraftelektronik AB

Automated production test of Flex Kraft rectifier module.

The system consists of of a LabVIEW program on a PC that control four Fluke 45 multimeters, an Oscilloscope TDS220, an I/O module NI-6008 and a Flex Kraft rectifier module. The LabVIEW program is created by following Kraftelektronik's manual test intruction for Flex Kraft and the measurements are performed in the same order. All measurement data are stored in a text file, Temp.txt. This file is read into an Excel program that presents the data in a test report.

To start the program, double click the FlexKraftProvning-icon on the desktop of the PC at the test site. To view the stored data, doubleclick the Excel-icon on the desktop.

Översiktsschema för "Automatiserat mätsystem - Flex Kraft produktionstest"



PC-USB connections
Layout seen from the rear
of the computer:

GPIB Interface USB 1	I/O mod NI-6008 USB 2
Rs-232 Hub USB 3	Mouse USB 4