

New feedback control for a scanning tunneling microscope

by

Adriaan H. Bredekamp

Thesis presented for the degree of
MASTER IN TECHNOLOGY
in the school of Electrical Engineering
Cape Technikon
January 1999

Declaration

I, Adriaan Hendrik Bredekamp declare that this thesis, for my Master's degree in Technology at the Cape Technikon, is my own work and that I have given credit to the authors of documents where I have referenced them. I also declare that this thesis has not been presented previously to obtain any other qualification.

Signed
Adriaan H. Bredekamp

Date : 31/1/1999

Acknowledgements

I would like to thank my supervisor, Dr. J. Tapson (Department of Electrical Engineering, UCT) for his advice and guidance during his research project. Thanks are also due to Dr. B. Mortimer (School of Electrical Engineering, Cape Technikon) for all the administration work that went into this thesis, also Mr. S. Schrire (Electrical Engineering, UCT) and to my family that gave me the emotional support that I needed. Thanks also goes to everybody that provided valuable technical support.

This project was financially supported by the Foundation for Research and Development (FRD) and research funds from the Cape Technikon Centre for Instrumentation Research.

Synopsis

This thesis describes the design and implementation of a new feedback controller for a scanning tunneling microscope or STM. The previous controller had several shortcomings when it came to the data throughput rate of the data acquisition system, the scan rate, and the way the data was stored and displayed.

The initial investigation was done to determine the most cost effective way to implement the data acquisition system. Various approaches such as DSP systems, analogue systems and microcontroller systems were looked at. The investigation also looked at the best way to get the data from the Z directional control loop to the PC for displaying the data. The final choice was to use an ultra fast microcontroller for the control loop implementation and to change the DOS based software for Windows based software.

The embedded system was divided into two parts. The first was the controller for the X and Y scan directions, and the second was for the Z scan direction. A digital PI control loop was implemented on the Z controller to control the height of the scan tip above the specimen surface. The microcontroller that was chosen for this was the Microchip PIC17c43. The data transfer to the PC was done with a PC-14 programmable digital input/output card. Two options for the implementation of the PC-14 software were considered. The first option was the software that was bundled with the card. This software proved to be very slow, so special device-driver-based software was developed to control the PC-14 card and the data transfer to and from the PC. The PC software was implemented using Visual C++.

Both the XY and the Z controllers proved to be working satisfactorily in the existing STM arrangement. It was discovered that the XY controller was overloaded with the many tasks that it has to perform, and a suitable alternative system to replace the XY controller is proposed. The selection of the PC that will be used for the data acquisition system is also discussed. It was found that this choice had a very big influence on the design of the final system because of the difference in PC bus design. Several proposals to increase the functionality of the PC software are also made.

Table of Contents

DECLARATION	I
ACKNOWLEDGEMENTS	II
SYNOPSIS.....	III
TABLE OF CONTENTS	IV
LIST OF ABBREVIATIONS	VII
LIST OF FIGURES.....	IX
CHAPTER 1: INTRODUCTION.....	1
CHAPTER 2: REVIEW OF STM TECHNOLOGY	4
2.2 <i>The tunneling tip</i>	5
2.2.1 <i>Opto-electronic tip design</i>	5
2.2.2 <i>Tip sharpness</i>	6
2.3 <i>The control system</i>	6
2.4 <i>Imaging systems</i>	8
CHAPTER 3: PREVIOUS WORK.....	9
3.1 <i>Introduction</i>	9
3.2 <i>Historical methods and current practice in STM design</i>	9
3.3 <i>Cantilever STMs</i>	12
3.4 <i>Dynamic behaviour of STM structures</i>	12
3.5 <i>A dual actuator STM design</i>	13
3.6 <i>The feedback of STM devices</i>	13
3.7 <i>The control System: implementation and performance</i>	14
CHAPTER 4: SYSTEM IMPLEMENTATION	16
4.1 <i>Introduction</i>	16
4.2 <i>System overview</i>	17
4.2.1 <i>Scan head</i>	17
4.2.2 <i>Vibrational isolation</i>	18
4.2.3 <i>Noise considerations</i>	19
4.2.4 <i>Electronic systems</i>	20
4.3 <i>DSP implementation</i>	20
4.4 <i>Multi processor system</i>	22
4.5 <i>Conclusion</i>	23
CHAPTER 5: CONTROL ACTION.....	24
5.1 <i>The tunneling junction</i>	24
5.2 <i>The control loop formula</i>	25
5.3 <i>Scanning bandwidth</i>	26
5.4 <i>Constant determination</i>	27
5.5 <i>Results</i>	28
5.6 <i>Conclusion</i>	29
CHAPTER 6: HARDWARE DESCRIPTION.....	30
6.1 <i>PIC Processor hardware</i>	30
6.2 <i>Communications Bus</i>	31
6.3 <i>Data Acquisition</i>	33
6.3.1 <i>High gain amplifier and ADC input</i>	35

6.3.2	Z controller 20-bit DAC.....	36
6.3.3	PCB layout for low noise in a data acquisition environment.....	36
6.4	Eagle Technology PC-14 card.....	38
6.5	Results.....	38
6.5.1	Timing problem.....	39
6.5.2	Opto-Couplers.....	39
6.5.3	PC-14 card.....	40
6.5.4	Lateral response.....	42
6.6	Recommendations.....	43
CHAPTER 7: SOFTWARE DESCRIPTION		46
7.1	Introduction	46
7.2	Z-control algorithm	46
7.3	XY control algorithm	47
7.4	PC software	48
7.4.1	Matlab routines.....	48
7.4.2	Graphic routines.....	48
7.4.3	Scan parameters.....	49
7.4.4	EDR software.....	49
7.4.5	Control toolkit software	50
7.4.6	User friendliness.....	51
7.4.7	Special device drivers	52
7.5	Software verification procedure	52
7.6	Results.....	53
7.6.1	The Z controller	53
7.6.2	The XY controller.....	54
7.6.3	The PC software.....	54
7.8	Recommendations.....	55
7.8.1	Embedded system.....	55
7.8.2	PC software updates	55
CHAPTER 8: FAULT DIAGNOSTICS ON THE STM		57
8.1	Introduction	57
8.2	Device driver and fault.....	57
8.3	Z controller.....	57
8.4	XY controller.....	58
8.5	The complete communications link.....	60
CHAPTER 9: RESULTS		62
CHAPTER 10: RECOMMENDATIONS		65
CHAPTER 11: FUTURE WORK		66
REFERENCES.....		68
APPENDIX A: CONTROL LOOP SOFTWARE LISTING		75
APPENDIX B: PIC CODE FOR CALCULATING Z_{OUT}		78
APPENDIX C: PIC17 CODE TO TEST THE X AND Y DACS.....		80
APPENDIX D: PIC17 CODE TO OPERATE THE ADC0820 ANALOGUE TO DIGITAL CONVERTER (ADC).....		82
APPENDIX E: TEST SOFTWARE TO TEST THE COMPLETE COMMUNICATIONS LINK.....		83
APPENDIX F: PROCEDURE TO TEST THE COMPLETE COMMUNICATIONS LINK.....		85
APPENDIX G: SCHEMATIC DIAGRAM OF THE Z CONTROLLER.....		86

List of abbreviations

Å	Angstrom
ADC	Analogue to Digital Converter
AFM	Atomic Force Microscope
BTC	Binary Two's Compliment
DAC	Digital to Analogue Converter
DC	Direct Current
DSP	Digital Signal Processor
EM	Electro-Magnetic
EMI	Electro Magnetic Interference
GUI	Graphic User Interface
i	Integral control constant
IBM	International Business Machines
IC	Integrated Circuit
Ir	Iridium
JPEG	Joined Picture Expert Group
k	Control scaling constant
kHz	kiloHertz
LPF	Low Pass Filter
LSB	Least significant bit
mA	milliAmpere
MHz	MegaHertz
MS	Microsoft
MSB	Most significant bit
nA	nanoAmpere
nm	nanometer
p	Proportional control constant
PC	Personal Computer
PGA	Pin Grid Array
PI	Combination of p and i controller variables
Pr	Platinum
PZT	Lead-Zirconate-Titanate
RF	Radio Frequency

Rh	Rhodium
RLL	Resonance Locked Loop
SHARC	Super Harvard Architecture Computer
Si	Silicon
SMT	Surface Mount Technology
STM	Scanning Tunneling Microscope
STP	Scanning Tunneling Potentiometry
TIF	Tagged Image Format

List of Figures

Figure 1.1: Block diagram of the system.

Figure 2.1: Block diagram of a complete STM system.

Figure 4.1: Photograph of the scanning tunneling microscope.

Figure 4.2: Photograph of the scanhead on its suspension rig.

Figure 5.1: Graph of current versus distance in Angstrom between tip and sample surface.

Figure 5.2: The Z control algorithm.

Figure 5.3: Graph of PZT-5H tube movement against signal frequency.

Figure 6.1: The Z controller block diagram.

Figure 6.2: XY controller block diagram.

Figure 6.3: Flowchart of PC data sending sequence.

Figure 6.4: Photograph of the XY controller PCB.

Figure 6.5: Photograph of the Z controller PCB.

Figure 6.6: Graph of the latching signal on the PCM1702P 20-bit DAC.

Figure 6.7: Block diagram of the new proposed system that will replace the existing XY controller.

Figure 8.1: DAC waveforms that were produced by the test software.

Figure 8.2: READ and WRITE control signal timing.

Figure 8.3: Z controller output signal showing a 'f80000' value.

Figure 8.4: Output of the Z controller showing the value of '7fff'.

Figure 9.1: STM scan of a gold alloy surface.

Chapter 1: Introduction

The first successful scanning tunneling microscope or STM was built in 1982 in the IBM Research Labs in Zurich, Switzerland. This new type of electronic microscope had the ability to obtain atomic resolution, and for the first time materials scientists had the ability to image atoms in their lattice structure using a tabletop size instrument ¹. It was no longer necessary to use very powerful electron microscopes (which could fill a room) to get atomic resolution. In 1986, the scientific community recognized the achievement of Binnig and Rohrer of IBM by presenting them with the Nobel Prize in Physics ².

It was a long and sometimes difficult road to get to the design of the first STM. Russel Young of the National Bureau of Standards in the USA realized that it is possible to get better resolution than available from an electron microscope by using the tunneling effect^{2,3}. This quantum mechanical effect allows an electron to cross a barrier which, according to classical physics, it cannot cross since it lacks sufficiently high energy. It makes its way so to speak through a potential mountain by quantum-mechanical tunneling; hence the name tunneling microscope. This means that if a conductive scanning tip is near enough to a conductive surface (10 Å, i.e. 3-4 atom diameters) a current will flow even at low voltages. Young did manage to build an instrument but only managed to obtain a resolution of 200 Å. This resolution was thus considerably poorer than other electron microscopes of that time.

Binnig and Rohrer succeeded with their STM^{1,4} but it was still very bulky. One cause of this was the vibrational isolation system that was used to isolate the scanner from disturbances in the environment. The microscope was built on a permanent magnet that floated freely on a disk of superconducting lead. Today, less bulky and equally effective isolation mechanisms have been developed^{5,6}. The Nanosurf Company of Switzerland has developed a STM that is a good example of this⁷. The horizontal resolution was approximately 2 Å and the vertical resolution was 0.1 Å for this instrument. In 1986, Pohl defined the required lateral resolution to be 0.1 Å and the vertical resolution to be 0.01 Å⁸. This is the resolution required to operate with atomic resolution.

The STM is a very simple instrument when considered in terms of its basic components. The primary component is the piezo ceramic scanner element into which a needle tip is fixed. The

needle tip can be a piece of noble metal wire (Pt 15% Rh or Pt-Ir alloys) that protrudes through the scanner tube. The movement of the scanner is controlled by a computer that gives out a control voltage that is applied onto the scanner to cause a controlled movement in the X and Y direction. To sense the topography of the specimen surface a control loop tries to maintain a constant current between the surface and the sample. The Z control voltage is then plotted to reveal the surface features of the sample.

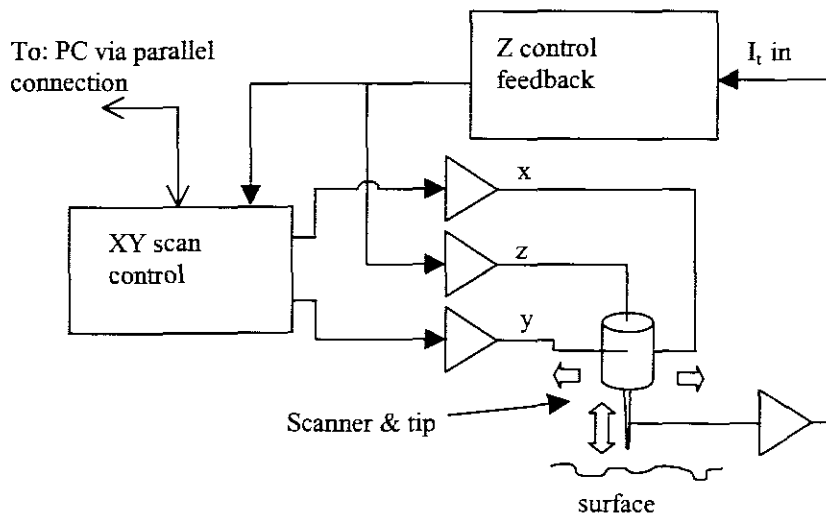


Figure 1.1: Block diagram of the system – showing the scanning tip and scanner from which the tunneling current is amplified and used as the input to the height feedback controller. The output of the Z controller is fed to high voltage amplifiers along with the X and Y signals from the XY scan generator to the scanner and tip. The Z control loop output is also sent to the XY controller from where it is sent to the PC for display.

Since 1982, there has been a considerable amount of new work done in this field, with surprising results. A whole group of sub-types has emerged from this single STM design. Instead of using current as a feedback variable, capacitance can be measured⁹, as well as resistance, elasticity, magnetic force and a host of other variables. Resistance can be measured by applying a voltage across the sample, and measuring the current orthogonally to the voltage field. This is called scanning tunneling potentiometry (STP)¹⁰. Elasticity measurements can be done using an atomic force microscope in which a stylus is used to profile the surface¹¹. In these techniques, the micrometer scale deflections of the AFM stylus can be measured using a focused laser beam.

The subject of this thesis is improving on the design of an existing STM that was built by Tapson for his Ph.D at the University of Cape Town¹². The two areas that were identified for

improvement were the imaging software and the control system. It was decided to upgrade the DOS based imaging software to Windows based software. The control system was also too slow for the purpose for which it was built. There were several communication bottlenecks between controller and the PC. Amongst these were the serial and parallel data links between the embedded system and PC, which were not built for a high data throughput. The new data links consist of an 8.25 Mbit/s serial link between the Z and the XY controller, and a plug-in PC card that is the interface between the XY controller and the PC. The data throughput of this link is in the order of 550 Kbytes/s. The lookup-table-based Z directional control was replaced with a fast microcontroller that uses a control algorithm to calculate the loop in real time. The reason for scanning the surface as fast as possible is because of the possibility of temperature variations that can cause the surface to expand, contract or drift, giving a distorted image. The old system scanned a surface of 256 by 256 data points in just over 15 minutes. The new system can scan a 512 by 512 data point surface in just over two minutes. This is a substantial improvement.

Chapter 2: Review of STM technology

2.1 Introduction

There are various routes to improving the performance of a STM. By redesigning the controller electronics, it is possible to make it faster and more intelligent. The signal processing can be improved, in terms of the way it corrects a distorted image, or cancels environmental noise. The instrument size can be reduced to make it more compact⁵. The scanner head can be redesigned to accommodate different scan heads, or different tips. The system cost can also be reduced to make it affordable for universities and small research companies⁶. Every field related to STM is open to improvement.

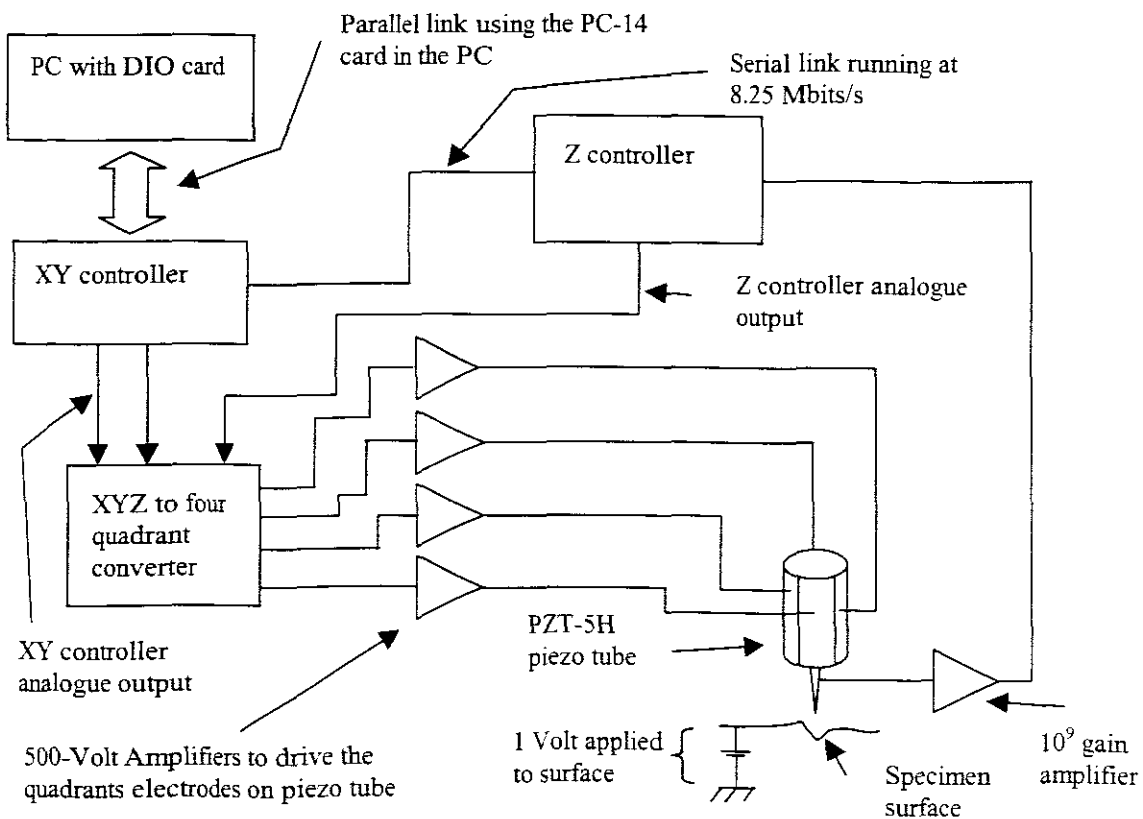


Figure 2.1: Block diagram of a complete STM system – showing the computer with the PC-14 card, the bi-directional parallel link running to the XY controller, the analogue outputs of the Z and XY controllers that go to the XYZ to 4 quadrant converter, the four 500 Volt amplifiers that drive the piezo tube, the 10⁹ gain stage that amplifies the 1nA tunneling current and feeds it to the Z controller and the 1 Volt surface biasing voltage on the specimen surface.

2.2 The tunneling tip

Almost all of today's STM instruments use some kind of piezo ceramic scanner to scan over the surface to obtain the image. The tip is normally a piece of noble metal wire that protrudes through the hollow centre of the piezo tube. There are also several other ways of mounting the tip. It can be mounted in such a way that it is still protruding through the hollow centre, but when the specimen is scanned, a piezo tube lengthening or shortening will give an X direction movement for the tip. Lateral deflections of the piezo material will give the Y and Z tip movement. The method of mounting the tip can be used when the surface features are very rough and a large scan area is required. The scan area will however not be square in shape, but rather a long strip. Three separate piezo tubes or bar elements can also be used to mount the tip. In this case each piezo tube is used only in the lengthwise direction. As each of the three piezo materials contracts or expands it produces the required movement in the XY and Z directions^{1,7}.

The tip is brought in close proximity to the surface (approximately one nanometer) of the specimen at which point a current (approximately one nano-ampere) will start flowing from the surface to the tip. The bandwidth at which the tip can be controlled is normally in the range of several kHz. There are however exceptions to the rule. One exception is the design of an optical switch type STM tip¹³, which is discussed below.

2.2.1 Opto-electronic tip design

The optical switch type STM tip was designed to give very fast temporal resolution. These tips are used primarily for the measurement of laser induced transients on transmission lines and photoconductors¹⁴. These optoelectronic tips are also used to scan very fast over a small area to make a movie of the movement of atoms in their lattice structure¹⁵. The opto-electronic tip was designed to try and increase the scan speed of a STM.

The basic structure of these special tips is a PtRh wire tip that is attached to a photoconductive Si switch. A laser beam is focused through the substrate to control the conductiveness of the silicon switch. The time between when the laser pulse is applied and when the tip current start flowing is then measured to obtain the required information about the specimen that is being scanned.

2.2.2 Tip sharpness

Tip sharpness is very important. When scanning at atomic resolution the surface features are created by atoms. This means that it would be ideal to have a single atom at the apex of the tip. If the tip is that sharp, the stream of electrons that the tip sends out or receives is coming from a single point. If a single atom is at the apex of the tip, then the tip can be modelled mathematically and the current can be characterised¹⁶. If the tip is not atomically sharp, it is possible to get tunnelling currents forming at different points on the tip. The result of these random tunnelling points is that it is not possible to determine with great accuracy what the tip was imaging and where it was focussing, and the result is a useless image, if an image was obtained at all. It is thus important to take good care of the tip and to understand the geometry of the tip¹⁷.

There are a lot of sources in the literature on how to prepare very sharp tips by etching and/or cutting^{18, 19, 20}. Most of these tips last about an hour before they start to go blunt again. This is normally due to oxidation of the surface of the tip, or tip crashes. Tips are made out of Platinum and 15% Rhodium or Platinum and Iridium alloy. The basic process of etching involves an etching solution and some mechanism to pull the etched wire from the solution at a constant rate. A typical rate is about 1mm per minute. This process gives tips that can have only a few atoms at the tip. Commercial tips are available with only one atom at the tip but these tips are very expensive. These are manufactured using a combination of etching and polishing.

2.3 The control system

The loop that controls the height above the specimen surface is the most important part in the whole system. This loop has to maintain a constant height above the surface by keeping a current value constant. In this lies the first problem with the height control loop. It is designed to control the current while it is actually intended to do something totally different. It must control the height above the surface of the specimen. It has only a superficial correspondence with the widely used *single-input-single-output* control loops. The problem is to decide what the tunneling current represents. Is it the input to the control loop, or is it a disturbance that is added to the system? Tapson¹² also noted this ambiguity.

There are various ways of implementing a PI controller for the height control above the specimen. Tapson covered the complete development of control systems in his thesis ¹². Tapson's work came down to a decision between using a sampled data system or an analogue system. The design of an analogue scan system tends to be more favoured by research groups ^{24, 26}. Commercial instruments tend to use digital control systems. An example of this is the *Discoverer* system from Topometrix ²⁷. Another example is the *CryoSTM* system from Oxford Instruments ²⁸. The drawback of a digital controller is that it takes many more man hours to build. Not only the software but also the physical circuit design and layout take up time. The only way to avoid the software design problem is to use a hardwired digital controller. This was done by Valenzuela-Benavides²⁶. This hardwired controller can be classified under the same heading as the analogue controller because once the controller parameters are built into the system, it is very difficult to change. It is better to go with a system where the control parameters can be easily changed in software.

The capability to change the control parameters in a digital control system makes the overall system very flexible. It is possible to scan a broad range of specimens without rewiring the controller; only the software parameters are changed. The decision for a digital control system can be subdivided further between a digital signal processor (DSP) system and a microcontroller-based system. The use of a DSP based system is very popular because DSP algorithms are very powerful ^{28, 29}. These powerful processors allow the implementation of any type of control strategy from a simple PI loop to a more complex multivariable or fuzzy control loop ³¹.

The other option to follow in a digital control system is to use a microcontroller based system. This was done by the present author²³ and also by Tapson ¹². The microcontroller solution has two possibilities. The first option was used by Tapson and consists of generating a lookup table as a substitute to calculating a real-time algorithm. The lookup table is calculated with a control algorithm on a PC or workstation and is downloaded to the microcontroller. The advantage of this is that the microcontrollers are very cheap and very fast throughput rates are still possible. The downside is that the lookup table download is time consuming. The table must also be recalculated each time the specimen is changed. A second option is to use fast microcontrollers. This is the route that this author took. The XY

and Z directional control is done on the microcontroller. The Z algorithm is calculated after each sample and a DSP comparable throughput is still obtained.

The use of floating point calculations in the control loop was later questioned by Tapson³². The control loop is basically an *integer-in-integer-out* loop. The use of a DSP processor to do the floating-point calculations in the control loop is thus questionable. The ADC and DAC values can all be represented by integer values. The calculations will also not produce fractions and can be done using integer mathematics. The bandwidth requirement of a STM is also low enough in the Z direction that a fast microcontroller can handle the control loop with ease²³.

2.4 Imaging systems

The imaging systems that were used in the earlier STM systems were mainly analogue storage oscilloscopes. The actual image was then photographed from the screen of the storage oscilloscope to get a hard copy of the image that was scanned. This was a cumbersome method at best. Later years brought combinations of data display, eg. using both a storage oscilloscope and a PC³³. In even later years, the various researchers and companies started to use personal computers to do the image display of the specimen surface^{21,22,29,30,34,35}. This is the method that is still used today^{23,36}. Some researchers have tried to use a different method of obtaining the image with the aid of sophisticated frame processors and frame grabbers³⁷. This was done to try and improve the throughput speed of the STM.

Most early STMs used the MS-DOS based operating system to display a line type image. Some early papers show examples of line type images^{38,39,40}. The lines then formed a type of 3-dimensional display that represented the topography of the surface. With the popularity of MS Windows⁴¹ products, Windows became the operating system of choice to do graphical displays of the data. A good example of this is Barchesi *et al*, which used a Pentium based computer system (PC) and Visual Basic under Windows to develop their application⁴². Some companies like Topometrix²⁷ have used Unix based systems, and others have opted for a Windows based modular software approach using Labview⁴³, eg. Oxford Instruments²⁸.

Chapter 3: Previous work

3.1 Introduction

The only STM related work that was done previously at UCT was by Tapson¹². The Materials Engineering Department needed a wide area STM that could be used for metallurgical research. Tapson designed and build a STM that filled that requirement.

3.2 Historical methods and current practice in STM design.

Designs in STM literature have always been filled with lots of detail. It is therefore easy to build a STM based on the technology that is described in these publications. This was true from the Topografiner of Young *et al*³, which showed the basic elements of a STM, up to the present day.

In its simplest form, the STM consists of a tip, a piezo scanner, a control system to control the height above a specimen, a data acquisition system and display, and a vibrational isolation assembly to isolated it from any outside vibrations. To construct a successful STM, all of these components must be integrated. An exact specification for building a STM was cited by Pohl⁸. Pohl stated that a resolution of 0.1 Å is sought in the Z direction and 1 Å in the lateral or X and Y direction. The resulting tolerance must be ten times better at 0.01 Å in the Z direction and 0.1 Å in the lateral direction. Pohl also stated four additional mechanical requirements that must be met. These are a Z regulation in the 0.1 to 1µm range, X and Y scanning capability in the same range, a tip approach mechanism in the 1 mm to 100nm range and a scan window selection in the same range.

The type of specimen that must be scanned dictates the design of the STM instrument. The thesis of Tapson¹² described the design of a STM that will be used on noble metal alloys. The tip that is used is made of a noble metal alloy wire like Platinum 15% Rhodium. It is used because it is hard, and easily available in the form of thermocouple wire. The method that is still very popular for sharpening the tips is simple cutting with a side cutter^{44,45,46}.

The piezo tube or scanning actuator as it is called, determines the accuracy of the movement of the scanning tip. Earlier designs used combinations of piezo washers³, beams^{1, 47} and bimorph structures⁴⁸. The tube scanner was invented by Binnig and Smith⁴⁹. These tubes are now the dominant piezo structures used in the construction of scanning actuators. The nonlinearities that are associated with the piezo materials are a source of concern. The problem was approached firstly by designing scanners with independent XY and Z actuators⁵⁰ and secondly by using feedback control on the tube to linearize (correct) the scanning motion. The later method can use optical^{51, 52} or capacitive methods^{53, 54}. The piezo tube that was selected for this study was the PZT-5H tube because this type has a higher electromechanical coupling⁵⁵.

The control electronics is another key element of a good STM design. Tapson's STM was used in constant current mode. This means that the control loop tried to maintain a constant current between the sample and the tip by varying the height of the tip above the sample. The high dynamic range that is required forced the designer to design the control electronics to have a very big gain. The best way to achieve this is to use an integrator in the loop design. Tapson went into considerable detail in as far as the different control strategies goes. The main consideration is a rule of thumb that states that the usable scan frequency in the lateral direction can only be a third of the resonant frequency of the scan tube. This was described by Pohl⁸. Also it is difficult to measure the dynamic response of the plant (tip, tunneling junction and tip) in a scan because it is impossible to maintain a stable tunneling junction in an open loop test. One solution to this is to use a closed loop with a very slow integral controller to perform the test^{25, 56}. The selection of the final controller is a choice between a digital implementation and an analogue design. Further aspects are discussed in Chapter 4.

The dynamic range problem has also complicated the STM designs. The dynamic range is that range over which the tip can be extended in the Z plane. The scanning tip can typically be moved over a range of 5 μm in the X and Y directions. The range in the Z plane is only a fraction of that. In many cases where the range was less than 1 μm in the Z direction, the digitisation range was only 16 bits or even less at 12 bits^{21,57,58}. One solution is to use separate low and high voltage control systems but to only digitise the low voltage control system output^{48,59}. A second way is to digitise the whole of the Z range to 17-20 bit accuracy. This is more in line with Pohl's cited requirement of 0.1 \AA , which means that the

actuator's range in the Z direction must be digitised to 1 part in 10^5 . This would require a 17-bit digitisation (one in 131072).

The problem of coarse approach in an STM has been the source of the greatest innovation in STM design. The problem that is faced is very simple. The full Z displacement is in the order of $1\mu\text{m}$. At this small range the implementation of a mechanical system becomes very difficult. A further requirement is that the scanner itself must be small and rigid. The specific object is usually to move the sample in the sub-micron range without causing a tip crash. A tip crash will result in a deterioration of the tip and also of the surface. A damaged tip has to be resharpened. A simple but effective method is to use a differential screw to bring the tip within range. This was first done by Young³ and later again by Dovek *et al*⁶⁰ and Stupian and Leung⁴⁴. The researchers at the IBM Zurich facility used miniature walking devices known as "louses" to do the close approach. These suffered from poor reliability^{61, 62, 63}. Later the commercial "Inchworm" walking devices⁶⁴ suffered the same problem. The more reliable inertial slider type⁶⁵ is now popular. A popular method is discussed^{66,67} in which the scanner head is suspended over the sample and two screws are used to do the coarse approach and also to act as the fulcrum in the lever. The third screw is then used to do the fine approach at the lever's end.

The XY translation stage has become an additional piece of equipment to include in a STM design. The design criteria are the same as the coarse approach system. It requires small size; small step size, simplicity, reliability and it must accommodate a wide range of samples. Basically the same methods for moving the tip in the Z direction are also applied to move the sample in the XY direction. The main method of interest here is also an inertial sliding method^{68, 69, 70}.

Scanning tunneling potentiometry (STP) has only been used in one direction previously. Tapson has however expanded it to two dimensions. The adding of a second scan over the surface at right angles to the first showed a lot more information than just a single scan image. The way that the STP image is obtained is by applying a current over the surface and measuring the voltage at right angles to that. Several systems were reported in the literature^{10,71,72,73}.

3.3 Cantilever STMs

In previous work at UCT a dual cantilever mechanism was constructed to give a tip to sample approach mechanism. It was found that thermal creep was the main reason why this design did not work. The slightest thermal expansion of the beams that were used for the cantilever caused instability. The tunneling current could not be maintained for more than a few seconds at a time.

A single cantilever STM was then constructed. As in the previous design it was the coarse approach mechanism that was under the spotlight. The single cantilever offered a good reduction ratio and the coarse approach was quickly performed. It was found to work satisfactorily. The main problems encountered now were related to digitisation.

3.4 Dynamic behaviour of STM structures

The behaviour of the scanning actuator is limited by the first resonant frequency of the piezo tube. A second limiting factor is the time it takes the tube to respond to a change in driving signal. To improve the scanning speed, these limiting factors must be changed, by either raising the resonant frequency of the scanner tube or by improving the control system to operate closer to the resonant frequency or even greater than the resonant frequency. The lowest resonant frequency has been increased by the use of tube scanners⁴⁹.

In previous work at UCT a two dimensional split cell arrangement was used to try and measure the resonant frequency of the tube. It was found that there are multiple sources of resonances in a STM scanner. A big problem was cross coupling of resonant frequencies from the base plate to the piezo tube. The need to isolate the system properly was shown with the 2D-split cell arrangement. A one-dimensional arrangement was also tried and it gave the required results. This was done by eliminating all possible sources of resonance and making the test rig as rigid as possible. The various resonant modes of a clamped piezo tube were measured. (See figure 5.3)

Next, a capacitive method was tried to measure the movement because it is a well-understood method⁷⁴. Research into measuring the small displacement gave a new method of measuring small capacitances. This new method is called capacitive measurement by resonance locked

loop (RLL)⁷⁵. The loop gives a DC voltage out, which is then used to plot the dynamic response of the system. Tapson went into considerable detail as to how the RLL works and the results that are obtainable with the method¹².

3.5 A dual actuator STM design

In a dual actuator design, the Z actuator is a piezo disc on which the sample is mounted. The tube scanner is used in the X and Y directional scans, but the high resonant frequency of the disk in thickness mode is used to give fast Z movement. The tube scanner can still be used to give slow Z movement. The problem with this set-up was that the disk movement was transmitted to the whole scanner structure (tube scanner, coarse and fine setscrews etc.) and excited resonance in the tube scanner. A method of filtering the output and later actively cancelling the vibrations was tried. The last method was not successful.

It was found that the conventional control of tube type actuators cannot be significantly improved upon without major increases on the complexity of the controller that is used, or further reductions in actuator size and range. With careful design of the control system for the separate actuator, the resonance that it induces can be overcome.

3.6 The feedback of STM devices

The first problem that was tackled was that of flying. This is when the tip momentarily loses the tunneling current on a downhill slope. It was mentioned by Scholl *et al*⁷⁶ that a conventional system without a logarithmic amplifier tends to “fly” over or lose contact on a downhill slope. Several methods for fixing the problem such as post-processing the image are given.

The next problem was the conversion from an analogue system to a digital system for the Z controller. Various questions were asked, such as the effect of discrete sampling on the input and the output; the effect of the time effect between input and output; the effect of quantisation of the input and output; the effect of calculation accuracy on the control result. These questions must be considered when the final control loop design is implemented.

Several methods for a digital PI implementation were considered. Popular algorithms are the standard rectangular approximation⁷⁷ and Tustin's method⁷⁷, which uses a trapezoidal approach. Of the several questions asked earlier in connection with the transition from an analogue to a digital system, only the output quantisation was important. If the output digitisation was too coarse, the output failed to settle. Instead it oscillated around a value. This is called limit cycling.

This limit cycling could have an adverse effect on the STM controller. It could generate noise that will appear as measurement artefacts on the image obtained. One way of minimising this limit cycling is to use a very fine output digitisation (eg. a 20-bit output DAC).

3.7 The control System: implementation and performance

The control system was implemented as a PC with a PC-30 card from Eagle Technology and a signal summer to change the XY and Z control signals to the appropriate four-quadrant signal. The PC-30 card is a PC plugin card that has several DACs and ADCs onboard. This configuration proved to be plagued by up to 50mV of noise on the signal-input lines. The source of the ground line noise was eventually traced to the analogue ground of the PC switching power supply. The inside of a PC itself is also a very noisy environment, because almost all of the signals in a PC are digital and it does not matter if there is a bit of noise on the signals. This ground line was also the ground line for the analogue data acquisition system. The system was then redesigned to try and isolated the control signals from the noise sources.

The next iteration of the electronics used a custom built data acquisition system based on Intel 8051 type microcontrollers. One was used for the Z control and one was used for the combined X and Y control. This configuration used a PC-14 digital input/output card to transfer the data via parallel links to the PC. The PC-30 card was not used in this new system. To get rid of the noise on the analogue ground line, separate power supplies were used for the data acquisition system. The data signals that were sent to the PC, were isolated from the PC with opto-couplers. The opto-couplers let the signal through but the two sides of the opto relied on different power supplies. This meant that the noisy PC digital and analogue ground lines did not get to the embedded system.

Tapson also opted to use a microcontroller system and not a DSP based system. A DSP system would have been much more expensive. A sampling speed of 55kHz was still achieved by using lookup tables. The use of 12-bit digitisation in the Z controller was also proven inadequate.

A solution to the dynamic range problem is also presented. The low voltage control signal is added to the high voltage (coarse Z) signal. This system allowed high precision digitisation of the Z signal. This method allows the designer to use only 16-bits but still get the required resolution.

Chapter 4: System implementation

4.1 Introduction

A literature study has shown that there are various ways of implementing a scanning tunneling microscope. The environment dictates a basic design drawn from the many different designs for the scanner arrangement. The scanner is the part that contains the piezo ceramic tube element, and often the first current amplifier and the vibrational isolation assembly. The scanner is used either in a high vacuum^{45,78,79}, low temperature^{35,81}, combinations of vacuum and low temperature^{62,70,82}, variable temperature^{46,67,80}, at room temperature¹² or in a fluid⁸³ environment. In all cases, the basic electronic control and imaging systems are more or less the same. Some designers used a dedicated video frame grabber to get the final image³⁷, while some have used specially written software that they developed to do the imaging. The control system is nearly always based on a PI control loop^{25,77,85}. The hardware for the control system can again be implemented in various ways^{29,30,31,42}.

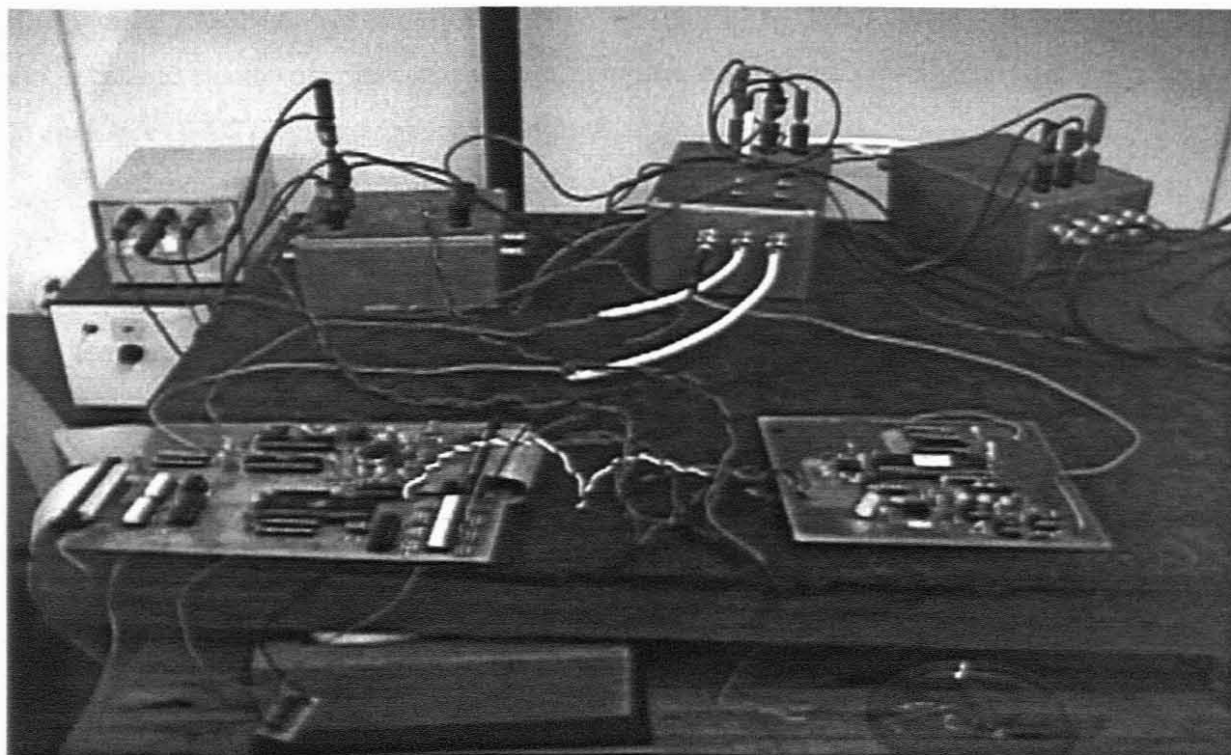


Figure 4.1: Photograph of the scanning tunneling microscope, showing the XY and Z controller in the foreground. In the background from right to left, are the 500V amplifiers, the XYZ to four-quadrant converter and the three power supplies, all resting on the vibrational isolation assembly. The two wires (X and Y) from the XY controller and the single wire from the output of the Z controller can be seen going to the XYZ to four-quadrant converter and then to the 500V amplifier from where the four signals can be seen emerging again.

This thesis is only concerned with the control system and a new user interface that is integrated with the new design. As with previous designs, this implementation also used a PI control loop that is implemented in the digital domain to make it easier to change the type and variables of the controller. The software is the only thing that is updated during operation. Several different hardware configurations were evaluated for the embedded system implementation.

4.2 System overview

The STM consists of several essential components that make up the system. These are the scan head, the control electronics, the PC and the vibrational isolation assembly. The scan head is mounted on the vibrational isolation assembly to isolate it from various kinds of movement. These include minute micron size building vibrations, vibrations caused by footsteps and even accidental bumps into the STM instrument as a whole.

The control electronics is mounted outside of the PC to make it easy to isolate it from electrical and electro-magnetic noise. It is here that the analogue-to-digital (ADC) and digital to analogue (DAC) conversions are performed.

4.2.1 Scan head

The scan head is the same head that was used in the previous iteration of the STM¹². It consists of a PZT-5H piezo ceramic scanner tube that is mounted inside a steel block. The block has set screws that are used to do the coarse approach of the tip to the sample. In the scanner tube is mounted the scan tip that is made of Platinum and Iridium. The tips can easily be damaged but is easy to sharpen them again^{18,19,20}.

On top of the steel block is mounted the first (transconductance) amplifier that converts the current to a voltage and which amplifies the voltage by 10^6 .

The steel block with the scanner tube rests on a second steel block onto which the sample is mounted, and which contains the fine approach screw. The fine approach screw has a

micrometer screw gauge thread with a pitch of 0.5 mm per turn. The block is suspended with rubber bands from the vibrational isolation assembly

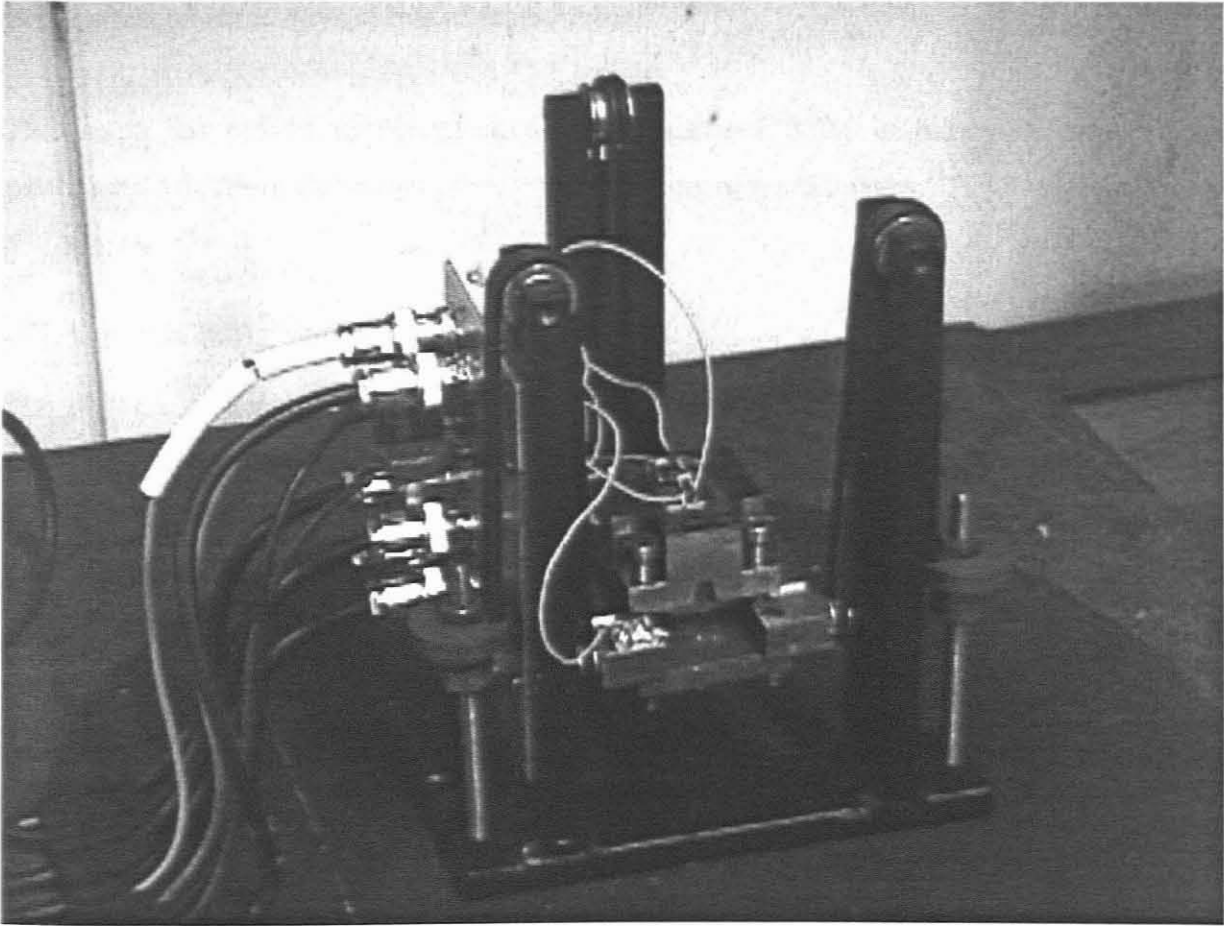


Figure 4.2: Photograph of the scanhead on its suspension rig. The scanhead with the 10^6 amplifier can be seen clearly along with the wires that connect it to the BNC connectors. The cables on the left supply the power to the 10^6 amplifier and the control signals to the piezo tube.

4.2.2 Vibrational isolation

The vibrational isolation assembly was constructed to isolate the STM head from any mechanical vibrations. These vibrations can be anything from micron size vibrations of the building caused by the wind or passing vehicles, to accidental bumps into the STM table. The isolation consists of a set of three rubber bands that suspend the scanner head from the scanner mount. The mount rests on a 5-millimeter sheet of foam rubber. The STM mount is covered with a bell jar to prevent movements of the scanhead by the surrounding air. The whole assembly rests on a very heavy slab of slate, measuring 5cm by 50cm by 100cm. The slab makes the assembly very heavy and helps to dampen any frequency that the assembly is

exposed to. The slab is supported at its corners by four sets of five tennis balls. This has the effect of blocking most if not all of the received vibrational energy. The tennis balls are contained in four tennis ball buckets which hold five tennis balls each. The tennis ball buckets rest on some more sheets of foam rubber.

This setup has proven to be effective in the previous STM in removing most of the mechanical vibrations that might prevent the obtaining of good images¹².

4.2.3 Noise considerations

The PC is a very noisy system and is always enclosed in a metal case to prevent it from radiating electro-magnetic radiation into the surroundings. The embedded control system was thus placed outside the PC to reduce the risk of it being influenced by EM noise. All connections to and from the PC are optically isolated to prevent noisy ground connections from affecting the control boards.

In the previous iteration of the STM electronics, it was found that the PC analogue ground line is a very big source of noise. The analogue ground line was used as the ground for the signal acquisition boards of the STM. The analogue ground line is attached to the ground point for the switching power supply of the PC. It was necessary to isolate the signals using opto-couplers in the previous STM. This stopped the ground signal noise from reaching the control PC boards.

The control boards are specially designed for minimum EM noise susceptibility. All connections to and from the PC boards are made with shielded cable to minimize the chance of EM noise appearing on the analogue signal lines of the control boards and influencing the imaging of the STM.

The single component that is most susceptible to noise in the system is the Burr-Brown PCM1702P 20-bit DAC. This chip gives out a 9.6 μV change in output for one least bit (LSB) change on the digital input. The voltage level is very small compared to the normal noise level of the environment. An unshielded measurement using an oscilloscope gives a noise level of between 2 to 30 mV. Electro-magnetic shielding is thus very important.

4.2.4 Electronic systems

The electronics of this study consist of the scan head, the PC, the control boards for the XY and Z controllers, and the various power supplies. The scan head needs a 500 Volt power supply to drive the piezo tube and a ± 15 V supply for the 10^6 current gain amplifier and the 10^3 voltage gain amplifier.

The control boards give out the XY and Z directional control voltages that are first split into a four-quadrant signal before passing it to the 500V amplifiers. The output of the 500V amplifiers are then used to drive the four quadrants of the piezo tube.

4.3 DSP implementation

This thesis is concerned with the implementation of a new control and imaging system. A literature study was done to determine the best solution for a DSP based control system implementation. First, an off-the-shelf system was looked at, using plug in boards that are available from *Data Translation*⁸⁵ and *National Instruments*⁴³.

The *Data Translation* board was designed to plug into a standard PC. It had a TMS320 DSP type processor with dual port memory and various DACs and ADCs onboard. It could easily be programmed in C to control the STM. Other research groups have used this kind of plug-in card with great success. It was decided not to use this well designed plug in board because of its very high cost, which was outside the available budget for this thesis. The placing of the analogue circuits in the PC is questionable since the PC produces a lot of electrical switching noise. The ADC and DAC circuits on these cards may at best be good up to 12 bits unless the analogue circuits are very well shielded. It is better to place the analogue circuits with their controllers outside the PC in a shielded box.

It must be noted that with all the available software that a company like *Data Translation* and *National Instruments* offer, it is possible to build an off the shelf system but at very high cost.

The next option was to use a small DSP development board that is available from Texas Instruments⁸⁶. These use the TMS320c30 processor with some outside logic, a DAC, an ADC and a serial port. The system was too restricted to allow easy interfacing to a STM system.

The DSP chip had the speed to run the system fast enough but was lacking in the environment that it was built into, namely the development board. It was available at a relatively low cost. A paper that became available near the end of the author's work confirmed that this option was the wrong path to follow. Paillard *et al*³⁶ used a TMS320c50 development board that is the same as the one that the author considered except for a more powerful DSP processor. The results that they obtained for their Z directional controller gave only a sampling rate of 25kHz. It must be noted that the onboard ADC and DAC were used, which were optimized for speech processing. Paillard also noted that the DSP processor still had lots of spare processing capacity. The system was priced at US \$250.00, which is too expensive when it was compared to the cost of producing the PCB with the microcontroller, ADC and DAC to form a Z controller. The cost of the latter worked out to about \$95.00.

A second DSP system was also evaluated. The Analog Devices SHARC 2106x development board is a very capable system⁸⁷. This board was available at US \$179 and came with the C compiler, C libraries and a simulator. It was finally also not selected because the cost in developing the PCB for this DSP chip would have been too high. It used a very fine pitch pin spacing on the DSP chip. To have the surface mount technology (SMT) device placed on the board would make the system too expensive. The DSP development board could be used along with circuits that interfaced to it, but one of the requirements of this thesis was to try to produce a commercially viable product at a low cost.

It was decided to consider building a system based on a DSP chip that can be placed using a Pin Grid Array (PGA) package. The PGA package style has the pins spaced 0.1 inches apart bit in an array style underneath the IC. The DSP chip would perform all the control algorithms and the interfacing to the PC via the parallel and serial ports. The STM Z directional control loop should run at a speed of exactly 60 kHz. This means that the Z control will be done at 60 kHz while the X and Y control will be done at 3 kHz. The reason for the X and Y bandwidth is fully explained in Chapter 5. This leaves just the short interval between XY control movements to send data to the PC via either a parallel or a serial connection. If the DSP system was tied up for some reason by a data transfer to or from the PC, the Z control loop timing might be influenced. The use of two DSP chips might solve the problem but the cost factor was creeping into the picture again.

The control loop speed is determined by the resonant frequency of the PZT-5H scan tube. It is vital that the Z control loop is not interrupted and is running continuously. More information on the bandwidth requirements of the scanning tube can be found in chapter 5.

4.4 Multi processor system

A multi processor system using cheap microcontrollers was also considered. In this configuration, each processor is given a simple task to do. The tasks are coordinated by inter processor communications or handshaking. The first processor performs the Z control functions and the second performs the communications to and from the PC and the XY control. The advantage of a multi processor system is that the control process can be chopped up into many pieces to give each processor a simple task to do. The processors can then communicate with each other to determine which is doing what. In this manner a parallel execution of instructions is obtained without using a fast processor that does time slicing to simulate a parallel process.

In the context of the STM system, a parallel process would occur when the Z control algorithm is running and data is read from the XY control system without an interruption to the Z control algorithm's execution.

The choice for the final selection was between an Intel 8051 style processor⁸⁸ and the much faster Microchip PIC17 series of microcontrollers⁸⁹. Both types of microcontrollers were well known by the author. The final choice was based on the fact that the Microchip PIC series has more peripherals available on the microcontroller and that the development tools were available from the Microchip company free of charge.

The PIC17c43 devices that were selected for the project had a few features that made them very suitable. The most important feature was the 8x8 bit unsigned multiplier that can multiply two bytes in a single machine cycle. This was very useful for calculating the Z directional control algorithm in real time. The second important feature was the high-speed serial port, which can be used in a synchronous mode. In this mode, it generates a data stream with a clock signal, which is what the serial 20-bit DAC needs for its input. The third feature was the 16 bit wide external bus that was used to load the 16-bit DACs for the X and Y directional signal generation. The PIC micro controller uses a RISC instruction set with only

58 instructions. These are easy to learn and very powerful when it comes to data manipulation. The PIC series has a divide by four internal clock structure which makes the machine cycle time very fast. The PIC17 series is also capable of doing low-end DSP functions⁹⁰. The PIC17 series of microcontrollers are ideal for this application.

In the current STM design, the embedded system is only divided between two processors, but it can be further divided. If the communication to and from the PC is separated from the XY control system, a third processor could control it. The X and Y control processes can also be split to put yet a fourth processor in. The inter-processor communications can manage the synchronization of the data but the communications protocol will probably be very complex to program. It will be a master-slave type protocol.

4.5 Conclusion

The combined effect of this is that cheap processors can be used while still maintaining a very fast execution time for the embedded system. The cost of the two PIC17c43 processors with their “glue” logic was still below the cost of a single DSP processor. The PIC processors offered a superior solution in both cost, and performance. The performance advantage comes from the ability to make the system modular. In this way components like the XY controller can be removed and replaced with a different module, without redesigning the rest of the system.

Chapter 5: Control Action

5.1 The tunneling junction

The tunnelling junction is the gap between the surface and the scanning tip in which electrons can tunnel from the one surface to the other. This distance is in the order of one nanometer and is called the tunnelling distance. It is this distance between the surface and the tip that must be controlled to a very high degree of accuracy. The controller must try and keep the distance at one nanometer exactly. Luckily a small change in distance causes a huge change in measured current. For this reason, the tunneling current can be digitised using only an 8-bit ADC. The current flow that is generated between the surface and the tip is converted to a voltage and then amplified by a factor of 10^9 times. This means that 1nA is represented by 1V after amplification. It is this voltage that the ADC reads in for the control loop.

In Figure 5.1 a typical current to distance graph is shown for a STM. The setpoint is typically at 1nA at a distance of 1 nm. It can be seen that the current to distance relationship is exponential. As the distance decreases the current increases. As the tip moves away from the surface the current change is not so great but is still easily perceived. All that the control loop has to do is to try and maintain the tunneling current at the setpoint.

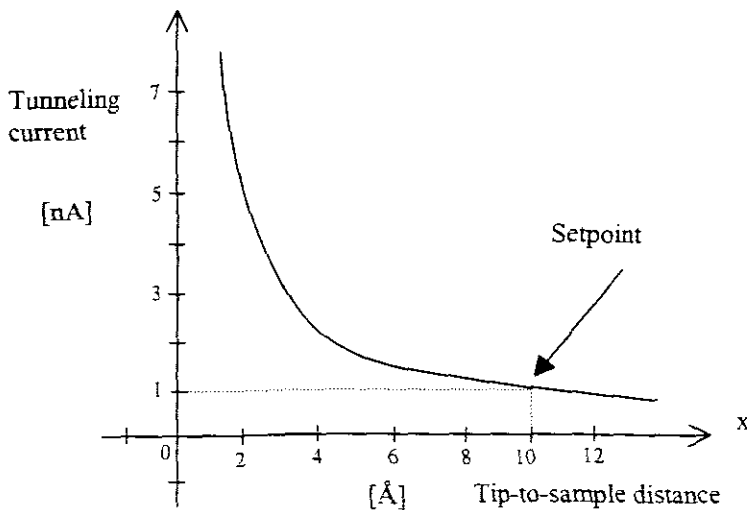


Figure 5.1 Typical graph of current vs. distance between tip and sample surface. A typical setpoint of 1nA corresponds to a specific distance of 10 Å and it is marked with a dashed line. This shows the typical exponential increase in current with decreasing tip-to-sample gap. The dotted lines illustrate the setpoint.

5.2 The control loop formula

The control loop is a digital PI control loop implementation. The use of differential control and to a certain extent, the use of proportional control is of questionable value in most STM systems. This means that the control implementation is usually no more than an integral controller. The author has included the proportional controller term in the control loop calculation because it can be set to zero in the PIC software if it is deemed unnecessary.

The basic formula for an analog PI controller is $V_Z = i_e \cdot p + \int k \cdot i_e \cdot dt$ where V_Z is the output voltage, i_e is the error current, and p and k are control constants. In a digital system we must calculate a discrete-time approximation. The calculation on the microcontroller is started by calculating the error current from the setpoint current and measured tunneling current.

$$i_e = i_{\text{set}} - i_{\text{measured}} \quad (1)$$

The simplest approximation uses stepwise increments:

$$V_Z(t_n) = p \cdot i_e(t_n) + V_Z(t_{n-1}) + k \cdot i_e(t_n) \quad (2)$$

where t_n represents the sampling instant, so that t_n is the sample after t_{n-1} (the time between t_n and t_{n-1} is the sampling time). The sampling time is inherently a factor in Eqn. (2), so the parameters p and k must be recalculated if the loop changes, as this will change the sampling time.

It can be seen that the approximation is very straightforward. There are only two multiplications and two additions to perform. It is very simple and easy to implement on a microcontroller like the PIC17c43.

The new 20-bit Z value can be calculated very quickly because the PIC17c43 has a dedicated x8-hardware multiplier onboard that does unsigned multiplication in a single machine cycle. This means that the new Z value can be calculated and sent to the XY control board without degrading the timing of the system.

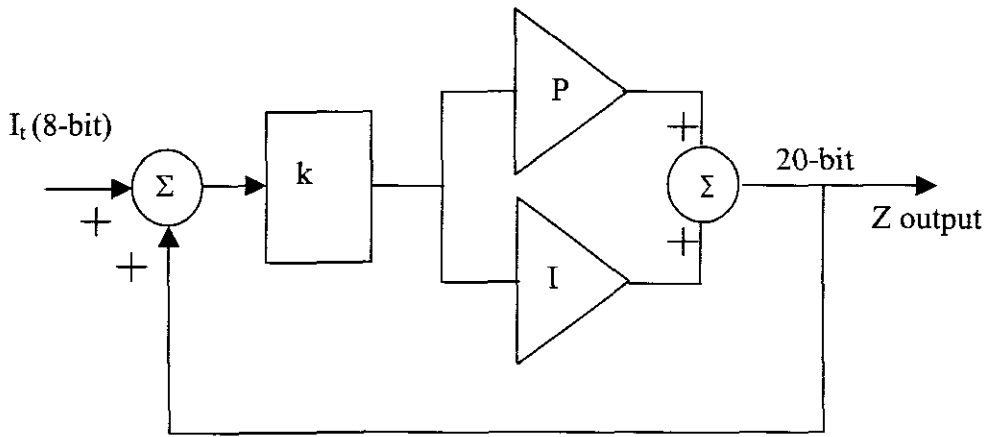


Figure 5.2: The Z control algorithm – showing the 8-bit tunneling current input, the scaling constant k , the p and i control constants and the 20-bit output. The reason why the feedback is indicated as an addition is because when the tunneling current increases, the control voltage in the Z direction must also increase. An increase in voltage in the Z direction on the piezo tube causes a tube compression and hence the tip will be pulled back and the tunneling current will decrease.

The process starts by measuring the current and calculating the error current. This value is multiplied by the p and k control variables. This result will not be bigger than 9 bits. The k control variable and the error current are multiplied together. These two 16 bit values are then added to the previous value of the loop to calculate the new value. It must be noted at this stage that the error current may be negative, causing a subtraction when an addition is done to get the new Z value by adding the newly calculated terms to the old Z value. The algorithm to calculate this loop is shown diagrammatically in Figure 5.2 .

5.3 Scanning bandwidth

The bandwidth of the Z directional scan is determined by the value of the first resonant frequency of the PZT-5H scanner tube. The scanner is the same as was used in the previous study¹². The process by which the Z directional bandwidth is determined is by driving the scanner tube at ever increasing frequencies and measuring the movement of the tube in the X and Y directions. The piezo tube must be clamped, preferably in its final mounting (eg. the scan head assembly). The driving signal should be such that it causes the tube to bend sideways only. This sideways movement of the scanner tube will reveal the resonant frequencies of the scanner. The movement can be measured using a LED and a photodetector. Tapson describes two methods that can be used¹². The scanner that is used in this study has its first resonant frequency at 10 kHz. This can be seen from Figure 5.3. The second and third resonant peaks are also visible. It must be noted that this graph is only valid

for the scan head used in this study. If the piezo tube or the scanhead is changed the resonant frequencies must be measured again.

The control of the tube is only possible up to a frequency of 3 kHz which is a third of the lowest resonant frequency.

The way that this is calculated is by looking at the lowest resonant frequency of the piezo tube. It is very important that the dynamic response of the piezo tube is known. The lateral or X and Y bandwidth is then a third of the resonant frequency. The reason for this is because control without sophisticated algorithms, is only feasible up to a third of the resonant frequency of a system. To obtain the bandwidth requirement for the Z control loop, the Z control signal must then be over sampled by a factor of two (Nyquist criterion), and then the common rule-of-thumb that sampling should exceed control bandwidth by a factor of 5-10 is applied. This gives the optimum bandwidth of 60 kHz for the Z control loop.

If a piezo tube with different dimensions is used the resonance must be measured again. If a longer tube is used a lower resonant frequency can be expected. If a shorter tube is used, the position of the first resonant frequency will be much higher up in the spectrum.

5.4 Constant determination

The p and k control constants must be established once the loop is running. They determine the response of the loop to a change in input. When the loop is first started the constants are set to low values and then slowly increased until a satisfactory response is reached. It is a slow process, which takes some hours to do. The response is measured by looking at the image that is obtained. When this image starts to deteriorate, then the controller variables must be reduced. It is a manual process, which is highly dependent on the operator of the STM.

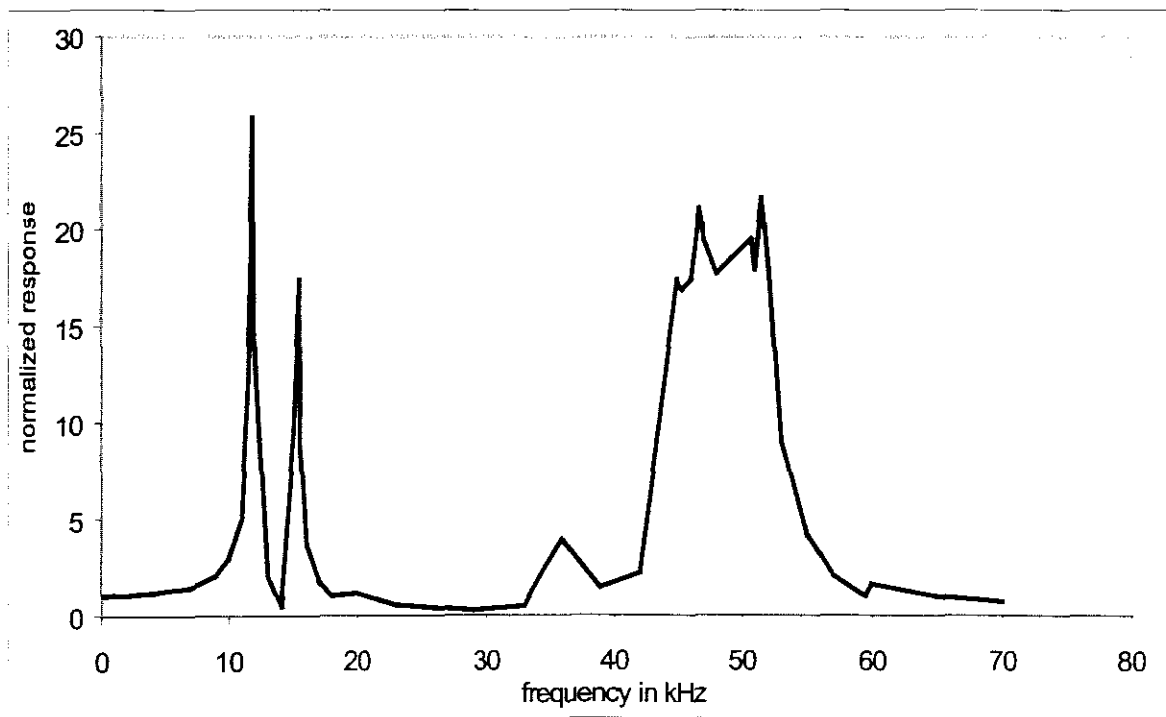


Figure 5.3: Graph of PZT-5H tube movement against signal frequency. The first resonant frequency can clearly be seen around 10kHz. This data was obtained by Tapson¹².

Currently the PC software makes provision for the tuning of the constants. In a later version of the PC software, that option might be removed to make the system easier to use. The constants should be set by somebody with a reasonable knowledge of control theory. The end user may not know what he is doing if he starts adjusting the constants. Setting the constants too high will result in an unstable system, which will prevent the obtaining of good images from the STM. Setting it too small will reduce the response time of the controller and hence increase the overall scan time. Therefore, the menu option may be removed from the PC software once the optimum values have been found.

5.5 Results

The control algorithm was implemented as designed and gave considerably better loop times than were expected. The loop frequency was measured at 69kHz. This can easily be improved by changing the assembler code to straight-line code. This means all the “goto”; “call” and “return” statements are removed. The loop time will then be 81kHz. The ADC0820 is currently sampling at 2.5 μ s but an early read can be performed which will

reduce the sampling time to $1.5\mu\text{s}$. This will improve the loop time by $1\mu\text{s}$, to give a frequency of 74 kHz. Again, the assembler code can be edited to give a straight-line version that gives an improvement in performance. The Microchip Corporation, manufacturer of the PIC17c43 is planning a 40 MHz version of their controller. If the 33 MHz version is swapped for 40MHz version, the sampling frequency can be improved to over 105kHz. It can probably be improved even further by doing the transmission of the results of the previous loop and the reading of the ADC at the same time and not separated as it is done now. By the use of clever software design, the loop frequency can be improved to an absolute maximum of 112.3 kHz for a 40MHz device and 99.3 kHz for a 33MHz device.

The control loop algorithm is now calculated in the controller and not on the PC as was the case in the previous design. This means that the system does not have to use a lookup table and slow download before each new scan, as was required in the previous system's implementation¹².

5.6 Conclusion

The design parameter of a Z controller with a bandwidth of at least 60kHz was easily reached. It was proved that the existing controller could easily be software upgraded to run at least at 87kHz. This corresponds to controlling a piezo tube with a first resonant frequency of 13.5kHz. The Z directional controller is no longer a bottleneck in the system and has spare capacity for future requirements.

Chapter 6: Hardware description

6.1 PIC Processor hardware

The Microchip PIC17c43 processor was chosen to implement the embedded control system. The PIC17c42A could also have been used but the 42 series is still plagued by some bugs in the silicon design. The PIC17c43 also has more SRAM and the EPROM data space is twice as big. The size of the EPROM and SRAM memory space might become important in future improvements to the system. Two of the processors are used, the first to control the Z displacement and the second to control the XY displacement and the communications ports.

The Z control board is kept in a reset condition while the XY control board waits for its initialization data from the PC. The XY control board must receive twelve bytes to set the initial scan position, the magnification and the data format. Immediately after the data is received, the Z control board is taken out of reset. It receives the p and k control variables from the XY controller via the high-speed serial link and starts to run the control loop after the setup phase is completed. Figure 6.1 shows the inputs to the Z controller PCB as the two 8-bit ADCs. The output is the 20-bit Burr-Brown DAC⁹¹ with its Butterworth low pass filter.

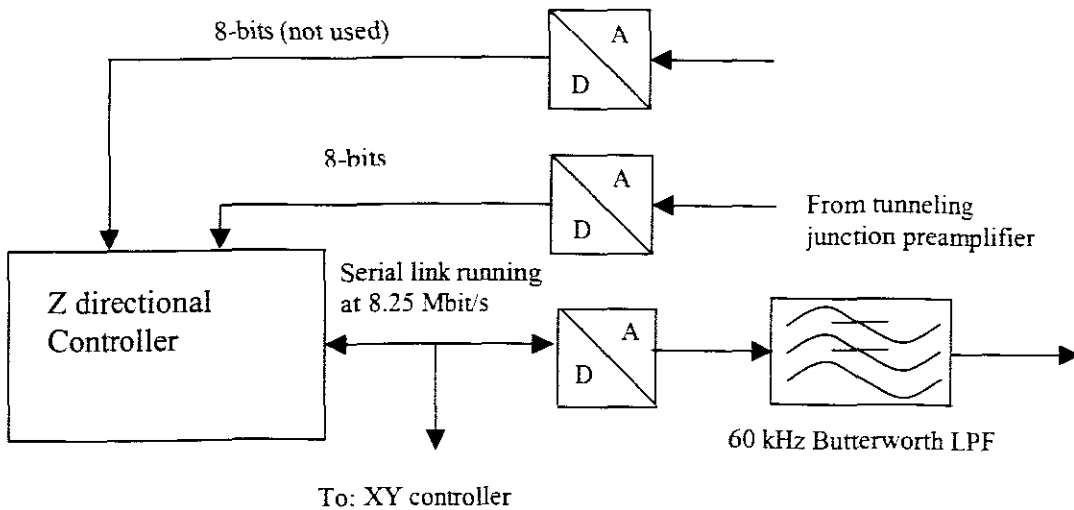


Figure 6.1: The Z controller block diagram, showing the data inputs and the DAC output with the 3rd order Butterworth LPF. The high-speed serial link is common to the DAC and the XY controller.

After each loop, it sends the results to the XY control board. There one of three things is done to the data. Either it is discarded, or it is placed in memory and run through an averaging function, or all twenty sets are sent to the PC. These data sets can be handled in different

ways by the system. The data that is sent by the Z controller to the XY controller is sent to the Burr-Brown PCM1702 DAC at the same time. The DAC requires the data in a Binary Two's Complement (BTC) bit reversed format. This means that the 20-bit data stream is transmitted MSB bit first instead of LSB first. The PC or the XY controller must reverse the bits again before it can be used by them.

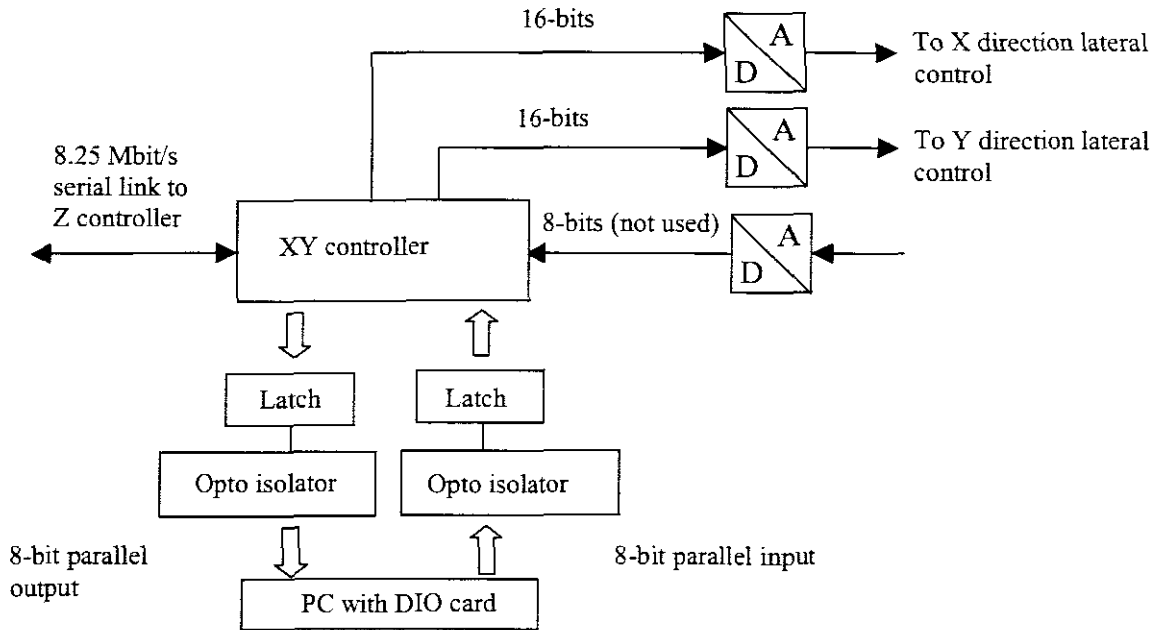


Figure 6.2: XY controller block diagram, showing the 16-bit X and Y DAC outputs which generates the analogue voltages for the XYZ to four-quadrant converter. From the XY controller the data is sent to a latch and then to the PC via a parallel link that is implemented with a PC-14 card in the PC. The serial link to the Z controller can also be seen.

6.2 Communications Bus

The communications bus consists of three separate parts. The first two are the optically isolated parallel ports that run to and from the PC. The third part consists of a synchronous serial link that runs between the two microprocessors.

The parallel links use a control line each to direct the flow of data. If the data comes from the parallel port of the PC, the PC will first place the data on the bus and then put the 'write' control line to the microcontroller 'LOW'. This control line is connected to one of the pins of PORTB on the PIC microcontroller. PORTB is configured to generate an interrupt if a change of state is detected on the pins. The software must scan the pins to determine the correct source of the interrupt. The PIC17c43 software activates the latches for the incoming data. This will cause the data to appear on the data bus where the microcontroller can read it.

After the data is read, the latch is closed again and the data bus goes to a floating or TRI-STATE condition. The microcontroller now uses the 'read' control line that is used for sending data to the PC to generate an acknowledge signal to the PC. The PC then asserts the 'write' control line high again to indicate that it has received the acknowledge signal. The microcontroller tests the control line coming from the PC until it sees it is back to a high condition. The 'acknowledge' signal is removed from the control line and the microcontroller ends the interrupt condition. A flow diagram for the writing of data sequence is shown below.

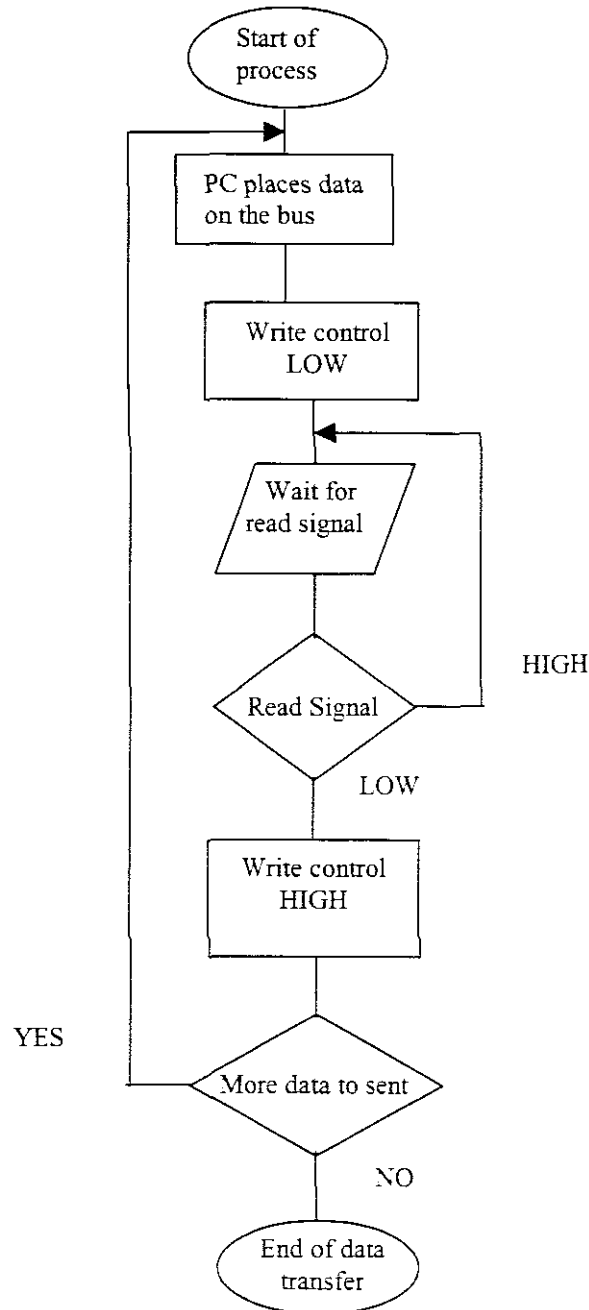


Figure 6.3: Flowchart of PC data sending sequence. Data is placed on the bus, write control line is asserted 'LOW' and it then waits for the PIC to respond by asserting the read control 'LOW'. The PC will then continue with normal program execution. The PIC will wait for the PC to assert the write control 'HIGH' before asserting the read control 'HIGH' again.

The speed requirement is that three bytes must be sent at 3 kHz. The parallel ports must be able to handle a throughput of at least 10kHz. This is fast enough even with some added signalling.

The serial port runs at 8.25 Mbits/s in a synchronous mode. The Z directional control board sends its three bytes at a frequency of 60kHz. To send the 24 bits takes $2.9\mu\text{s}$ ($24 \text{ bits} \div 8.25 \times 10^6 \text{ bits/s}$) which is about 17.7% of the time taken for each scan at 60 kHz assuming that the control loop is 135 instructions long. The Z control board is the master and the XY control board is the slave in this system. The serial bus is common between the 20-bit Burr-Brown DAC and the XY control board and is a two wire system which only allows a simplex data protocol. This means that one has to send data while the other one has to listen for data. The reason for making the Z controller the master in this system is the way the bus is driven when it comes to clock and data signals. If the XY controller was the master and was set to receive, the synchronous serial port will generate a continuous clock signal to synchronize any incoming data. This continuous 8.25MHz clock signal is a source of noise. It is better to have the Z controller as the master in transmit mode and the XY controller in slave receive mode. In this configuration, the clock signal is only generated when it is needed by the Z controller. If no data is transmitted, both clock and data output pins from the serial port are driven low by the microcontroller. This configuration helps to reduce noise in the sensitive Z controller.

The speed of the parallel data link to the PC is currently 12.3 kbytes/s which is fast enough for the current application. It would however be nice to have more bandwidth available to send additional information to the PC like the status of the embedded system. The speed of the synchronous serial port is also more than adequate. We need a speed of 60kHz times 3 bytes which is 180 kbits/s. It is however better to transmit the data as fast as possible to minimize noise in the Z controller.

6.3 Data Acquisition

The data acquisition system can be divided in two categories. The first is the acquisition system that is built onto the Z directional control board. It consists of two 8-bit ADC's and a 20-bit serial DAC from Burr-Brown. The XY control board has one 8-bit ADC and two 16-

bit Burr-Brown DACs. The ADC is a National Semiconductor ADC0820 IC in each case. The two 16-bit DACs have full bus control and are loaded in parallel, 16 bits at a time using a table write instruction from the PIC controller. The PIC microcontrollers are 8-bit controllers but have a 16-bit external data bus. At the moment, only the one ADC on the Z controller PCB is used.

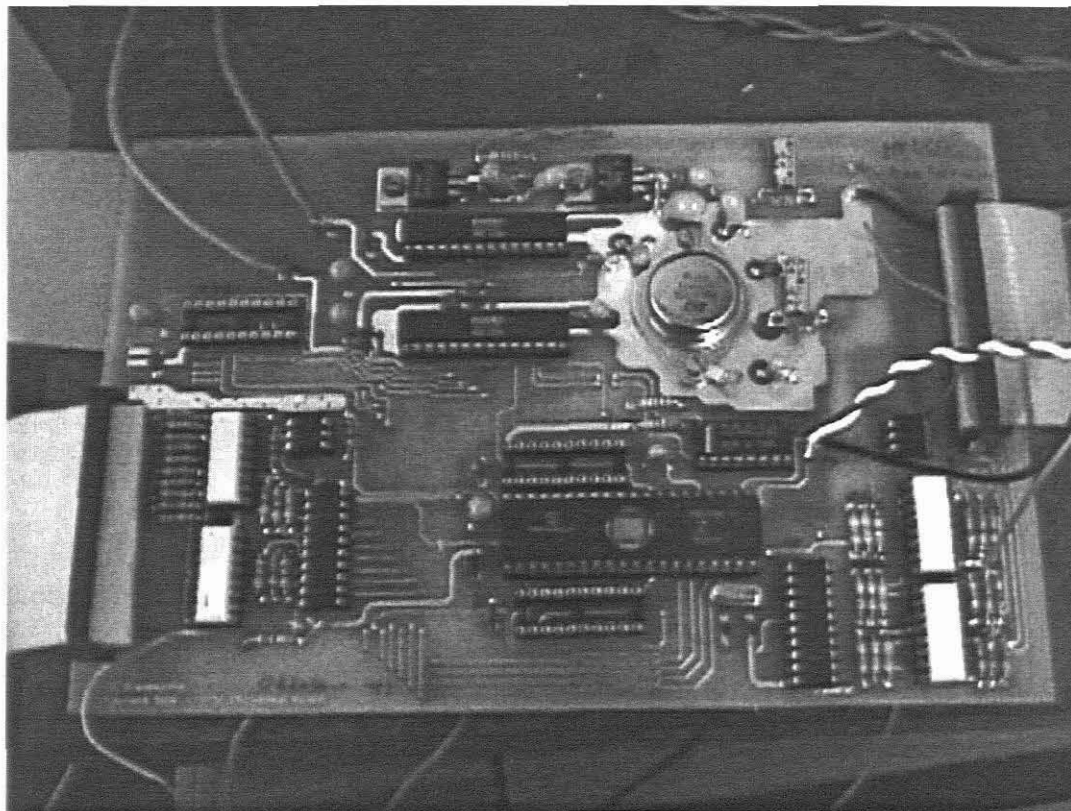


Figure 6.4: Photograph of the XY controller PCB, showing the two parallel links which runs to the PC-14 card in the PC (ribbon cable and the write opto-couplers), and the two 16-bit DACs in the top left corner which generate the X and Y control voltages.

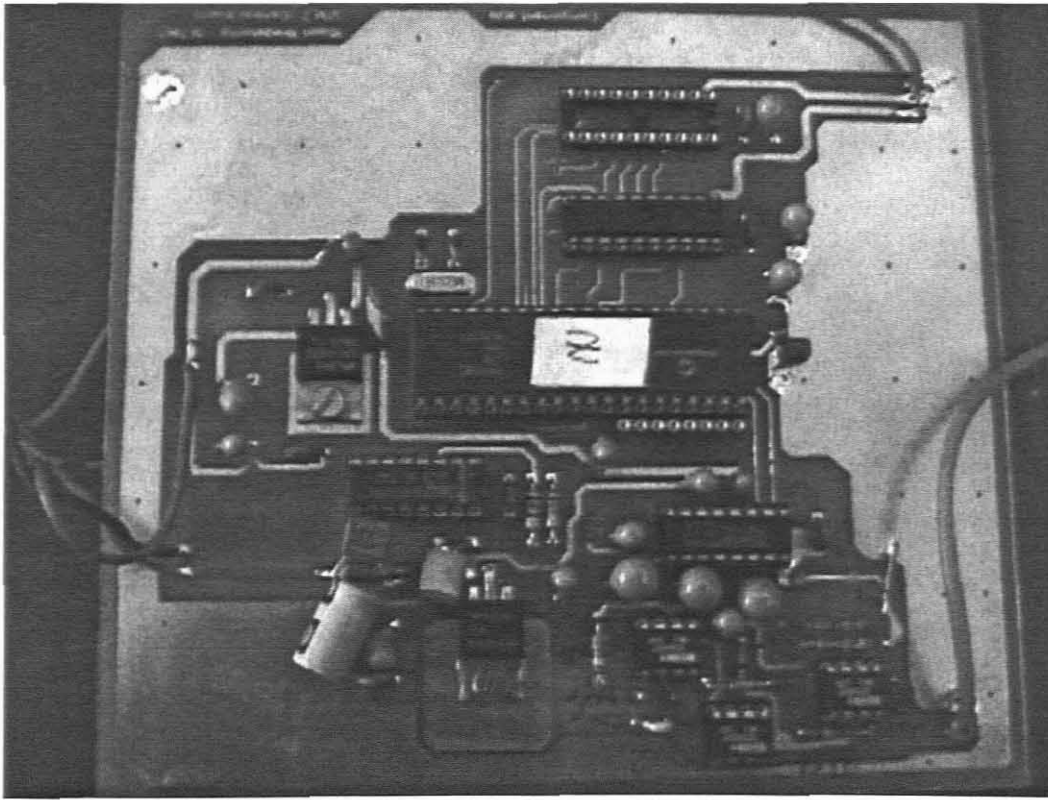


Figure 6.5: Photograph of the Z controller PCB, showing the 20-bit DAC with the LPF in the bottom right hand corner.

The input format of these devices is BTC (binary two's complement) code. Binary two's complement is a natural way for computers to do calculations using integer numbers. The values used in the microcontroller are all integer values, although the integer value might represent a current of only 1nA.

6.3.1 High gain amplifier and ADC input

The tunneling current has to be amplified from 1nA to give an input range for the ADC0820 of 0-5V. This is done by first converting the current to a voltage and amplifying it by a factor of 10^6 at the same time. It must then be amplified by a further factor of 10^3 in a separate voltage amplifier. The problem is that the tunneling current is very small. The operational amplifier (op-amp) will use a certain amount of current for the base drive of the transistors that it is made of in order to operate itself. This input current is in the order of 50pA for the LF356 op-amps that are used. It is essential to use a FET input based op-amp for the current to voltage converter to minimize the input current requirement. FETs generally use very little input current. The 10^3 amplifier has an additional offset adjust for when there is no input

current. If the input to the amplifier is 0V, then the output must be zero also. An offset adjustment potentiometer was added to ensure that a zero output voltage is obtained for a zero input.

6.3.2 Z controller 20-bit DAC

The 20-bit serial DAC requires the MSB first in its data stream. The binary value is swapped (mirrored) around on the Z control board by writing it to PORTC and reading it from PORTA. This is the fastest way to swap all 24-bits in the 3 bytes that are needed for the output. This is done for each byte. The PIC serial port will send 24 bits worth of data but the 20-bit DAC can be latched after the 20th bit. The first four bits in the serial data stream are bit numbers 23 to 20. The next 20 bits are the ones that the DAC needs to produce the output waveform. This means that when the latch signal goes low the last 20 bits received will be latched into the DAC. The DAC needs four extra clock pulses to produce the output waveform. These clock pulses can be obtained by waiting for the next 24 bit data stream and using the first four clock pulse to clock out the last analogue value. The problem with this approach is that the next clock pulses only occur 17 μ s later. It is better to send a 00h byte to the DAC and let it use those clock pulses to produce an immediate output. The total length of the serial data stream is thus 32-bits long. The downside of this method is that the extra 8-bits generate nearly 1 μ s of extra high frequency noise. The serial lines goes to the 20-bit DAC which is very noise sensitive. The latch can go high anytime after the second bit has been sent but has to go down on the 20th-bit. This is shown in figure 6.6 below.

The Z controller has a low pass Butterworth filter on the output after the trans-impedance amplifier that converts the current output of the 20-bit DAC to a voltage. This filter has a cut-off frequency of 60 kHz. The schematic diagrams for the Z controller and the output filter is shown in Appendix G and H.

6.3.3 PCB layout for low noise in a data acquisition environment

The PC board layout for the DACs needed special attention due to the high accuracy of these devices. A small thing like contact resistance becomes very important. To give an example, a 16-bit converter with a 20-volt full-scale range has a one LSB value of 305 μ V. With a load of 5 mA, series wiring and connecting resistance of only 60 m Ω will cause a voltage drop of

300 μV . The resistivity of a typical 1-ounce (28.349gram) copper-clad printed circuit board is 0.5 $\text{m}\Omega$ per square inch (3.2258 square millimeter). For a 5mA load, a 10 milli-inch (0.254mm) wide printed circuit conductor 60 milli-inches (1.524 millimeter) long will result in a voltage drop of 150 μV . This corresponds to a half LSB⁹¹.

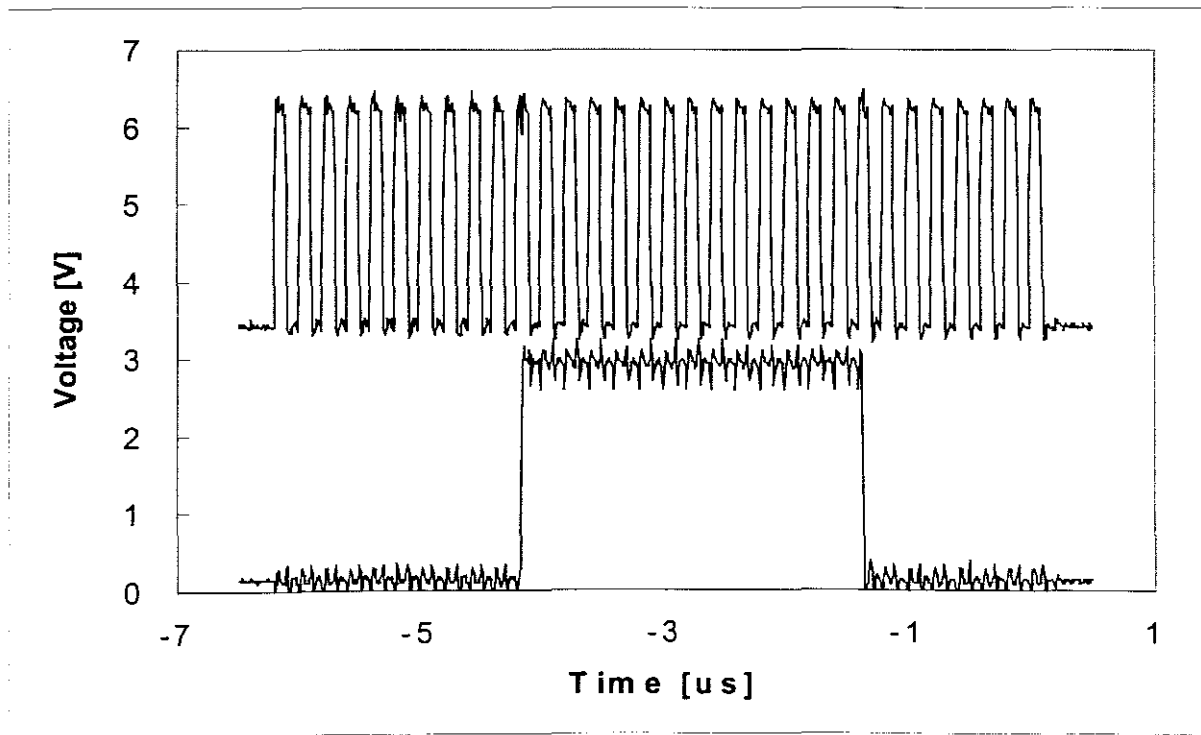


Figure 6.6: Graph of the latching signal on the Burr Brown PCM1702P 20-bit DAC output. These graph shows the latching signal(bottom graph) and the clock pulses(top graph). The extra '0h' byte that is sent to generate a clock for the output of the DAC can be seen clearly. These graphs were obtained by plotting the data points of the scope using MS-Excel. The top graph has been shifted up to make it stand out more clearly.

The power supply to the PCB must be regulated on the chip for the best results and lowest noise. The input to the regulators must have ferrite inductors to clamp incoming high frequency noise, and de-coupling capacitors must be added liberally around the power supply and close to the power supply pins of all the ICs. The best capacitors to use for optimum results are those with ceramic and tantalum dielectrics.

The input to the XY controller can also be supplied with very large electrolytic capacitors. This will act as energy reservoirs to supply the circuit with extra current incase the opto-couplers need it. A condition like this would arise if the output of the opto-couplers switches

from all 'LOW' to all 'HIGH'. This will generate a lot of switching noise on the ground lines, which is undesirable.

Wiring to high-resolution D/A converters should be routed to provide optimum isolation from sources of electro-magnetic interference (EMI). The key to eliminating RF radiation or pickup is small loop area. Signal leads and their return conductors should be kept close together so that they present a small capture area for any external field. The ground layout and the distinction between analogue and digital ground must be kept in mind when a PCB layout is designed. The digital ground has a lot more switching noise, which can be filtered using many ceramic capacitors. The respective data sheets of the ADC and DAC components will usually give some information as to which items might cause problems and what are possible solutions⁹¹.

6.4 Eagle Technology PC-14 card

The PC uses a plug in card with two Intel 8255 parallel peripheral interface chips to communicate with the control electronics. This PC-14 card is available at very little cost from Eagle Technology⁹². In the STM system, one 8255 chip on the PC-14 card is used to send data to the XY controller and the other is used to receive data from the controller. The card is very easy to install and use and is sold with a set of software drivers for Windows NT and Windows '95. It has some system and DLL files that are installed on the PC Windows directory. It has *include* files for both Borland Delphi and Visual C++ environments. The *system* files contain the device drivers and the *DLL* files contain the actual code that is referenced to in the header file. The device driver is the interface software between the operating system and the actual hardware of the computer.

6.5 Results

The results of the hardware design were mixed. Several faults were discovered during the testing procedure and corrective action was implemented to minimise the effect of these problems. Each one the potential problems and solutions is discussed below.

6.5.1 Timing problem

Each component of the hardware was tested and an unfortunate hardware-timing problem was discovered on the XY controller. The NOT write pulse that the PIC microcontroller generates during its 'table latch write' operation is only 30ns long. The Burr Brown DAC712 16-bit DAC requires minimum write pulse duration of 50ns. This problem can be overcome by lowering the clock frequency of both XY and Z controllers to 20MHz. This lowering of the clock frequency will solve the timing problem on the XY controller. If the Z controller is still run at 33MHz, the serial port speed will have to be lowered to match the maximum speed that the XY controller can receive at 20MHz. It was found that the serial bit error rates are only acceptable below 1Mbit/s. If the speed of the Z controller is also lowered to 20MHz to match the XY controller's clock frequency, the serial ports can run at 5 Mbit/s with no bit errors.

Additional circuitry could have been added to extend the pulse length to the DAC. This could have been implemented in various ways but it was decided to rather use a reduced clock speed as a quick fix. At the time when the mistake was discovered, it was also suspected that the XY controller could be overloaded with the many tasks it had to perform. It was thus unwise to try and do a new PCB layout with the associated costs involved to try and fix the timing problem. Instead the system was fully tested at the lower clock frequency to gain an understanding of the system for when the next iteration of the electronics is designed.

The reduced sampling frequency for the Z controller means that the control loop now runs at 42kHz for the unoptimized code. With complete optimized code, the scan can still be performed at 60kHz. Due to time constraints the code was not optimized. This will form part of the future work that will be done later by the author.

6.5.2 Opto-Couplers

It was found that the opto couplers for the XY controller were not fast enough to give good square shaped waveforms at 100kHz. The opto coupler used was the 4N32. This opto coupler uses a Darlington pair for an output stage. These opto couplers were changed for 4N25 devices, which use only a single BJT transistor for their output stage. Slightly better results were obtained with the 4N25. They were still not good enough because the waveforms began to lose the square shape around only 40kHz. A future design will have to use fast

digital opto isolators like the 6N137. The 6N137 can handle a data throughput of 10Mbit/s. This means that the waveform will still be perfectly square shaped at 10 MHz. At the moment the throughput that is obtained with the 4N25 devices is adequate because the embedded system is only transmitting the Z DAC values. A future system might want to transmit more data to the PC like the exact value of the tunneling current and system status information.

Again it was decided not to implement the new digital opto-couplers straight away but rather to wait and do all the corrections at the same time in a completely new PCB layout. The cost of producing a single double sided through hole plated board is in the region of R200,00 and it is therefore not a good idea to build new PCBs for every discovered mistake. The 4N32 and 4N25 devices are pin compatible but the 6N137 is not pin compatible with the existing PCB layout. The new PCB layout will also split the XY controller into three parts. This is described in more detail below.

6.5.3 PC-14 card

The parallel port communications bus formed a major bottleneck. The problem here is more software related than hardware related. The Windows NT system has too many security checks before the information gets down to the hardware level or from the hardware level, to the user level. The second part of this problem is that the Eagle Technology software is written to comply with a wide range of hardware and is not designed with high throughput speed in mind. The Eagle software is called as functions from the Visual C++ or Visual Basic software that the programmer uses. The only thing the user sees of the Eagle software is an include file with the function names in it.

The hardware was tested and a throughput of only 250 Hz was obtained with the Eagle EDR software under Windows NT 4. The throughput calculation is a bit arbitrary here because it is taken as the time taken for a low to high to low transition on the output of the card. This 250 Hz throughput figure shows up as a 125 Hz oscillation on an oscilloscope. It was found that if the debug information was removed from the Visual C++ program that the hardware response increased from 250 Hz to 300 Hz. The obtained throughput rate with the EDR software was unacceptable for the embedded-system-to-PC data link. The software support section of Eagle Technology said that this is an expected figure for the configuration that was used. The

configuration that is referred to here is the combination of Windows NT and the Eagle software. Eagle's software support said that the throughput would have been a lot faster under Windows '95. Windows NT was however chosen as the operating system for this study because it is a lot stabler than Windows '95.

At this stage, a cursory investigation was made of the PC-14 card and it was found that with the available hardware it should be possible to run the card at speeds in excess of 1MHz. This ruled the hardware out as a possible cause for the low throughput.

Windows NT uses a system of device drivers to communicate between different levels of the operating system. It is possible to write data straight to any memory location in a Windows '95 system. In Windows NT, the operating system will simply not allow this memory write operation without working through a device driver. Improving the throughput of the card required a new, purpose-specific device driver to be written. This type of driver requires expert programming. The author is grateful for the help received from Mr. Bernard Kuc in this part of the work. A device driver was developed to enable the end user to write directly to any memory location in a Windows NT system. The test software that was developed along with the device driver, proved that the throughput could be as high as 50kHz on a 166MHz MMX Pentium machine. If the port-write-routines for the Intel 8255 chips that make up the PC-14 card are hard-coded in the device driver itself, the throughput jumps up to 550 kHz. This 550 kHz shows up as a 275kHz oscillation on an oscilloscope. Hard-coding the port-read and write routines in the device driver means that the code is in the device driver and not in the user program. The data that the device driver collects during a data scan, can then be sent as an array up through the NT security layers from the device driver to the user level. The user only gets the results and does not control the actual read routine. Most of the software had to be rewritten to accommodate the new device driver based interface to the PC-14 card. This means that all the EDR software had to be removed.

On the 486 DX4-100 MHz machine that is used for the STM development the speed of the later method drops to 170 kHz due to the difference in hardware design between the two generations of PCs. The way that the 550 kHz throughput and the 170 kHz throughput on the PC were obtained is to just switch any port on the PC-14 card on and off as fast as possible. These numbers are arbitrary and are just used to learn more about the capability of the system under test.

It was found that the opto couplers deformed the output waveforms and that a full speed device-driver-based read was impossible to do. The PC-14 card will read in different values due to the deformation of the waveform. The problem was solved by doing several reads of the port and only taking the last read as a valid read. The throughput that was obtained was 12.3 kbytes/s with the handshaking signals between the PC and the embedded system included. The read sequence is delayed by a factor of three and the three bytes are then assembled in a double word. On a Pentium 166 MMX the speed should be 3.23 times faster at 40.4 kbytes/s. It must be noted that the 12.3 kbytes/s throughput on the existing system is enough and no more speed is really needed. It would however be nice to send other information to the PC also like the actual value of the tunneling voltage or the position of the tip in the X and Y space. These features are again not required for the operation of the STM but would be nice to have in a future iteration of the hardware and software.

The 3.23 times number is derived from previous testing and is an empirical number again. If the device driver is used to just switch a port on and off a throughput of 550kHz is observed. This drops to 170kHz on a 486DX4-100. Dividing the 170kHz into the 550kHz gives the 3.23 times faster number.

If the opto-couplers are replaced, the device driver can be recompiled to do fewer reads on the device driver level and give an even faster data throughput. The data is then just read once instead of the three times it is done now. This means that the 40.4 kbytes/s will jump up to 121.2 kbytes/s. The 121.4 kbytes/s throughput is an absolute maximum for the system. This figure is only a calculated figure and has not been tested.

6.5.4 Lateral response

The lateral response of the controller must be tested to determine whether the first sets of data are taken on the correct X and Y position, or whether the scanner tube was still moving to the new position. If it was still moving, the Z position value might be very low or very high giving the impression that there is a depression or a hill on the surface of the sample. It will also influence the average value if the samples are averaged over say 5 to 20 samples. If the scanner tube is still moving, the first few sets may be discarded. The amount to discard and

the amount of data sets to send to the PC is set during the initialization download to the XY board. It is set by the data format byte.

6.6 Recommendations

The best way to implement a STM control system in hardware would be to split the XY controller into more parts. This means that the XY DACs and the communications part must be split. Ideally, the X and Y controllers must also be placed separately.

The two DAC controllers must be run in microcontroller mode and not in extended microcontroller mode as it is done now. The extended microcontroller mode is one of four modes that the PIC17 series of microcontrollers can be operated in. In this mode the microcontroller uses the internal EPROM space for program memory but configures some of the port pins to operate as a data and address bus with basic control lines. The microcontroller then generates the 'read' and 'write' signals to operate additional memory or in the case of the XY controller, a DAC. The programmer has no control over the timing of these control lines. If the PIC17c43 is run in microcontroller mode, the microcontroller's pins can be controlled and it can write the required length pulse out to the DAC712 write pin. This will solve the XY controller's timing problem. With the DAC712 timing problem solved, the PIC micro controllers can be run at 33MHz again.

It will also place extra demands on the X and Y controllers if a separate image correction control algorithm must be implemented in the X and Y directions. For an image correction algorithm, it would be best if separate X and Y directional controllers were used. The image correction would be done using feedback control based on Leatt's work⁹⁴. The standard way to do image correction is to post-process the image data on the PC using standard image correction algorithms.

The slow opto-coupler can be replaced with special high speed opto isolators that use special waveform shaping circuits and that can handle up to 10 Mbits/s. An example of such a chip is the 6N137. If a controller board is constructed to handle just the communications interface, it can use such a chip to isolate it from the PC. The Z controller will send its serial data to this board and not to the XY DAC boards. This communications interface will then send the data to the PC. The communications interface can even do the bit reversal for the Z controller. It

receives the data in a MSB first format from the Z controller. The parallel port can be wired in reverse order so to swap the data around. This will make it unnecessary for the PC to do this in software.

In short, the XY controller must be split to form a high-speed interface and separate X and Y directional controllers.

The Z controller will need an extra delay loop to slow the control loop down to exactly 60kHz. The capacity of the Z controller is also enough to implement a few additional improvements to the system. This could be something like giving the Z controller the capacity to work with fractions when it comes to implementing the p and i control.

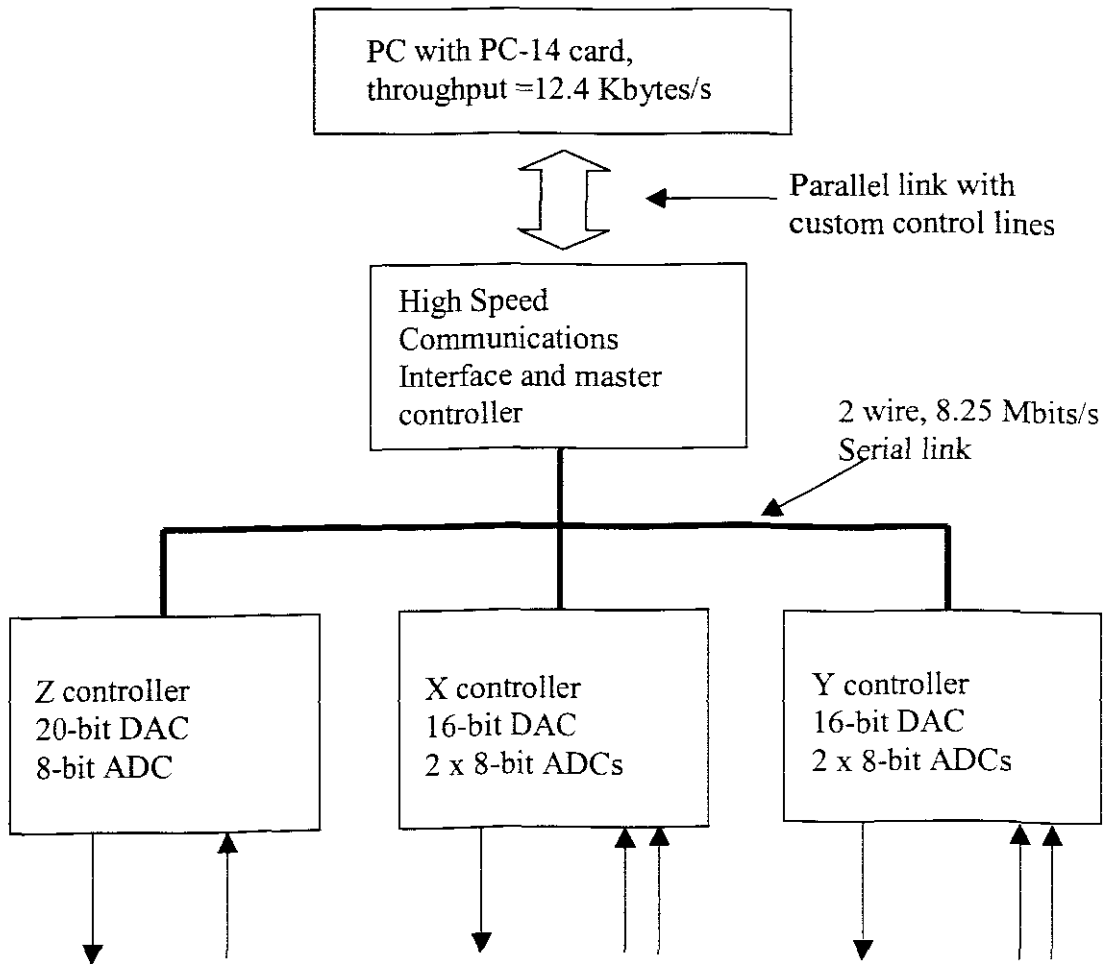


Figure 6.7: Block diagram of the new proposed system that will replace the existing XY controller. The system is split up into a high-speed interface that does the communications between the PC and the various sub-systems. The Z controller stays the same from the previous iteration but the X and Y control signals are now generated from two separate control boards. The X and Y control boards are supplied with two 8-bit ADCs each to implement a possible image correction algorithm or an STP function.

The next generation of the X, Y and Z directional controllers must be very carefully designed to minimize noise in the system. It would be best to construct the whole system inside a metal enclosure. The individual components can also be built inside their own metal enclosures and all power and signal connections can be made via shielded cable. An example of this would be to enclose the sensitive DAC output and filter stage of the Z controller in a separate metal enclosure that is mounted on the PC board.

This forms part of the future work that the author will attempt as part of a MSc degree in 1999.

Chapter 7: Software description

7.1 Introduction

The software for the STM is the most important part of the instrument. If the hardware is made very general in nature with lots of spare capacity, it can accommodate many software upgrades and even a minor hardware addition. The software for this project was written in assembler for the PIC and Microsoft Visual C++ 5 for the PC software. The PC software was developed under Windows NT 4 on a 486DX4-100 machine.

7.2 Z-control algorithm

The Z directional control algorithm was implemented in assembler on a Microchip PIC17c43 processor. The control loop is free running at a speed of 60 kHz and is a single input single output (SISO) digital PI loop. The control equation (2) is very simple and can be implemented on a microprocessor.

In the previous STM design, the Z control calculations were done on the PC before the sample was scanned. The control values were then stored as a look-up table on the Z control processor. The problem with this setup was that if the sample changed the control lookup table had to be changed. In practice, this was not always the case. The same lookup table was sometimes used for different samples because changing the lookup table was very difficult. The advantage of the lookup table was that execution of the control loop was still very fast at about 55kHz if using the Intel 8051 type processors. It is however more desirable to calculate the value each time the control loop is executed. This eliminates the need to reload the lookup table, because none is used.

The reason why the Z control loop runs at 60kHz has been discussed in Chapter 5. (See figure 5.3). The inputs for the control loop are the 8-bit ADC tunneling current value, and the previous value of the 20-bit DAC. These two values are then used to calculate a new DAC value that is used to position the scanner at the correct height above the sample. The software transmits the 20-bit result to the XY control board as soon as the loop is finished.

The final result is then placed on the serial data bus that is common to the DAC and the XY control board. The control loop then assigns the new DAC value to the old DAC value and starts the control loop again.

7.3 XY control algorithm

The XY control loop simply steps through a set of values as it receives the information from the Z control board. This stepping function is done to generate a raster pattern. After the initialization is done and the Z control board is taken from its reset condition the X and Y output values are placed on the DACs and the scanner tube is driven to its correct position. It stays there until it has received the correct number of data sets from the Z control board. It then moves on until it reaches the end of the line, where it flies back to start a new line.

The exact process is a little more complicated. The scanner is placed at a point say (a;a), where it will stay while the Z control board sends the 20 data sets. Depending on the setup that was fixed during the initialization, the microcontroller will do one of several things. It will send all 20 data sets to the PC for analysis, or it will average the set of 20 sets and send only the average value. Other options such as sending every fifth set, or ignoring the first five sets, are also available. The more sets that are sent to the PC in the allocated time frame, the slower the scan will be. The fastest scan with a cycle time of 3 kHz is obtained when only the very last data set is sent to the PC. After it has sent the last data set to the PC, the raster position is incremented. The scanner is placed at (a;a+1) and the whole process repeats itself. This continues until the scanner reaches (a;a+511) where the scanner is advanced to (a+1;a) and the whole process keeps going until the final point is reached at (a+511;a+511). The raster area is normally 512 by 512 data points, but the microcontroller makes provision for the scan size to be a rectangle also. It is thus possible to scan a thin strip in the horizontal or vertical direction. The magnification is a factor between 1 and 128 where 1 gives the largest magnification and the smallest area. The maximum scan area is divided into 65536 steps. The 512 by 512 scan point image is a subset of this 65535-point matrix. By adding the magnification factor to each scan point, the distance that the tip is moved is determined.

The PC software will take care of the factoring automatically. The operator will just select one of several scan area sizes to work with. The operator must enter the step size to give the required magnification into the PC program. A future version of the software will use the

mouse to select the segment of the image that is of interest on the PC screen and start the scanning with the click of a button.

7.4 PC software

The PC software was written in Microsoft Visual C++ version 5. The PC application was compiled for speed of execution in a 32-bit format. The Visual C++ option was also set to give an executable that was split into separate code and Dynamic Link Library (DLL) files. This makes the separate files smaller and more portable.

Windows messaging is how MS-Windows communicates between different classes. The two main classes in Visual C++ are the document class and the view class. All data processing must be done in the document class. To display any data on the screen in text or graphic format, the data must be sent from the document class to the view class for display. This is done using Windows messaging. This structure is part of the object-orientated method of programming.

7.4.1 Matlab routines

All Matlab routines can be converted into C routines using conversion software that is readily available. This makes them ideal to add into the STM software to do special functions like taking an FFT of the image or to convert the raw data into a graphic format like TIF or JPEG. The Matlab routines will only add functionality to the system but do not influence the operation of the system in any way. Adding the extra functions in the form of Matlab routines was an extra specification of the thesis but was left out because of time constraints. It will also be part of the future work that will be attempted later.

7.4.2 Graphic routines

The display format for the graphics is a line scan method. Each line that is scanned is displayed on the screen as a line at position (x,y) and as height z. To get a three-dimensional image the y position is incremented from the top to the bottom of the screen. The problem with this approach is that the individual lines have to be spaced slightly apart (3 pixels) to see

the information that is presented in the previous line. This could result in the generation of image artifacts like the display of a thicker line next to a thin line.

One possible solution to this would be to generate a colour-coded image of the specimen surface. The higher the surface feature the more blue the coding and the lower the feature the more red the coding of the pixel.

The image itself is not saved, only the array of Z values that the embedded controller sends to PC. To load any image from the hard disk, the array is loaded with the data and the graphic routine is called to display the image.

7.4.3 Scan parameters

The scan parameters are loaded from the PC to the embedded system before each scan. The parameters that are loaded are the initial coordinates of the scan, and then the magnification of the scan. The user will only select the magnification in terms of a scan size between 10 nm x 10 nm and 5 μm x 5 μm . The software will select the correct step size to send to the XY control microcontroller. The p and k control constants are also loaded, along with the end position of the scan in the x and y direction.

7.4.4 EDR software

The EDR software kit from Eagle Technology is used to control the PC-14 card. The PC-14 card handles the input and output from the PC. Eagle has developed the software to be used in a Windows NT or 95 environment and for both Borland Delphi and Visual C++. The development engineer selects the correct software platform and integrates the EDR software into his software. All the software that Eagle Technology has developed and made available are general-purpose software and is not very fast.

Under a Windows NT environment there are four files to install under the WinNT directory, of which two go under the System32 directory. These files contain the EDR software routines that are declared in a C++ header file. The DLL and SYS files contain the code for the control of several different types of Eagle Technology cards. The only EDR software that the programmer uses is the C header file. It contains all the error codes; all the memory and

string functions, and the device access control functions. It is very easy to use and there are many software examples for each Eagle card.

Under Windows NT the EDR installation program puts an icon in the control panel. This is the shortcut to the EDR applet that controls the plug in cards. Each installed card is auto detected or manually selected. The base address is also selected in memory, along with the interrupt vectors and DMA channels. It must be noted at this stage that the PC-14 card does not use interrupts or DMA. As each card is installed it is allocated a board number or board handle which the programmer can use to select different cards in a multi-card environment.

Special care must be taken with the position in memory at which the EDR interrupt vector table is loaded. The user must find a seven byte open space between 200 and 340 (hex) in the PC memory map and set the EDR base address there. With the aid of the EDR software and a suitable plugin card it is very easy to write a program that will communicate to the external embedded system.

7.4.5 Control toolkit software

Warren Carew as part of his Ph.D project wrote the Control Toolkit in control systems⁹³. The toolkit is basically a set of Visual C++ routines to simplify the writing of software in a control engineering environment. It includes everything from routines to initialise real-time threads, to graphic routines and data storage routines. The most important routine that was used from the kit was the routine to initialise a thread to run the control loop.

A thread is a program segment that runs in parallel with other program segments. The code segments only appear to run in parallel, but in reality, there are not multiple processors to do the work but only one. The one available CPU uses time slicing to give every thread a turn. Under Windows NT, each thread gets a maximum of 20 milliseconds processing time from the main CPU before it is paused to give the next program some processing time. If a process or thread does not require any CPU time it is skipped and the next one gets its share of time. If a process utilises only half of its allotted time, the CPU will start to process the next thread. Theoretically any number of threads can be started but in practice it is only a couple of hundred.

Each thread uses up memory space and the more threads running, the slower the system response is because there is more code that is waiting to be executed. The number of threads can vary from one to a couple of hundred. If the code segment of a thread is very short, many copies can be made in the available PC memory. If the code segment is very complex and long, the thread might not complete the complete segment of code in its allocated 20 ms time slot. This means that if there are many threads running trying to finish their assigned tasks, then the PC will struggle. The number of threads also depends on the priority that is assigned to a thread. If its priority level is very high, it will be called very often and the other threads or system resources will have to wait. In a control environment there is usually only one very high priority thread that does all the computations.

The PC software for this thesis use only one thread from which the STM is controlled.

With the control toolkit, it was easy to implement a separate thread to run the control loop. If it had not been done in this way, the CPU would have been stuck in one loop and would not have had the time to service other parts of the STM application if a mouse button was clicked. The PC uses only one thread to run the control loop. The rest of the software will appear as a separate thread.

7.4.6 User friendliness

The software of the previous STM had to be changed every time it was used. Scan parameters like scan speed had to be changed in the source code. The source code was written in C++, which made it relatively easy to work with the STM, if the operator was a programmer.

This visual environment makes it much easier for the user to interact with the new STM. Things like file handling, graphic editing and system scan parameters are changed using windows. On the previous system the raw data had to be moved to a workstation to post process the image. On the new system, all the functions are available by mouse-accessible software.

7.4.7 Special device drivers

The Eagle technology software is developed for general-purpose use and is not very fast because of that requirement. As stated in Chapter 6, a contract programmer was employed to develop a special set of software tools for the use of the STM in general. These routines were developed to be fast and to aid in the data download from the embedded system to the PC.

The contractor also developed the device driver in Visual C++. It was made available as two C++ classes called CDevice and CDriver and a file called 'STMdev.sys', which is the device driver. The files are all stored in the directory from which the software is run. It is not necessary to place any files in the Windows System directory. The special STM device driver is working satisfactory and was tested completely.

7.5 Software verification procedure

The Z control board was tested using a sine and square wave at the input to the Z directional control loop. The input voltage was varied between 0V and 5 V. 0V indicates a total absence of a tunneling current and 5V indicates a tip crash. A tip crash occurs when the tip actually touches the sample surface. The control loop will try to keep the input voltage at a level of 1V, as was discussed in Chapter 5.

When a square wave is used as input, the time response of the loop can be measured in response to the step input. With the sine wave, the ability of the control loop to track a changing surface can be measured. Both will give important information for the characterization of the loop.

The Z controller was first tested without the integral controller term. The integral controller will just charge up to some arbitrary value and do nothing further. Therefore it was only used once the loop is controlling the STM and not during test input.

The XY microcontroller software was verified using a special emulator that is available from Microchip⁸⁹. It is called the PicMaster and has a head that plugs into the socket where the processor normally sits. It simulates the real processor, but it can be single stepped and a signal trace can be obtained at any stage from any pin on the probe head. It is a very versatile

tool and makes the verification process of the microcontroller software very easy. To verify that the correct voltage levels are available from the DAC outputs, a digital storage scope was used.

The PC software was more difficult to test. The main problem is the absence of test data to test with. The embedded system that is installed outside could supply some data but the XY controller is programmed to wait for a complete data set download before incrementing its own position. This means that it will wait until all three values in a set are downloaded to the PC. The XY controller will however keep servicing its own interrupt structure. These interrupts are generated when data is received from the Z controller. The best way to test the PC software in the absence of data is to make extensive use of the IDE debugger system that is supplied with Visual C++. Using the debugger, it is possible to simulate errors that might occur in the software and to test some of the basic functions of the device driver.

7.6 Results

The results for each section will be considered separately. The software was the most important section of this thesis and it took the longest time to complete. The more complex the software, the longer it took to get it working properly.

7.6.1 The Z controller

The Z control loop is only 119 instructions long and the result is that it is very easy to program and test. The process still took a reasonable amount of time because the software development environment was unfamiliar. Once the IDE was understood, the development was easy, and program bugs were easily found.

The main problem that was found with the Z directional controller software was when the input clock frequency was lowered to 20MHz from 33MHz. The reason for the lowering of the clock frequency was due to a hardware-timing problem and is explained in Chapter 6. The problem is that the execution speed dropped to 42 kHz from 69 kHz. The software will have to be optimized now to compensate for the lower execution speed if the XY controller is going to be used continuously at the lower clock frequency of 20 MHz. The optimization can be done by changing the code to a straight-line version. This means that there will be no

function calls made. All the “goto” instructions are removed and the ADC0820 timed read is reduced from 2.5 μ s to 1.5 μ s. This will bring the Z loop time back up to 60kHz even with a 20MHz input clock. The other option was the use a system of lookup tables as was done previously¹². This would also have given the required 60kHz bandwidth at the lower clock frequency.

The optimization has not yet been done at the time of writing this thesis because it was decided to rather solve the problem by designing a totally new system with the associated PCB layout by splitting the existing XY controller into three separate parts as was described in chapter 6 in the next iteration. The splitting of the XY controller will have the indirect effect of bringing the clock speed of the Z controller back up to 33 MHz and thus the control loop execution speed back up to 69kHz.

7.6.2 The XY controller

The amount of data that the XY controller has to handle, coupled with the amount of signaling in the 3 kHz time frames is the downfall of the XY controller. The XY controller has to do the X and Y code generation for the raster pattern and write it out to the 16-bit DACs. It has to receive 20 data sets of three bytes each from the Z controller, which have to be kept or discarded. The 20th set of three bytes must be transmitted to the PC with the communications protocol that is described in Chapter 6.

It was found that the existing XY controller slows the whole system because it has to do too many tasks. The XY controller should be split into three separate components, namely separate X and Y controllers and a high-speed interface that controls them and does the complete PC interfacing. The full hardware solution is described in Chapter 6.

7.6.3 The PC software

The PC software is working satisfactorily and achieved the goal of providing a user-friendly environment for viewing the data and using the STM with ease. It also provided easy storage and retrieval of data, and the facility for easy upgrading of the PC software. This is done means of object orientated programming.

7.8 Recommendations

Several problems can be solved with the next version of the STM software. Both the embedded software for the Z and XY controllers, and the PC can be improved. The main software upgrades are described below.

7.8.1 Embedded system

The PIC microcontroller software was written to produce software modules that can easily be used to create new software for the STM embedded system. If the XY controller is split into three separate parts as was described in Chapter 6, the software can easily be developed for each relevant controller. The relevant software modules can be copied into the new programs and assembled. It is suggested that the X and Y control is done on separate microcontrollers. The X, Y and Z controllers will then be controlled by a master controller that will also do the interfacing to the PC.

The communications interface can be equipped with a circular buffer to hold the incoming data from the Z controller. It will keep sending the data from the buffer unless it has some other housekeeping tasks to do.

7.8.2 PC software updates

The PC software has only the very minimum of functionality. The online help file system must still be constructed along with a better system for setting the various variables of the STM. It is possible to place menu options that are used a lot on the shortcut menu list, or to make the menu available with a click of the secondary mouse button.

More functionality can be added by using the functions that are available via Matlab. These would include converting the image from raw data to a TIF or JPEG formatted image. A colouring scheme can also be added to show differing heights. By means of the Matlab functions, image-processing routines could be added to give functions like contrast adjusting, brightness adjusting, and FFT functions to filter the image. It is also highly recommended that the line-scan type image be transformed into a relief type image. This type of image will show the surface topography better.

Presently, the embedded system has to be reset by hand before each scan. The PC-14 card can be used to reset the embedded system from the PC with a software command. One of the control lines on the PC-14 card can be tied to the reset pin of the XY controller to control it.

It is also possible to program the embedded system to restart itself without a reset by just forcing the controller to wait for a download when it has finished scanning the required number of data points.

The bottom line with the PC software is that it needs a lot more functionality and the way the information is presented must be improved.

Chapter 8: Fault diagnostics on the STM

8.1 Introduction

This chapter is added to give any future user of the system some idea of what must be tested and what kind of output waveforms must be obtained if the system develops a fault condition. The level of complexity of this system makes this a requirement. This chapter does not give any results of the system, it only explains what must be checked to determine which component is malfunctioning. The system has some basic diagnostic capability that is built in. More capability will be added later by the author during future work.

8.2 Device driver and fault

The device driver is the only part with built in software diagnostics. Because the device driver (STMdev.sys) is programmed in C, the complexity of the software can be very high. This allows for the creation of software counters to give time-out counts and conditions where the data that is read is not valid. The device driver expects the handshaking signals from the embedded system otherwise it will give a time-out. If a time-out occurred, the device driver will generate a popup message on the user screen telling the user that there is a fault condition and what kind of fault condition occurred. Examples of fault conditions are 'invalid I/O address', 'device timeout' and 'STMdev not loaded'. A total of 72 different error messages can be given by the device driver. Most of them are operating system related and has nothing to do with the operation of the STM but these error messages cannot be ignored because they can influence the functioning of the operating system. The device driver has been tested extensively and is stable. At the time of writing this thesis it gave virtually no problems.

8.3 Z controller

The Z controller has various sections that must be tested to make sure it is working properly. The most important thing is the serial link that runs between the PIC17c43 controller, the PCM1702P 20-bit DAC and the XY controller. This can be tested as a complete unit when a general communications test is performed. This is described later in this chapter.

The input to the Z controller board is read by the ADC0820. This part of the input circuit is design to take an input voltage range of 0 to 5V. It is important to check that it does not go outside of this range. This voltage range is generated by the 10^3 pre-amplifier stage. If the input is outside of this range, the amplifier stage must be tested.

8.4 XY controller

The XY controller has two main sections that must be tested. These are the two 16-bit DACs and the 'READ' and 'WRITE' handshaking signals that are part of the data download process to and from the PC.

The two 16-bit DACs are very difficult to test if they are running a scan. The X DAC produces a positive going ramp that cycles 512 times and the Y DAC is produces a single ramp during the whole scan. It is better to use a special test program to test that these DACs are working correctly. The special software is supplied in Appendix C. The software produces two sawtooth waveforms. The ramp slope for the X DAC is positive and for the Y DAC is negative. These signals should be about 20V peak to peak. The oscilloscope image is shown in figure 8.2. From the oscilloscope image, the time delay between the X and Y DAC write routines can be seen. This is part of the test software. The microcontroller clock frequency for this test was 20 MHz.

The next function to be tested is that the microcontroller receives and generates the correct 'WRITE' and 'READ' control signals. This is shown in figure 8.3. From this can be calculated the throughput of the system. In this case it is 12.3 kHz. This will change if the 20 MHz crystal is exchanged for a 33 MHz crystal in the next iteration of the system.

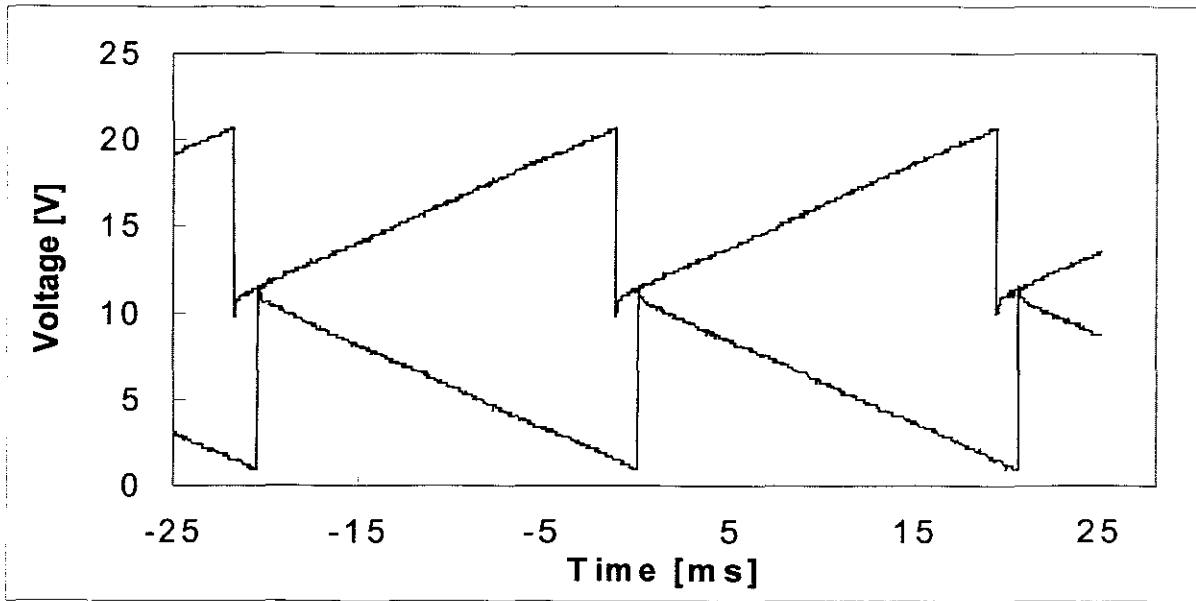


Figure 8.1: DAC waveforms that were produced by the test software. The top graph is the X DAC output and the bottom trace is the Y DAC output. The waveform data was obtained from the scope and plotted using MS-Excel and then shifted apart to make it stand out more clearly.

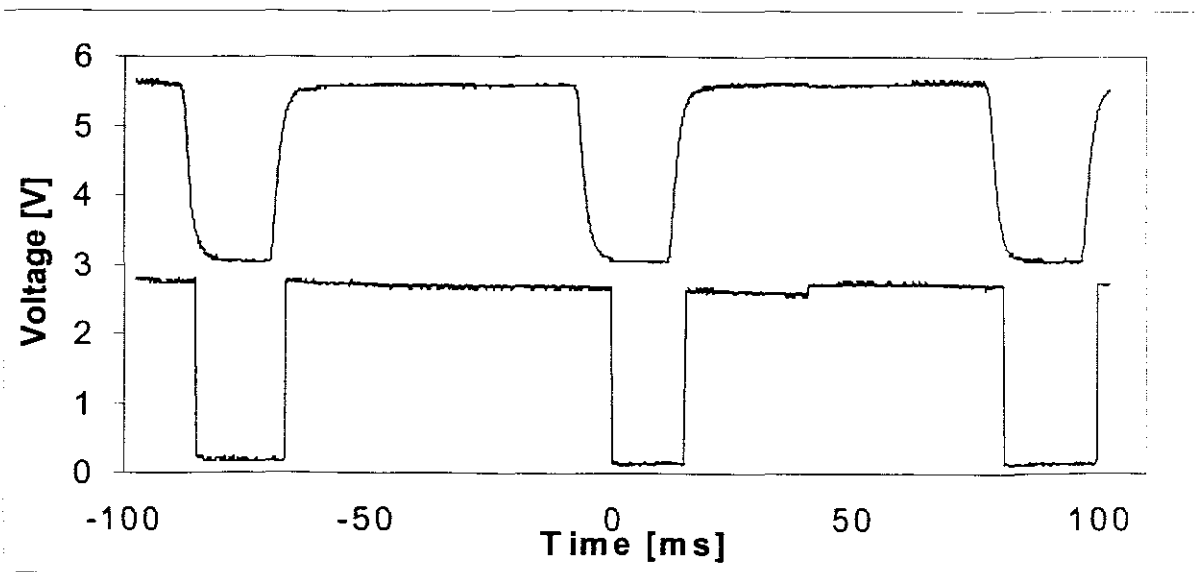


Figure 8.2: READ and WRITE control signal timing. It can be seen that the 'READ' control line is being deformed by the opto-coupler that is too slow. The bottom signal is the 'WRITE' control line that goes from the PIC to the opto-coupler and then to the PC. The waveform data was obtained from the scope and plotted using MS-Excel and then shifted apart to make it stand out more clearly.

8.5 The complete communications link

The complete system communications link can be tested by using one test lead and a piece of special test software. The listing for the special test software is given in Appendix E. The procedure to test the complete link is listed in Appendix F.

If the value of F80000 is produced, the Z output must look like the one in figure 8.4. If it is writing 7FFFF, the Z output must look like the one in figure 8.5.

This test is the best test to run on the STM communications system because it shows whether the control loop is working and whether all the communications links are in good health. It is a two-part test and both tests must be performed to ensure that everything is in order.

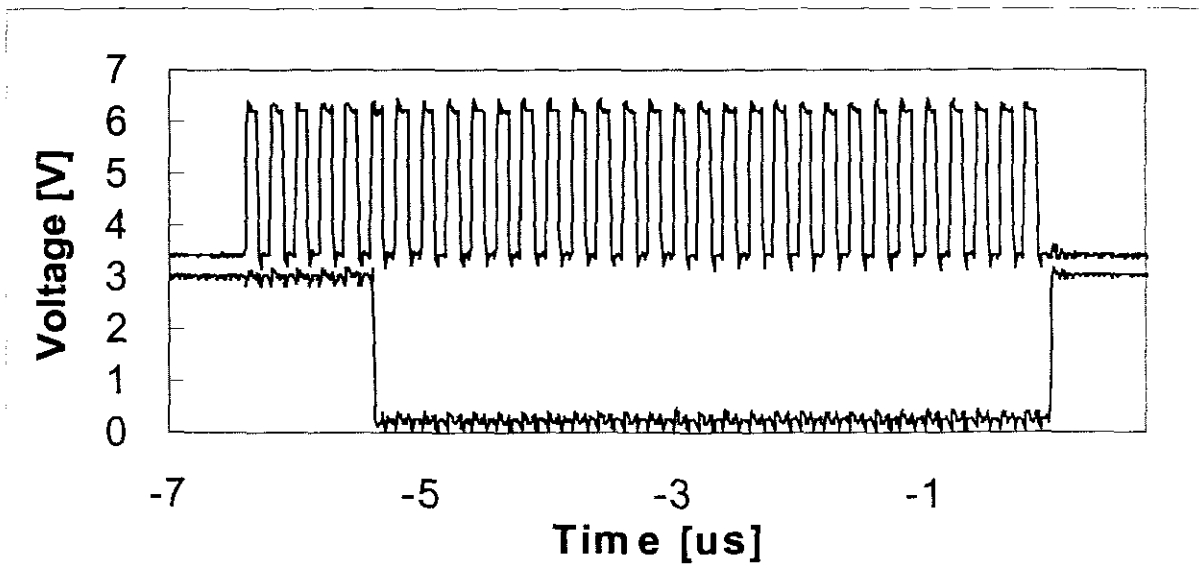


Figure 8.3: Z controller output signal showing a 'F80000' value. The bottom signal is the data signal and the top signal is the clock signal. The waveform data was plotted using MS-Excel and then shifted apart to make it stand out more clearly.

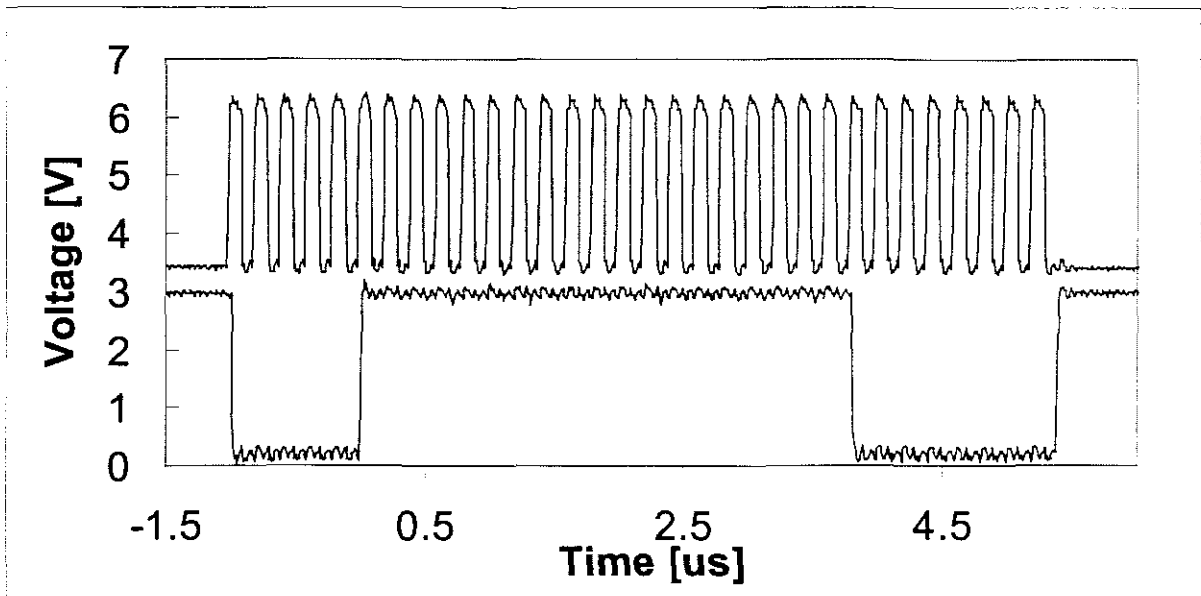


Figure 8.4: Output of the Z controller showing the value of '7FFFF'. The top signal is the clock signal and the bottom signal is the data signal. The waveform data was obtained from a scope and plotted using MS-Excel and then shifted apart to make it stand out more clearly.

Chapter 9: Results

The result for this thesis is very good in that it succeeded in producing a system with a potential for a superior throughput when compared to the previous system¹². A faster throughput of the data translates to the ability to control the rate of scanning at a faster rate and hence to an improvement in the feedback controller in the Z direction. This was done using cheap components and custom designed software. It was also proved that the embedded design does not have to use a DSP chip to do all the processing. A distributed system using several cheap microcontrollers works just as well and there is the added benefit of making the system modular. Any part can be removed and replaced with another part with a different capability. The schematic diagrams for the Z controller is included in Appendix G and H but the schematic diagrams for the XY controller is not included because the existing XY controller will be replaced by the system that is described in chapter 6.

The Z controller was implemented using a Microchip PIC17c43 microcontroller. This combination of a fast microcontroller and a good understanding of the software problem led to the design of a Z controller that was able to calculate the PI control loop in real-time. The loop executed at a frequency of 69 kHz and it is possible to increase the execution speed to an absolute maximum of 112.3 kHz.

A hardware error was discovered in the XY controller. It is a timing error that is related to the use of the two 16-bit DACs. The 'NOT WRITE' pulse is too short and was lengthened by lowering the clock frequency of both the XY and Z controllers. This is a quick fix to save on development cost. This means that the sampling frequency for the Z controller is now lower at 42 kHz and is a trade-off based on the available budget. It is currently slower than the lookup table based system that was used by Tapson¹² but it is calculated in realtime. The timing problem will be corrected in the new system design that is shown in figure 6.7 and described in chapter 6 during the next iteration of the STM electronics.

It was found that the opto-couplers for the XY controller were not fast enough to give good square shaped waveforms at 100kHz. These opto-couplers were changed for 4N25 devices, which use only a single BJT transistor for their output stage. Slightly better results were

obtained with the 4N25. They were still not good enough because the waveforms began to lose the square shape around 40kHz. The result of this was that the device driver which controls the PC-14 card in the PC has to do three separate reads of the data ports to make sure that the data on the ports is stable and not changing. The throughput with handshaking signals that was obtained with the PC-14 card was only 12.3 kHz. If the opto-couplers are changed for faster digital opto-couplers, the throughput speed on the XY controller to PC link will jump up to 40.4 kHz. Continued testing of the device driver and the signals that were obtained, helped in the understanding of the difference between a 486 PC bus and a Pentium PC bus. If the latter is used a maximum throughput of 121.4 kbytes/s can be obtained. Again the new opto-couplers will be implemented on PC board when the new system that is shown in figure 6.7 is implemented.

The use of a digital input/output card like the PC-14 card is definitely necessary in the design of a high-throughput system like this. The standard PC parallel port is just too restricted to handle the requirement of high speed bi-directional data transfer with control lines. The PC-14 card has six programmable input/output ports and a programmable timer on the card.

The software for the XY controller is working satisfactorily but at the moment it is only sending every 20th group of three bytes that is received from the Z controller to the PC. The XY controller is just fast enough to handle the data transfer to the PC and to handle the X and Y DAC operations and to receive the data from the Z controller. If more has to be done in the X and Y directions, the software tasks will have to be split into more parts. This is also true for the hardware of the XY controller.

The results for the PC software are also satisfactory. The PC software created a separate thread under Windows NT that controlled all the functions of the STM. The software used a device driver to control a PC-14 card that was used to handle the data transfers to and from the embedded system. The graphic user interface (GUI) also proved to be very useful in operating the STM.

A colour image of one of the first scans that was obtained is shown below. It can be seen that the STM produces a result in the form of a usable image and that the new feedback controller is thus working. The following is visible on the image, a tip switch (shifting of image along a horizontal plane), adjusting the tunneling current with the manual setscrew to get a stable

tunneling current (top left of the image), and a image artifact caused by the piezo tube itself. The tip switch is caused when the tunneling current shifts from being focused on one region of the tip to a totally different region. The only way to overcome this is to use very sharp etched tips. The tip in this study was sharpened by using a method of cutting with a sidecutter^{18,19,20}. This is a method that gives a reasonable tip if no time can be wasted on tip preparation. The region that is formed at the top left corner is caused by a tip that is out of tunneling range. It must be manually adjusted until a tunneling current is obtained. The lightly coloured band on the left side was only observed once the first image was obtained. It is caused when the piezo tube moves back to start the scan of a new line. It is still “flying back” to the start of the new line but the embedded system is already reading values from the location at the beginning of the next line. The result is that the digitized values are very “high” and that appears as orange to white on the false colour image in figure 9.1. This problem is not serious at all and can easily be rectified by adding a small delay in the XY controller software to force it to wait some milliseconds before starting to scan the next line after the tip is moved to the new location at the beginning of the next line.



Figure 9.1: STM scan of a gold alloy. The surface is flat and does not show a lot of surface features. A gold alloy surface was chosen because the focus is not on the surface topography but rather in any image artifacts caused by the scanning process. Some features of note in this image are the tip-switch which occurs a third of the way down the image; and the light band on the left hand side. The tip-switch shows where tunneling switched from one protrusion of the tip to another. The light band is a function of insufficient settling time after flyback.

Chapter 10: Recommendations

The recommendations for this thesis are concerned with the redesign of the XY controller. This controller must be split into three separate parts to remove the last communications and data processing bottle necks.

The first part must handle all the communications to and from the PC and it must act as the master controller of the embedded system. Freeing the communications microcontroller of the tasks that are related to the X and Y DACs will enable it to carry out more data transfers to the PC in a given timeslot. It can be equipped with a circular buffer to hold the data as it is received from the Z controller. The X and Y controllers will have lots of spare capacity to do additional functions such as image correction if they are implemented separately. The existing XY controller software was written as assembler modules and can easily be changed to develop the code for the three separate controller boards. A block diagram of the proposed system is shown in figure 6.7.

The PC software still needs a lot of work. Special attention must be given to the creation of the help files and the context sensitive help that goes with that. The user interface layout can be improved along with the way the data is displayed. At the moment it is displayed using a line type display but is proving to be impractical for the general use of the STM. A colour display using a false colour image will work better and will give a better idea of what the surface topography looks like.

Chapter 11: Future Work

The future course of this project will be determined by the results that are obtained from this thesis and the thesis of Leatt⁹⁴. The results of this thesis outline the shortcomings in the existing XY controller. As a first step, the XY controller will have to be split into smaller parts. These will most probably be a separate X and Y controller and a high-speed communications interface that will act as the master in a master slave situation with the XY and Z directional controllers for the STM scanner.

Leatt is working on the accurate measurement of the movement of a piezo ceramic tube or disk by means of capacitive measurement. The results that will be obtained will aid in the design of an image correction system where a control loop is used to correct the image as to the use of an image correction algorithm that is applied after the scan is completed. The results of Leatt will help to do the corrections in real time. A big help in these corrections will be if it was known what kind of image distortion was generated in the system. The easiest way to characterize this would be to get a calibration grid with a line spacing of 1 μm and to get STM images at various magnifications of the calibration grid. The image correction algorithm can be implemented on the proposed X and Y controller boards, which will be separate. Each will have two ADCs, with which the position value in the X and Y directions can be measured. The X and Y controllers can then calculate the correction value and move the piezo tube until the required position is reached.

The PC software requires more work. The main piece of software that must be added is to give the system more versatility. Examples of this would be to show each scan line as it is generated in a separate window. Other examples are the compression of the image into a usable size using say the JPEG algorithm.

The response of the scanner at the high speed must still be measured to obtain the best response possible. This must be done in the XY and Z directions. In the case of the Z directional controller, it is the p and k constants of the control loop that must be optimized for different scan sizes. Running the Z controller too fast could result in noise being generated and appearing as image artifacts on the scanned image.

The data display format that the PC is using can also be improved. Currently a line display is used but a false colour display will show the height information better. The way that this is done is to code the different heights with different colours.

One corner of the display can also be used to give the user a status display of what is happening with the STM system. This would be very useful if something is going wrong in the system. A fault diagnostics routine can be of great help here. A second window can also display the voltage that is applied to the input of the control loop. This will replace the oscilloscope that is currently used. These additional diagnostics and oscilloscope functions can be implemented on a fifth PIC17c43 based system. Instead of using the PC-14 card to send its data to the PC, it can use the serial port of the PC. Normally every PC has at least two serial ports available.

References

1. Binnig and H. Rohrer. The Scanning Tunneling Microscope. *Sci. Amer.*, 253(50):40-46, 1985.
2. Office of the Nobel Foundation, Box 5232, S-102 45, Stockholm, Sweden.
3. Russel Young, John Ward and Fredric Scire. The Topographiner : An Instrument for Measuring Surface Microtopography. *Rev. Sci. Instrum.*, 43(7):999-1011, 1972.
4. G. Binnig and H. Rohrer. Scanning Tunneling Microscopy. *Helv. Phys. Acta.*, 55:726,1982.
5. Ch. Gerber, G. Binnig, H. Fuchs, O. Marti, H. Rohrer. Scanning tunneling microscope combined with a scanning electron microscope. *Rev. Sci. Instrum.*, 57(2):221-224, 1986.
6. Robert D. Braun. Scanning Tunneling Microscopy of Silicon and Carbon. *J. Chem. Edu.*, 69(3):A90-A93, 1992.
7. NanoSurf AG, Austrasse 4, CH-4410 Liestal, Switzerland.
8. D.W. Pohl. Some design criteria in scanning tunneling microscopy. *IBM J. Res. Develop.*, 30(4):417-427, 1986.
9. C.D. Bugg and P.J. King. Scanning capacitance microscopy. *J. Phys. E: Sci. Instrum.*, 21:147-151, 1988.
10. P. Muralt and D.W. Pohl. Scanning tunneling potentiometry. *Appl. Phys. Lett.*, 48(8):514-516, 1986.
11. G. Binnig, C.F. Quate and Ch. Gerber. Atomic Force Microscope. *Phys. Rev. Lett.*, 56(9):930-933, 1986.
12. J. Tapson. A Wide-Area Scanning Tunneling Microscope Configured for Metallurgical Research. Ph.D thesis, University of Cape Town, 1994.
13. R.H.M. Groeneveld, Th. Rasing, L.M.F. Kaufmann, E. Smalbrugge, J.H. Wolter, M.R. Melloch and H. van Kempen. New optoelectronic tip design for ultrafast scanning tunneling microscopy. *J. Vac. Sci. Technol. B*, 14(3):861-863, 1996.
14. U.D. Keil, J.R. Jensen and J.M. Hvam. Transient measurements with an ultra fast scanning tunneling microscope. *Appl. Phys.*, A 66: S23-S26, 1998.
15. D.Botkin, S. Weiss, D.F. Ogletree, J. Beeman M. Salmeron and D.S. Chemla. Design considerations in an ultrafast scanning tunneling microscope. *Rev. Sci. Instrum.*, 66(8):1894-1897, 1995.
16. A.P. Stamp, G.C. McIntosh, and Xue-Wen Liu. Current characteristics for the scanning tunneling microscope. *J. Vac. Sci. Technol. B.*, 12(3): 2175-2178, 1994.

17. J.S. Villarrubia. Morphological estimation of tip geometry for scanned probe microscope. *Surf. Sci.*, 321:287-300, 1994.
18. V. Weinstein, M. Slutzky, A. Arenshtam, and E. B. Jacob. A method for the preparation of Pt-Ir tips for the scanning tunneling microscope. *Rev. Sci. Instrum.*, 66 (4):3075-3076, 1995.
19. A.I. Oliva, A. Romero G., and J.L. Peña, E. Anguiano and M. Aguilar. Electrochemical preparation of tungsten tips for a scanning tunneling microscope. *Rev. Sci. Instrum.*, 67 (5):1917-1921, 1996.
20. R. Kazinczi, E. Szöcs, E. Kálmán and P. Nagy. Novel methods for preparing EC STM tips. *Appl. Phys.*, A 66: S535-S538, 1998.
21. J. Becker. Scanning tunneling microscope computer automation. *Surf. Sci.*, 181:200-209, 1987.
22. A. Brown and R.W. Cline. A low cost, high performance imaging system for scanning tunneling microscopy. *Rev. Sci. Instrum.*, 61 (5):1484-1489, 1990.
23. A.H. Bredekamp and J. Tapson. A scanning tunneling microscope control system based on fast microcontrollers. In *COMSIG'98: IEEE Conf. on Communications and Signal processing*, P225-228, Cape Town, South Africa, September 1998.
24. J.W. Gerritsen, E.J.G. Boon, G. Janssens and H. van Kempen. Design of a simple high-resolution scanning tunneling microscope with an analogue generator. *Appl. Phys.*, A 66:S79-S82, 1998.
25. D. Jeon and R.F. Willis. Feedback system response in a scanning tunneling microscope. *Rev. Sci. Instrum.*, 62 (6): 1650-1651, 1991.
26. J. Valenzuela-Benavides. An X-Y scan generator for a scanning tunneling microscope. *Rev. Sci. Instrum.*, 65 (8): 2733-2734, 1994.
27. STM and AFM systems from TopoMetrix Corporation. 5403 Betsy Ross Drive, Santa Clara, CA 95054-1162, USA.
28. Oxford Instruments. Old Station Way, Eynsham, Witwney, Oxon, OX81 TL, UK.
29. T.M.H. Wong and M.E. Welland. A digital control system for scanning microscopy and atomic force microscopy. *Meas. Sci. Technol.*, 4: 270-280, 1992.
30. D.R. Baselt, S.M. Clark, M.G. Youngquist, C.F. Spence and J.H. Baldeschwieler. Digital signal processor control of control of scanned probe microscopes. *Rev. Sci. Instrum.*, 64(7):1874-1882, 1993.

31. F.M. Battiston, M. Bammerlin, Ch Loppacher, M. Guggisberg, R. Lüthi, E. Meyer, F. Eggimann and H.J. Güntherodt. Combined scanning tunneling and atomic force microscope with fuzzy controlled feedback. *Appl. Phys. A*, 66:S49-S53, 1998.
32. J. Tapson. Stochastic resonance and cooperative behaviour in scanning probe microscopy. *Appl. Phys. A*, 66:S17-S22, 1998.
33. K. Kajimura, H. Bando, K. Endo, W. Mizutani, H. Murakami, M. Okano, S. OkaYama, M. Ono, Y. Ono, H. Tokumoto, F. Sakai, K. Watanabe and S. Wakiyama. Construction of an STM and the observation of 2H-NbSe_2 atomic images. *Surf. Sci.*, 181:165-173, 1987.
34. C.W. Snyder and A.L. de Lozanne. Concentric tube tunneling microscope. *Rev. Sci. Instrum.*, 59 (4):541-544, 1988.
35. A.P. Fein, J.R. Kirtley, and R.M. Feenstra. Scanning tunneling microscopy for low temperature, high magnetic field, and spatially resolved spectroscopy. *Rev. Sci. Instrum.*, 58 (10):1806-1810, 1987.
36. B. Paillard and R. Tang and P. Rowntree. Digital linearization and cancellation of capacitive coupling for a scanning tunneling microscope. *Rev. Sci. Instrum.*, 69 (4):1770-1780, 1998.
37. A. Brown and R.W. Cline. A low cost, high performance imaging system for scanning tunneling microscopy. *Rev. Sci. Instrum.*, 61(5):1484-1489, 1990.
38. G.E. Poirier and J.M. White. Diffraction grating calibration of scanning tunneling microscope piezoscanners. *Rev. Sci. Instrum.*, 61(12):3917-3920, 1990.
39. H.M. Marchman and G.C. Wetsel, Jr. Optically guided large-nanostructure probe. *Rev. Sci. Instrum.*, 64(5):1248-1252, 1993.
40. K. Habib and A. Abdullah. Scanning tunneling microscopy of a magnetic shielding material in amorphous and in crystalline forms. *J. Mat. Sci. Lett.*, 9:1055-1057, 1990.
41. Microsoft Corporation. One Microsoft Way, Redmond, Washington 98052-6399 USA.
42. C. Barchesi, A. Cricenti, R. Generosi, C. Giammichele, M. Luce and M. Rinaldi. A flexible implementation of scanning probe microscopy utilizing a multifunction system linked to a PC-Pentium controller. *Rev. Sci. Instrum.*, 68 (10):3799-3802, 1997.
43. National Instruments. 11500 N. Mopac Expwy, Austin TX 78759-3504, USA.
44. G.W. Stupian and M.S. Leung. A scanning tunneling microscope based on a motorised micrometer. *Rev. Sci. Instrum.*, 60(2):181-185, 1989.
45. G.E. Poirier and J.M. White. A new ultra-high vacuum scanning tunneling microscope for surface studies. *Rev. Sci. Instrum.*, 60(2):3113-3119, 1989.

46. M.A. Dubson and J. Hwang. Simple, variable temperature, scanning tunneling microscope. *Rev. Sci. Instrum.*, 63(7):3643-3645, 1992.
47. G. Binnig and H. Rohrer. Scanning tunneling microscopy. *IBM J. Res. Develop.*, 30(4):355-369, 1987.
48. S.-I. Park and C.F. Quate. Scanning tunneling microscope. *Rev. Sci. Instrum.*, 58(11):2010-2017, 1987.
49. G. Binnig and D.P.E. Smith. Single-tube three dimensional scanner for scanning tunneling microscopy. *Rev. Sci. Instrum.*, 57(8):1688-1689, 1986.
50. Wide-area scanners from TopoMetrix use tripod designs for improved linearity. TopoMetrix Corporation. 5403 Betsy Ross Drive, Santa Clara, CA 95054-1162, USA.
51. R.C. Barrett and C.F. Quate. Optical scan-correction system applied to atomic force microscopy. *Rev. Sci. Instrum.*, 62(6):1393-1399, 1991.
52. Park Scientific Instruments produces a scanning system ("ScanMaster") with optical position feedback for linearity correction. Park Scientific Instruments, 1171 Borregas Ave. Sunnyvale California 94089.
53. G.L. Miller, E.R. Wagner, and T. Sleator. Resonant phase shift technique for the measurement of small changes in grounded capacitors. *Rev. Sci. Instrum.*, 61(4):1267-1272, 1990.
54. R. Nerino, G. Barbato, S. Desogus, and G.B. Picotto. Capacitive sensors coupled to an STM piezo-scanner for accurate measurements of the tip displacements. In *STM '93: Int. Conf. on STM*, Beijing, China, August 1993.
55. Vernitron Ltd. *Five Modern Piezoelectric Ceramics*.
56. D.P. DiLella, J.H. Wandass, R.J. Colton, and C.R.K. Marrian. Control systems for scanning tunneling microscopes with tube scanners. *Rev. Sci. Instrum.*, 60(6):997-1002, 1989.
57. F Besenbacher, E. Laegsgaard, K. Mortensen, U. Nielsen, and I Stensgaard. Compact, high-stability "thimble-size" scanning tunneling microscope. *Rev. Sci. Instrum.*, 59(7):1035-1038, 1988.
58. M.A. Player and G. Ping. A simple scanning tunneling microscope. *Meas. Sci. Technol.*, 3:1205-1210, 1992.
59. S. Grafström, J Kowalski, and R. Neumann. Design and detailed analysis of a scanning tunneling microscope. *Meas. Sci. Technol.*, 1:139-146, 1990.

60. M.M. Dovek, M.J. Heben, C.A. Lang, N.S. Lewis, and C.F. Quate. Design of a scanning tunneling microscope for electrochemical applications. *Rev. Sci. Instrum.*, 59(11):2333-2340, 1988.
61. S. Rubel, M. Trochet, and A.L. de Lozanne. STM-CVD nanofabrication with a UHV STM/SEM. In *STM '93: Int. Conf. On STM*, Beijing, China, August 1993.
62. S.A. Elrod, A.L. de Lozanne, and C.F. Quate. Low-temperature vacuum tunneling microscopy. *Appl. Phys. Lett.*, 45:1240-1242, 1984.
63. S.-I. Park and C.F. Quate. Theories of the feedback and vibration isolation systems for the scanning tunneling microscope. *Rev. Sci. Instrum.*, 58(11):2004-2009, 1987.
64. Inchworm linear motors from Burleigh Instruments, Burleigh Park, Fishers, NY 14453, USA.
65. G.C. Rosolen and M.E. Welland. A combined scanning electron microscope and scanning tunneling microscope for studying nanostructures. *Rev. Sci. Instrum.*, 63(9):4041-4045, 1992.
66. W.J. Kaiser and R.C. Jaklevic. Reliable and versatile scanning tunneling microscope. *Rev. Sci. Instrum.*, 59(4):537-540, 1988.
67. O. Züger, H.P. Ott, and U. Dürig. Variable-temperature ultrahigh vacuum scanning tunneling microscope: Mechanical and electronic instrumentation. *Rev. Sci. Instrum.*, 63(12):5634-5643, 1992.
68. K. Besocke. An easily operable scanning tunneling microscope. *Surf. Sci.*, 181:145-153, 1987.
69. J. Frohn, J.F. Wolf, K. Besocke, and M. Teske. Coarse tip adjustment and positioner for a scanning tunneling microscope. *Rev. Sci. Instrum.*, 60(6):1200-1201, 1989.
70. S. Pan, S. Behler, M. Bernasconi and H.-J. Güntherodt. A high vacuum low temperature STM with a novel coarse-approach mechanism. In *STM '93 Int. Conf. On STM*, Beijing, China, August 1993.
71. P. Mural, D.W. Pohl, and W. Denk. Wide-range, low-operating-voltage, bimorph STM: application as potentiometer. *IBM J. Res. Develop.* 30(5):443-450, 1986.
72. J.P. Pelz and R.H. Koch. Extremely low-noise potentiometry with a scanning tunneling microscope. *Rev. Sci. Instrum.*, 60(3):301-305, 1989.
73. J.R. Kirtley, S. Washburn and M.J. Brady. Direct Measurement of Potential Steps at Grain Boundaries in the Presence of Current Flow. *Phys. Rev. Lett.*, 60(15):1546-1549, 1988.
74. R. Whiddington. The Ultra-Micrometer: an application of the Thermionic Valve to the measurement of very small distances. *Phil. Mag.*, 14:634-639, 1920.

75. J. Tapson and J.R. Greene, Improved capacitance measurement by means of resonance locking. *Meas. Sci. Technol.*, 5:20-26, 1994.
76. D. Scholl, M.P. Everson, R.C. Jaklevic and W. Shen. Improved scanning tunneling microscope feedback for investigation of surfaces with micron-scale roughness. *Rev. Sci. Instrum.*, 62(9):4046-4048, 1992.
77. G.F. Franklin, J.D. Powell, and A Emani-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, Massachusetts, 1986.
78. U. Memmert, U. Hodel, and U. Hartmann. Combined ultrahigh vacuum tunneling microscope scanning electron microscope system. *Rev. Sci. Instrum.*, 67(6):2269-2273, 1996.
79. R. Curtis, T. Mitsui and E Ganz. An ultrahigh vacuum high speed scanning tunneling microscope. *Rev. Sci. Instrum.*, 68(7):2790-2796, 1997.
80. M. Bott, T. Michely and G. Comsa. Design principles of a variable temperature scanning tunneling microscope. *Rev. Sci. Instrum.*, 66(8):4135-4139, 1995.
81. V. Weinstein, A. Arenshtam, M. Slutzky and E. Ben-Jacob. STM system for spectroscopic and topographic studies at liquid-helium temperatures. *Rev. Sci. Instrum.*, 66(6):3580-3583, 1995.
82. A.R. Smith and C.K. Shih. New variable low-temperature scanning tunneling microscope for use in ultra-high vacuum. *Rev. Sci. Instrum.*, 66(3):2499-2503, 1995.
83. B. Drake, R. Sonnenfeld, J. Schneir, and P.K. Hansma. Tunneling microscope for operation in air or fluids. *Rev. Sci. Instrum.*, 57(3):441-445, 1986.
84. K.H. Kim, S. Eah, B. Lee, C.H. Cho and W. Jhe. A novel digital feedback scheme of shear-force control in near-field scanning optical microscope. *Rev. Sci. Instrum.*, 68(7):2783-2786, 1997.
85. Data Translation Inc., 100 Locke Drive, Marlboro, MA, 01752-1192, USA.
86. <http://www.ti.com/> is the web address for Texas Instruments. They are the manufacturers of a large range of DSP products.
87. Analog Devices, Corporate Headquarters, One Technology Way, P. O. Box 9106, Norwood, MA 02062-9106, USA.
88. <http://www.intel.com/> is the corporate website address for the Intel Corporation. They manufacture a large range of microcontrollers and microprocessors.
89. Microchip Technology Inc., 2355 West Chandler Blvd., Chandler, AZ 85224-6199, USA.
90. A. Paracherla. Implementation of Fast Fourier Transforms. *Microchip Embedded Control Handbook.*, 4:145-147, 1994/5.

91. Data acquisition products from the Burr-Brown Corporation, P. O. Box 11400, Tucson, AZ 85734-1400, USA.
92. Data acquisition PC cards from Eagle Technology, 31-35 Hout Street, PO Box 4376, Cape Town, 8000, South Africa.
93. W.D. Carew. Multivariable Control of a Nonlinear Process. Ph.D thesis, University of Cape Town, 1998.
94. M. Leatt, MSc thesis in preparation concerning the measurement of very small displacements by means of capacitive measurements. University of Cape Town, 1998.

Appendix A: Control loop software listing

This software listing is the code segment that is used in the control thread in Windows NT.

```
//-----  
UINT CESTMDoc::GetSamples()  
{  
  
    long counter,b=0,c=0,y_dist,x_dist;  
  
    ULONG data[256];    // array where data is store from XY controller  
    m_updateDisplay = false;  
  
    x_dist = def_xstopPos - def_xstartPos;  
    y_dist = def_ystopPos - def_ystartPos;  
  
    m_xCount = x_dist;  
    m_yCount = y_dist;  
  
    // Let any other thread complete it's task before  
    // starting the loop by passing zero to the Sleep instruction  
    Sleep(0);  
  
    /*      Since we are getting DWords in @ 1024 per shot, we need to  
           calc how many times we need to call ReadSTMBuffer1k. */  
    counter = (x_dist*y_dist); // 512*512 scan  
  
    while (ControlThread.Started() && !ControlThread.Finished()  
           && m_pointIndex <= counter)  
    {  
        /*      The ReadSTMBuffer1k routine does a read of 1024 bytes  
              of the XY controller. It assembles groups of three bytes  
              into a DWord space and makes it available as a 256 size  
              array called data.          */  
  
        Result = dev.ReadSTMBuffer1k(&data[0]);  
    }  
}
```

```

if (Result.GetLength() != 0)
{
    ::MessageBox(NULL, (LPCTSTR)Result, 0, 0);
    ControlThread.Terminate();
}

// Check for negative number and strip bits 21-24 off
for (c=0; c<256; c++)
{
    if (data[c] > 0x7ffff)
        data[c] -= 0x0f00000;
    m_convert[m_pointIndex++] = data[c];
}

// Read another 1024 bytes
Result = dev.ReadSTMBuffer1k(&data[0]);
if (Result.GetLength() != 0)
{
    ::MessageBox(NULL, (LPCTSTR)Result, 0, 0);
    ControlThread.Terminate();
}

// Check for negative number and then strip bits 21-24 off
// Otherwise just save into array
for (c=0; c<256; c++)
{
    if (data[c] > 0x7ffff)
        data[c] -= 0x0f00000;
    m_convert[m_pointIndex++] = data[c];
}

m_yCount +=2;
m_updateDisplay = true;
theApp.PostThreadMessage(ID_COMM_INVALIDATE,0,0);

Sleep(5);

```

```
} // end while
Sleep(10);
// Suspend thread until next run
ControlThread.Stop();

return 1;
}
```

Appendix B: PIC code for calculating Z_{out}

This code shows how to calculate a 24-bit result in 8-bit registers.

```

;-----
Zcalc      movfp  p,WREG          ; Ie * P ->
           mulwf  Ie             ; PRODH:PRODL
           movfp  PRODH,ram_h    ; Save
           movfp  PRODL,ram_l

           movfp  k,WREG        ; Ie * K ->
           mulwf  Ie             ; PRODH:PRODL
           movfp  PRODH,arg1H    ; Save
           movfp  PRODL,arg1L

           movfp  oldres0,WREG
           movwf  res0
           movfp  oldres1,WREG
           movwf  res1
           movfp  oldres2,WREG  ; Save old values for just incase.....
           movwf  res2

           movfp  Is,WREG       ; move Is into WREG
           cpfslt Ia            ; Ia < Is ?
           goto   plus          ; No !

;-----Procedure seperator for ease of reading-----
minus      ; Ie is NEGATIVE !
           movfp  ram_l,WREG    ; (Old Z) - (Ie * k) => oldres0:2
           subwf  oldres0,f     ; (20-bit)
           movfp  ram_h,WREG    ; I part of controller
           subwfb oldres1,f
           movfp  dummy,WREG
           subwfb oldres2,f

           movfp  arg1L,WREG    ; [Old Z - (Ie * k)] - (Ie * p)
           subwf  oldres0,f     ; => oldres0:2(20-bit)
           movfp  arg1H,WREG    ; P part of controller
           subwfb oldres1,f
           movfp  dummy,WREG
           subwfb oldres2,f

m_next     btfss  res2,3        ; test if below or above BPZ
           goto   calc_end      ; -> above Bi-Polar Zero
           movfp  oldres2,WREG  ; -> below
           cpfsgt test          ; test for rollover on 20th bit
           goto   calc_end1     ; ->no rollover
           goto   old_vall      ; ROLLOVER -> reload 080000h

;-----Procedure seperator for ease of reading-----
plus       ; Ie is POSITIVE !

```



```

        movfp  arg1L,WREG          ; (Old Z) + (Ie * k) => oldres0:2
        addwf  oldres0,f          ; (20-bit)
        movfp  arg1H,WREG          ; I part of controller
        addwfc oldres1,f
        movfp  dummy,WREG
        addwfc oldres2,f

        movfp  ram_1,WREG          ; (Ie * P) + [Old Z + (Ie * k)]
        addwf  oldres0,f          ; -> 20 bit result
        movfp  ram_h,WREG          ; P part of controller
        addwfc oldres1,f
        movfp  dummy,WREG
        addwfc oldres2,f

p_next   btfs   res2,3
        goto   calc_end
        movfp  oldres2,WREG
        cpfslt res3
        goto   calc_end1
        goto   old_val            ; ROLLOVER -> reload 07ffffh

;-----Procedure seperator for ease of reading-----
old_val   movlw  0ffh              ; Replace lower byte with full range
        movwf  oldres0            ; count
        movwf  oldres1            ; Restore previous values of oldres0:2
        movlw  07h
        movwf  oldres2

        return

old_val1  movlw  00h              ; Replace lower byte with full range
        movwf  oldres0            ; count
        movwf  oldres1            ; Restore previous values of oldres0:2
        movlw  0f8h
        movwf  oldres2

        return

;-----Procedure seperator for ease of reading-----
; This section is here to make the code equal length
calc_end  nop
        nop
        return

calc_end1 return

```

Appendix C: PIC17 code to test the X and Y DACs.

This code segment for the PIC17c43 will run both X and Y DACs through their full range of values. The X DAC will generate a positive going slope and the Y DAC a negative going slope. The reason for this is just to distinguish between the output waveforms of the two different DACs.

```
-----  
loop      call    DACup          ; Increment X DAC to give a positive ramp  
          call    DACdown       ; Decrement Y DAC to give a negative ramp  
          goto   loop  
  
-----  
DACdown  
          movlw   00h  
          movwf  PORTA          ; Select X DAC  
  
          movfp  yh_temp,WREG   ; Move upper part of 16-bit DAC value  
          tlwt   1,WREG         ; to table latch  
          movfp  yl_temp,WREG   ; Move lower part in WREG  
          tablwt 0,0,WREG       ; and write to program memory, non  
  
          movfp  step_size,WREG ; Step Size = magnification factor  
          subwf  yl_temp,f      ; used, just add to the stepping code  
          movfp  dummy,WREG     ; Add carry to increment  
          subwfb yh_temp,f  
  
          return  
  
-----  
DACup  
          movlw   04h  
          movfp  WREG,PORTA     ; Select Y DAC  
  
          movfp  xh_temp,WREG   ; Move upper part of 16-bit DAC value  
          tlwt   1,WREG         ; to table latch  
          movfp  xl_temp,WREG   ; Move lower part in WREG  
          tablwt 0,0,WREG       ; and write to program memory  
  
          movfp  step_size,WREG ; Step Size = magnification factor  
          addwf  xl_temp,f      ; used, just add to the stepping code
```

```
movfp    dummy,WREG    ; Add to increment
addwfc   xh_temp,f
return
```

;------

Appendix D: PIC17 code to operate the ADC0820 analogue to digital converter (ADC).

This code segment is the function call to read a value for the tunneling current into the PIC17c43 based Z controller. This code segment takes about 2.5 μ s to execute at 33 MHz.

```

;-----
GetIa    bcf      PORTB,5      ; Select CS for ADC1
         movlb   01h          ; Bank 1
         clrf   PORTD,1      ; Port latches cleared. Pins should
         clrf   PORTC,1      ; read 0
         movlw  0ffh         ; Must set Port D as input otherwise
         movwf  DDRD         ; any data on Port D will influence
                             ; a ADC read on port C

         bcf    PORTE,2      ; Write signal LOW
         nop
         nop
         nop
         nop
         nop
         bsf    PORTE,2      ; Write signal HIGH
         nop
         nop
         nop
         nop
         nop
         bcf    PORTE,1      ; (end of 600 ns wait after Write high)
                             ; Read signal LOW! , INT will go low
                             ; in response to Read going low !
         nop
         nop
                             ; Delay of 240 ns
         movfp  PORTC,WREG   ; Get data from bus
         movlb  00h
         movwf  Ia
         movlb  01h
         bsf    PORTE,1      ; Read signal HIGH
                             ; end of current read !
         movlw  00h          ; Must set Port D as output again
         movwf  DDRD

         movlb  00h          ; Select bank 0
         bsf    PORTB,5      ; DE-select ADC
         return
;-----

```

Appendix E: Test software to test the complete communications link.

```
void CSTMView::OnTest()
{
    CDriver drv;
    CString Result;

    // Note :
    // A Started service does not always die immediately. Sometimes if a service
    // does not want to start make sure that a previous copy is still not running.
    // One can check it in
    // Administrative tools
    // Windows NT Diagnostics
    // Services
    // Devices

    // Start the device running at the specified base address, and specified port count.
    drv.ExampleStart((ULONG) 0x00000330, (ULONG) 0x0000000c, "STMdev",
"STMdrv.sys");

    // Create an instance of the device
    CDevice dev;

    // Connect to the device driver, i.e. almost a init function
    Result = dev.Connect("STMdev");
    if (Result.GetLength() != 0)
    {
        ::MessageBox(NULL, (LPCTSTR)Result, 0, 0);
        return;
    }

    // Set STM mode
    dev.SetSTM();
    ULONG ar[256];
    for (int i = 0 ; i < 256; i++) ar[i] = 0;

    // Ask for a hard reset of the STM system !
    ::MessageBox(NULL, "Reset the STM !!", 0, 0);

    for (int j = 0; j < 1024; j++)
    {
        Result = dev.ReadSTMBuffer1k(&ar[0]);
        if (Result.GetLength() != 0) ::MessageBox(NULL, (LPCTSTR)Result, 0, 0);
    }
    CString Dat;
    CString Tmp;
```

```

Dat = "";

for (i = 0; i < 32; i++)
{
    for (int j = 0 ; j < 8; j++)
    {
        Tmp.Format("%x\t",ar[j+i*8]);
        Dat += Tmp;
    }
    Dat += "\n";
}
::MessageBox(NULL, (LPCTSTR)Dat, 0, 0);
Result = dev.ReadSTMBuffer1k(&ar[0]);
if (Result.GetLength() != 0) ::MessageBox(NULL, (LPCTSTR)Result, 0, 0);

//////////

for (long q =0; q <=256;q++)
{
    if (ar[q]>0x7ffff)
        ar[q] -= 0x0f00000;
}

//////////

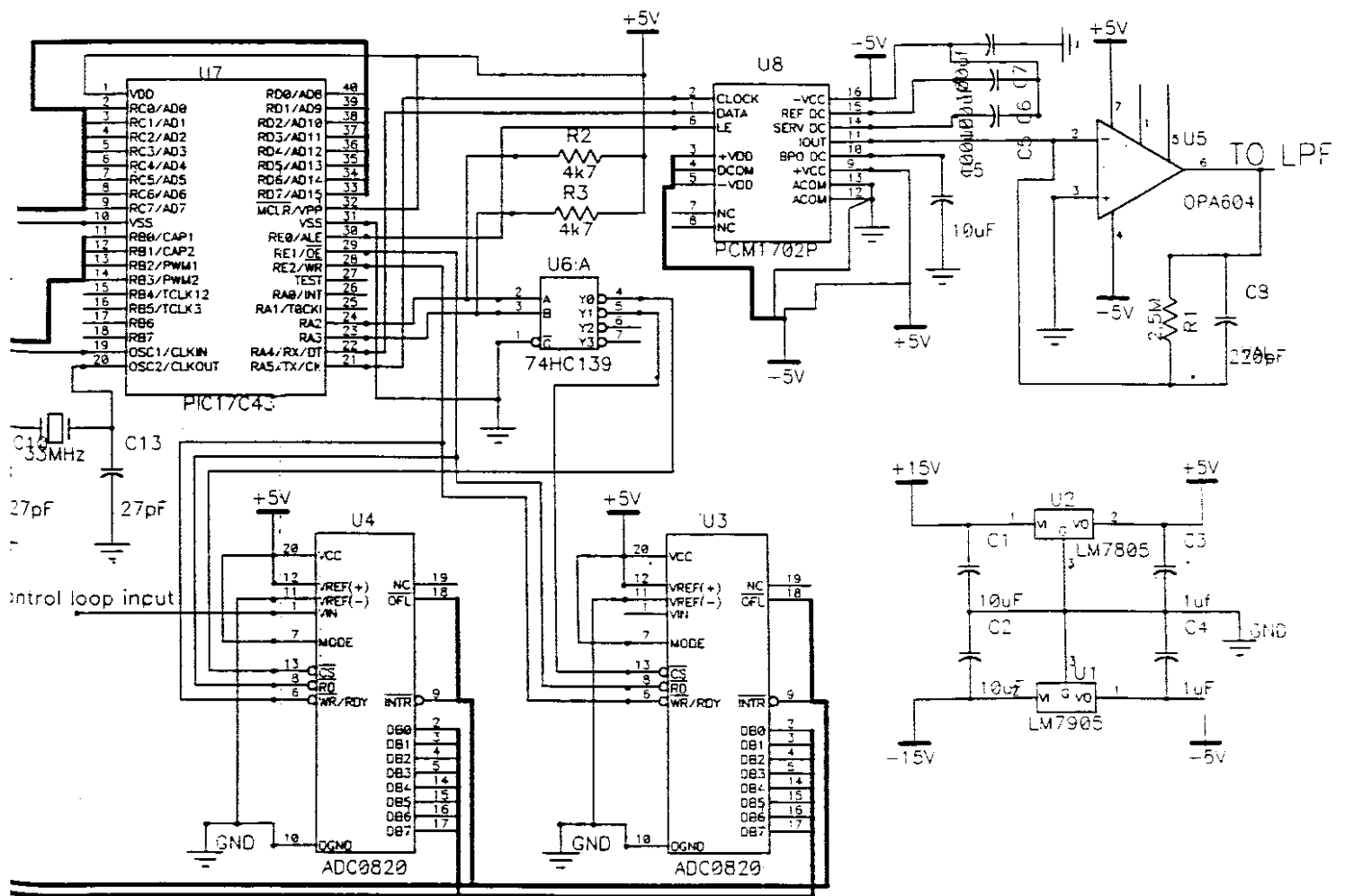
Dat = "";
for (i = 0; i < 32; i++)
{
    for (int j = 0 ; j < 8; j++)
    {
        Tmp.Format("%x\t",ar[j+i*8]);
        Dat += Tmp;
    }
    Dat += "\n";
}
::MessageBox(NULL, (LPCTSTR)Dat, 0, 0);
drv.ExampleStop();
}

```

Appendix F: Procedure to test the complete communications link.

Disconnect the BNC connector that comes from the 10^6 tunneling current amplifier and use the test lead to connect the BNC cable input that goes to the 10^3 amplifier to +5Volt. Run the test routine now. The PC will simulate a scan, but at the end, it will display two screens full of values that were read from the system. These values would be the values that the Z controller generated in response to a 0Volt input on the control loop. The values are displayed in a pop-up screen and all of the values should be 'f80000'. If the other part of the test is performed, the input to the 10^3 amplifier is connected to ground. In this case, the output should show '7ffff', which corresponds to the control loop trying to extend the tube to the maximum length to re-acquire the signal. The signal is inverted in the XYZ to 4-Quadrant converter in the working STM, the more voltage that is apply to a piezo tube to change the Z length, the shorter it will become. This is not the case for this system test condition. At the same time this test is done a oscilloscope can be use to check that the Z controller is writing the correct waveform to the 20-bit DAC.

Appendix G: Schematic diagram of the Z controller.



Appendix H: Schematic diagram of the Butterworth LPF on the Z controller.

